hohner

Elektrotechnik Werne

User Manual

Absolute Rotary Encoder with Ethernet/IP-Interface

EtherNet/IP™ conformance tested

Your partner for standard and special designs
- precise, reliable and fast -

hohner
Elektrotechnik Werne

ABSOLUTE IXARC ROTARY ENCODER WITH ETHERNET/IP INTERFACE
USER MANUAL

1. Introduction 4
1.1 Control and Information Protocol (CIP) 5
1.2 Object model 5
2. Data Transmission 6
2.1 Implicit Messaging I/O Connection 9
2.1.1 I/O Assembly Instances 9
2.1.1.1 Data Attribute Format 9
2.1.2 Data Mapping 9
2.1.3 Data Mapping (Parameter) 10
2.1.3.1 Data Offset 10
2.1.4 Connection Path 11
2.2 Explicit Messaging 11
2.2.1 CIP Common Services for Position sensor object (Class $0x23_{hex}$) 12
Save / Restore 12
2.2.2 Position Sensor Objects 13
2.3 TCP/IP Interface Object 14
2.3.1 Status Instance Attribute (01_{hex}) 14
2.3.2 Configuration Instance Attribute (02_{hex}) 15
2.3.3 Configuration Control Inst. Attribute (04_{hex}) 15
2.3.4 Physical Link Object (05_{hex}) 15
2.3.5 Interface Configuration (06_{hex}) 16
2.3.6 Host Name 16
2.4 Ethernet Link Object 17
2.4.0 Instance Attributes 18
2.4.1 Interface Flags 19
2.4.2 Common Services 19
2.4.3 Link Object Instances 20
2.5 Setting parameters with scanners 20
2.5.1 Read out position value 21
2.5.2 Set preset value 22
2.5.3 Get preset value 22
3 Diagnostic 23
4 Programmable Parameters 25
4.1 Encoder parameters for Position Sensor Object Class 23hex 25
4.1.1 Direction counting 25
4.1.2 Scaling function control 25
4.1.3 Resolution per revolution 25
4.1.4 Total resolution 26
4.1.5 Preset value 26
4.1.6 Velocity Format 27
4.1.7 Velocity Filter 27
4.1.8 Endless Shaft 27
5. Installation 28
5.1 Electrical connection 28
5.2 Ethernet cables 28
6 Power On 29
7 Installation 29
7.1 Rockwell configuration tools 29
7.1.1 Setting IP-Address (BOOTP/DHCP) 29
7.1.2 Configuration RSLinx Classic™ 31
7.1.3 RSNetWorx™ 33
7.1.4 Configuration RSLogix 5000 36
7.2 Schneider configuration tools 43
7.2.1 Setting configuration 43
7.2.2 Online configuration 46
7.3 BOOTP/DHCP and IP configuration tool ... 48
8 FAQ 50
9 Glossar 51
10. Technical Data 53
10.1 Electrical Data 53
10.2 Mechanical Data 53
10.3 Minimum (mechanical) lifetime 53
10.4 Environmental Conditions 53
10.5 Mechanical Drawings 53
11 Models/Ordering Description 53
12 Accessories and Documentation 53
13 Revision index 53

hohner
Elektrotechnik Werne

ABSOLUTE IXARC ROTARY ENCODER WITH ETHERNET/IP INTERFACE USER MANUAL

1. Introduction

Absolute rotary encoders provide a definite value for every possible position. All these values are reflected on one or more code discs. The beams of infrared LEDs are sent through code discs and detected by Opto-Arrays. The output signals are electronically amplified and the resulting value is transferred to the interface.

The absolute rotary encoder has a maximum resolution of 65536 steps per revolution (16 Bit). The Multi-Turn version can detect up to 16384 revolutions (14 Bit). Therefore the largest resulting resolution is 30 Bit = 1.073.741.824 steps. The standard Single-Turn version has 13 Bit, the standard Multi-Turn version 25 Bit.

The integrated Ethernet interface of the absolute rotary encoder supports all necessary the EtherNet/IP functions.
The protocol supports the programming of the following additional functions in several ways:
Code sequence (Complement)
Resolution per revolution
Total resolution
Preset value
IP-Address

The general use of absolute rotary encoders with EtherNet/IP interface is guaranteed. The data will transmit in a standard Ethernet frame in the data section, see at the bottom of this side the pink field with the blue frame.

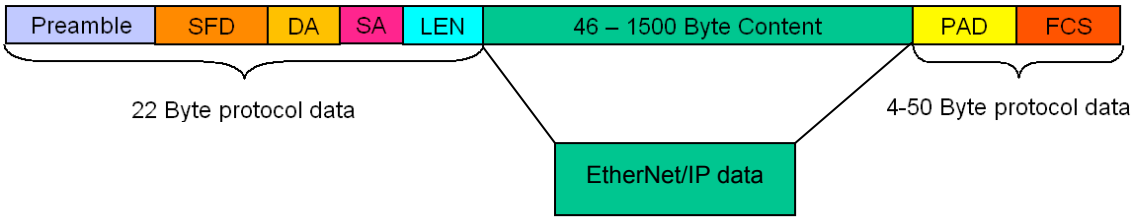
The MAC Address for each encoder is available on the type label.

The IP address can be programmed with DHCP or BOOTP by configuration tools of the PLC.

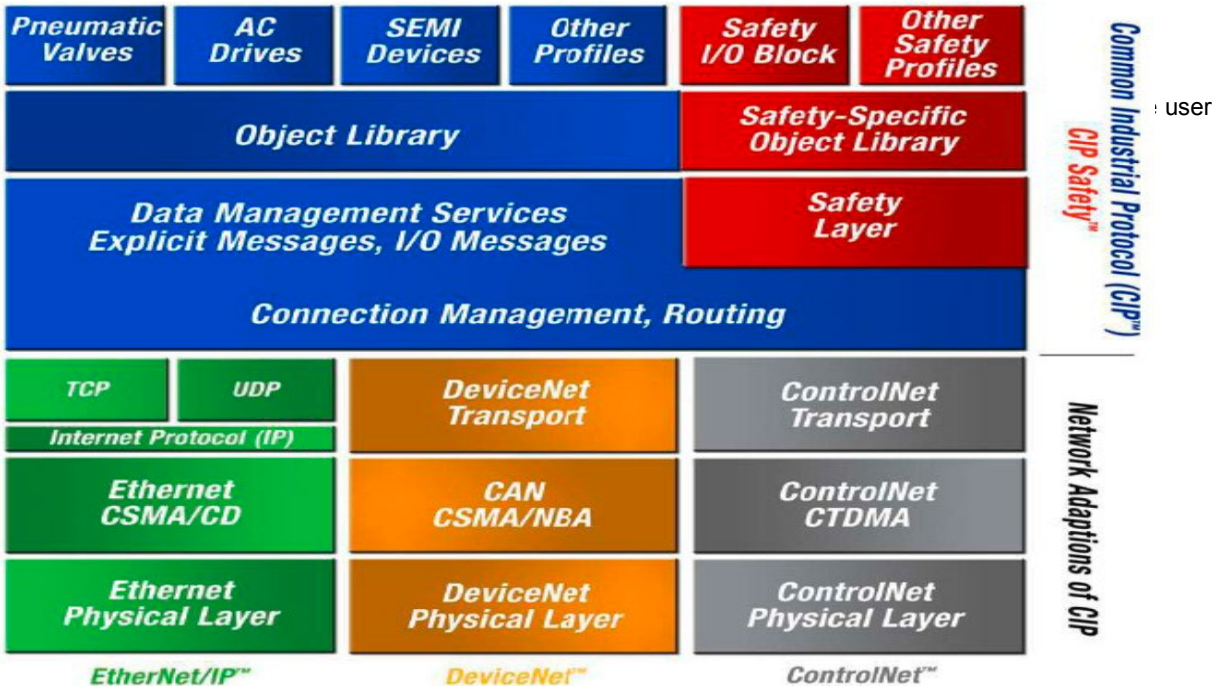
The physical interface support Autonegotiation and Autocrossing.

