# AFS/AFM60 EtherNet/IP

Absolute encoder





#### **Described product**

AFS/AFM60 EtherNet/IP

#### Manufacturer

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

#### Legal information

This work is protected by copyright. Any rights derived from the copyright shall be reserved for SICK AG. Reproduction of this document or parts of this document is only permissible within the limits of the legal determination of Copyright Law. Any modification, abridgment or translation of this document is prohibited without the express written permission of SICK AG.

The trademarks stated in this document are the property of their respective owner.

© SICK AG. All rights reserved.

#### **Original document**

This document is an original document of SICK AG.



# Contents

1	Abou	ut this d	ocument	6				
	1.1	Scope		6				
	1.2	Purpose of this document						
	1.3	Target group						
	1.4	Further	information	6				
	1.5	Symbols	s and document conventions	7				
	1.6	Abbrevia	ations used	7				
2	Safe	ty infor	mation	9				
	2.1	General	notes	9				
	2.2	Intende	d use	9				
	2.3	Require	ments for the qualification of personnel	9				
3	Prod	luct des	cription	11				
	3.1	Product	identification	11				
	3.2	Specific	features	12				
	3.3	Operating principle of the encoder						
		3.3.1	Scalable resolution	13				
		3.3.2	Preset function	13				
		3.3.3	Round axis functionality	14				
	3.4	Integrati	ion in EtherNet/IP	15				
		3.4.1	EtherNet/IP architecture	15				
		3.4.2	Communication in EtherNet/IP	16				
	3.5	CIP object model						
		3.5.1	Supported classes	18				
		3.5.2	Identity object	20				
		3.5.3	Assembly object	22				
		3.5.4	Position sensor object	27				
	3.6	Integrati	ion and configuration options	34				
		3.6.1	Integration in EtherNet/IP	34				
		3.6.2	Configuration	34				
	3.7	Parame	terizable functions	36				
		3.7.1	Saving and resetting configuration	36				
		3.7.2	IP address	38				
		3.7.3	Slave sign of life	38				
		3.7.4	Code sequence	38				
		3.7.5	Scaling	39				
		3.7.6	Increments per revolution	39				
		3.7.7	Total resolution/measuring range	39				
		3.7.8	Preset function	40				
		3.7.9	Speed measurement unit	40				
		3.7.10	Round axis functionality	40				
	3.8	Operatir	ng elements and status indicators	41				

4	Con	nmission	ing	43				
	4.1	Electrica	al installation	43				
		4.1.1	Encoder connections	43				
	4.2	Settings	on the hardware	44				
		4.2.1	IP address setting	44				
		4.2.2	Triggering a preset value with the preset pushbutton	45				
5	Configuration using a PLC							
	5.1	Delivery	state	46				
	5.2	IP addre	ess of the encoder	46				
		5.2.1	Without DHCP server	46				
		5.2.2	Assigning the IP address via DHCP	46				
		5.2.3	Freezing the assigned IP address	49				
		5.2.4	Checking integration into EtherNet/IP via RSLinx Classic	50				
	5.3	Creating	a project in the control software	51				
	5.4	Integrati	on and configuration using an EDS file	52				
		5.4.1	Requirements	52				
		5.4.2	Setting up communication	53				
		5.4.3	Configuration	55				
	5.5	Installin	g the ladder routine	56				
		5.5.1	Import of the ladder routine	56				
		5.5.2	Integration as SubRoutine in MainRoutine	61				
		5.5.3	Using the SubRoutine	62				
		5.5.4	Reading out and changing parameters of the encoder	63				
	5.6	Functior	n block	66				
		5.6.1	Requirements	66				
		5.6.2	Import and wiring	66				
	5.7	Integrati	on of the encoder as generic module	67				
		5.7.1	Module settings	68				
		5.7.2	Downloading the configuration to the control	70				
		5.7.3	Checking communication	71				
	5.8	Program	ming examples	71				
		5.8.1	Reading out temperature	72				
		5.8.2	Setting preset value	82				
6	Con	figuratio	n using the integrated web server	94				
	6.1	Home		95				
		6.1.1	Device	95				
		6.1.2	Position	95				
		6.1.3	Speed	95				
		6.1.4	Temperature	95				
		6.1.5	Timer	95				
	6.2	Paramet	terization	96				
		6.2.1	Overview	97				
		6.2.2	Units	97				
		6.2.3	Changing preset value	98				

		6.2.4	Triggering preset	98
		6.2.5	Scaling	99
		6.2.6	Round axis functionality	99
		6.2.7	Limits	100
		6.2.8	Reset	101
	6.3	Diagnosti	CS	101
		6.3.1	Status	101
		6.3.2	Speed	102
		6.3.3	Temperature	102
		6.3.4	Time	102
		6.3.5	Cycles	103
		6.3.6	Heartbeat	103
	6.4	Tools		103
		6.4.1	EDS	103
		6.4.2	Ladder routine	103
		6.4.3	Update	104
		6.4.4	Address switch	105
		6.4.5	Fault header information	105
	6.5	Test note	S	105
7	Troul	oleshooti	ng	106
	71	Response	e to errors	106
	7.2	SICK-STE	GMANN Support	106
	7.3	Diagnosti	CS.	106
		7.3.1	Error and status indications of the LEDs	106
		7.3.2	Self test via EtherNet/IP	107
		7.3.3	Warnings, alarms and errors via EtherNet/IP	107
		7.3.4	Error messages of the Allen Bradlev control system	110
				-
8	Anne	ЭX		112
	8.1	Conformi	ty	112

# **1** About this document

# 1.1 Scope

<sup>7</sup> These operating instructions apply to the AFS60/AFM60 EtherNet/IP absolute encoder with the following type designations:

- Singleturn encoder = AFS60A-xxlx262144
- Multiturn encoder = AFM60A-xxlx018x12

## 1.2 Purpose of this document

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

- Electrical installation
- Commissioning
- Parameterization
- Operation
- Maintenance

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

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

# 1.3 Target group

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

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

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

#### NOTICE

1

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

### **1.4** Further information

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

The following information is available via the Internet:

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

# 1.5 Symbols and document conventions

# Safety notes

# DANGER

A safety note informs you of real-world specifications for safely mounting and installing the absolute encoder.

This is intended to protect you against accidents.

Read the safety notes carefully and follow them.

# Information on property damage/general advice

# NOTICE

Indicates important information and possible property damage.

#### 

Indicates useful tips and recommendations.

#### Instructions

- Instructions requiring specific action are indicated by an arrow. Carefully read and follow the instructions for action.
- 1. The sequence of instructions for action is numbered.
- 2. Numbered instructions for action are to be followed in the given order.

### Status indicators

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

- The LED is illuminated continuously.
- The LED is flashing.
- O The LED is off.

# 1.6 Abbreviations used

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

I/O Input and Output Data (from the point of view of the master)

IP in TCP	/IP	Internet Protocol
	/ ••	

IP in EtherNet/IP

- et/IP Industrial Protocol MAC Media Access Control
- ODVA Open DeviceNet Vendor Association
  - PLC Programmable Logic Controller
  - TCP Transmission Control Protocol
- **UDP** User Datagram Protocol = connectionless network protocol

# 2 Safety information

# 2.1 General notes



#### DANGER

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

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

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

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

#### 2.2 Intended use

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

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

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

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

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

### 2.3 Requirements for the qualification of personnel

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



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

The following qualifications are necessary for the various tasks:

#### Table 1: Authorized personnel

Task	Qualification
Mounting (see mount- ing instructions)	<ul><li>Basic practical technical training</li><li>Knowledge of the current safety regulations in the workplace</li></ul>
Electrical installation and device replace- ment	<ul> <li>Practical electrical training</li> <li>Knowledge of current electrical safety regulations</li> <li>Knowledge of the operation and control of the devices in their particular application (e.g., industrial robots, storage and conveyor systems)</li> </ul>
Commissioning, oper- ation, and configura- tion	<ul> <li>Knowledge of the current safety regulations and of the operation and control of the devices in their particular application</li> <li>Knowledge of automation systems (e.g. Rockwell ControlLogix con- troller)</li> <li>Knowledge of EtherNet/IP</li> <li>Knowledge of the use of automation software (e.g. with Rockwell RSLogix)</li> </ul>

# **3 Product description**

# 3.1 Product identification



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

#### Solid shaft type code

Table 2: Solid shaft type code

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

Table 3: Solid shaft type code - explanation

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

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

#### Blind hollow shaft type code

Table 4: Blind hollow shaft type code

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

Table 5: Blind hollow shaft type code - explanation

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

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

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

# 3.2 Specific features

Table 6: Special features of the encoder variants

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

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

# 3.3 Operating principle of the encoder

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

#### The AFS60 EtherNet/IP is a singleturn encoder

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

#### The AFM60 EtherNet/IP is a multiturn encoder

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

#### 3.3.1 Scalable resolution

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

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

#### 3.3.2 Preset function

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



Figure 1: Setting a preset value

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

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

#### 3.3.3 Round axis functionality

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

#### 

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

#### Example with transmission ratio

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



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

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

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

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

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

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

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

#### Example without transmission ratio



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

- ① Rotating table with nine fillers
- 2 1000 steps
- 3 Encoder

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

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

1000 steps are configured as the total resolution.

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

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

### 3.4 Integration in EtherNet/IP

#### 3.4.1 EtherNet/IP architecture

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

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



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

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



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

#### 3.4.2 Communication in EtherNet/IP

#### MAC address

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

- 3 byte ident number
- 3 byte device identifier

#### TCP/IP and UDP/IP

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

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

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

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

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

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

#### Common industrial protocol (CIP)

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



Figure 6: CIP and other services

- Process level
- 2 Communication levels
- 3 Physical level

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

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

#### EtherNet/IP communication

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



Figure 7: Ethernet frame

- ① Transmission order
- 2 Header

- 3 Data field
- ④ Trailer

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

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

IP-Header	TCP/UDP-Header	CIP-Header		CIP-Daten
		_	1	
		-	2	-
			3	-

Figure 8: Ethernet data field

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

# 3.5 CIP object model

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

The most important terms are explained below:

- Class A class contains related objects of a device, organized into instances.
- Instance An instance consists of various attributes, which describe the properties of this instance. Different instances in a class have the same services and the same attributes. They can, however, have different attribute values.
- Attribute Attributes represent the data which a device makes available via EtherNet/IP. This data contains the current values of a configuration or an input, for example. Typical attributes are, e.g., configuration or status information.
- **Service** Services are used to access classes or attributes of a class and to generate certain events. These services perform specified actions, e.g., reading attributes.

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

Table 7: Example CIP object model

#### 3.5.1 Supported classes

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



Figure 9: Supported classes

Table 8: Supported classes

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

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

#### 3.5.2 Identity object

The device information or parameters are retrieved via the instances.



Figure 10: Connections for the identity object

Table 9: Class services of the identity object

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

Table 10: Class attributes of the identity object

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

# NOTE

The class attribute 5 has not been implemented.

i

#### Table 11: Instance services of the identity object

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

Table 12: Instance attributes of the identity object

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

Table 13: Bits of the "Status" instance attribute

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

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

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

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

#### 3.5.3 Assembly object

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

Table 15: Class services of the assembly object

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

Table 16: Class attributes of the assembly object

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

The class attributes 4 and 5 have not been implemented.

The encoder supports only input and listen-only connections.

Table 17: Instance :	services of	the assembly	object
----------------------	-------------	--------------	--------

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

Table 18: Instance attributes of the assembly object

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

# **i** NOTE

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

#### I/O assembly

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



Figure 11: Connections for the I/O assembly

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

Instanc e	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	0			Position	value (lea	ast signific	ant byte)	·	
	1				Positio	n value			
	2				Positio	n value			
	З			Position	value (mo	ost signific	ant byte)		
2	0			Position	value (lea	ast signific	ant byte)		
	1				Positio	n value			
	2				Positio	n value			
	3			Position	value (mo	ost signific	ant byte)		
	4							Warning	Alarm
3	0			Position	value (lea	ast signific	ant byte)		
	1				Positio	n value			
	2				Positio	n value			
	3			Position	value (mo	ost signific	ant byte)		
	4			Speed	value (lea	st significa	ant byte)		
	5		Speed value						
	6				Speed	d value			
	7			Speed	value (mo	st significa	ant byte)		

Instanc e	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
101/	0		Fault head	der (least	l significant	t byte, <mark>see</mark>	table 33,	page 108)		
101WS	1		Fault header							
	2				Fault	header				
	3			Fault h	eader (mo	st signific	ant byte)			
	4			Position	value (lea	ast signific	ant byte)			
	5				Positio	n value				
	6				Positio	n value				
	7			Position	value (mo	ost signific	ant byte)			
102/ 102WS	0			Fault h	eader (lea	st signific	ant byte)			
	1				Fault	header				
	2				Fault	header				
	3			Fault h	eader (mo	st signific	ant byte)			
	4			Position	value (lea	ast signific	ant byte)			
	5				Positio	n value				
	6				Positio	n value				
	7			Position	value (mo	ost signific	ant byte)			
	8							Warning	Alarm	
103/ 103WS	0		Fault head	der (least	significant	t byte, <mark>see</mark>	table 33,	, page 108)		
	1				Fault	header				
	2				Fault	header				
	3			Fault he	eader (mo	st signific	ant byte)			
	4			Position	value (lea	ast signific	ant byte)			
	5				Positio	n value				
	6				Positio	n value				
	7			Position	value (mo	ost signific	ant byte)			
	8			Speed	value (lea	st significa	ant byte)			
	9				Speed	d value				
	10				Speed	d value				
	11			Speed	value (mo	st significa	ant byte)			

# Configuration assembly

The encoder can be configured via the configuration assembly.



Figure 12: Connections for the configuration assembly

I) NOTE

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

			-		-				
Instanc e	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit O
100	0				Not	used			
	1				Not	used			
	2				Not	used			
	3				Not	used			
	4		Ste	ps per rev	olution CF	PR (least s	ignificant	byte)	
	5				С	PR			
	6				С	PR			
	7			CP	R (most si	gnificant l	oyte)		
	8		Тс	otal resolu	ition CMR	(least sig	nificant by	rte)	
	9				С	MR			
	10				С	MR			
	11			CM	R (most s	ignificant	byte)		
	12				Not used				cw/ccw <sup>1</sup>

Table 20: Data format of the configuration assembly attributes

Instanc e	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit O
	13				Not used				scf <sup>2</sup>
	14				Not used				raf <sup>3</sup>
	15				Not	used			·
	16	Nume	erator of t	he numbe	er of revolu	itions CNF	R_N (least	significan	it byte)
	17				CN	R_N			
	18				CN	R_N			
	19			CNR	_N (most	significant	t byte)		
	20	Denon	ninator of	the numb	er of revo	lutions CN	IR_D (leas	t significa	nt byte)
	21				CN	R_D			
	22				CN	R_D			
	23			CNR	_D (most :	significant	t byte)		
	24		Spe	ed measu	irement ui	nit (least s	ignificant	byte)	
	25		Spe	ed measu	irement ur	nit (most s	ignificant	byte)	
	26				Not	used			
	27				Not	used			

1 cw = clockwise

ccw = counterclockwise

 $^2$  scf = scaling function

<sup>3</sup> raf = round axis functionality

# I NOTE

•

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

#### 3.5.4 Position sensor object

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



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

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

Table 22: Class attributes of the position sensor object

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

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

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

Table 24: Instance attributes of the position sensor object

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

Attribute ID	Acces s	V/NV <sup>1</sup>	Name	Description	Data type	Min. max. (default value)
19h	Set	NV	Velocity format	Speed unit 1F04h = counts/s 1F05h = counts/ms 1F0Eh = turns/s 1F0Fh = turns/min 1F10h = turns/h	ENG UInt	(1FOFh)
1Ah	Set	NV	Velocity resolution	Minimum resolution of the speed measure- ment	DUInt	(0000001h)
1Bh	Set	NV	Minimum velocity setpoint	Lower/upper limit of the speed in turns/min. <sup>4</sup>	DInt	(-12,000)
1Ch	Set	NV	Maximum velocity setpoint	If the speed under- cuts/exceeds this value, the warning flag (attrib- ute 2Fh) is set.	DInt	(+12,000)
1Dh	Get	V	Acceleration value	Current acceleration. The format is deter- mined by attributes 1Eh and 1Fh.	DInt	00000000h FFFFFFFFh
1Eh	Set	NV	Acceleration for- mat	Acceleration unit $0810h = counts/ms^2$ $0811h = counts/s^2$ $0812h = turns/s^2$ $0813h = rad/s^2$	ENG UInt	(0810h)
1Fh	Set	NV	Acceleration reso- lution	Minimum resolution of the acceleration meas- urement	DUInt	(1)
20h	Set	NV	Minimum acceler- ation setpoint	Lower/upper limit of acceleration in counts/	DInt	(C000001h)
21h	Set	NV	Maximum acceler- ation setpoint	1 ms <sup>2, 5</sup> If the acceleration undercuts/exceeds this value, the warning flag (attribute 2Fh) is set.	DInt	(3FFFFFFh)
29h	Get	V	Operating status	Operational status of the encoder Bit 0: Direction 0 = Counting up 1 = Counting down Bit 1: Scaling 0 = Off 1 = On Bit 2 4: Reserved Bit 5: Diagnostics on/off 0 = Off 1 = On Bit 6, 7: Reserved	Byte	
2Ah	Get	NV	Physical resolution span	Physical resolution per revolution = 18 bits	UDInt	(40000h)
2Bh	Get	NV	Physical resolution Number of span	Physical number of revo- lutions 0001h = Singleturn 1000h = Multiturn	Uint	(0001h) or (1000h)
2Ch	Get	V	Alarms	Bit field with flags for alarms and errors (see table 34, page 109)	WORD	-
2Dh	Get	NV	Supported alarms	Supported alarms and errors	WORD	3003h
2Eh	Get	V	Alarm flag	0 = No alarm/error 1 = Alarm/error	BOOL	-

Attribute ID	Acces s	V/NV 1	Name	Description	Data type	Min. max. (default value)
2Fh	Get	V	Warnings	Bit field with flags for warnings (see table 35, page 110)	WORD	-
30h	Get	NV	Supported warn- ings	Supported warnings	WORD	67C3h
31h	Get	V	Warning flags	0 = No warning 1 = Warning	BOOL	-
32h	Get	NV	Operating time	Stored operating time in 0.1 h = 6 min	UDInt	0
33h	Get	NV	Offset value	Offset value is calcu- lated when initializing the preset function.	DInt	00000000h
64h	Get	V	Temperature value	Current temperature with ± 5° accuracy -40 to +100 °C or -40 to +212 °F	INT	F060h 2710h
65h	Set	NV	Temperature value format	Temperature unit 1200h = °C (Celsius) 1201h = °F (Fahren- heit)	ENG UINT	(1200h)
66h	Set	NV	Temperature reso- lution	Smallest resolution of temperature (°C/100 or °F/100)	UDINT	(00000064h)
67h	Set	NV	Minimum temper- ature setpoint	Lower/upper limit of temperature in °C. 6	INT	F060h - (F060h = -4,000)
68h	Set	NV	Maximum temper- ature setpoint	If the temperature undercuts/exceeds this value, the warning flag (attribute 2Fh) is set.	INT	- 2710h (2710h = +10,000) or (52D0h = +21,200)
69h	Get	V	Fault header	see table 33, page 108	DWORD	(0000000h)
6Ah	Set	V	Special encoder functionalities	Bit field with flags for special encoder func- tions Bit 0: Slave sign of life (On/Off) Bit 1 7: Not used Bit 8 15: Update fac- tor (2 127) Bit 16: Storage mode (Auto (0)/Manual (1)) Bit 17 23: Not used Bit 24 31: Activa- tion/deactivation of FTP firmware update func- tionality (default: active (1))	DWORD	(01000500h)
6Bh	Get	NV	Encoder motion time	Stored movement time in seconds (will be increased when mov- ing).	UDINT	-
6Ch	Get	NV	Encoder operating time	Stored operating time in seconds (is increased as soon as the encoder is in operation).	UDINT	-
6Dh	Get	NV	Max. velocity	Highest speed that the encoder has reached since commissioning. <sup>7</sup>	UDINT	-
6Eh	Get	NV	Max. acceleration	Highest acceleration the encoder has expe- rienced since commis- sioning. <sup>8</sup>	UDINT	-

31

Attribute ID	Acces s	V/NV <sup>1</sup>	Name	Description	Data type	Min. max. (default value)
6Fh	Get	NV	Max. Temp	Operating temperature in °C/100	UDINT	-4,000
70h	Get	NV	Min. temp	Lowest operating tem- perature reached in °C/100	UDINT	10,000
71h	Get	NV	Number of start- ups	Number of start-ups (power on) of the encoder	UDINT	-
72h	Get	V	LED current value	Momentary internal LED current of the sensor in µA	UINT	200 25,000 (0)
73h	Get	NV	Max. current value	Maximum internal LED current of the sensor system in µA	UINT	200
74h	Get	NV	Min. current value	Minimum internal LED current of the sensor system in µA	UINT	25,000
75h	Get	V	Direction change counter	Number of changes in the direction of rota- tion (the numerator increases when the encoder changes the direction of rotation).	UDINT	0
76h	Get	V	Revolution counter forward	Number of clockwise starts (The numerator increases when the encoder moves clock- wise).	UDINT	0
77h	Get	V	Revolution counter backwards	Number of counterclock- wise starts (The numer- ator increases when the encoder moves counter- clockwise).	UDINT	0
78h	Get	V	Power supply volt- age	Current supply voltage in mV	UINT	9,500 30,500 (24,000)
79h	Get	V	Max. power supply voltage	Maximum supply volt- age in V (stored in EEPROM).	UINT	0 33 (0)
7Ah	Get	V	Preset offset value	Offset value calculated from the preset value. <sup>9</sup>	DINT	(0000000)
7Dh	Set	NV	Endless shaft functionality	Activates round axis functionality 0 = Off 1 = On	BOOL	(0)
7Eh	Set	NV	Number of revolu- tions, nominator	Numerator for the num- ber of revolutions	UDINT	1 2,048 (2,048)
7Fh	Set	NV	Number of revolu- tions, divisor	Denominator for the number of revolutions	UDINT	1 65,535 (1)
80h	Set	NV	Velocity filter inte- gration time	Number of measured values from which a mean value is formed (scanning frequency of the position as the basis for calculating the speed = 1 kHz). <sup>10</sup>	UDINT	0 128 (1)

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

<sup>1</sup> V = Volatile (volatile data), NV = Non-volatile (non-volatile data).

<sup>2</sup> Area monitoring can be implemented with the lower and upper limit of the position. It is not an electronic cam.

<sup>3</sup> The maximum speed depends on the "solid shaft" or "blind hollow shaft" mechanical interface used (see data sheet).

<sup>4</sup> The unit changes with the velocity format (attribute ID 19h). The limits must then be converted accordingly, e.g. 12,000 turns/min = 200 turns/s.

<sup>5</sup> The unit changes with the acceleration format (attribute ID 1Eh). The limits must then be converted accordingly, e.g. 2 counts/ms<sup>2</sup> = 2,000,000 counts/s<sup>2</sup>.

<sup>6</sup> The unit changes with the temperature value format (attribute ID 65h). The limits must then be converted accordingly.

<sup>7</sup> The value is output in the format defined in attribute ID 19h.

<sup>8</sup> The value is output in the format defined in attribute ID 1Eh.

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

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

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

#### 

The filters are each applied in the following order:

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

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

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

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

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

#### 3.6 Integration and configuration options

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

#### 3.6.1 Integration in EtherNet/IP

The encoder can be integrated into EtherNet/IP:

 As a generic module (see "Integration of the encoder as generic module", page 67):

All module settings must be entered manually.

 Using an EDS file (see "Integration and configuration using an EDS file", page 52):

The module settings of the absolute encoder are already predefined.

#### 3.6.2 Configuration

The following options are available for configuring the encoder:

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

#### Case 1: When integrated as generic module

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

 If the configuration assembly is activated under Connection Parameters, then the configuration assembly is to be used for configuration (see "Module settings", page 68).

In addition, the parameters that are not included in the configuration assembly can be configured with the web server integrated in the encoder.

 If the configuration assembly is not activated under Connection Parameters, then the web server can be used to configure all parameters (see "Configuration using the integrated web server", page 94).

# NOTE

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

#### Case 2: When integrating using the EDS file

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

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

#### Case 3: When using ladder routine for configuration mapping

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

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

# NOTE

i

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

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

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



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

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

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



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

# 3.7 Parameterizable functions

#### 3.7.1 Saving and resetting configuration

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

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

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



Figure 14: Configuration memory

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

#### **Reset: Resetting to factory settings**

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

Or:

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

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

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

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

#### 

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

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

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

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

### Save: Saving parameters to non-volatile memory

Use the Save class service (service code 16h) from the position sensor object (see table 21, page 28).

The parameters are saved to the non-volatile memory. The following table shows which parameters are saved.

Attribute ID in the Parameter **Factory setting** position sensor object 0Ch Code sequence cw 0Eh Scaling Off 10h Increments per revolution 262.144 11h 1,073,741,824 Total resolution Preset value 13h 0 16h 0 Lower position limit 17h Upper position limit 1,073,741,823 19h Speed unit turns/min -12,000 1Bh Lower speed limit 1Ch Upper speed limit 12,000 1Eh Acceleration unit counts/ms<sup>2</sup> 20h Lower limit of acceleration -1,073,741,82321h 1,073,741,823 Upper limit of acceleration 65h °C Temperature unit Off 7Dh Round axis functionality 7Eh Numerator for the number of revolutions 2.048 7Fh Denominator for the number of revolutions 1 80h 1 Number of measured values from which a mean value is formed 100 81h Bandwidth of the low pass filter

Table 26: Parameters that are saved or reset

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



The following parameters are not reset:

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

# 3.7.2 IP address

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

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

# 3.7.3 Slave sign of life

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

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

The bit changes its value in the configured update cycle.

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

Update cycle = RPI × update factor × 6

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

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

## 3.7.4 Code sequence

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

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



# DANGER

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

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

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

# 

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

## 3.7.5 Scaling

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

#### 

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



# DANGER

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

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

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

#### 

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

### 3.7.6 Increments per revolution

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

# NOTE

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

### 3.7.7 Total resolution/measuring range

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

### Table 27: Examples for total resolution

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

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



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

### 3.7.8 Preset function

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

The preset value can be set in the following way:

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

# DANGER

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

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

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

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

### 3.7.9 Speed measurement unit

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

Possible units are:

- counts/s<sup>1</sup>
- counts/ms<sup>1)</sup>
- turns/s
- turns/min
- turns/h

The factory setting is turns/min.

<sup>1)</sup> Depending on the configured resolution.

Example: Resolution = 2,000 steps; the encoder rotates 0.5 times per second = 1,000 counts/s or 1 counts/ms.

### 3.7.10 Round axis functionality

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

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

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

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

### Number of revolutions, numerator

The numerator (attribute ID 7Eh, nominator "CNR\_N") is scalable from 1 ... 2,048 in whole numbers. Factory setting for the numerator is 2,048.

### Number of revolutions, denominator

The denominator (attribute ID 7Fh, divisor "CNR\_D") is scalable from 1 ... 65,535 in whole numbers. Factory setting for the denominator is 1.

### **Resulting total resolution**

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

### The following restrictions must be observed:

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

# NOTE

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

# i NOTE

i

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

#### 

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



### DANGER

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

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

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

# NOTE

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

# 3.8 Operating elements and status indicators

The absolute encoder has five LEDs.

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



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

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

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

The following control elements are located under the screw cover:

- Address switch
- Preset pushbutton

# 4 Commissioning

# 4.1 Electrical installation



# Risk of injury from electrical voltage.

Disconnect the system from the voltage supply to prevent the system from starting unintentionally.

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

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

# 4.1.1 Encoder connections

The encoder connections are located on the rear side.



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

- ① Voltage supply
- ② Ethernet port 1
- 3 Ethernet port 2





Figure 17: Ethernet port M12 × 4, female connector Figure 18: Voltage supply M12 × 4, male connector nector

### 

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

Table 28: Pin a	assignment of the	voltage supply	connection
-----------------	-------------------	----------------	------------

PIN	Signal	Wire color <sup>1</sup>	Function
1	V <sub>S</sub>	Brown	Supply voltage 10 30 V DC
2	-	White	Do not use
3	GND	Blue	0 V DC (ground)

PIN	Signal	Wire color <sup>1</sup>	Function
4	-	Black	Do not use

<sup>1</sup> When using pre-assembled cables.

#### 

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

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

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

<sup>1</sup> When using pre-assembled cables.

i) NOTE

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

# 4.2 Settings on the hardware

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

- Three address switches
- Preset pushbutton
- Open the screw cover using a screwdriver for slotted screws with a blade width of min. 10.0 mm.

# 4.2.1 IP address setting



Figure 19: Address switch and preset pushbutton

- ① Address switch
- 2 Preset pushbutton
- 3 Hundreds place
- (4) Tens place
- S Ones place

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

Table 30: Address switch - meaning of the adjustable values

### Fixed IP address via address switch

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

### Obtaining the IP address via DHCP

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

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

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

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

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

### 4.2.2 Triggering a preset value with the preset pushbutton

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



# DANGER

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

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

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



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

- The preset value must be within the configured measuring range.
- Do **not** press the preset pushbutton for longer than 5 seconds, as this may reset the encoder to the factory settings.

# 5 Configuration using a PLC

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

# I) NOTE

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

# 5.1 Delivery state

The encoder is delivered with the following parameters:

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

# 5.2 IP address of the encoder

# 5.2.1 Without DHCP server

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

#### 

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

# 5.2.2 Assigning the IP address via DHCP

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

5	BootP DHCP EtherNet/IP Con	nmissioni	ng Tool				_		$\times$
File	Tools Help								
	Add Relation		Discovery H	listory				Clear Hist	ory
E	thernet Address (MAC)	Туре	(hr:min:sec)	#	IP Address		Hostna	me	
			Entered Re	lations					
E	thernet Address (MAC)	Туре	IP Address		Hostname	Descr	iption		
Erro	rs and warnings							Rela	ations —

### Figure 20: Empty window in BootP

ress Hostname
ame Description

Figure 21: Contact structure of the encoder

- As soon as contact with the encoder is established via the Ethernet address (MAC), but there is no link to the specific IP address yet, the message "Unable to service DHCP request" is displayed in BootP.
   This means that the encoder has started to periodically send DHCP requests to get an IP address assigned to its MAC address. However, the server (laptop/PC) cannot yet respond accordingly.
- 7. Add encoder to the list via the Add Relation button.
- Assign the last three digits of the IP address (client IP Address) for the encoder. Use the same IP address range as specified by the server (laptop/PC) (server IP address).

Add Relation		Discovery H	History		Clear Histor
Ethernet Address (N	AC) Type	(hr:min:sec)	# IP Addr	ess	Hostname
0:06:77:07:C3:92	New Entry			$\times$	
	Server	IP Address: 10.2.1	00.1		
	Client Add	ress (MAC): 00:06:	77:07:C3:92		
	Client	IP Address:   10	. 2 . 100 .	10	
Ethernet Address (I	ì	Hostname: Description:			ption
		ОК	Cancel		

Figure 22: IP address assignment

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

This typically takes up to 30 seconds.