General information's about EtherNet/IP are available:
www.ethernetip.de (German)
www.odva.org/default.aspx?tabid=67 (English)

Setup of an Ethernet data package on layer 2



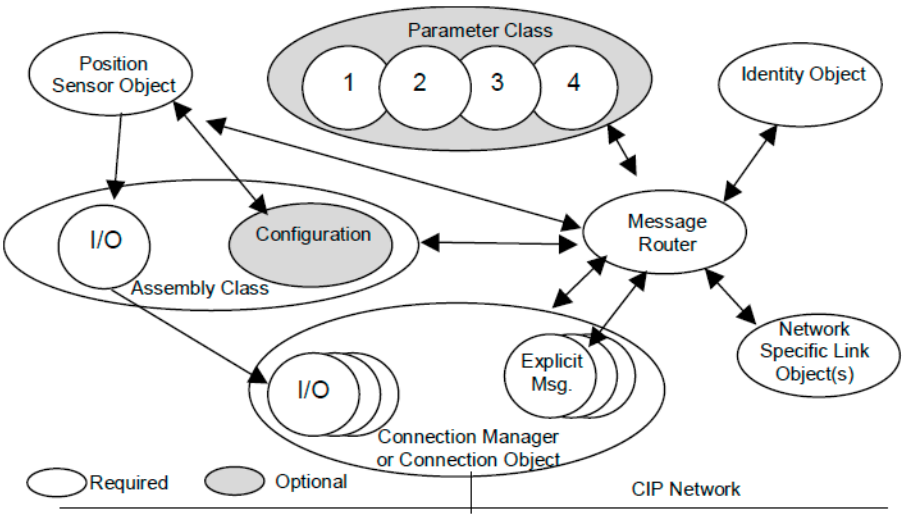
1.1 Control and Information Protocol (CIP)



: user

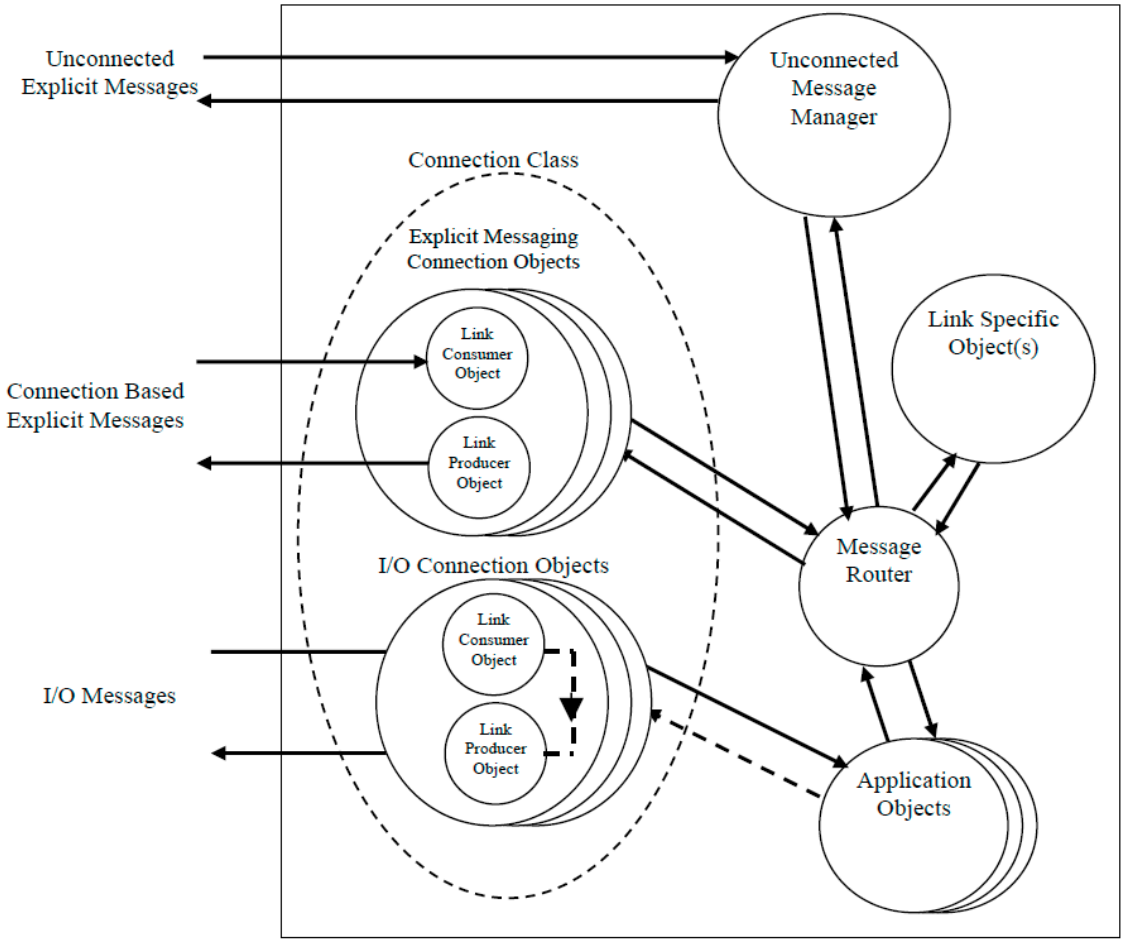
1.2 Object model

EtherNet/IP describes all data and functions of a device considering an object model. By means of that object-oriented description, a device can be defined complete with single objects. A object is defined across the centralization by associated attributes (e.g. process data), its functions (read- or write access of a single attribute) as well as by the defined behavior. The absolute rotary encoder support the Encoder Device Type: 22_{hex} or Generic Device Type: 0_{hex}. This is programmable, see chapter 4.1.6. All parameters will be used with Big Endian notation.



2. Data Transmission

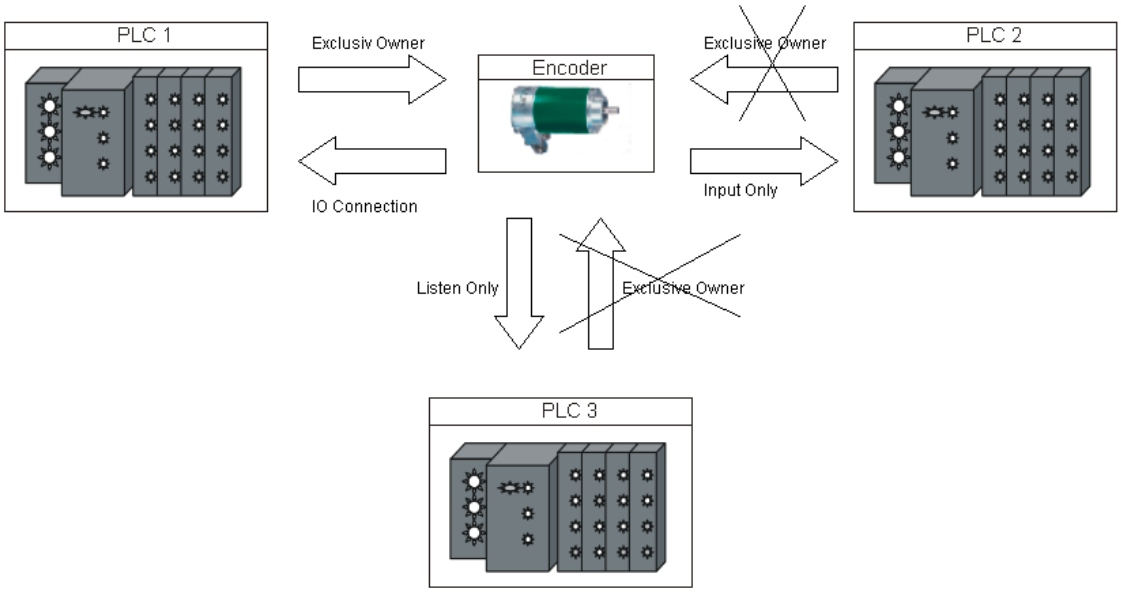
The data transmission in the EtherNet/IP network is realized by implicit or explicit messaging. Explicit messages are split in unconnected and connection based versions. Unconnected messages will be use i.e. by EtherNet/IP scanners.