		Discovery H	listory			Clear History
Ethernet Address (MAC)	Туре	(hr:min:sec)	#	IP Address	Hostna	ате
0:06:77:07:C3:92	DHCP	10:15:24	4			
		Entered Re	lation	3		
	-			1	<b>n</b>	
thernet Address (MAC)	Туре	IP Address		Hostname	Description	
10:06:77:07:C3:92		10.2.100.10				

Figure 23: Sending the IP address to the encoder

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

Add Relation		Discovery H	listory			Clear Histor
Ethernet Address (MAC)	Туре	(hr:min:sec)	#	IP Address	Hostna	ame
JU:U6:77:U7:C3:92	DHCP	10:16:28	b	10.2.100.10	J	
Ethernet Address (MAC)	Туре	Entered Re	lations	Hostname	Description	
00:06:77:07:C3:92	DHCP	10.2.100.10				

Figure 24: IP address assigned successfully

# 5.2.3 Freezing the assigned IP address

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

**Deactivating DHCP server** 

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

Add Relation		Discovery H	listory			Clear History
Ethernet Address (MAC)	Туре	(hr:min:sec)	#	IP Address		Hostname
00:06:77:07:C3:92	DHCP	10:18:49	8	10.2.100.10		
Delete Relation		Entered Re	lations	Enable BOOTP	/DHCP	Disable BOOTP/DHCF
Delete Relation Ethernet Address (MAC)	Туре	Entered Re IP Address	lations	Enable BOOTP/	/DHCP	Disable BOOTP/DHCF
Delete Relation Ethernet Address (MAC) 00:06:77:07:C3:92	Туре DHCP	Entered Re IP Address 10.2.100.10	lations	Enable BOOTP/	/DHCP	Disable BOOTP/DHCF
Delete Relation Ethernet Address (MAC) 00:06:77:07:C3:92	Type DHCP	Entered Re IP Address 10.2.100.10	lations	Enable BOOTP/	/DHCP	Disable BOOTP/DHCF
Delete Relation Ethernet Address (MAC) 00:06:77:07:C3:92	Type DHCP	Entered Re IP Address 10.2.100.10	lations	Enable BOOTP/	/DHCP	Disable BOOTP/DHC
Delete Relation Ethernet Address (MAC) 00:06:77:07:C3:92	Туре DHCP	Entered Re IP Address 10.2.100.10	lations	Enable BOOTPA	/DHCP	Disable BOOTP/DHC

Figure 25: Disable BOOTP/DHCP button

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

5	BootP D	HCP EtherNet/IP Con	nmissioni	ng Tool				— I	×
File	Tools	Help							
	1	Network Settings				1		Clea	ar History
Ē		Add Relation				<u> </u>			
		Clear Discovery Histor	y			dress		Hostname	
	1	Delete Relation				100.10	)		
	1	Enable BOOTP/DHCP				L			
	1	Disable BOOTP/DHCP							
	1	Reset Module's Netwo	rk Setting	s to Factory Defaults					
	1	Properties				e BOOT	р/DHCP	Disable BOOT	P/DHCP
Ē			-						
	thern	et Address (MAC)	Туре	IP Address	Host	name	Descri	iption	
0	0:06:7	7:07:C3:92	DHCP	10.2.100.10					
	orsand v v 1021	varnings 00.10 to Ethernet addres	- 00·0E·77	07.02.02					Helations
Jei	ic 10.2.1	oo, to to Ethemet addles	\$ 00.00.77	.01.03.32					101230

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

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

	<b>T</b>	Discovery H	listory +	10 4 4 4		-	
D0:06:77:07:C3:92	DHCP	10:18:49	# 8	10.2.100.10		nosulai	
Delete Relation		Entered Re	elations	Enable BOOT	P/DHCP	Disable I	300TP/DHCF
Ethernet Address (MAC)	Туре	IP Address		Hostname	Descr	iption	
00:06:77:07:C3:92	DHCP	10.2.100.10					

Figure 27: Disable DHCP status message

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

# 5.2.4 Checking integration into EtherNet/IP via RSLinx Classic

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

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



Figure 28: RSWho button in RSLinx Classic

- 3. Open the AB\_ETHIP-1  $\rightarrow$  Ethernet path.
- ✓ The encoder is visible under its IP address.



Figure 29: Encoder in the AB\_ETHIP-1 of RSLinx Classic path

# 5.3 Creating a project in the control software

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

O New Project		?	×
1769-L18ERM-B AFM60_manual_1	B1B CompactLogix™ 5370 Controller		
Revision:	31 ×		
Expansion I/O:	0 Modules Y		
	▲ <b>Danger:</b> When online, if the modules present do not match the modules specified in the project, unexpected control may occur. The Expansion I/O setting must match the actual number of modules.		
Security Authority:	No Protection ~		
	Use only the selected Security Authority for authentication and authorization		
Secure With:	Logical Name <controller name=""></controller>		
	<ul> <li>Permission Set</li> </ul>		
Description:			
	Cancel Back Next	Fini	sh

Figure 30: Projecting the hardware

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

# NOTE

i

Type and Chassis Type must match your controller.

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

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

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

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

# 5.4 Integration and configuration using an EDS file

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

# 5.4.1 Requirements

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

### 5.4.2 Setting up communication

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

⊿ 🚍 Poin @ ⊿ 😭	ntlO [0] 11 Emb [1] Expan	769-L18ERM-B edded I/O I] Embedded D nsion I/O, 0 Mo	B1B AFM60_manual_` Discrete_IO pdules
⊿ 🚠 Ethe	ernet 1	New Module.	
-		Import Modu Discover Mod	le lules
	ß	Paste	Strg+V
		Properties	Alt+Eingabe
Bus Size		Print	•

Figure 31: Integrating encoder using EDS

- ✓ The Select Module Type dialog opens.
- 2. Select the respective encoder type in the Catalog tab.

Enter Search Text fo	r Module Type	Clear Filters		Show Filters ≈
Catalog Number	Description	Vendor	Category	^
842E-S	Single Tum Encoder	Rockwell Autom	Specialty	
A3XX	Thermo Vision Camera	FLIR Systems	Specialty	
ACT350 Ether	Weight Transmitter	Mettler-Toledo	Communication	
ACT350-2P E	Weight Transmitter	Mettler-Toledo	Communication	
ACU	Air Conditioner	Hoffman Enclos	Specialty	
AFM60A-Eth/IP	AFM60A-Eth/IP	SICK AG	Encoder	
AFS60A-Eth/IP	AFS60A-Eth/IP	SICK AG	Encoder	
AX8	ThermoVision Camera	FLIR Systems	Specialty	
Bus Module	Bus Module EtherNet/IP Interfa	Advanced Ener	Communication	
CEP7-ETN	CEP7 EtherNet/IP	Rockwell Autom	Motor Overload	
CNB	FLOWSERVE IPS - 400/480V	Rockwell Autom	DPI to EtherNet	
CNB	FLOWSERVE IPS - 208/240V	Rockwell Autom	DPI to EtherNet	
CNB	FLOWSERVE IPS - 600V	Rockwell Autom	DPI to EtherNet	
DACS EtherN	Dialight EtherNet/IP Adapter	Dialight	Communication	

Figure 32: Select module

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

- AFS60A-Eth/IP for the AFS60 EtherNet/IP
- AFM60A-Eth/IP for the AFM60 EtherNet/IP
- 3. Click **OK** .
- ✓ The Module Properties [Modulname] dialog opens.

ieneral*	Connect	ion Mod	ule Info	Internet Protoc	ol Port Config	guration	Network		
Type: Vendor: Parent:	A S L	FM60A-Et ICK AG ocal	h/IP AFN	160A-Eth/IP					
Name: Descript	ion:	Encoder			^		ernet Address ) Private Network: ) IP Address: ) Host Name:	192.168.1. 192 . 168	▲ ▼ 8 . 1 . 11
Module Revisi Electro Conne	e Definitio ion: onic Keyir ections:	n 2.00 ng: Com Inpu	1 patible M t Only (1	odule 01): Fault Head	er + Positio				
					Change	]			

Figure 33: Entering module properties

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

-	lf you	want	to change	this inst	ance, cl	ick on C	hange
---	--------	------	-----------	-----------	----------	----------	-------

Module Definition*						Х
Revision: 2 v 001 - Electronic Keying: Compatible Module v						
Connections:						
Name		Size		Tag Su	ffix	
Input Only (103): Fault Header + Position Value + Velocity	Input: 12		Encoder:I1			
×	Output:	0	SINT	1	<none></none>	
Input Only (101): Fault Header + Position Value Input Only (102): Fault Header + Position Value + Velocity Listen Only (101): Fault Header + Position Value + Elags Listen Only (102): Fault Header + Position Value + Flags Listen Only (103): Fault Header + Position Value + Velocity Input Only (13): Position + Flags Input Only (3): Position + Flags Listen Only (3): Position Value + Listen Only (3): Position Value Listen Only (3): Position Value + Velocity Listen Only (3): Position Value + Velocity Input Only (101/WS): Fault Header + Position Value Input Only (101/WS): Fault Header + Position Value Input Only (102/WS): Fault Header + Position Value + Velocity Input Only (103/WS): Fault Header + Position Value + Velocity						
				ОК	Cancel Hel	p

Figure 34: Changing connections

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

Size	
12	SINT
0	Ĭ
	SINT
	DINT
	REAL

Figure 35: Changing data format

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

# **Checking communication**

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

Controller Organizer 🛛 👻 🕂 🗙		Module Properties: Local (AFM60A-I	th/IP 2.001)		
a 11	Scope: @AFM60_manual_ ~ Show: All Tags				
▲	News	ant Value	Farmer March	Chile	Data Tuna
Controller Tags	Name	value	rorce mask	Style	Data type
Controller Fault Handler	Local: 1:C	{	{}		AB:Embedded_Discre
Power-Up Handler	Local:1:1	{]	{}		AB:Embedded_Discre
Tasks	▶ Local:1:0	{]	{}		AB:Embedded_Discre
Main lask	▲ EncodenC	{]	{}		_0328:AFM60A_EthIP
Unscheduled	Encoder:C.Measuring_Units_per_Span	262144		Decimal	DINT
A 🛁 Motion Groups	Encoder:C.Total_Measuring_Range	1073741824		Decimal	DINT
Ungrouped Axes	Encoder:C.Direction_Counting_Toggle	C		Decimal	BOOL
The Logical Model	Encoder:C.Scaling_Function_Control_SFC	C		Decimal	BOOL
🔺 🚄 I/O Configuration	Encoder:C.Endless_shaft_functionality	C		Decimal	BOOL
PointlO     Ol 17(0   10(0) A DD10 A D10(0)	Encoder:C.Number_of_rotations_Nominator	2048		Decimal	DINT
Embedded I/O	Encoder:C.Number_of_rotations_Divisor	1		Decimal	DINT
[1] Embedded Discrete_IO	Encoder:C.Velocity_Format	7951		Decimal	INT
Expansion I/O, 0 Modules	▲ EncodenI1	{]	{}		_0328:AFM60A_EthIP
A The Ethernet	Encoder:11.ConnectionFaulted	C		Decimal	BOOL
AFM60A-Eth/IP Encoder	▲ Encoder:I1.Data	{]	{}	Decimal	DINT[3]
	Encoder:I1.Data[0]	C		Decimal	DINT
	Encoder:I1.Data[1]	1037225780		Decimal	DINT
	Encoder:11.Data[2]	252		Decimal	DINT

Figure 36: Checking communication

- 1. In Controller Organizer, open the Controller test  $\rightarrow$  Controller Tags folder.
- 2. In Controller Tags, in the Name column, open the AFx60\_EIP:I1 → AFx60\_EIP:I1.Data item.

Displayed data in the example:

- AFx60\_EIP:I1.Data[0]: Fault header: 0
- AFx60\_EIP:I1.Data[1]: Position: 1037225780
- AFx60\_EIP:I1.Data[2]: Speed: 252

# 5.4.3 Configuration

▲ Encoder:C	{}	{}		_0328:AFM60A_EthIP
Encoder:C.Measuring_Units_per_Span	262144		Decimal	DINT
Encoder:C.Total_Measuring_Range	1073741824		Decimal	DINT
Encoder:C.Direction_Counting_Toggle	0		Decimal	BOOL
Encoder:C.Scaling_Function_Control_SFC	0		Decimal	BOOL
Encoder:C.Endless_shaft_functionality	0		Decimal	BOOL
Encoder:C.Number_of_rotations_Nominator	2048		Decimal	DINT
Encoder:C.Number_of_rotations_Divisor	1		Decimal	DINT
Encoder:C.Velocity_Format	7951		Decimal	INT

Figure 37: Encoder configuration

- 1. In Controller Tags, in the Name column, open the AFx60\_EIP:C item.
- 2. Enter the parameters of the encoder (see "Parameterizable functions", page 36).

# 5.5 Installing the ladder routine

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

Use the following ladder routine depending on the selected instance:

- SickAFx\_A101WS\_A103WS\_FB\_Enc1\_GetSet.L5X for instances 101WS and 103WS
- SickAFx\_A102WS\_FB\_Enc1\_GetSet.L5X for instance 102WS

Requirements for the installation of the ladder routine are:

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

Cogis Designer - APM60_manual_1 (1769-L135RM-8818	saula	8 X
FILE EDIT VIEW SEARCH LOGIC COMMUNICAT	DORS TOOLS WINDOW HELP	
14 🖆 🐸 🖶 🗴 8 5 7 9 9		
II FUN		
PBN: A8_ETHP-11192.1		
III LO Offline B. No Forces	F. WEBS B. ST PRVINTES Associt sately waters or intercence speculatin cargor cargores intercepter revise, resent sep- intercence and the second sately waters or intercence speculatin cargore cargorement intercepter revise.	_
Controller Organizer V 9 X	Controller Tags - ATM30, namual, literatroller) 3 Module Properties Local (ATM60A-Cth/P 2.001) ×	•
Control (2000) Annual (2000)     Control (2000) Annual (2000) Annual (2000)     Control (2000)	Source Touristic Marke House Processing Market House       Touristic Market House Processing Market House       Variation Status       Variation Status       Designer       Designe	
Madar Diffeel Tap Constraint Designing Bada Office Madar Fast C Southern Organica Distribution Organica Distribution Distri		

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

• An executable project with the absolute encoder in the RSLogix 5000.

The following steps must be performed:

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

# NOTE

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

### 5.5.1 Import of the ladder routine

i

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

💰 Logix Designe	r - AFM60_manual_	1 [1769-L18ERM	-BB18	8 31.11]*									
FILE EDIT VI	EW SEARCH L	OGIC COMM	UNICA	TIONS TOOLS WINDOW	HELP								
5 🖕 🗎 🗧	808	2 6		v 🐤 🍫	, T	h. 6 h. #	1 📸 👘 (	2 (2					
	` <b>∥</b> ′ ⊧	Path: AB_ETHIP-	1\192.	. 168. 1. 3*		옥 랆 8	• H		⊢ -( )(U)(L)				
III Energy Storage	Offline	🗄 🗸 No F	orces	▶ No Edits	₽.		<ul> <li>Favo</li> </ul>	rites Add-On Sa	ety Alarms Bit	Timer/Counter	Input/Output	Compare	Compute/Math
Controller Organiz	ter	-	ąх	Controller Tags - AFM60	manual	1(controller) $\times$							
<u>ت</u>				Scope: Scope: Scope:	v s	now: Al Tags							
🔺 🚄 Controller	AFM60_manual_1			Name			- 01 -	Value	Eorce Mark	Style		Data Tune	
Control	ller Tags			h Local:1:C			-01-	value	1 I I I I I I I I I I I I I I I I I I I	/ J		AB-Embed	Ided Discre
Power	-Up Handler			b Local:1:1					3	1.1		AB:Embed	Ided Discre
🔺 📹 Tasks				b Local:1:0					3	1.1		AB:Embed	ded Discre
🔺 🖓 MainTa	ask			4 Encoderil					3	()		0328-AFM	1604 Ethip
	Add	•		New Routine	Fai	ulted			0	Decimal		BOOL	
🖌 📹 Motio 💡	Cut	Street	0	New Local Teg Stra	-14/					{} Decimal		DINT[3]	
n n n 🖷	Copy	Stra+C	×	New Parameter	··· -				0	Decimal		DINT	
P ■ Assets The Logical ■ ■ Pool ■ ■ Pool ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■	Paste	Strg+V		Second Second	-			10737418	23	Decimal		DINT	
	Delete	Entf	-	P Encoder:11.Dat	121				0	Decimal		DINT	
	Verify Cross Reference Browse Logic Find in Logical O Online Edits Print Export Program Properties	Strg+E Strg+L irganizer • • • • • • •	Jal_1										
Description													
Status	Scheduled												
Number of Routin	nes 1 MainPautina												
Fault Routine	MainKoutine												
Max Scan													
Last Scan													
Parent Schoduled In	MainTack												
scheduled in	iviain lask			-									

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

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

- 2. Select the Sick-AFx\_A102WS\_FB\_Enc1\_GetSet.L5X file via Add and click on Import Routine.
- ✓ The Import Configuration dialog opens.