Exclusive-Owner, Input Only Listen Only

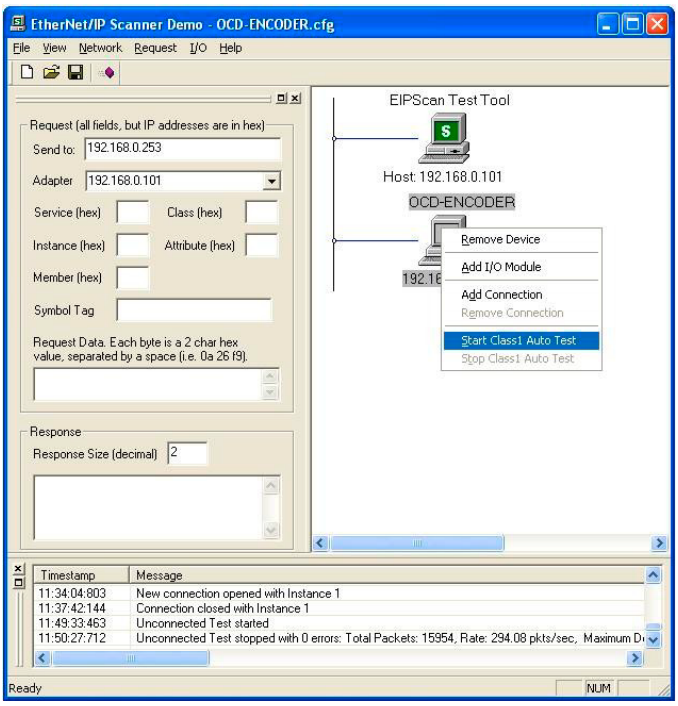
It is possible to open 256 connections to the encoder. One could be an Exclusive Owner connection, 255 additional connections can be realized mixed in Input Only or Listen Only.
With an Exclusiv Owner connection can be transmit the parameters (cycle time, configuration and Assembly Instances) to the encoder.

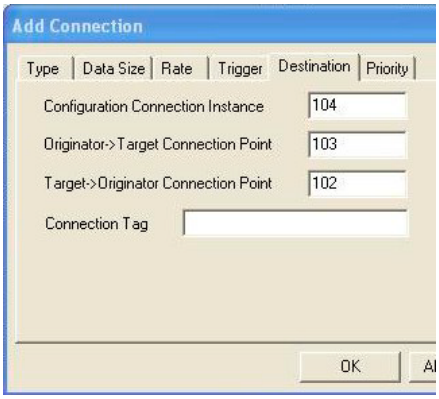
Input Only connections can only work if all of the parameters are according to the encoder parameters.
Listen Only need an connection of Excusiv Owner or Input Only.

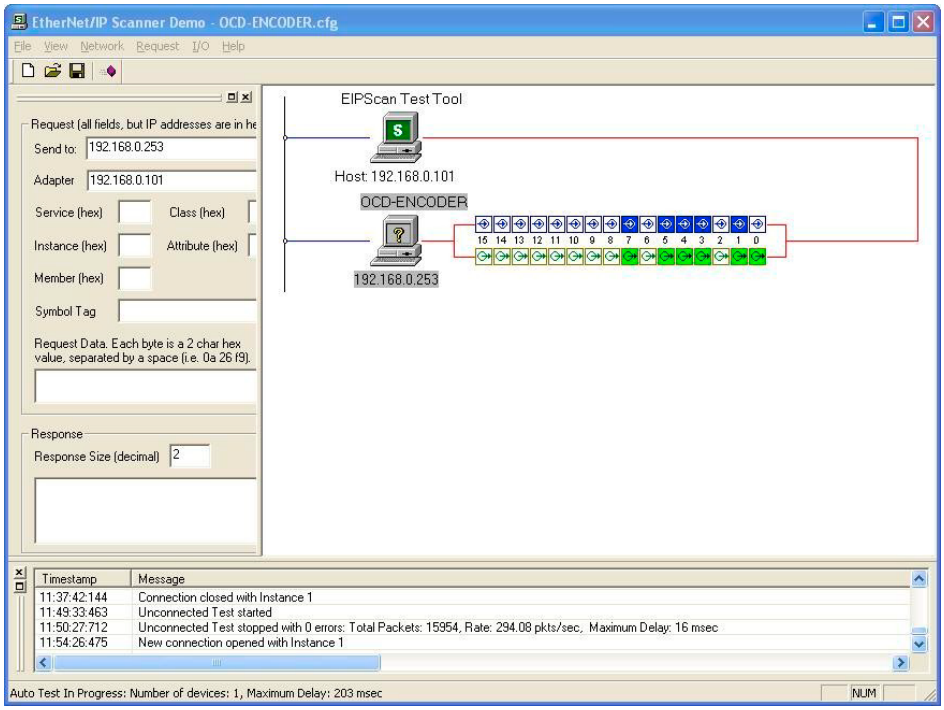


Assembly	Config	Output Instance	Input Instance
Connection Manager	Config	Connection Point 1	Connection Point 2
Exclusiv-Owner	$0x6A_{hex}$ (106)	$0x69_{hex}$ (105)	0x01 Position value
			0x03 Position value + velocity
Input Only	$0x6A_{hex}$ (106)	$0x64_{hex}$ (100)	0x01 Position value
			0x03 Position value + velocity
Listen Only	-	$0x65_{hex}$ (101)	0x01 Position value
			0x03 Position value + velocity
Demo-Scanner	$0x68_{hex}$ (104)	$0x67_{hex}$ (103)	$0x66_{hex}$ (102)

Communication check







2.1 Implicit Messaging I/O Connection

Provide dedicated, special-purpose communication paths between a producing application and one or more consuming applications for the purpose of moving application-specific data. This is often referred to as implicit messaging. Class 0 and 1 are supported.

2.1.1 I/O Assembly Instances

Instance	Type	Name
1	Input	Position Value
3	Input	Position Value and Velocity

2.1.1.1 Data Attribute Format

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	0	Position Value (low Byte)							
	1								
	2								
	3	Position Value (high byte)							
3	0	Position Value (low Byte)							
	1								
	2								
	3	Position Value (high byte)							
	4	Velocity (low Byte)							
	5								
	6								
	7	Velocity (high byte)							

2.1.2 Data Mapping

Data Component	Class		Instance	Attribute	
Name	Name	Number	Number	Name	Number
Position Value	Position Sensor	23_{hex}	1	Position Value	$0A_{hex}$
Velocity	Position Sensor	23_{hex}	1	Velocity	18_{hex}

2.1.3 Data Mapping (Parameter)

On every Forward Open Request, the following parameters, will be sent from the controller to the encoder.

Assembly Instance Configuration: 7, size 12 Bytes

Configuration Parameter Name	Class		Instance Number	Attribute	
	Name	Number		Name	Number
Direct Counting Toggle	Position Sensor	23_{hex}	1	Direct Counting Toggle	$0C_{hex}$
Scaling Function Control	Position Sensor	23_{hex}	1	Scaling Function Control	$0E_{hex}$
Measuring units per Revolution	Position Sensor	23_{hex}	1	Measuring Units per Span	10_{hex}
Total Measuring Range in measuring units	Position Sensor	23_{hex}	1	Total Measuring Range in measuring units	11_{hex}
Velocity Format	Position Sensor	23_{hex}	1	Velocity Format	19_{hex}

2.1.3.1 Data Offset

Byte Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Direction Counting Toggle							
1	Scaling Function Control							
2	Measuring units per Revolution (low byte)							
3								
4								
5	Measuring units per Revolution (high byte)							
6	Total Measuring Range in measuring units (low byte)							
7								
8								
9	Total Measuring Range in measuring units (high byte)							
10	Velocity Format (low byte)							
11	Velocity (high byte)							

2.1.4 Connection Path

Is made up of a byte stream that defines the application object to which a connection instance applies.

This path will be created from the configuration tools and are available in the EDS file too. This path will sent during power up to the encoder. For some tools it is necessary to use the connection path as parameter:

[20] [04] [24 6A] [2C 69] [2C 01] [80 06 00 01 00100000 00200000 041F]

Segment Groups	Segment	Description
Application Path	20 04	Assembly object class
	24 6A	Instance segment type with Assembly Instance $0x6A_{hex}$ (105) (Configuration)
	2C 69	Assembly Instance $0x69_{hex}$ (106) (Output controller to encoder)
	2C 01	I/O Assembly Instance 1 (Position value)
	80 06	Data segment with lenght of 6 Bytes
	00 01 00100000 00200000 041F	Configuration Data, see chapter 2.1.3.1 for details

2.2 Explicit Messaging

Provide generic, multi-purpose communication paths between two devices. These connections often are referred to as just Messaging Connections. Explicit Messages provide the typical request/response-oriented network communications. Class 2 and 3 are supported.