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

ort Content:			
- 🗐 Programs	Configure Routin	e Properties	
MainProgram	Import Name:	SickAFx_A101WS_A103WS_FB_Enc1_GetSet	
References	Operation:	Create 🗸 🗋	
Add-On Inst	ruction	References will be imported as configured in the References folders	
Data Types     Other Compo	Final Name:	SickAFx_A101WS_A103WS_FB_Enc: V Properties	
- Errors/Warnings	Description:	^	
		~	
	Type:	🗏 Ladder Diagram	
	In Program:	🔓 MainProgram 🗸 🗸	
	Number of Rungs	: 3	

Figure 40: Import Configuration dialog

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



Figure 41: Encoder selection

	elect the Use Existing option.	column,	In the Operation	7.
--	--------------------------------	---------	------------------	----

🎽 Find:	~ 🏭 🐴	Find/Replace			
Find Within: Final Name					
ort Content:		4			
- Programs	Configure Compone	nt References			
MainProgram	Import Name	Operation	Final Name 🔺	Class Name	
References	SickAFX_Enc1_xy	z Use Existing	Encoder 🗸 🗸	Module	
Tags					
11 Add-On Instruction:					
Other Components					
Errors/Warnings					
	t.				
>					

Figure 42: Selection of the operation for the component

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

Content	ame	Con	figu	re Tag References							
- Ь М	ainProgram			Import Name	Operation	P	Final Name	۶	Usage	Alias For	Data Type
harr E	SickAFx_A101W5_A103		5	Local:1:C	Use Existing		Local:1:C		Local		AB:Embedd
	Taos		1	Local:1:1	Use Existing		Local:1:1 2		Local		AB:Embedd
			1	Local:1:0	Use Existing		Local:1:0		Local		AB:Embedd
		<b>T</b>	8	SickAFX_Enc1_xyz:11	Undefined		Encoder:11		Local		
	Other Components		1	SickAFxWS_Enc1_AOI	Create	D	Foter Name Filter	-	Tage		SICK_AFXE
Errors	/Warnings			SickAFxWS_Enc1_Cyclic	Create		1		raga		BOOL
			1	SickAFxWS_Enc1_GetData	Create		Name <u>==</u> Data Type		Usage	Descripti ^	SICK_AFX6
			8	SickAFxWS_Enc1_Init_Get	Create	D	Encoder:I1 _0328:AFN	160	<controller></controller>		BOOL
			F	SickAFxWS Enc1 msgGet	Create	D	■ Local:1:C 3. AB:Embed	led.	<controller></controller>	•	MESSAGE
			Ē	SickAFxWS Enc1 msgSet	Create		■ Local:1:1 AB:Embe	032	8:AFM60A_E	thIP_5DA4C79D	SSAGE
			F	SickAFxWS Enc1 SetData	Create	D	Local:1:0 AB:Embed	led.	<controller></controller>		SICK AFXE
			F	SickAFxWS Enc1 Timer G	Create	n				~	TIMER
						-	Show controller tags				
		<					-				
							Show Main Program tags				
							Show parameters from other program:				
							<none></none>		~		
											_

Figure 43: Selection of the tag of the instance used

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

Find Within: Final Name		~	Find/Replace	П						
rt Content:				$\checkmark$						
Programs	Con	figur	e Tag References							
MainProgram			Import Name	Operation	1	Final Name	- 8	Usage	Alias For	Data Type
SICKAFX_AIUIWS_AIU:		•	Local:1:C	Use Existing		Local:1:C		Local		AB:Embedd
Tags		•	Local:1:1	Use Existing		Local:1:1		Local		AB:Embedd
		•	Local:1:0	Use Existing		Local:1:0		Local		AB:Embedd
	•	•	SickAFX_Enc1_xyz:I1	Use Existi 🗸		Encoder:11		Local		_0328:AFM
Other Components		•	SickAFxWS_Enc1_AOI	Use Existing		SickAFxWS_Enc1_AOI		Local		SICK_AFX
_o Errors/Warnings		•	SickAFxWS_Enc1_Cyclic	Create		SickAFxWS_Enc1_Cyclic_GetSet		Local		BOOL
		•	SickAFxWS_Enc1_GetData	Create		SickAFxWS_Enc1_GetData		Local		SICK_AFX
		•	SickAFxWS_Enc1_Init_Get	Create		SickAFxWS_Enc1_Init_GetSet		Local		BOOL
		•	SickAFxWS_Enc1_msgGet	Create		SickAFxWS_Enc1_msgGetData		Local		MESSAGE
		•	SickAFxWS_Enc1_msgSet	Create		SickAFxWS_Enc1_msgSetData		Local		MESSAGE
		•	SickAFxWS_Enc1_SetData	Create		SickAFxWS_Enc1_SetData		Local		SICK_AFX
		•	SickAFxWS_Enc1_Timer_G	Create		SickAFxWS_Enc1_Timer_GetSet		Local		TIMER
	٢									

Figure 44: Selection of the operation for the tag references

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

2 2	Find	ı: []	l.		~	<b>8</b> 6	Find/Replace			_				
	Find	Wit	hin: Final Name							1 1.				
port Cor	ntent									V				
- 🗐 P	rogra	ms		Cont	figure	Tag	References				-			
	Ma	inPr	ogram			port	Name	Operation	6	Final Name	- J.F	Usage	Alias For	Data Type
	<b>H</b>	Sie	ckAFx_A101W5_A103	H	F	Loc	al:1:C	Use Existing		Local:1:C		Local		AB:Embedd
3		1	C Tage		F	Loc	al:1:1	Use Existing		Local:1:1		Local	_	AB:Embedd
			-1+t Add-On Instructions			loc	al:1:0	Use Existing		Local:10		Local	_	AB Embedd
1			- 101 Data Types	-		Sid	AFX Encl world	Use Existing		Encoder 11		Local		0328 AFM
		Ι.,	Other Components	-	F	Sid	AFxWS Enc1 AOI	Create	h	SickAExWS_Enc1_AOL		Local		SICK AEX6
- 🔽 E	rrors	Wa	rnings		1	Sid	AFxWS Encl Cyclic	Create		SickAFxWS Enc1 Cyclic GetSet		Local		BOOL
				Н		Sid	AFxWS_Enc1_GetData	Create		SickAFxWS_Enc1_GetData		Local		SICK AFX6
				Н	6	Sid	AFxWS Enc1 Init Get	Create	R	SickAFxWS Enc1 Init GetSet	1	Local		BOOL
				H		Sid	AEvWS Enc1 megGet	Create	H	SickAFyWS_Enc1_magGatData	1	Local		MESSAGE
				Н		Sid	AFvWS Enc1 megSet	Create		Sick AFvW/S Enc1 megSetData	-	Local		MESSAGE
				Н	5	Cial	AEvWC Encl CotData	Create	H	SideAEvW/S_Enc1_SatData	-	Local		SICK AEVE
				Н	8	Cial	AFWS_DICI_Jerbala	Create	H	Cicle A Full C Fac 1 Tenas Cat Cat	-	Local	-	TIMED
				$\vdash$		510	CAPXWS_Elici_lillier_G	Cleare		SickArxW3_Enc1_nimer_delSet		LUCA		TIMEN
	_			<										
			Import Name	Ope	ration	9	Final Name	- F						
		8	Local:1:C	Use	Existing		Local:1:C							
		8	Local:1:1	Use	Existing		Local:1:1							
		8	Local I.U	Use	Existing		Local: 1:0							
	-	-	SICKAPA_Enc1_xyz.11	Use	Existing		Encoder.II			1				
	-		SickArxWS_Enc1_AUI	Crea	se	븡	SickArxWS_Enc2_AUI	C-11						
	-	0	SickArXWS_Enc1_Cyclic	Crea	se	븡	SickArXWS_Enc2_Cyclc_det	Set						
	-		SickArXWS_Enc1_decoata	Creat	ac .	븑	SickAPAWS_Enc2_GetData							
_	÷	0	SickAFrWS_Enc1_Int_Get.	Crea	ate .	분	SickAFyWS_Enc2_me_Get3et							
	•	8	SickAFxWS_Enc1_msgGet.	Crea	te	H	SickAFrWS_Enc2_msgGetDat	a						
	-	1	SickAFxWS_Enc1_SetData	Crea	te	ň	SickAFxWS_Enc2_SetData						_	
1	-	8	Ciel AEvWC Engl Tager G	Con	10	F	Cial ADVING Engl Timer Cat			2		OK	Cancel	Holo

Figure 45: Changing the tag names

- 13. Click OK .
- $\checkmark$  The ladder routine is imported.

roller Organizer 🚽 🖛 🗙	Controller Tags - AFM60_manual_1(controller) ×	Andule Properties: Local	(ASMAGAN SHE (ID 2 000)	an a		
Tentralian APAKA annual 1			he most crisis 2001)	MainProgram - SickAFx_A	101WS_A103WS_FB_Enc1_GetSet	
Controller AFA460 menual 1	Scope: [E0AFM60_manual_ V Show: Al Tags					V Exter Name Filter
controller Anwoo_manual_1	Name	Value	Force Mask	* Style	Data Type	Description
Controller Tags	Elocal:1:C	101-	(_)	()	AB:Embedded Discre	o composition
Power-Up Handler	h Localita		( )	()	AB-Embedded Discret	
fasks	h localdi		1-1	()	AR/Embedded_Discre	
🔁 MainTask	P Cocarito		1-7	()	AB:Embedded_Discre	
A 5 MainProgram	P Encoderii		()	()	_0328:AFMBUA_EthiP	
Parameters and Local Tags In Main Parating	SickAfxWS_Enc2_AOI		{}	{}	SICK_AFX60_A101WS	
SickAEx A101WS A103WS FB Enc1 Ge	SickAFxWS_Enc2_Cyclic_GetSet		0	Decimal	BOOL	
Unscheduled	SickAFxWS_Enc2_GetData		()	()	SICK_AFX60_WS_DATA	Data structure for AFS60 / AFM60 WebSen
Motion Groups	SickAFxWS_Enc2_Init_GetSet		0	Decimal	BOOL	
Ungrouped Axes	SickAFxWS_Enc2_msgGetData		{}	{}	MESSAGE	
Assets	SickAFxWS_Enc2_msgSetData		{}	{}	MESSAGE	
A SICK AFX60 A101WS A103WS A01	SickAFxWS_Enc2_SetData		{}	{}	SICK_AFX60_WS_DATA	Data structure for AFS60 / AFM60 WebSen
Parameters and Local Tags	SickAFxWS_Enc1_Timer_GetSet		{}	()	TIMER	
Istings       Image: Isting isting istication istin a streation istication istication istication istication istica						

Figure 46: Project structure after import

# 5.5.2 Integration as SubRoutine in MainRoutine

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

🧳 Logix Designer - AFM60_manual_1 (1769-L18ERM-8818	3 31.11J*	- 0 ×
FILE EDIT VIEW SEARCH LOGIC COMMUNICAT	mons tools withdow HELP	
200 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	<b>&gt; &gt; // b/k B/k 2 (G // 6 (G // 7)</b>	
Path: A8_ETHIP-1(2)2.	158.13" • • • • • • • • • • • • • • • • • • •	
III Energy Storage Offline II , No Forces	F. No Edits B. 4 Feworites Add-On Alianne Bit Timer/Counter Input/Output Compute/Math. MoveLogical FileMisic. File/Shift Sequencer P.	
Controller Organizer • 0 ×	Hainfrogram - MainRostine* ×	•
8 11	4. 4. 11 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.	
Controller Tags		^
Controller Fault Handler		
A Casks	(n)	
A C MainTask		
<ul> <li>Parameters and Local Tags</li> </ul>		
E MainRoutine		
Unscheduled		
Generation Groups		
Image of the second		
Logical Model		
A B PointiO		
[0] 1769-L18ERM-BB1B AFM60_manual_1		
[1] Embedded Discrete_IO		
Expansion I/O, 0 Modules		
1769-LISERM-8B1B AFM60_manual_1		
AFM60A-Eth/IP Encoder		
< >		
Description		
Program MainProgram		
Number er Kungs		
The Controller Department Polytomizer	4	· · ·
Finans Search Results & Watch		

Figure 47: MainRoutine without SubRoutine

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

![](_page_61_Picture_2.jpeg)

Figure 48: MainRoutine with SubRoutine

# 5.5.3 Using the SubRoutine

1. Switch the controller to online mode.

Logix Designer - AFM60_manual_1 [1769-L18ERM-BB1B	31.11]*			
FILE EDIT VIEW SEARCH LOGIC COMMUNICAT		WINDOW HEIP		
15 🖕 🖶 🖶 🖉 히 기 연 📃		> > ≠ ■ 10 h B b ± 0 % C C		
Program Mode				
Controller OK Paul. Ab_EliPuP*1/192.	100.1.3		LEC JOR JOR HET SUR THE HER OTE OTE SPR SPP ENERT EET	HP1 HOP F
No Forces	. ►_ N	Edits 🔒 Alarms Bit Timer/C	counter Input/Output Compare Compute/Math Move/Logical File/Misc	File/Shift Sequencer Program Control Fo
Controller Organizer 🗢 🤻 🛪	🗎 MainPro	gram - MainRoutine 🔋 MainProgram - SickAFx_A101WS_A103WS_FB_Enc1	_GetSet ×	
	1 en e		(E) E and Eg al. + and	
▲ S Controller AFM60 manual 1	2.111.11	a second as an as as as as as as as		
Controller Tags	545	SickAFxWS_Enc2_Init_GetSet	SickAFxWS_Enc2_Cyclic_GetSet SickAFxWS_Enc2	AOLManualGetData SickAFxWS_Enc2_Init_GetSet
Controller Fault Handler	0			
Power-Up Handler		Part A Full (P. Fand), Curlin, Califord, Part A Full (P. Fand), Tamas, Califord (M)		700
🔺 🚍 Tasks	10			Timer_SickAFxWS_Enc1_Timer_GetSet (EN)
🔺 🔿 MainTask				Preset 100 + (DN)-
A L MainProgram				Accum
Parameters and Local Tags				
SickAFx A101WS A103WS FR Fpc1 Ge		SickAFxWS_Enc1_Timer_GetSet.DN	SICK_AFX60_A101WS_A103WS_A01	SickAFxWS_Enc2_AOI.ManualGetData
Unscheduled	2		SICK_AFX60_A101WS_A103WSSickAFXWS_Enc2_	(U)
🔺 🚄 Motion Groups			msgGetData SickAFxWS_Enc2_msgGetD	ata
Ungrouped Axes			msgSetData SickAFxWS_Enc2_msgSetD	ata
Assets			SetData SickAFxWS_Enc2_GetD	ata
b. Logical Model				
<ul> <li>W Configuration</li> <li>Configuration</li> </ul>				
FU IOI 1759-I 18EPM-BB1B AEM60 manual 1	(End)			
A S Embedded I/O	0.000			
[1] Embedded Discrete_IO				
Expansion I/O, 0 Modules				
4 💑 Ethernet				
1769-L18ERM-BB1B AFM60_manual_1				
APMBUA-Eth/IP Encoder				
< >>				
Type Ladder Diagram				
Description				
MainProgram				
Number of Kungs 5				
< >				
1= Controller Organizer	4=			

Figure 49: Imported SickAFx ladder routine in online mode

2. In the MainProgram, switch to SickAFx\_A101WS\_A103WS\_FB\_Enc1\_GetSet .

A Loois Decision	AEX460 manual 1 11760 11050M 0010	24 4416			
S rus con un					
tu 🖆 🗎 🖨	× ゴ	ېر څ څ 🗸 🗸			
Program Mode Controller OK Energy Storage OF	Path: AB_ETHIP-1\192.	168. 1.3*	× 25 Β ← → → →	UMP LEL JSR JOR RET SER THE HCR UID UIE SFR SFP MAHT EET AFI HE HERCounter Inpublicutput Compare ComputerNath MoveR.ogical FileNtisc. FileSh	P b ift Sequencer Program Control Fo
Cantraller Orazaira					
@ 911		e ex es iv is is i	< to be to be to be B G Fri to	TT TT an Pa do a con	
4 Controller A	EM60 manual 1				
Q Controlle	er Tags	SickAFxWS_Enc	2_Init_GetSet	SickAFxWS_Enc2_Cyclic_GetSet SickAFxWS_Enc2_	AOI.ManualGetData SickAFxWS_Enc2_Init_GetSet
Controlle	er Fault Handler		X Cut Instruction Stra+X	<ul> <li>(L)</li> </ul>	(U)
Power-U	lp Handler		Convinstruction Stra+C		
🖌 📹 Tasks	A	SICKAPXWS_EI	C Danks (Sec. 1)		TON
🔺 🔿 MainTas	k	1	11 Pasce Sugev		Preset 100 - (DN)-
🔺 🔓 Main	Program		Delete Instruction Entf		Accum 0+
🥥 P	arameters and Local Tags		Add Ladder Element Alt+Einfg		
10 N	AsinRoutine		Edit Main Operand Description Stro+D		
🗧 S	ickAFx_A101WS_A103WS_FB_Enc1_Ge	SIGKAFXWS_EI		SICK_AFX60_A101WS_A103WS_A01	SickAFXWS_Enc2_AOI/VanualGetData
Unsched	luled	2.	Save Instruction Defaults	inputAssembly_101WS_103WS Encoder.II.Dr	ata (0)
🔺 🚄 Motion Grou	ups		Clear Instruction Defaults	msgGetData SickAFxWS_Enc2_msgGetData	sta
Ungroup	and Axes		Trank Dit Cha T	msgSetData SickAFxWS_Enc2_msgSetData SickAFxWS_Enc2_msgSetData	sta
Assets			loggie bit Strg+1	SetData SickAFXWS Enc2 SetDa	sta
b. Logical Mod	fel		Force On		
🔺 🖳 VO Configu	ration				
A PointIO			Force Off		
ES [0] 11	769-L18ERM-BB1B AFM60_manual_1	(End)	Remove Force		
🖌 🖳 Emb	edded I/O				
19 1	I] Embedded Discrete_IO		Go To Strg+G		
Expan	nsion (/U, U Modules		Instruction Help		
A ST Etnemet	1105014 0010 451460				
1709	LISERM-BEIE AFMOU_manual_1				
Arm	book-Etry in Encoder				
<	>				
lype	Ladder Diagram				
Description					
Program Number of Purson	MainProgram				
Number of Kungs	3				
<	>				
1 Controller Organ	izer by Logical Organizer	-			

Figure 50: Initialization and start of the SubRoutine

- 3. In the context menu of SickAFxWS\_Enc1\_Init\_GetSet, activate the Toggle Bit command.
- ✓ This completes the integration and parameterization of the encoder can be performed both on the controller side and via the web server.