2.2.1 CIP Common Services for Position sensor object (Class $0x23_{hex}$)

Supported Service Code	Service Name	Comment
05_{hex}	Reset	Boot up of the encoder, the programmed parameter from the customer will use again
$0E_{hex}$	Get_Attribute_Single	Read out attribute from the encoder
10_{hex}	Set_Attribute_Single	Write attribute to the encoder
15_{hex}	Restore	Restore the saved parameters. Use instance 0 of position sensor class to restore all configuration parameter at once. To restore single parameter use instance 1 of position sensor class with attribute number as argument (see next table).
16_{hex}	Save	Save the parameters from chapter 2.1.3 to the nonvolatile memory. Use instance 0 of position sensor class to save all configuration parameter at once.

2.2.2 Position Sensor Objects

Instance Attributes (Get: read, Set: write + read)

Class Code: 23_{hex}

Attrib. ID	Access	Name	Data Type	Description
01_{hex}	Get	Number of Attributes	USINT	Number of supported Attributes
02_{hex}	Get	Attribute List	Array of USINT	List of supported Attribute
$0A_{hex}$	Get	Position Value Signed	DINT	Current position signed
$0B_{hex}$	Get	Position Sensor Type	UINT	Specifies the device type
$0C_{hex}$	Set	Direction Counting Toggle	Boolean	Controls the code sequence clockwise or counterclockwise
$0E_{hex}$	Set	Scaling Function Control	Boolean	Scaling function on/off
10_{hex}	Set	Measuring units per Span	UDINT	Resolution for one revolution
11_{hex}	Set	Total Measuring Range in Measuring Units	UDINT	Total resolution
13_{hex}	Set	Preset Value	DINT	Setting a defined position value
18_{hex}	Get	Velocity Value	DINT	Current speed in format of attribute 19_{hex} and $2A_{hex}$
19_{hex}	Set	Velocity Format	ENGUINT	Format of the velocity attributes
29_{hex}	Get	Operating Status	BYTE	Encoder diagnostic operating status
$2A_{hex}$	Get	Physical Resolution Span	UDINT	Resolution for one revolution
$2B_{hex}$	Get	Number of Spans	UINT	Number of revolutions
33_{hex}	Get	Offset Value	DINT	Shift position value with the calculated value
64_{hex}	Set	Device Type	DINT	Encoder device = 22_{hex} Generic device = 0 (default)
65_{hex}	Set	Endless Shaft	DINT	Off = 0, On = 1, Auto = 2
66_{hex}	Set	Velocity Filter	DINT	Fine = 0, Middle = 1, Raw = 2

2.3 TCP/IP Interface Object

The TCP/IP Interface Object provides the mechanism to configure a device's TCP/IP network interface. With this parameter it is possible i.e. to read or write the device's IP Address and Network Mask.

Class Code: $F5_{hex}$

Attribute ID	Access	Name	Data Type	Description
01_{hex}	Get	Status	DWORD	Interface status, details in chapter 2.3.1
02_{hex}	Get	Configuration Capability	DWORD	Interface capability flags, details in chapter 2.3.2
03_{hex}	Set	Configuration Control	DWORD	Interface control flags, details in chapter 2.3.3
04_{hex}	Get	Physical Link Object	STRUCT of:	Path to physical link object
		Path size	UINT	Size of path
		Path	Padded EPATH	Logical segments identifying the physical link object
05_{hex}	Set	Interface Configuration	STRUCT of:	TCP/IP network interface configuration
		IP Address	UDINT	The device's IP address
		Network Mask	UDINT	The device's network mask
06_{hex}	Set	Host Name	STRING	

2.3.1 Status Instance Attribute (01_{hex})

Bit(s)	Called	Definition	
0-3	Interface Configuration Status	Indicates the status of the Interface Configuration attribute.	0 = The Interface Configuration attribute has not been configured. 1 = The Interface Configuration attribute contains valid configuration obtained from BOOTP, DHCP or nonvolatile storage. 2 = The Interface Configuration attribute contains valid configuration, obtained from hardware settings (e.g.: pushwheel, thumbwheel, etc.) 3-15 = Reserved for future use.
4	Mcast Pending	Indicates a pending configuration change in the TTL Value and/or Mcast Config attributes. This bit shall be set when either the TTL Value or Mcast Config attribute is set, and shall be cleared the next time the device starts.	
5-31	Reserved	Reserved for future use and shall be set to zero.	

2.3.2 Configuration Instance Attribute (02_{hex})

Bit(s)	Called	Definition
0	BOOTP Client	1 (TRUE) shall indicate the device is capable of obtaining its network configuration via BOOTP.
1	DNS Client	Not supported
2	DHCP Client	1 (TRUE) shall indicate the device is capable of obtaining its network configuration via DHCP.
3	DHCP-DNS Update	Not supported
4	Configuration Settable	1 (TRUE) shall indicate the Interface Configuration attribute is settable. Some devices, for example a PC or workstation, may not allow the Interface Configuration to be set via the TCP/IP Interface Object.
5-31	Reserved	Reserved for future use and shall be set to zero.

2.3.3 Configuration Control Inst. Attribute (04_{hex})

Bit(s)	Called	Definition	
0-3	Startup Configuration	Determines how the device shall obtain its initial configuration at start up.	0 = The device shall use the interface configuration values previously stored (for example, in non-volatile memory or via hardware switches, etc). 1 = The device shall obtain its interface configuration values via BOOTP. 2 = The device shall obtain its interface configuration values via DHCP upon start-up. 3-15 = Reserved for future use.

2.3.4 Physical Link Object (05_{hex})

This attribute identifies the object associated with the underlying physical communications interface (e.g., an 802.3 interface). There are two components to the attribute: a Path Size (in UINTs) and a Path. The Path shall contain a Logical Segment, type Class, and a Logical Segment, type Instance that identifies the physical link object. The maximum Path Size is 6 (assuming a 32 bit logical segment for each of the class and instance).

The physical link object itself typically maintains link-specific counters as well as any link specific configuration attributes. If the CIP port associated with the TCP/IP Interface Object has an Ethernet physical layer, this attribute shall point to an instance of the Ethernet Link Object (class code = $F6_{hex}$). When there are multiple physical interfaces that correspond to the TCP/IP interface, this attribute shall either contain a Path Size of 0, or shall contain a path to the object representing an internal communications

interface (often used in the case of an embedded switch).
For example, the path could be as follows:

Path	Meaning
0-3	[20] = 8 bit class segment type; [F6] = Ethernet Link Object class; [24] = 8 bit instance segment type; [01] = instance 1.

2.3.5 Interface Configuration (06_{hex})

Name	Meaning
IP Address	The device's IP address. Value of 0 indicates no IP address has been configured. Otherwise, the IP address shall be set to a valid Class A, B, or C address and shall not be set to the loopback address (127.0.0.1).
Network mask	The device's network mask. The network mask is used when the IP network has been partitioned into subnets. The network mask is used to determine whether an IP address is located on another subnet. Value of 0 indicates no network mask address has been configured.

2.3.6 Host Name

Name	Meaning
Host Name	ASCII characters. Maximum length is 64 characters. Shall be padded to an even number of characters (pad not included in length). A length of 0 shall indicate no Host Name is configured.