## 5.5.4 Reading out and changing parameters of the encoder

Under **Controller Tags**, the parameters of the encoder can be read out in the **Sick-AFxWS\_Enc1\_GetData** node.

Logix Designer - AFM60_manual_1 [1769-L18ERM-BB1B 31	.11]*			
	NS IOOLS WINDOW HELP			
	·····································	cá cár		
Controller OK Path: AB_ETHIP-1\192.168	8.1.3* 🧏 🛔 👌 🛏	He He JHP LBL JSR JXR	RET SBR TND MCR UID	UIE SFR SFP EVENT EOT P
Energy Storage OK	No Edits	ms Bit Timer/Counter Input/Ou	tput Compare Compute/Mat	Move/Logical File/Misc.
Controller Organizer	MainProgram MainPoutine MainProgram SickAEv A101WS	A103W/S EB Encl GatSat	oller Tage - AEM60 manual 1/c	ontroller) ×
are		Nows_rb_ener_detser	oner rags - Ar woo_mandal_r(	
✓ Controller AFM60 manual 1	Scope: [g:AFM60_manual_ V] Show: Air rags		and the second	line of the
Controller Tags	Name III +	Value • Force Ma	sk 🔹 Style	Data Type
🛑 Controller Fault Handler	SickAFxWS_Enc2_AOI	{}	{}	SICK_AFX60_A101WS
Power-Up Handler	SickAFxWS_Enc2_Cyclic_GetSet	0	Decimal	BOOL
A Carlos	<ul> <li>SickAFxWS_Enc2_GetData</li> </ul>	{}	{}	SICK_AFX60_WS_DAT
A to Main lask	SickAFxWS_Enc2_GetData.Acceleration_Format	2064	Decimal	INT
Parameters and Local Tags	SickAFxWS Enc2 GetData.Acceleration HighLimit	1073741823	Decimal	DINT
D MainRoutine	SickAExWS Enc2 GetData Acceleration Low imit	-1073741823	Decimal	DINT
SickAFx_A101WS_A103WS_FB_Enc1_Ge	SickAFr/WS Enc2 GetData CMR	1073741824	Decimal	DINT
Unscheduled	N Sick/AFA//S Ene2 GetData CNR D	1	Desimal	DINT
General Annual Annua	<ul> <li>SickAFAWS_Enic2_GetData.CNR_D</li> <li>SickAFAWS_Enic2_GetData.CNR_D</li> </ul>	2040	Decimal	DINT
D Assets	V SICKAPXWS_Enc2_GetData.CINK_IN	2048	Decimal	
The Logical Model	SickAFxWS_Enc2_GetData.CPR	262144	Decimal	DINT
🔺 🚅 I/O Configuration	SickAFxWS_Enc2_GetData.CS	0	Decimal	SINT
A E PointlO	SickAFxWS_Enc2_GetData.ESF	0	Decimal	SINT
[0] 1769-L18ERM-BB1B AFM60_manual_1	SickAFxWS_Enc2_GetData.DirectionChange_Limit	1000000	Decimal	DINT
Ill Embedded //O	SickAFxWS_Enc2_GetData.MotionTime_Limit_seconds	630720000	Decimal	DINT
Expansion I/O. 0 Modules	SickAFxWS_Enc2_GetData.PowerTime_Limit_seconds	630720000	Decimal	DINT
▲ 윪 Ethernet	SickAFxWS_Enc2_GetData.StartsCCW_Limit	1000000	Decimal	DINT
1769-L18ERM-BB1B AFM60_manual_1	SickAFxWS_Enc2_GetData.StartsCW_Limit	1000000	Decimal	DINT
AFM60A-Eth/IP Encoder	SickAFxWS_Enc2_GetData.Position_HighLimit	1073741823	Decimal	DINT
< >	SickAFxWS Enc2 GetData.Position LowLimit	0	Decimal	DINT
	SickAExWS Enc2 GetData.PresetValue	0	Decimal	DINT
	SickAFrWS Enc2 GetData Scaling	0	Decimal	SINT
	SickAEvWS Enc2 GetData SlaveSignOfLife	1280	Decimal	DINT
	SickAEvWS Enc2 GetData Temperature Format	4609	Decimal	INIT
	SickAFAWS_Enc2_GetData.heimperature_ronnat	7051	Decimal	INT
	V SICKAPAWS_Enc2_GetData.velocity_Pormat	12000	Decimal	DUIT
	SickAFxWS_Enc2_GetData.Velocity_HighLimit	12000	Decimal	DINT
	SickAFxWS_Enc2_GetData.Velocity_LowLimit	-12000	Decimal	DINT
I	SickAFxWS_Enc2_GetData.xErrorCode	0	Decimal	DINT
	SickAFxWS_Enc2_GetData.xMsgRecordArray	{}	{} Decimal	SINT[4]
	SickAExWS Enc2 GetData.SerialNo	184614935	Decimal	DINT
	SickAFxWS_Enc2_Init_GetSet	1	Decimal	BOOL
	<ul> <li>SickAFxWS_Enc2_msgGetData</li> </ul>	{}	<b>{}</b>	MESSAGE
	SickAFxWS_Enc2_msgSetData	{}	()	MESSAGE
< >	SickAFxWS_Enc2_SetData	{}	<i>()</i>	SICK_AFX60_WS_DATA
T= Controller Organizer Logical Organizer	Monitor Tags / Edit Tags /		<	

Figure 51: Reading out the parameters under GetData

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

				× ୭ ୭ ♬ <u>b</u> № № 8 ± 0 %	Ci Ci		
JCK				68.1.3* % Å # 4		ISR JOR RET SER THD MCR UID	UIE SFR SFP ENERT EOT AFI
ensor Intelligence.				h hardta 2 d h Ab	arms Bit Timer/Counter	Input/Output Compare Compute/Ma	th Move/Lopical File/Misc. Fi
				MainProgram - MainRoutine MainProgram - SirkAFy A101WS	A103WS FR Encl GetSet	O Controller Tags - AFM60 manual 1	(controller) ×
				Correct Contraction of the Correct All Trans			
				Scope: [[::Armou_manual_ V] Show: [Arrays	1. 11. Sec. 12.	and the second second	
Home Parameterization Diagnostics 1	Tools			Name III	▲ Value 🔹 F	orce Mask 🔹 Style	Data Type
Overview Units Preset				SickAFxWS_Enc2_AOI	{}	{}	SICK_AFX60_A101WS
				SickAFxWS_Enc2_Cyclic_GetSet	0	Decimal	BOOL
verview				SickAFxWS_Enc2_GetData	{}	{}	SICK_AFX60_WS_DATA
				SickAFxWS_Enc2_GetData.Acceleration_Format	2064	Decimal	INT
	Current	Default I	) hex	SickAFxWS_Enc2_GetData.Acceleration_HighLimit	1073741823	Decimal	DINT
Code sequence	CW	CW I	0x0C	SickAFxWS_Enc2_GetData.Acceleration_LowLimit	-1073741823	Decimal	DINT
Preset	0	0	0×13	SickAFxWS_Enc2_GetData.CMR	1073741824	Decimal	DINT
Lower limit for the position	123456	0	0x16	SickAExWS Enc2 GetData.CNR D	1	Decimal	DINT
Upper limit for the position	1073741823	1073741823	0x17	SickAEvWS Enc2 GetData CNR N	2048	Decimal	DINT
Lower limit for the velocity	-9000	-9000	×18	N SickAEdWS Enc2 GetData CPP	262144	Decimal	DINT
Lower limit for the acceleration	-1073741823	-1073741823		<ul> <li>Sicker Wisgenergereinergereiner Ge</li> <li>Sicker Wisgenergerein Erendergereinergere</li></ul>		Desired	CINT
Upper limit for the acceleration	1073741823	1073741823	w21	P SICKAPXWS_Enc2_GetData.CS	0	Decimal	SINT
Velocity unit	1070741020	rpm	19	SickAlxWS_Enc2_GetData.ESF	0	Decimal	SINT
Acceleration unit	cpmss	comss	DALE	SickAFxWS_Enc2_GetData.DirectionChange_Limit	123	Decimal	DINT
Temperature unit	°C	°C	0xe5	SickAFxWS_Enc2_GetData.MotionTime_Limit_seconds	630720000	Decimal	DINT
Limit for the motion time in hours	175200	175200	)x86	SickAFxWS_Enc2_GetData.PowerTime_Limit_seconds	630720000	Decimal	DINT
Limit for the operating time in hours	175200	175200	DXF	SickAFxWS_Enc2_GetData.StartsCCW_Limit	1000000	Decimal	DINT
Limit number of changes in the direction of rotation	123	1000000	0x88	SickAFxWS_Enc2_GetData.StartsCW_Limit	1000000	Decimal	DINT
Limit number of clockwise starts	1000000	1000000	98x0	SickAFxWS_Enc2_GetData.Position_HighLimit	1073741823	Decimal	DINT
Limit number of counterclockwise starts	1000000	1000000	0x8A	SickAFxWS_Enc2_GetData.Position_LowLimit	123456	Decimal	DINT
Scaling	Off	Off	Dx0E	SickAFxWS_Enc2_GetData.PresetValue	0	Decimal	DINT
Round axis functionality	Off	Off I	ix7D	SickAFxWS Enc2 GetData.Scaling	0	Decimal	SINT
				b SickAEvWS Enc2 GetData SlaveSignOfLife	1280	Decimal	DINT
				N Sick/AE-WS Enc2 GetData Temperature Format	4609	Decimal	INT
				Sight Full (Card Card Card Mala Mala and Card Card Card Card Card Card Card Car	7051	Desimal	INIT
				CLASSING C. A CUB CALL A LIGHT A	1001	Decima	Date:
			-	SICKAPAWS_Enc2_GetOata.velocity_PiighCimit	12.54	Decimal	Direi
			$\neg$	P SICKAPXWS_Encz_GetUata.Velocity_LowLimit	-12000	Decimal	DINT
				SickAFxWS_Enc2_GetDataxErrorCode	0	Decimal	DINT
				SickAFxWS_Enc2_GetData.xMsgRecordArray	()	{} Decimal	SINT[4]
				SickAFxWS_Enc2_GetData.SerialNo	184614935	Decimal	DINT
				SickAFxWS_Enc2_Init_GetSet	1	Decimal	BOOL
				SickAFxWS_Enc2_msgGetData	{}	{}	MESSAGE
				SickAFxWS_Enc2_msgSetData	{}	{}	MESSAGE
				SickAFxWS_Enc2_SetData	()	{}	SICK_AFX60_WS_DATA
				A Monitor Tags / Edit Tags /		<	

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

# NOTE

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

i

💰 Logix Designer - AFM60_manual_1 [1769-L18ERM-BB1B	31.11]*			
FILE EDIT VIEW SEARCH LOGIC COMMUNICAT	IONS TOOLS WINDOW HELP			
ち 🖕 🖶 😸 🗇 ብ 🗇 🔍 🔛		66		
Program Mode				
Controller OK Path: AB_ETHIP-1(192.)	108.1.3* 後春日 4	I JHP LBL JSR JXR	RET SBR TND MCR UID	UIE SFR SFP EVENT EOT I
I/O OK Rem Prog I No Forces	No Edits it Ala	ms bit Timer/Counter Inpul/Ou	tput Compare Compute/Mati	n move/Logical File/Misc.
Controller Organizer 🗸 🕈 🗙	MainProgram - MainRoutine 📙 MainProgram - SickAFx_A101WS_	A103WS_FB_Enc1_GetSet 🗸 Contr	oller Tags - AFM60_manual_1(c	ontroller) ×
0 H	Scope: @AFM60_manual_ ~ Show: AI Tags			
Controller AFM60_manual_1	Name	Value Force Ma	sk 🗢 Style	Data Type
Controller Tags	SickAFxWS_Enc2_AOI	{}	{}	SICK_AFX60_A101WS
Power-Up Handler	SickAFxWS Enc2 Cyclic GetSet	0	Decimal	BOOL
🔺 🖳 Tasks	SickAExWS Enc2 GetData	()	()	SICK AFX60 WS DAT
🔺 🔅 MainTask	SickAEvWS Enc2 Init GetSet	1	Decimal	BOOL
MainProgram     A Basemeters and Local Tags	b SickAEvWS Enc2 mrcGetData	1.1	1.1	MESSAGE
MainRoutine	<ul> <li>SickAEdV/S Enc2 mcsSctData</li> </ul>	()	()	MESSAGE
SickAFx_A101WS_A103WS_FB_Enc1_Ge	SickArtwo_Ence_msgbetbata	1	11	SICK AEVED INC DAT
iii Unscheduled	SickArxws_Enc2_SetData	[]	1007	SICK_AFX00_WS_DAT
Motion Groups	V SICKAPXWS_Enc2_SetData.Acceleration_Pormat	2004	Decimal	DINT
Ongrouped Axes	SickAFxWS_Enc2_SetData.Acceleration_HighLimit	10/3/41823	Decimal	DINI
The Logical Model	SickAFxWS_Enc2_SetData.Acceleration_LowLimit	-1073741823	Decimal	DINT
🔺 🚅 I/O Configuration	SickAFxWS_Enc2_SetData.CMR	1073741824	Decimal	DINT
A PointlO	SickAFxWS_Enc2_SetData.CNR_D	1	Decimal	DINT
[0] 1/69-L18ERM-BB1B AFM60_manual_1	SickAFxWS_Enc2_SetData.CNR_N	2048	Decimal	DINT
Ill Embedded Discrete IO	SickAFxWS_Enc2_SetData.CPR	262144	Decimal	DINT
Expansion I/O, 0 Modules	SickAFxWS_Enc2_SetData.CS	0	Decimal	SINT
▲ 🚡 Ethernet	SickAFxWS_Enc2_SetData.ESF	0	Decimal	SINT
[] 1769-L18ERM-BB1B AFM60_manual_1	SickAFxWS_Enc2_SetData.DirectionChange_Limit	1000000	Decimal	DINT
AFM60A-Eth/IP Encoder	SickAFxWS_Enc2_SetData.MotionTime_Limit_seconds	630720000	Decimal	DINT
< >	SickAFxWS_Enc2_SetData.PowerTime_Limit_seconds	630720000	Decimal	DINT
	SickAFxWS_Enc2_SetData.StartsCCW_Limit	1000000	Decimal	DINT
	SickAFxWS_Enc2_SetData.StartsCW_Limit	1000000	Decimal	DINT
	SickAFxWS_Enc2_SetData.Position_HighLimit	1073741823	Decimal	DINT
	SickAFxWS Enc2 SetData.Position LowLimit	0	Decimal	DINT
	SickAFxWS Enc2 SetData.PresetValue	0	Decimal	DINT
	SickAEvWS Enc2 SetData Scaling	0	Decimal	SINT
	SickAEvWS Enc2 SetData SlaveSignOfLife	1280	Decimal	DINT
	SickAExWS Enc2 SetData Temperature Format	4608	Decimal	INT
	b SickAEvWS Enc2 SetData Velocity Format	7051	Decimal	INT
	N SickAEd//S Enc2 SetData Velocity_Format	12000	Decimal	DINT
	<ul> <li>Sicker Avs_cricz_second.velocity_highLimit</li> <li>Sicker Avs_cricz_second_velocity_highLimit</li> </ul>	12000	Decimal	DINT
	<ul> <li>SICKAFXWS_Enc2_SetData.velocity_LowLimit</li> </ul>	-12000	Decimai	DINT
	SickAFxWS_Enc2_SetData.xErrorCode	0	Decimal	DINT
	SickAFxWS_Enc2_SetData.xMsgRecordArray	{}	{} Decimal	SINT[4]
N State Stat	SickAFxWS_Enc2_SetData.SerialNo	184614935	Decimal	DINT
Logical Organizer	Monitor Tags / Edit Tags /		<	

Under Controller Tags, the parameters of the encoder can be changed in the Sick-AFxWS\_Enc1\_SetData node.