2.4 Ethernet Link Object

Class Code: $F6_{hex}$

Attribute ID	Access	Name	Data Type	Description	Semantics of Values
01_{hex}	Get	Revision	UINT	Revision of this object	The minimum value shall be 1. Shall be 2 or greater if instance attribute 6 is implemented. Shall be 3 if any instance attributes 7-10 are implemented. The maximum value shall be 3.
02_{hex}	Get	Max Instance	UINT	Maximum instance number of an object currently created in this class level of the device	The largest instance number of a created object at this class hierarchy level
03_{hex}	Get	Number of Instances	UINT	Number of object instances currently created at this class level of the device	The number of object instances at this class hierarchy level

2.4.0 Instance Attributes

ID	Access	Name	Data Type	Description of Attribute	Semantics of Values
1	Get	Interface Speed	UINT	Interface speed currently in use	Speed in Mbps (e.g., 10, 100
2	Get	Interface Flags	DWORD	Interface status flags	See chapter 2.4.1
3	Get	Physical Address	ARRAY of 6 USINTs	MAC layer address	Displayed format "XX-XX-XX-XX-XX-XX"
6	Set	Interface Control	STRUCT of:	Configuration for physical interface	
		Control Bits	WORD	Interface Control Bits	See table below
		Forced Interface Speed	UINT	Speed at which the interface shall be forced to operate	Speed in Mbps (10 or 100)
7	Get	Interface Type	USINT	Type of interface	1 = The interface is internal to the device, i.e. in the case of an embedded switch 2 = Twisted-pair (e.g. 100Base-TX)
8	Get	Interface State	USINT	Current state of the interface	0 = No link 1 = The interface is enabled and is ready to send and receive data
10	Get	Interface Label	SHORT_STRING	Human readable identification	„Internal switch" or „External Port 1" or „External Port 2"

Control Bits

Bit(s)	Called	Definition
0	Auto-negotiate	802.3 link Auto-negotiation: 0 = disabled, 1 = enabled (standard) If Auto-negotiation is disabled then the device shall use the settings indicated by the Forced Duplex Mode and Forced Interface Speed bits.
1	Forced Duplex Mode	If Auto-negotiation bit = 0 the Forced Duplex Mode bit indicates whether the interface shall operate in full or half duplex mode. 0 = Half Duplex, 1 = Full Duplex
2-15	Reserved	Shall be set to zero

Example

Use on Transmit data size double (4 bytes) 00000064
for Auto-negotiation = disable on 100 MBaud

2.4.1 Interface Flags

Bit(s)	Called	Definition
0	Link Status	Indicates whether or not the Ethernet 802.3 communications interface is connected to an active network. 0 indicates an inactive link; 1 indicates an active link. The determination of link status is implementation specific. In some cases devices can tell whether the link is active via hardware/driver support. In other cases, the device may only be able to tell whether the link is active by the presence of incoming packets.
1	Half/Full Duplex	Indicates the duplex mode currently in use. 0 indicates the interface is running half duplex; 1 indicates full duplex. Note that if the Link Status flag is 0, then the value of the Half/Full Duplex flag is indeterminate.
2-4	Negotiation Status	Indicates the status of link auto-negotiation 0 = Auto-negotiation in progress. 1 = Auto-negotiation and speed detection failed. Using default values for speed and duplex. Default values are product-dependent; recommended defaults are 10Mbps and half duplex. 2 = Auto negotiation failed but detected speed. Duplex was defaulted. Default value is product-dependent; recommended default is half duplex. 3 = Successfully negotiated speed and duplex. 4 = Auto-negotiation not attempted. Forced speed and duplex.
5	Manual Setting Requires Reset	0 indicates the interface can activate changes to link parameters (auto-negotiate, duplex mode, interface speed) automatically. 1 indicates the device requires a Reset service be issued to its Identity Object in order for the changes to take effect.
6	Local Hardware Fault	0 indicates the interface detects no local hardware fault; 1 indicates a local hardware fault is detected. The meaning of this is product-specific. Examples are an AUI/MII interface detects no transceiver attached or a radio modem detects no antennae attached. In contrast to the soft, possible self-correcting nature of the Link Status being inactive, this is assumed a hard-fault requiring user intervention.
7	Reserved	Shall be set to zero

2.4.2 Common Services

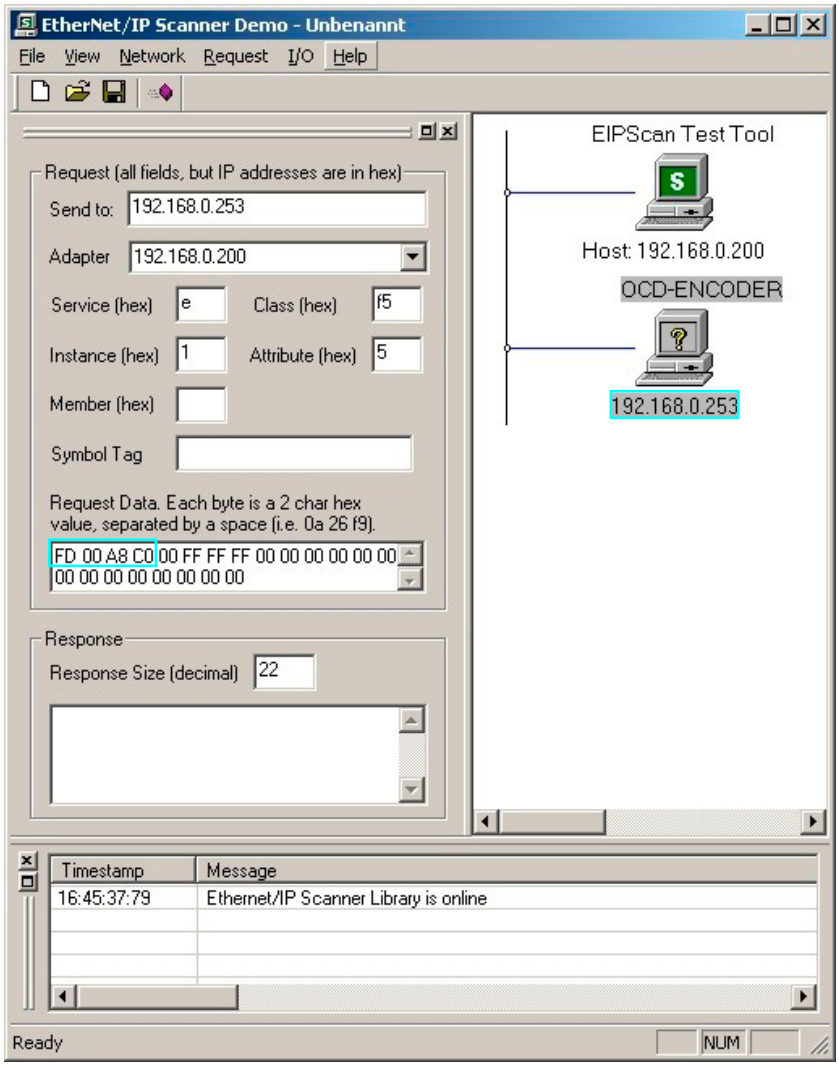
Service Code	Class	Instance*	Service Name	Description of Service
$0E_{hex}$	Condi-tional	Required	Get_Attribute _Single	Returns the contents of the specified attribute
10_{hex}	n/a	Conditional	Set_Attribute _Single	Modifies a single attribute

2.4.3 Link Object Instances

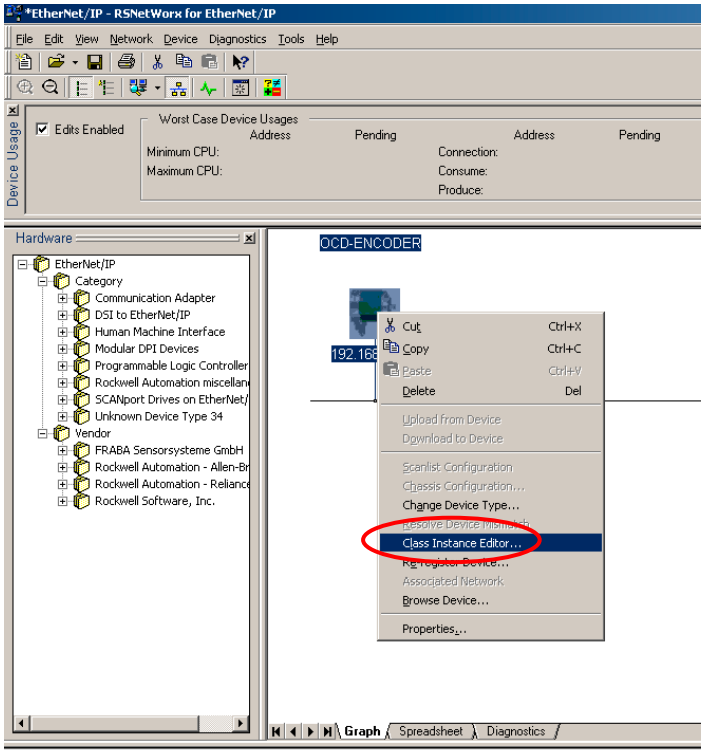
Instance	Description
1	Internal interface
2	Intern switch Port 1
3	Intern switch Port 2

2.5 Setting parameters with scanners

There are several external scanners for EtherNet/IP available. RS-NetWorks™ has one such scanner. In the figure is an example where the IP-Address (FD 00 A8 C0 complies 192.168.0.253), the Subnet (00 FF FF FF complains 255.255.255.0), Gateway (00 00 00 00), DNS1 (00 00 00 00), DNS2 (00 00 00 00) and Domain Name = "" (ASCII Character max length = 48 bytes) was read out of the encoder.