Figure 53: Changing parameters under SetData

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

### 

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

				🔷 🐤 🎓 🖉 📴 🛌 🗅 🕹 🐇	C & C C		
NCK				3* % # 8		JSR JAR RET SER THO HCR UI	D UIE SFR SFP EMENT EOT
nsor Intelligence.				▶, No Edits ₽.	Alarms Bit Timer/Counter	Input/Output Compare Compute/	Math Move/Logical File/Misc.
				MainProgram - MainRoutine 🛛 🧮 MainProgram - SickAF)	_A101WS_A103WS_FB_Enc1_GetSet	Controller Tags - AFM60_manual	1(controller) ×
				cope: BAFM60_manual_ ~ Show: Al Tags			
Nome Paramelarization Disconstice	Toolo			Name	Ella Value	Force Mask 🔹 Style	Data Type
Home Formation Diagnosities	roois			SickAFxWS Enc2 AOI	{}	{}	SICK AFX60 A101W
Overview Units Preset				SickAFxWS_Enc2_Cyclic_GetSet	0	Decimal	BOOL
andour				SickAFxWS Enc2 GetData	()	()	SICK AFX60 WS DA
<u>or view</u>				SickAFxWS Enc2 Init GetSet	1	Decimal	BOOL
	Current	Default	ID hex	SickAFxWS Enc2 msgGetData	{}	{}	MESSAGE
Code sequence	CW	CW	0x0C	SickAFxWS Enc2 msgSetData	()	{}	MESSAGE
Preset	0	C	0x13	<ul> <li>SickAFxWS Enc2 SetData</li> </ul>	()	()	SICK AFX60 WS D
Lower limit for the position	0	c	0x16	SickAExWS Epr2 SetData Acceleration Format	2064	Decimal	INT
Upper limit for the position	1073741823	1073741823	0x17	SickAEvWS Enc2 SetData Acceleration HighLimit	1073741823	Decimal	DINT
Lower limit for the velocity	-9000	-9000	0x1B	SickAEvWS Enc2 SetData Acceleration Low limit	-1073741823	Decimal	DINT
Upper limit for the velocity	9000	9000	0x1C	<ul> <li>SickAFXWS_Enc2_SecBata.Acceletation_cowerning</li> <li>SickAFXWS_Enc2_SecBata.Acceletation_cowerning</li> </ul>	1073741023	Decimal	DINT
Upper limit for the acceleration	1073741823	1073741823	0x20	<ul> <li>SICKAPIWS_Enc2_setData.CMR</li> <li>SICKAPIWS_Enc2_setData.CMR</li> </ul>	10/3/41024	Decimal	DINT
Velocity unit	1075741025	rpm	0x19	SickAhxWS_Enc2_SetData.CNR_D	1	Decimal	DINT
Acceleration unit	cpmss	comss	0x1E	SickAExWS_Enc2_SetData.CNR_N	2048	Decimal	DINT
Temperature unit	°C	0°C	0x65	SickAFxWS_Enc2_SetData.CPR	262144	Decimal	DINT
Limit for the motion time in hours	200	175200	0x86	SickAFxWS_Enc2_SetData.CS	00 0	Decimal	SINT
Limit for the operating time in hours	100	175200	0x87	SickAFxWS_Enc2_SetData.ESF	0	Decimal	SINT
Limit number of changes in the direction of rotation	1000000	1000000	0x88	SickAFxWS_Enc2_SetData.DirectionChange_Limit	1000000	Decimal	DINT
Limit number of clockwise starts	1000000	1000000	0x89	SickAFxWS_Enc2_SetData.MotionTime_Limit_sec	onds 720000	Decimal	DINT
Limit number of counterclockwise starts	1000000	1000000	0x8A	SickAFxWS_Enc2_SetData.PowerTime_Limit_seco	nds 350000	Decimal	DINT
Scaling	Off	Off	0x0E	SickAFxWS_Enc2_SetData.StartsCCW_Limit	1000000	Decimal	DINT
Round axis functionality	Off	Off	0x7D	SickAFxWS_Enc2_SetData.StartsCW_Limit	1000000	Decimal	DINT
				SickAFxWS_Enc2_SetData.Position_HighLimit	1073741823	Decimal	DINT
				SickAFxWS_Enc2_SetData.Position_LowLimit	0	Decimal	DINT
		~	_	SickAFxWS_Enc2_SetData.PresetValue	0	Decimal	DINT
		<-		SickAFxWS Enc2 SetData.Scaling	0	Decimal	SINT
		×		SickAEvWS Enc2 SetData SlaveSignOfLife	1280	Decimal	DINT
				SickAExWS Enc2 SetData.Temperature Format	4608	Decimal	INT
				SickAEvWS Enc2 SetData Velocity Format	7951	Decimal	INT
				SickAEvWS Enc2 SetData Velocity Highlight	12000	Decimal	DINT
				<ul> <li>Score Arro Lince Second relocity Flight limit</li> <li>Score Arro Lince Second relocity Land inside</li> </ul>	12000	Desimal	DINT
				<ul> <li>SICKARAWS_ENC2_SELDATA.VEIOCITY_LOWLIMIT</li> <li>SICKARAWS_ENC2_SELDATA.VEIOCITY_LOWLIMIT</li> </ul>	-12000	Decimal	DINT
				<ul> <li>SICKARXWS_Enc2_setData.xErrorCode</li> </ul>	0	Decimal	UNVI CONTEN
				SickAExWS_Enc2_SetData.xMsgRecordArray	()	{} Decimal	SINF[4]
				SickAFxWS_Enc2_SetData.SerialNo	184614935	Decimal	DINŤ

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

![](_page_65_Picture_3.jpeg)

# DANGER

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

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

# 5.6 Function block

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

# 5.6.1 Requirements

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

### 5.6.2 Import and wiring

In order to be able to use the function block in the RSLogix 5000 software, import the component into a project as an add-on instruction (file name: SICK\_AFx60\_Vxxx.L5X).

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

- SICK AFS60 / AF	M60 A01	
SICK_AFX60	?	(bReadDone)
iTimeout	?	-(bWriteDone)
	??	-(bReadError)
GetMessage	2	-(bWriteError)
SetMessage	2	36 58
stData	2	
bRead	2	
bWrite	2	
iReadErrorcode	?	
iWriteErrorcode	2	

Figure 55: Function block in the Rockwell control

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

# 5.7 Integration of the encoder as generic module

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

![](_page_66_Picture_6.jpeg)

Figure 56: Integrating encoder

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

generic		Clear	Filters			Hide Filters	*
Module Type       Analog       CIP Motion Ci       Communication       Communication       Communication	Category Filters onverter nn nns	^ ~	Mod     Adv     Diali     Endi     FAN	dule Type Vendor Filt anced Energy Indust ight ress+Hauser IUC CORPORATION	ters ries, Inc.		~
Catalog Number	Description	Vendor		Category			
ETHERNET-	Generic EtherNet/IP CIP Bridge	Rockwe	Autom	Communication			
ETHERNET	Generic Ethernet Module	Rockwe	ell Autom	Communication			
ETHERNET	Generic EtherNet/IP Safety and	Rockwe	ell Autom	Safety,Other			

Figure 57: Select module

- 4. Click Create .
- ✓ The Module Properties dialog opens.

# 5.7.1 Module settings

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

New Module	É.				×
Type: Vendor: Parent:	ETHERNET-MODULE Generic Ethem Rockwell Automation/Allen-Bradley Local	et Module	ametere		
Name: Description:	AFM60_Encoder	Connection ran	Assembly Instance:	Size:	
Decemption		Input:	103	3	(32-bit)
	×	Output:	198		
Comm Format	Input Data - DINT V	Configuration:	100	28 🛓	(8-bit)
<ul> <li>IP Address / I</li> </ul>	ess: 192 . 168 . 1 . 11	Status Input:			
⊖ Host Na	me:	Status Output			
Open Mod	ule Properties	ОК	Can	cel I	Help

Figure 58: Entering module properties

Example:

- Name: AFM60\_Encoder (name is freely selectable)
- Comm Format: Input data DINT
- o IP Address: 192.168.1.123
- Input: Assembly instance: 103; size: 3
  - This selects instance 103 of the assembly object (see table 18, page 23). The size is 3 × 32 bits ( = 12 bytes).

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

# i NOTE

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

3. Click OK .

### Example data for a configuration assembly

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

These can be viewed at Controller Tags in the following column: Name at AFM60\_Encoder:C  $\rightarrow$  AFM60\_Encoder:C.Data.

# NOTE

i

The low byte is displayed before the high byte.

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

Figure 59: Example data for a configuration assembly

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

# 5.7.2 Downloading the configuration to the control

1. Load the configuration for the controller.

Offline	🖞 🚽 No Forces	▶ <sub>+</sub> No	Edits
	Go Online		perties P
	Upload		FM60 ma
460_manual_1	Download		-
Tags Fault Handler Handler	Program Mode Run Mode Test Mode		60_Enco M60_En M60_En
ogram ed	Clear Faults Go To Faults		M60_En M60_En
s	Controller Prope	erties	M60_En

Figure 60: Loading configuration

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

🗳 Logix Designer - Af	FM60_manua	al_1 (1769	9-L18ERM-BB1B 31.1	1]*		
FILE EDIT VIEW	SEARCH	LOGIC	COMMUNICATION	S TOOLS	WINDOW	HELP
🏷 🛳 🔛 🖶 🛛	¥ 🗇 🖞	26			~ <b>*</b> *	<b>ا ا</b> هر ه
Program Mode     Controller OK     Ecorru Storage OK	`₽	Path: Al	B_ETHIP-1\192.168.	1.3*		
I/O OK	Rem Prog	1	🖳 No Forces	▶ <sub>↓</sub> No	Edits	Đ.

Figure 61: Communication status

### 5.7.3 Checking communication

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

Controller Organizer	• 4 ×	Module Properties Report: Local (ETHERNET-MC	DOULE 1.001) 🗸 Con	troller Tags - AFM60_rr	ianual_1(controller)	) × (				
6 <sup>9 0</sup>		Scope: CAFM60_manual_ ~ Show: Al Tags							✓ T <sub>v</sub> Bater Name Alter	
Controller AFM60_manual_1     Controller Tags		Name	<u>=8</u> ] + 1	/alue •	Force Mask	•	Style	Data Type	Description	6
Controller Fault Handler		AFM60_Encoder:C		{]		{}		AB:ETHERNET_MOD		
iii Power-Up Handler		▲ AFM60_Encoden!		{)		<b>{}</b>		AB:ETHERNET_MOD		
a 🗐 Tasks		▲ AFM60_Encoder:I.Data		()		()	Decimal	DINT[3]		
P & MainProgram		AFM60_Encoder:I.Data[0]		0			Decimal	DINT	Fault header	
<b>Unscheduled</b>		AFM60_Encoder:I.Data[1]		15130			Decimal	DINT	Position	
A G Motion Groups		AFM60_Encoderil.Data[2]		270			Decimal	DINT	Velocity	
Ungrouped Axes Assets		Local:1:C		()		()		AB:Embedded_Discre		
h Logical Model		▶ Local:1:1		{)		{}		AB:Embedded_Discre		
Genfiguration     Genfiguration		Local:1:0		{}		<b>{}</b>		AB:Embedded_Discre		

Figure 62: Checking communication

- 1. In Controller Organizer, open the Controller AFM60\_manual\_1  $\rightarrow$  Controller Tags folder.
- 2. Under Name, open the AFM60\_Encoder:I → AFM60\_Encoder:I.Data item in the Controller Tags column.

Displayed data in the example:

- AFM60\_Encoder:I.Data[0]: Fault header: 0
- AFM60\_Encoder:I.Data[1]: Position: 15130
- AFM60\_Encoder:I.Data[2]: Speed: 270 turns/min

# 5.8 Programming examples

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

During programming, the controller must be in offline mode.

![](_page_71_Figure_3.jpeg)

Figure 63: Controller in offline mode

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

# 5.8.1 Reading out temperature

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

### Defining and declaring variables

First variables TEMP\_Trigger, TEMP\_OneShot, TEMP\_Value and TEMP\_Message have to be defined and declared for the program.

First, variable TEMP\_Trigger is created, which controls the readout process.

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

![](_page_71_Picture_15.jpeg)

Figure 64: Create a new variable

✓ The New Tag dialog opens.

72
New Tag		×
Name:	TEMP_Trigger	Create 🗸 🔻
Description:	^	Cancel Help
	×	
Usage:	<controller></controller>	
Туре:	Base ~ Connection	
Alias For:	~	
Data Type:	BOOL	
Parameter Connection:	~	
Scope:	🖸 AFM60_manual_1 🗸 🗸	
External Access:	Read/Write ~	
Style:	Decimal ~	
Constant		
Sequencin	g	
Open Confi	iguration	
Open Para	meter Connections	

Figure 65: Definition of variable TEMP\_Trigger

Enter TEMP\_Trigger in the Name field, select the BOOL data type in the Data Type field and click on OK.
 To trigger the process only once, another element, in this case edge-sensitive, must be defined and declared. This causes the process to be triggered only when

an edge change of variable TEMP\_Trigger from 0 to 1 occurs.

3. Select New Tag again.

New Tag		×
Name:	TEMP_OneShot	Create 🛛 🕶
Description:	<u>^</u>	Cancel
		Help
Usage:	<controller></controller>	
Туре:	Base ~ Connection	
Alias For:		
Data Type:	BOOL	
Parameter Connection:		
Scope:	😳 AFM60_manual_1 🛛 🗸	
External Access:	Read/Write ~	
Style:	Decimal $\checkmark$	
Constant		
Sequencing	9	
Open Confi	guration	
Open Parar	meter Connections	

Figure 66: Definition of variable TEMP\_OneShot

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

New Tag		×
Name:	TEMP_Value	Create
Description:	· · · · · · · · · · · · · · · · · · ·	Cancel
		Help
		1
Usage:	<controller></controller>	Y.
Туре:	Base ~ Connection.	
Alias For:		~
Data Type:	INT	
Parameter Connection:		~
Scope:	C AFM60_manual_1	~
External Access:	Read/Write	~
Style:	Decimal	<b>~</b>
Constant		
Sequencing	9	
Open Confi	guration	-
Open Para	meter Connections	

Figure 67: Definition of variable TEMP\_Value

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

New Tag		×
Name:	TEMP_Message	Create 🗸 🔻
Description:		Cancel
		Help
		,
Usage:	<controller></controller>	1
Туре:	Base V Connection	
Alias For:		
Data Type:	MESSAGE	]
Parameter Connection:		~
Scope:	PAFM60_manual_1	·
External Access:	Read/Write	·
Style:		*
Constant		
Sequencing	9	
Open MES	SAGE Configuration	
Open Parar	meter Connections	

Figure 68: Definition of variable TEMP\_Message

- 8. In the **New Tag** dialog, enter TEMP\_Message in the **Name** field, select the MESSAGE data type in the **Data Type** field and click on **OK**.
- The following figure shows the resulting variable structure for acyclic reading of the temperature:

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx									
Name 📰 🔺 Alias For	Base Tag	Data Type	Description	External Access	Constant	Style			
Local:1:C		AB:Embedded_DiscreteIO:C:0		Read/Write					
Local:1:1		AB:Embedded_DiscretelO:I:0		Read/Write					
Local:1:0		AB:Embedded_DiscretelO:O:0		Read/Write					
TEMP_Trigger		BOOL		Read/Write		Decimal			
TEMP_OneShot		BOOL		Read/Write		Decimal			
TEMP_Value		INT		Read/Write		Decimal			
TEMP_Message		MESSAGE		Read/Write					
Encoder1:C		_0328:AFM60A_EthIP_BEF003F5:C:0		Read/Write					
Encoder1:11		_0328:AFM60A_EthIP_5DA4C79D:I:0		Read/Write					

Figure 69: Variable structure for reading out the temperature

#### Defining process flow

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

1. Open the MainRoutine window under Tasks  $\rightarrow$  MainTask  $\rightarrow$  MainProgram.



Figure 70: Opening MainRoutine

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

68.1.3ª						<b>€</b>	5 🖻		нь		++ +/+ -( )()	U)(L)-				
	▶	No Edit	s	₽.					▶ Favorite	s Add-	On Examine On Bit	Timer/Counter	Input/Output	Compare	Compute/Math	Move/Logical
目 N	lainl	Program	n - MainR	toutine*	× 🥏	Contro	ller Ta	igs - A	FM60_manua	L_1(contro	lle?					
•	Q				abed ab	] ab '	<ab< th=""><td>&gt;</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></ab<>	>								
0	8		?													
(End	1)															

Figure 71: Inserting ExamineOn module

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

	TEN	IP_	Trigger 🧹					
	T,	E	nter Name Filter		Show: All	Tags		~
		Na	ame	==	Data Type	Usage	Descripti	^
-	•	٥	Local:1:0		AB:Embedded	<controller></controller>		
		۵	TEMP_Message		MESSAGE	<controller></controller>		
	•		TEMP_OneShot		BOOL	<controller></controller>		
	Ē		TEMP_Trigger		BOOL	<controller></controller>		
	Ē		TEMP_Value		INT	<controller></controller>		~
	√ 9	Sho	w controller tags					
	<u>_</u>	Sho	w Main Program ta	ags				
	Sho	w p	arameters from ot	her p	rogram:			
	<no< td=""><td>ne</td><td>&gt;</td><td></td><td></td><td>~</td><td></td><td></td></no<>	ne	>			~		
Ц		_						

Figure 72: Assignment of variable TEMP\_Trigger to ExamineOn

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



Figure 73: Inserting ONS block

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

TEN	IP_	OneShot 🧹			
<b>T</b> .	E	nter Name Filter	Show: All	Tags	~
	Na	ame _=	Data Type	Usage	Descripti 🔺
Ē	٥	Local:1:0	AB:Embedded	<controller></controller>	
Ē	Þ	TEMP_Message	MESSAGE	<controller></controller>	
-		TEMP_OneShot	BOOL	<controller></controller>	
•		TEMP_Trigger	BOOL	<controller></controller>	
Ē		TEMP_Value	INT	<controller></controller>	~
<b>⊘</b> 9	Sho	w controller tags			
<b>⊘</b> 9	iho	w MainProgram tags			
Sho	w p	arameters from other p	rogram:		
<no< td=""><td>ne</td><td>&gt;</td><td></td><td>~</td><td></td></no<>	ne	>		~	

Figure 74: Assignment of variables TEMP\_OneShot to ONS

- 6. Click on the question mark .
- ✓ A drop-down menu will open.
- 7. Select variable TEMP\_OneShot.

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

d, None d, None			8	•	ны	Here MS	GSV S	SV 1	IOT				
•	No Edits	a.		_ § • 1	Favorites	Add-On	Alarms	Bit	Timer/Counter	Input/Output	Compare	Compute/Math	Move/Logical
目 Ma	inProgram - MainRo	outine 🗙 🛷 Co	ntroller T	ags - AFM	160_manual_	1(controller)							
•	9, 11 12 13	10 E ates 22	ab +	16>									
0	TEMP_Trigg	er TEMP_OneSh [ONS]	ot										

Figure 75: Inserting MSG block

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

	MSG			_	^
	Message C	Control TEN	IP_Message	~	
Enter Name Filter	~	Show: All	Tags		~
Name	== Data	Туре	Usage	Descr	ipti 🔺
TEMP_Message	MES	SAGE	<controller></controller>		
					_
					~
Show controller tags					
Show Main Program t	ags				
Ch	1				
Snow parameters from o	iner program	n:	_		
<none></none>			$\sim$		

Figure 76: Assignment of variables TEMP\_Message to MSG

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

MSG			
Message	Control	TEMP_Message	 -(EN)

Figure 77: Opening configuration dialog of the MSG block

- 10. Click on the button with the three dots.
- ✓ The Message Configuration dialog opens.

lessage C Configuratio	onfiguratio	unication T	Aessage ag			>
Message	Туре:	CIP Gener	ic	~	•	
Service Type: Service Code: Instance:	Get Attribu	ute Single lex) Class: Attribute:	23 (Hex) 64 (Hex)	Source Element Source Length: Destination Element:	0	<ul> <li>(Bytes)</li> <li></li> </ul>
) Enable ) Error Co irror Path: irror Text:	⊖ Enabl	le Waiting Extende	⊖ Start ed Error Code:	🔾 Done	Done Length: 0	

Figure 78: Configuration dialog of the MSG block

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

#### 

TEMP\_Value is the fourth variable created. The value of the temperature is written into this when the example program is executed.

12. Open the Communication tab.

Message Conf	iguration - TEM	P_Message	Х
Configuration*	Communication	Тад	
Path:		Browse	

Figure 79: Communication tab

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

Path: Encoder1 Encoder1 PointIO PointIO PointIO Embedded I/O Expansion I/O, 0 Modules Expansion I/O, 0 Modules AFM60A-Eth/IP Encoder1	Message Path Browser	×
Encoder1	Path: Encoder1	
<ul> <li>I/O Configuration</li> <li>Image: PointIO</li> <li>Image: PointI</li></ul>	Encoder1	
I AFM60A-Eth/IP Encoder1	□····⊆       I/O Configuration         □····⊡       PointIO         □····⊡       [0] 1769-L18ERM-BB1B AFM60_manual_1         □····⊡       Embedded I/O         □····⊡       [1] Embedded Discrete_IO         □···□⊡       Expansion I/O, 0 Modules         □···□⊡       1769-L18ERM-BB1B AFM60_manual_1	
	AFM60A-Eth/IP Encoder1	

Figure 80: Selecting encoder

Configuration*	Communication*	Tag	
Path:	Encoder1		Browse

Figure 81: Selected encoder

- ✓ The encoder is transferred to the **Path** field.
- 15. End the Message Path Browser dialog with OK .

#### Transmitting program to controller

The program is then transmitted to the controller.

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



Figure 82: Transmitting the program to the controller

2. Confirm the next message.

#### **Testing program**

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

Scope: @AFM60_manual_ ~ S	Show: All Tags				
Name	== .	Value 🔶	Force Mask 🗧 🗧	Style	Data Type
Local:1:C		{}	{}		AB:Embedded_Discre
Local:1:1		{}	{}		AB:Embedded_Discre
Local:1:0		{}	{}		AB:Embedded_Discre
TEMP_Trigger		1		Decimal	BOOL
TEMP_OneShot		1		Decimal	BOOL
TEMP_Value		3900		Decimal	INT
TEMP_Message		{}	{}		MESSAGE
Encoder1:C		{}	{}		_0328:AFM60A_EthIP
▲ Encoder1:I1		{}	{}		_0328:AFM60A_EthIP

Figure 83: Display of the temperature value in TEMP\_Value

#### 5.8.2 Setting preset value

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

#### **Defining and declaring variables**

First variables PRESET\_Trigger, PRESET\_OneShot, PRESET\_Value and PRESET\_Message must be defined and declared for the program.

First, variable PRESET\_Trigger is created to trigger the process.

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



Figure 84: Create a new variable

The New Tag dialog opens.

New Tag		×
Name:	PRESET_Trigger	Create 🛛 🕶
Description:		∧ Cancel
		Help
		~
Usage:	<controller></controller>	$\sim$
Туре:	Base V Connection	
Alias For:		~
Data Type:	BOOL	
Parameter Connection:		~
Scope:	CAFM60_manual_1	~
External Access:	Read/Write	~
Style:	Decimal	~
Constant		
Sequencing	g	
Open Confi	guration	
Open Parar	meter Connections	

Figure 85: Definition of variable PRESET\_Trigger

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

New Tag		×
Name:	PRESET_OneShot	Create 🛛 🕶
Description:		Cancel
		Help
Usage:	<controller></controller>	1
Type:	Base V Connection	
Alias For:		
Data Type:	BOOL	]
Parameter Connection:		·
Scope:	PAFM60_manual_1	·
External Access:	Read/Write ~	,
Style:	Decimal v	·
Constant		
Sequencing	9	
Open Confi	guration	
Open Parar	meter Connections	

Figure 86: Definition of variable PRESET\_OneShot

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

New Tag		×
Name:	PRESET_Value	Create 🛛 🔻
Description:	^	Cancel
		Help
	~	
Usage:	<controller></controller>	2
Type:	Base ~ Connection	
Alias For:		
Data Type:	DINT	]
Parameter Connection:		~
Scope:	PAFM60_manual_1	·
External Access:	Read/Write	·
Style:	Decimal	/
Constant		
Sequencing	g	
Open Confi	iguration	
Open Para	meter Connections	

Figure 87: Definition of variable PRESET\_Value

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

New Tag		×
Name:	PRESET_Message	Create 🛛 🕶
Description:	^	Cancel
		Help
Usage:	<controller></controller>	
Type:	Base ~ Connection	
Alias For:	~	
Data Type:	MESSAGE	
Parameter Connection:	~	
Scope:	PAFM60_manual_1	
External Access:	Read/Write ~	
Style:	~	
Constant		
Sequencing	3	
Open MES	SAGE Configuration	
Open Parar	meter Connections	

Figure 88: Definition of variable PRESET\_Message

8. In the New Tag dialog, enter PRESET\_Message in the Name field, select the MES-SAGE data type in the Data Type field and click on OK.

The following figure shows the resulting variable structure for setting a preset value:

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

Figure 89: Variable structure for setting a preset value

#### Defining process flow

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

1. Open the MainRoutine window under Tasks  $\rightarrow$  MainTask  $\rightarrow$  MainProgram .



Figure 90: Opening MainRoutine

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

None	Move/Logical File/Misc. File/Shift Sequencer
🗄 MainProgram - MainRoutine* 🗙 🥔 Controller Tags - AFM60_manual_1(controller)	
<sup>8</sup> Q, Q, <sup>10</sup> 12 <sup>10</sup> <sup>10</sup> <sup>10</sup> <sup>10</sup> <sup>10</sup> <sup>10</sup> <sup>10</sup> <sup>10</sup>	
TEMP_Trigger TEMP_OneShot      [(ONS)	MSG Message Control TEMP_Message(EN) (ON) (ER)
(and a second	

Figure 91: Inserting Rung block

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

d, Nor d, Nor	ne ne ▶_	No Edits	a.	٩	0 4 4	⊢ ⊢ Favorites	Add-On	+/+ -( ) Alarms	(U)(L)- Bit Timer/Counter	Input/Output	Compare	Compute/Math	Move/Logical	File/Misc.	File/Shift	► Sequencer
目	Mair	Program - I	MainRoutine*	🗙 🥏 Controll	ler Tags - AF	M60_manual_	1(controller)									
0		L Hi Li		* 40 E	(0)											
0	6	3 —	P_Trigger TEN	IP_OneShot -[ONS]									MSG Message C	ontrol TEMF	_Message	
i	C	3 -														
(6	nd)	$\vdash$														

Figure 92: Inserting ExamineOn module

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

	?		$\sim$							
	<b>T.</b>	E	nter Name Filter		Show:	AII 1	Tags			~
		Na	ame	==	Data Type		Usage		Descripti	^
1	Ē	Þ	Local:1:0		AB:Embedde	d	<contro< td=""><td>oller&gt;</td><td></td><td></td></contro<>	oller>		
	•	Þ	PRESET_Messag	e	MESSAGE		<contr< td=""><td>oller&gt;</td><td></td><td></td></contr<>	oller>		
	Ē		PRESET_OneSho	ot	BOOL		<contr< td=""><td>oller&gt;</td><td></td><td></td></contr<>	oller>		
	Ē		PRESET_Trigger		BOOL		<contro< td=""><td>oller&gt;</td><td></td><td></td></contro<>	oller>		
	Ē		PRESET_Value		DINT		<contro< td=""><td>oller&gt;</td><td></td><td>v</td></contro<>	oller>		v
[	<u>_</u>	Sho	w controller tags	Na Da	ame: PRESET ata Type: BO(	_Tr OL	igger			
[	Description:									
	Show parameters from other program:									
	<no< td=""><td>ne</td><td>&gt;</td><td></td><td></td><td>`</td><td>~</td><td></td><td></td><td></td></no<>	ne	>			`	~			

Figure 93: Assignment of variable PRESET\_Trigger to ExamineOn

- 4. Click on the Fragezeichen .
- ✓ A drop-down menu will open.
- 5. Select variable PRESET\_Trigger.

For the edge sensitivity of the process flow, the ONS block must be inserted.

d, None d, None				8 🗉 🖣	ны	heri H	- +1+ -(.)-	-(L)(U)- ONS	OSR OSF							F
	▶ <sub>+</sub> Na	o Edits	a.	•	Favorites	Add-On	Alarms Bit	Timer/Counter	Input/Output	Compare	Compute/Math	Move/Logical	File/Misc.	File/Shift	Sequence	er F
∃м	ainPro	ogram - MainRo	utine* 🗙 🥏 Cor	ntroller Tags - A	FM60_manual	1(controlle	r)									
<b>e</b> ,	Q,		1. E1. and 22 a	š., * (00)												
0	8	TEMP_Trigge	er TEMP_OneShot									MSG Messa	ge Control 1	EMP_Messa	ge() -() -()	EN)
<u>t</u>	0	PRESET_Trie	gger ? 													
(End																

Figure 94: Inserting ONS block

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

?		$\sim$					
<b>T</b> .,	E	nter Name Fil	ter	Show:	All Tags		$\sim$
	Na	ame	-8	Data Type	Usage	Descripti	^
•	Þ	Local:1:0		AB:Embedde	d <controller></controller>		
6	Þ	PRESET_Me	essage	MESSAGE	<controller></controller>		
Ē		PRESET_Or	neShot	BOOL	<controller></controller>		
Ē		PRESET_Tri	igger	BOOL	<controller></controller>		
Ē		PRESET_Va	Name:	PRESET_One	Shot pontroller>		v
Data Type: BOOL Description:							
Show MainProgram tags							
Show parameters from other program:							
<none> ~</none>							

Figure 95: Assignment of variables PRESET\_OneShot to ONS

- 7. Click on the Fragezeichen .
- ✓ A drop-down menu will open.
- 8. Select variable PRESET\_OneShot.

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

i, None	No Edits	Ð.	° ∎ ∢ ∢	Favorites	Add-On Ala	SV SSV I arms Bit	OT Timer/Counter	Input/Output	Compare	Compute/Math	Move/Logical	File/Misc.	File/Shift	> Sequencer
<mark>ा Maini</mark> ि् ्	ImainProgram - MainRoutine x         ImainProgram - MainRoutine x         ImainProgram - MainRoutine x           Q         Imain Program - MainRoutine x         ImainProgram - MainRoutine x													
0	TEMP_Trigge	TEMP_OneSho [ONS]	t								MSG Message Co	ntrol TEMP_	Message 🗔	   
া	PRESET_Trig	ger PRESET_On [ONS]	eShot								MSG Message Cont	ol PRESET_	Message [	.] -(EN)
(End)														

Figure 96: Inserting MSG block

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

	MSG						
	Mess	age C	ontrol	PRE	SET_Message	~	
Enter Name Filter		$\sim$	Show:	All	Tags		~
Name	==	Data	Туре		Usage	Descripti	^
PRESET_Messa	ge	MESS	SAGE		<controller></controller>		
							~
Show controller tags							
Show Main Program t	ags						
Show parameters from o	ther n	morar	<b>.</b>				
Show parameters from 0	iner p	logial	n.	_			
<none></none>					~		
							_

Figure 97: Assignment of variables PRESET\_Message to MSG

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



Figure 98: Opening configuration dialog of the MSG block

- 11. Click on the button with the three dots.
- $\checkmark$  The Message Configuration dialog opens.