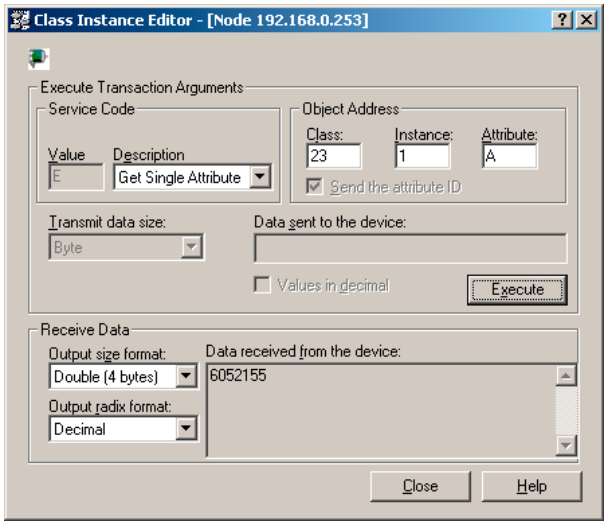


In RSNetWorx is a scanner available too. In the next section is a sample to set the Preset value.



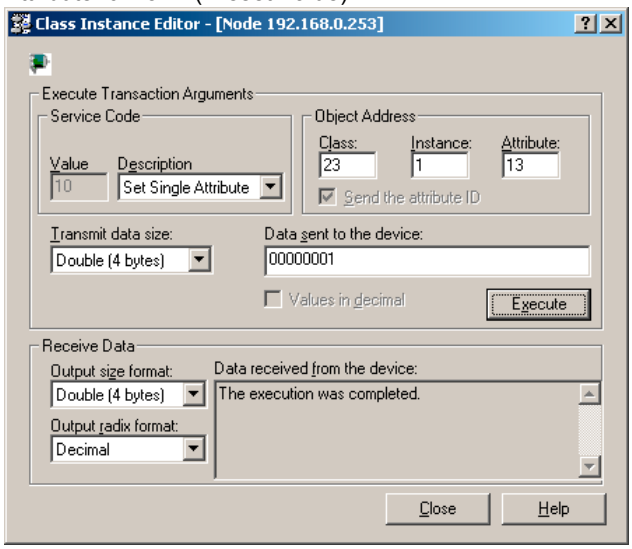
2.5.1 Read out position value

Get Single Attribute Position sensor value:
Class: 0x23 (Position sensor object)
Instance: 0x01
Attribute: 0x0A (Position Value)



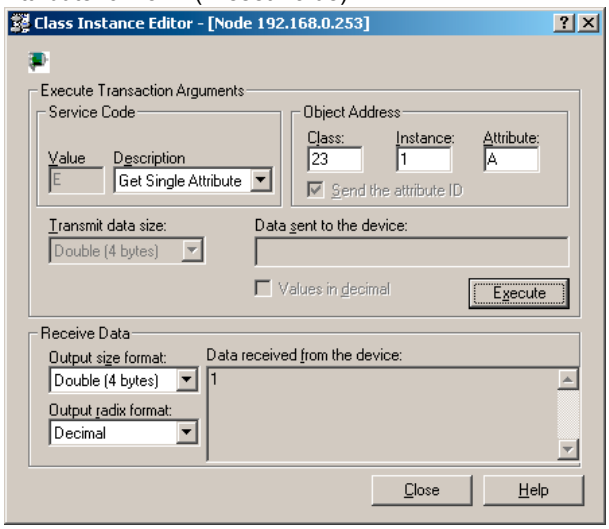
2.5.2 Set preset value

Set Single Attribute Position Preset Value to 1
Class: 0x23 (Position sensor object)
Instance: 0x01
Attribute: 0x13 (Preset Value)



2.5.3 Get preset value

Get Single Attribute Position Value
Class: 0x23 (Position sensor object)
Instance: 0x01
Attribute: 0x13 (Preset Value)



3 Diagnostic

LED	Color	EtherNet/IP name	Description
Active1	Yellow	Network Status Indicator 1	Details in table 2
Link1	Green		
Active2	Yellow	Network Status Indicator 2	Details in table 2
Link2	Green		
Stat1	Green	Module Status Indicator	Details in table 1
Stat2	Red		

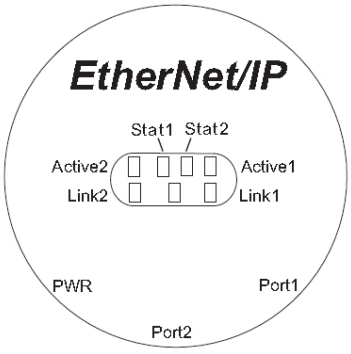


Table 1: Module Status Indicator Stat1/Stat2

LED	Summary	Requirements
Steady Off	No power	
Steady On Green	Device operational	If the device is operating correctly, the module status indicator shall be steady green
Flashing Green 1	Standby	If the device has not been configured including the IP-Address, the module status indicator shall be flashing green with 1 Hz
Flashing Green 2	Missing IP	If the device does not have an IP-Address, the module status indicator shall be flashing green with 4 Hz
Flashing Red	Minor fault	If the device has detected a recoverable minor fault. I.e. an incorrect or inconsistent configuration
Steady On Red	Major fault	If the device has detected a non-recoverable major fault
Flashing Red + Green	Self-test	While the device is performing its power up testing, the Stat1 and Stat2 LED shall be flashing red / green

Table 2: Network Status Indicator Stat2

LED	Summary	Requirements
Steady Off 	No power, no IP address	If the device does not have an IP address or is powered off
Steady Green 	Connected	If the device has at least one established connection (even to the Message router)
Flashing Green 	No connection	If the device has no established connections, but has obtained an IP address
Flashing Yellow 	Connection timeout	If one or more of the connections in which this device is the target has timed out. This shall be left only if all timed out connections are reestablished or if the device is reset
Steady Yellow 	Duplicate IP	If the device has detected that its IP address is already in use
Flashing Yellow / Green 	Self-test	While the device is performing its power up testing, the Stat1 and Stat2 LED shall be flashing yellow / green

4 Programmable Parameters

4.1 Encoder parameters for Position Sensor Object Class 23hex

4.1.1 Direction counting

This operating parameter can be used to select the code sequence. The parameter can set with Configuration Assembly and Explicit Messaging

Attribute ID	Default value	Value range	Data Type
$0C_{hex}$	0_{hex}	0_{hex} - 1_{hex}	Boolean

The parameter code sequence (complement) defines the counting direction of the process value **as seen on the shaft** (clockwise or counter clockwise). The counting direction is defined in the attribute $0C_{hex}$:

Bit 0	Counting direction	Position values
0	CW	Increase
1	CCW	Decrease

Bit 0	Scaling function on/off
0	on
1	off

4.1.2 Scaling function control

If the Scaling function control is deactivated then complains the output value the physical resolution.

Attribute ID	Default value	Value range	Data Type
$0E_{hex}$	1_{hex}	0_{hex} - 1_{hex}	Boolean

This parameter can be set with Configuration Assembly and Explicit Messaging

4.1.3 Resolution per revolution

The parameter resolution per revolution is used to program the encoder to set a desired number of steps per revolution. Each value between 1 and the maximum (see type label) can be realized. The parameter can set with Configuration Assembly and Explicit Messaging. Scaling function control **must be switch on** for customer parameters!

Attribute ID	Default value	Value range	Data Type
10_{hex}	(*)	0_{hex} - 10000_{hex}	Double Integer32

(*) see type label, Maximum resolution:
16Bit Encoder: $10{,}000_{hex}$ (65,536)

When the value is set larger than 8192 for a 13Bit encoder, the process value of the encoder will not be single stepped and values will be skipped while rotating the shaft. So, it is recommended, to keep the measuring steps per revolution below 8192 measuring steps.

4.1.4 Total resolution

This value is used to program the desired number of measuring steps over the total measuring range. This value must not exceed the total resolution of the encoder with 25 bit = 33,554,432 steps. Please note the value written on the type shield. The parameter can set with Configuration Assembly and Explicit Messaging Scaling function control **must be switch on** for customer parameters!

Attribute ID	Default value	Value range	Data Type
11_{hex}	(*)	0_{hex} - $40,000,000_{hex}$	Unsigned Integer 32

(*) see type shield
Maximum total resolution
30 Bit Encoder: $40,000,000_{hex}$ (1,073,741,824)

4.1.5 Preset value

The preset value is the desired position value, which should be reached at a certain physical position of the axis. The position value of the encoder is set to the desired process value by the parameter preset. The preset value must not exceed the parameter total measuring units. The parameter can set with Explicit Messaging.
Set the preset value only in standstill! Use the save commando from chapter 2.2.1 to save the preset value in the non volatile memory.

Attribute ID	Default value	Value range	Data Type
13_{hex}	0_{hex}	0_{hex} - total measuring range	Unsigned Integer 32

4.1.6 Velocity Format

Default value for Velocity Format is steps per second. This parameter can be set with Configuration Assembly and Explicit Messaging.

Attribute ID	Default value	Value range	Data length
19_{hex}	$1F04_{hex}$	$1F04_{hex}$	Steps per second
		$1F05_{hex}$	Steps per millisecond
		$1F06_{hex}$	Steps per microsecond
		$1F07_{hex}$	Steps per minute
		$1F0F_{hex}$	RPM

4.1.7 Velocity Filter

To manage the noise of the velocity it is possible to switch between three classes.

Attribute ID	Default value	Value range	Description	Data Type
66_{hex}	0_{hex}	0_{hex} / 1_{hex} / 2_{hex}	0 = Fine, 1 = Middle, 2 = Raw	Double Integer

4.1.8 Endless Shaft

Normally the period, i.e. “Total resolution” / “measuring units” per revolution must be an integer and it must fit an integer number of times (integer multiple) into 4096 for an encoder with 12 Bit for the revolutions. So the following equation must apply:

(4096 x measuring units per revolution) / Total resolution = integer

But with this EtherNet/IP encoder it is possible to solve this problem. If the Endless Shaft is activated then this problem will be solved by the encoder. The default value is Auto. In this case the encoder checks if the parameters need the endless shaft. The parameter can be set only with Explicit Messaging.

Note: The internal software routine only works if the encoder is in operation. If it is necessary to turn the encoder shaft more than 1024 revolutions without power supply this can lead to problems (the internal routine will not work without power supply). In this case the rule ahead should be observed even with new devices.

Attribute ID	Default value	Value range	Description	Data Type
65_{hex}	2_{hex}	0_{hex} / 1_{hex} / 2_{hex}	0 = Off, 1 = On, 2 = Auto	Double Integer

5. Installation

5.1 Electrical connection

The rotary encoder is connected by a 4 pin M12 connector for the power supply and two 4 pin, D-coded M12 connector for Ethernet.

The Encoder uses a second D-coded connector and provides integrated switch functionality. On or in the packaging of the connector is the mounting description.

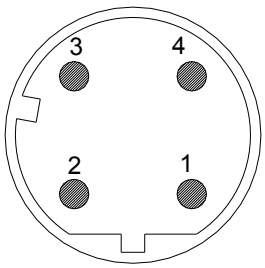
Connector Ethernet
4 pin female, D-coded

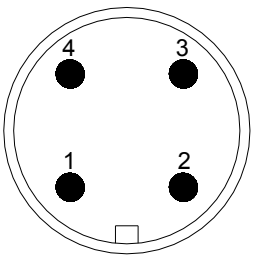
Pin Number	Signal
1	Tx +
2	Rx +
3	Tx -
4	Rx -

Connector power supply
4 pin male, A-coded

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

Sketch as seen on the encoder





5.2 Ethernet cables

5.2.1 RJ45 – M12 crossed

Signal	RJ45 Pin	M12 Pin
Tx+	1	2
Tx-	2	4
Rx+	3	1
Rx-	6	3

5.2.2 RJ45 – M12 straight

Signal	RJ45 Pin	M12 Pin
Tx+	1	1
Tx-	2	3
Rx+	3	2
Rx-	6	4

5.2.3 M12 – M12 straight

Signal	M12 Pin	M12 Pin
Tx+	1	1
Tx-	2	2
Rx+	3	3
Rx-	4	4

6 Power On

After power on the LED's on the absolute rotary encoder will flash between green and red or yellow.

7 Installation

7.1 Rockwell configuration tools

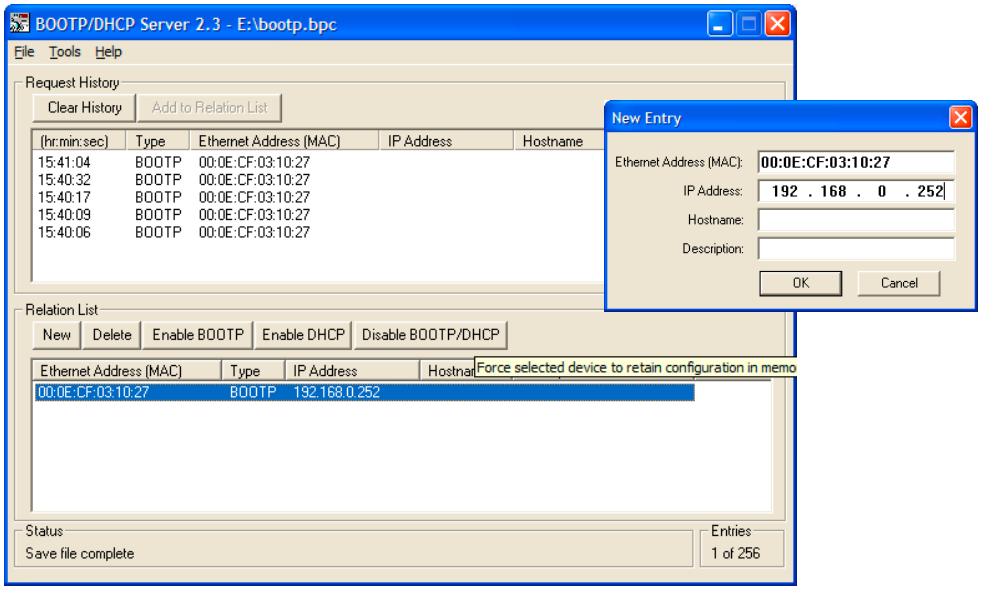
7.1.1 Setting IP-Address (BOOTP/DHCP)

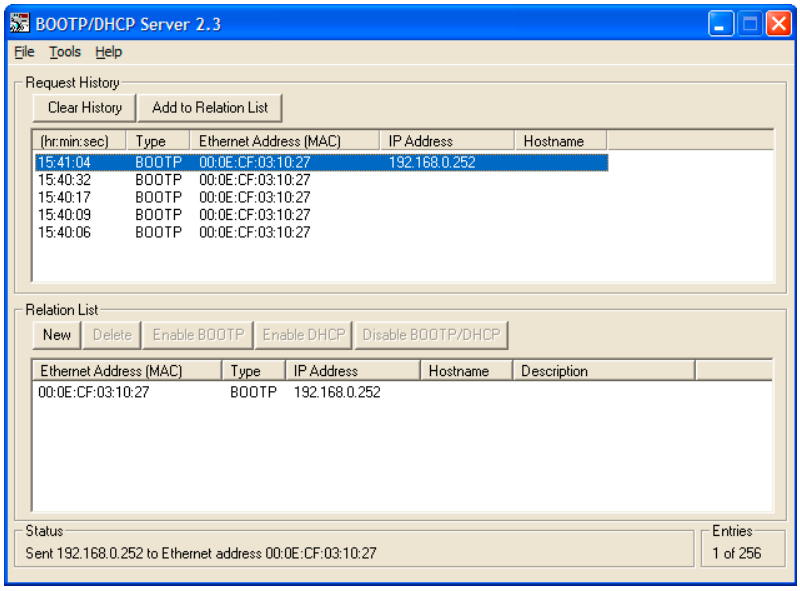
To set the IP Address there are special tools available. I.e. the BOOTP/DHCP Server is installed with the software package from RSNetWorx™. The server scan the network for the MAC Addresses of all products with active BOOTP or DHCP. If one MAC address is selected in the Request History then the IP Address can be set by the "Add to Relation List" button. The MAC Address of each EtherNet/IP encoder is available on the type label. Note: After a power up the encoder send the BOOTP or DHCP request often. But after several time comes no answer the frequency of requests decrease. A power up after a longer pause could solve the missing requests.