Message Configuration - PRESET_Mes	sage			×
Configuration* Communication* Tag				
Message Type: CIP Generic		~		
Service Type: Service Code: 10 (Hex) Class: 23 Instance: 1 Attribute: 13	V Sou Sou (Hex) Des (Hex)	rce Element: rce Length: tination ment:	PRESET_Value	✓ (Bytes)
Enable     Enable     Enable     Enror     Code:     Extended     Error     Path:     Error     Text:	Start O	Done Dor	ne Length: 0 Timed Out 🗲	
E	OK	Cancel	Apply	Help

Figure 99: Configuration dialog of the MSG block

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

#### 

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

13. Open the Communication tab.

Message Configuration - PRESET_Message					
Configuration*	Communication	Tag			
Path:		Browse			

Figure 100: Communication tab

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

Message Path Browse	r		×			
Path: Encoder1						
Encoder1						
PointIO						
[] Emiliaria [] Emiliaria	III Embedded I/O					
Ethemet     1769-L18ERM-BB1B AFM60_manual_1     AFM60A-Eth/IP Encoder1						
	ОК	Cancel	Help			

Figure 101: Selecting encoder

Configuration*	Communication*	Tag	
Path:	Encoder1		Browse
	Encoder1		

Figure 102: Selected encoder

- ✓ The encoder is transferred to the **Path** field.
- 16. End the Message Path Browser dialog with OK .

#### Transmitting program to controller

The program is then transmitted to the controller.

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



Figure 103: Transmitting the program to the controller

2. Confirm the next message.

## **Testing program**

Encoder1:11	{}	{}	_0328:AFM60A_EthIP
Encoder1:I1.ConnectionFaulted	0	Deci	mal BOOL
Incoder1:11.Data	{}	{} Deci	mal DINT[3]
Encoder1:I1.Data[0]	0	Deci	mal DINT
Encoder1:I1.Data[1]	500	Deci	mal DINT
Encoder1:I1.Data[2]	0	Deci	mal DINT
PRESET_Trigger	1	Deci	mal BOOL
PRESET_OneShot	[ 1]	Deci	mal BOOL
PRESET_Value	500	Deci	mal DINT
PRESET_Message	{}	{}	MESSAGE

Figure 104: Display of the preset value in PRESET\_Value

- 1. To test the example program, in variables **PRESET\_Value**, enter a value in **Controller Organizer** (500 in the example).
- 2. Set variable **PRESET\_Trigger** from 0 to 1.
- ✓ In position date AFM60\_EIP:I.Data[1], the value now jumps to 500.

# 6 Configuration using the integrated web server

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

## NOTE

i

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

# SICK

Home Parametrierung E	liagnose Tools	
Gerät Position Geschwi	indigkeit Temperatur Timer	
rät		
Gerätename	AFM60 EtherNet/IP	Benutzer
Firmware-Version	2.01	AuthorizedClient Abmelden Ändere
DHCP	•	Passwort
Stellung der Adressschalter	111	
MAC-Adresse	00:06:77:07:00:2B	Sprache
Seriennummer	0B01002B	German
Protokoll	Ethernet/IP CIP Position Sensor Object	
Position	311429	
[		
Status	•	

Figure 105: Web server interface

#### Requirements

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

## Language

The web server starts in English.

User User:		
Password:	Log on	
Language English ▼		
German		

Figure 106: Select language

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

#### 6.1 Home



All displayed values are updated about once a second.

#### 6.1.1 Device

This page lists the basic data about the encoder.

In addition, an LED symbol indicates the following status:

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

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

#### 6.1.2 Position

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

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

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

#### 6.1.3 Speed

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

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

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

#### 6.1.4 Temperature

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

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

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

#### 6.1.5 Timer

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

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

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

# 6.2 Parameterization

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

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

#### 

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

The following parameterization options are available without logging in:

- Overview
- Units
- Preset

The following parameterization options are available after logging in as the "Authorized-Client" user:

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

#### Log in

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

- User: AuthorizedClient
- Password: enc123

<b>User</b> User: Password:	Log on	
Language English English German		

Figure 107: Log in

### Changing the password

## NOTE

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

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

i

Benutzer & AuthorizedClient	<u>Abmelden</u>	Ändere Passwort
Sprache German ▼		

Figure 108: Changing the password

✓ The Ändere Passwort dialog opens.

# Ändere Passwort

Altes Passwort	
Neues Passwort	
Neues Passwort noch einmal eingeben	
	Ändere Passwort

Figure 109: Dialog for changing the password

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

# i NOTE

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

### 6.2.1 Overview

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

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

#### 6.2.2 Units

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

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

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

## 6.2.3 Changing preset value

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



## DANGER

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

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

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

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



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

#### 6.2.4 Triggering preset

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

#### Preset

der Preset-Funktion, ob eine age ausgeht, in die der Encoder
470
0

Figure 110: Triggering preset

- 1. Click PRESET.
- $\checkmark$  The position value is set to the preset value.

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

#### 6.2.5 Scaling

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

- Skalierung (attribute ID 0Eh)
  - o on
  - o Off

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

<u>Skalierung</u>

on 💌

	1
CPR	262144
Umdrehungen	2 -
Gesamtauflösung (CMR)	524288

Figure 111: Figure 110: Scaling

- CPR, number of steps per revolution (attribute ID 10h)
- Umdrehungen, number of revolutions of the total resolution (This is not an attribute of the position sensor object).
   Only the following values can be selected: 1, 2, 4, 8, 16, 32, 64, 128, 256, 512,

Unly the following values can be selected: 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1,024, 2,048 and 4,096.

 The Gesamtauflösung (CMR) field shows the value of attribute ID 11h "Total Measuring Range, Total Resolution" of the position sensor object (see table 24, page 29).

## NOTE

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



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

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

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

#### 

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

### 6.2.6 Round axis functionality

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

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

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

99

#### Rundachsfunktionalität

on	•
----	---

Zähler für die Anzahl der Umdrehungen	137
Nenner für die Anzahl der Umdrehungen	10
Gesamtauflösung (CMR)	3600

Figure 112: Round axis functionality

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

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

# NOTE

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

# DANGER

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

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

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

#### 

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

### 6.2.7 Limits

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

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

# I NOTE

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

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

If these limits are exceeded, then the consequence is:

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

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

- Limit of movement time in hours <sup>1)</sup>
- Operating time limit in hours <sup>1)</sup>
- Limit of the number of changes in the direction of rotation
- Limit of the number of clockwise starts
- Limit of the number of counterclockwise starts
- **1**) The movement time and the operating time are always calculated from the first commissioning of the encoder. When configuring the limit, note that the encoder may already have some movement time and operating time.

#### 6.2.8 Reset

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

#### Saving parameters to non-volatile memory

► Click -S-.

The function uses the **Save** class service (service code 16h) of the position sensor object.

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

#### Resetting to factory settings

# DANGER

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

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

# NOTE

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

Click -D-.

The function uses the **Reset** class service (service code 05h) of the position sensor object (Data = 01h).

The parameters are reset to the factory settings and the encoder is restarted.

### Restarting

Click -R- . The encoder is restarted.

## NOTE

i

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

# 6.3 Diagnostics

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

#### 6.3.1 Status

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

#### <u>Status</u>

Aktueller Status Limit Betriebszeit des Encoders überschritten Statusspeicher Keine Einträge Bewegungszeit des Encoders Innerhalb tolerierbarer Werte Betriebszeit des Encoders Außerhalb tolerierbarer Werte

Figure 113: Diagnostic status

#### Aktueller Status

The last three messages since switch-on are displayed (after switching off and switching back on, the memory is empty).

Statusspeicher

The texts for warnings, alarms and errors from the Fault header are displayed (see table 33, page 108). If no warning, no alarm and no error has occurred yet, the displayed text is Keine Einträge.

- Bewegungszeit des Encoders
   Indicates whether the movement time is within the tolerated values (see "Limits",
- page 100).
  Betriebszeit des Encoders Indicates whether the operating time is within the tolerated values (see "Limits", page 100).

#### 6.3.2 Speed

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

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

#### 6.3.3 Temperature

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

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

#### 6.3.4 Time

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

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

#### 6.3.5 Cycles

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

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

#### 6.3.6 Heartbeat

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

Heartbeat	
on 💌	
•	
Aktueller RPI in ms	5
Aktueller Update-Faktor (2 127)	5
Aktueller Update-Zyklus in ms	150



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

An LED symbol indicates the heartbeat:

Gray



Green Active Not active

#### NOTE i

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

The Aktueller RPI in ms column shows the RPI.

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

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

#### 6.4 Tools

#### 6.4.1 EDS

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

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

#### 6.4.2 Ladder routine

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

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

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

### 6.4.3 Update

A firmware update via FTP can be performed as follows.

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

Host: 192.168.1.124	Username: host	Password: ••	••••	Port		Quickconnect 💌				
Command: PASV lesponse: 227 Enteri Command: LIST lesponse: 150 Here i lesponse: 226 Trans itatus: Directory l itatus: Disconnec	ng Passive Mode (192,168,1,124; t comes er OK, Closing connection sting successful red from server	4,3)								
.ocal site: C:\Documents a	nd Settings\benjamin\Desktop\			*	Remote site:	/FIRMWARE_UPDATE	DRIVE			
	Desktop Favorites Local Settings My Documents				■ <mark>()</mark> /	- FIRMWARE_UPDATE_C	RIVE			
	hardcopy 1980/01/06	0:05:40	1	-						
Filename A	FERNUNIH-32A738	Files	ize Filetype	-						
🛄 🔁 3nn			File Folder							
5mn			File Folder		Filename V	7	1	Filesize	Filetype	Last modified
FileZillaPortable			File Folder		<u> </u>					
FirefoxPortable			File Folder		🕑 FupFile.bi	in		1.863.440	BIN File	01.01.1980 (
Copy of Orginal_Messun	20_04_2011.xls	882.1	76 Microsoft Ex	ce	C FLASH_II	NFO			File Folder	01.01.1980 (
FupFile.bin		1.863.4	40 BIN File		C FLASH_C	CONTENT			File Folder	01.01.1980
🐏 K8Messreihe.xls		49.1	52 Microsoft Ex	ce						
MSMP_03032011.xls		226.3	IO4 Microsoft Ex	CE						
KSMP_18042011.xls		432.6	i40 Microsoft Ex	CE						
MAE_Data.txt		1	69 Text Docum	er						
MessprogrammDaten		. 4	13 File	-						
•				<u>کا</u>	1					
Selected 1 file. Total size: 1.8	63.440 bytes				1 file and 2 di	rectories. Total size: 1.86	3.440 bytes			
Server/Local file	Direction Remote file		Si	ze F	riority Sta	atus				
Owned Glass	1 Summetican Commented Summetic	(21)								
E COLORES E COLORES	Judi bi ei s Successi ul tralisi i	15(21)								

Figure 115: Example for the firmware update

- 4. Open the FIRMWARE\_UPDATE\_DRIVE folder.
- 5. Transmit the update file  $(*.bin)^{1}$  to this folder.

The firmware update takes about 3 minutes.

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

Following the firmware update, the encoder performs a reboot.

• The Encoder LED then lights up green again.

# NOTE

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

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

i

#### 6.4.4 Address switch

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

#### 6.4.5 Fault header information

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

# 6.5 Test notes



# DANGER

No commissioning without inspection by authorized personnel!

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

# 7 Troubleshooting

# 7.1 Response to errors



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

# 7.2 SICK-STEGMANN Support

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

## 7.3 Diagnostics

#### 7.3.1 Error and status indications of the LEDs

#### Mod, Net and Encoder status LEDs

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

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

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

D	lisplay	Description
0	Off	No supply voltage or No IP address
÷.	Green	Warning Incorrect parameter
٠	Green	Device in operation
÷.	Red	Warning, but device still ready for operation or Firmware update in progress
•	Red	Error Encoder error or Restart after firmware update in progress
· <b>.</b>	Red/green	Self-test when switching on

#### Link 1 and 2 Ethernet link LEDs

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

Table 32: Meaning of the Link 1 and 2 LEDs

Display		Description
0	Off	No supply voltage
		or
		No Ethernet connection
$\bullet$	Green	Ethernet connection established
	Yellow	Interface port locked
÷.	Green	Data transmission TxD/RxD
-	Yellow	Data collisions

#### 7.3.2 Self test via EtherNet/IP

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

#### 

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

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

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

#### 7.3.3 Warnings, alarms and errors via EtherNet/IP

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

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

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

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

#### Fault header

Table 33: Fault header

Byte	Bit	Description	Alarm (A) / Warning (W)
0	0	Operating temperature of the microcontroller outside the per- missible range	W
	1	Operating temperature of the encoder outside the permissible range	W
	2	Permissible internal LED current in the sensor system exceeded	W
	3	Supply voltage outside the permissible range	W
	4	Frequency error, maximum speed is exceeded	W
	5	The lower/upper limit of the speed configured with the attribute IDs 1Bh or 1Ch has been undercut/exceeded (see table 24, page 29).	W
	6	The lower/upper limit of the acceleration configured with the attribute IDs 20h or 21h has been undercut/exceeded (see table 24, page 29).	W
	7	The lower/upper limit of the position configured with the attribute IDs 16h or 17h has been undercut/exceeded (see table 24, page 29).	W
1	8	Position error (amplitude error of singleturn measurement)	A
	9	Position error (amplitude error of multiturn measurement)	А
	10	Position error (vector error Sin <sup>2</sup> + Cos <sup>2</sup> of singleturn measure- ment)	A
	11	Position error (vector error Sin <sup>2</sup> + Cos <sup>2</sup> of multiturn measure- ment)	A
	12 14	Reserved	-
	15	The "start bit" has not yet been reset or a parameter has been changed via the web server. Bit 15 is set after each restart. It can be reset using the following command: Attrib- ute ID 8Bh (see table 24, page 29).	W
2	16	Singleturn position error (error in sensor)	А
	17	Multiturn position error (synchronization MA single)	А
	18	Multiturn position error (synchronization quad single)	A
	19	Multiturn position error (internal interface)	A
	20	Multiturn position error (FRAM)	-
	21	Limit of the number of changes of the direction of rotation exceeded	W
	22	Limit of the number of clockwise starts exceeded	W
	23	Limit of the number of counterclockwise starts exceeded	W
Byte	Bit	Description	Alarm (A) / Warning (W)
------	---------------------------	---	----------------------------
3	24	Memory error (EEPROM checksum)	А
	25	Memory error (EEPROM IRQ)	А
	26	Error during commissioning (start-up)	А
	27 Error during self test		A
	28	Limit of the movement time of the encoder is exceeded	W
	29	So-called "sanity check flag". The flag is set when the encoder has detected an incorrect speed or a position error. Is reset when the device is switched on again.	-
	30	Slave sign of life. Active if attribute ID 0Dh is set (see table 24, page 29). The bit changes its value in the configured update cycle.	-
	31	Limit of the operating time of the encoder is exceeded	W

#### Alarms

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



### DANGER

Alarms in the application must be evaluated!

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

In addition, the Encoder LED permanently lights up red.

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

Table	34:	Alarm	s
Table	34:	Alarm	s

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

#### Warnings

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

In addition, the Encoder LED flashes red.

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

### NOTE

i

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

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

Table 35: Warnings

#### 7.3.4 Error messages of the Allen Bradley control system

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

RUN     OK     Energy Storage	I/O Forces: Disabled, None					
I/O	Offline		Ē .	No Forces	$\rightarrow$	No Ec
ontroller Organizer				Go Online		
a •=				Upload		
🔺 🚄 Controller AFM	✓					
<ul> <li>Controller Tags</li> <li>Controller Fault Handler</li> <li>Power-Up Handler</li> <li>Tasks</li> <li>MainTask</li> <li>MainProgram</li> </ul>				Program Mode		
				Run Mode		
				Test Mode		
				Clear Faults		
<ul> <li>Parameters and Local Ta</li> <li>MainRoutine</li> </ul>			Go To Faults			
				Controller Prop	erties	

Figure 116: Example of an error message in RSLogix

The following error messages come from the RSLogix 5000 software.

Table 36: Error messages from the RSLogix 5000 software

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

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

# 8 Annex

### 8.1 Conformity

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

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

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

# ANNEX 8

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Detailed addresses and further locations at www.sick.com

#### Hungary

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

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

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

Italy Phone +39 02 27 43 41

E-Mail info@sick.it Japan Phone +81 3 5309 2112

E-Mail support@sick.jp

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

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

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

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

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

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

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

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

Singapore Phone +65 6744 3732 E-Mail sales.gsg@sick.com Slovakia Phone +421 482 901 201 E-Mail mail@sick-sk.sk

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

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

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

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

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

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

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

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

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

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

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

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

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