If not all encoders are listed in the BOOTP/DHCP Server then check the following points:

- LED status of the encoder OK?
- Is the Network setting correct?
- Is the BOOTP and/or DHCP enabled?

If the encoder has got his IP-Address, the BOOTP and DHCP must be disabled with the corresponding button. Otherwise the encoder start up to get a new IP-Address again. After setting the IP-Address the Status LED is flashing with 1 Hz. But in this case save the configuration in the File menu, because the products cannot be found by the BOOTP/DHCP Server. After loading this file the MAC Addresses and IP-Addresses are available and BOOTP or DHCP can be activated by the corresponding button. Possible IP-Range:
Class A-C (0.0.0.0 – 223.255.255.255) without Loopback range (127.x.x.x)
Referenced IP-Address range: 192.168.0.x





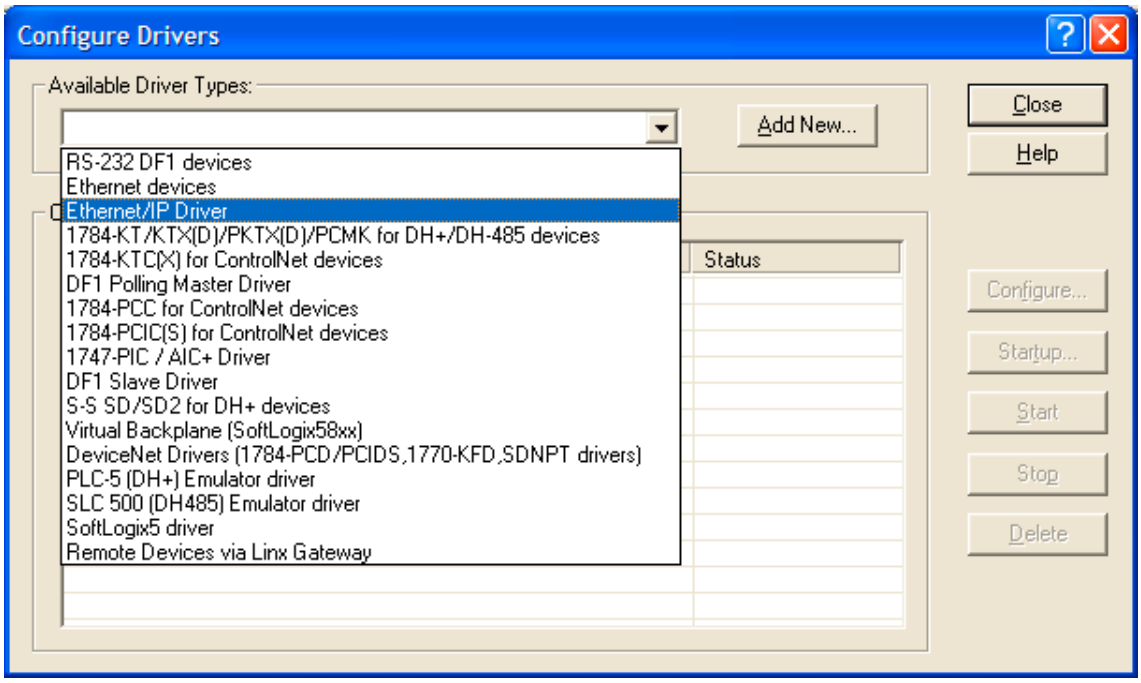
After setting the IP-Address with this tool the IP-Address will be available only after the next BOOTP request.

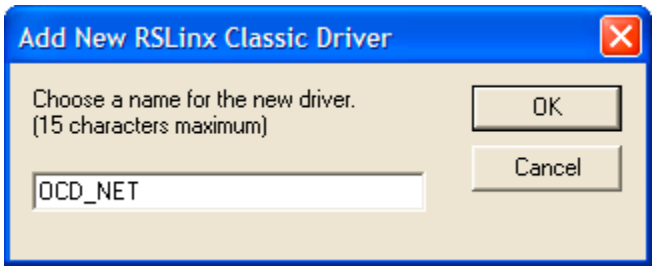
If the IP-Address is not known and BOOTP and DHCP are deactivated it is possible with a special tool to find the IP-Address or to activate BOOTP or DHCP. See details in chapter 7.3.

7.1.2 Configuration RSLinx Classic™

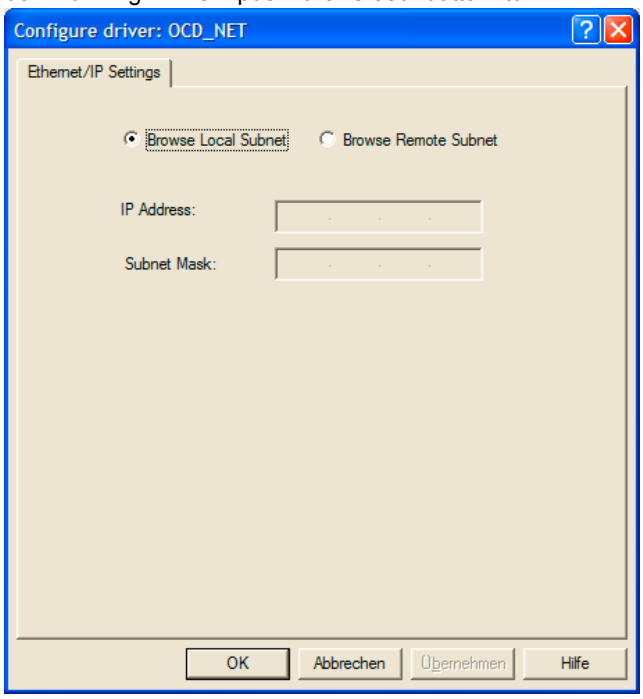
RSLinx™ is a complete communication server providing plant-floor device connectivity for a wide variety of Rockwell Software applications such as RSLogix™, RSNetWorx™,…

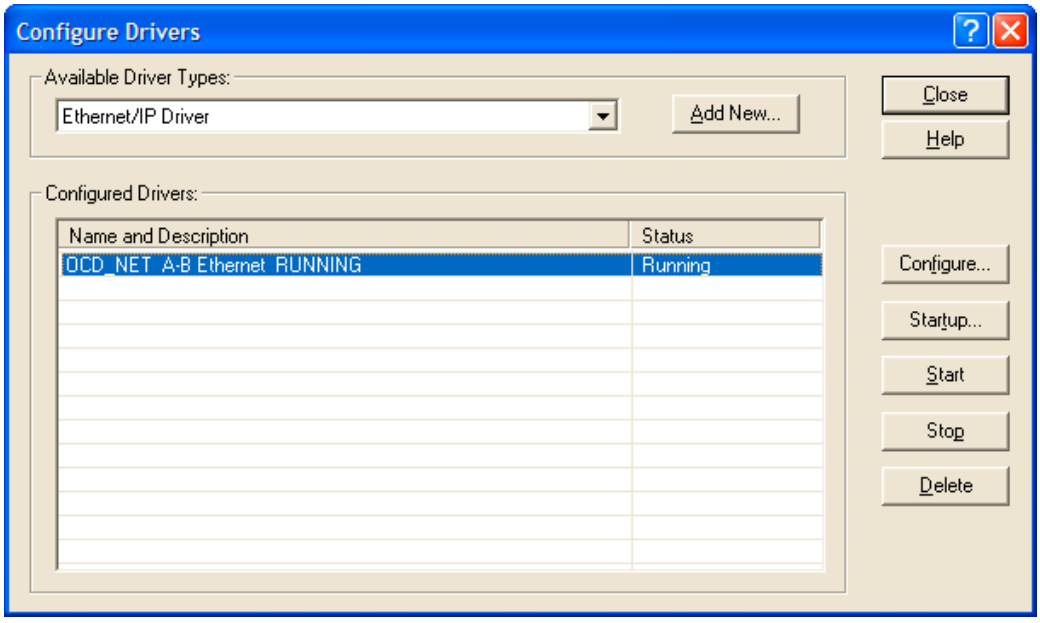
To start a new project add first a new RSLinx Classic™ Driver for EtherNet/IP under *Communications Configuration Drivers* and input the name.





Use *Browse Local Subnet* to find the EtherNet/IP components in the network. The status should be “Running”. Then push the *Close* button to finish this configuration.





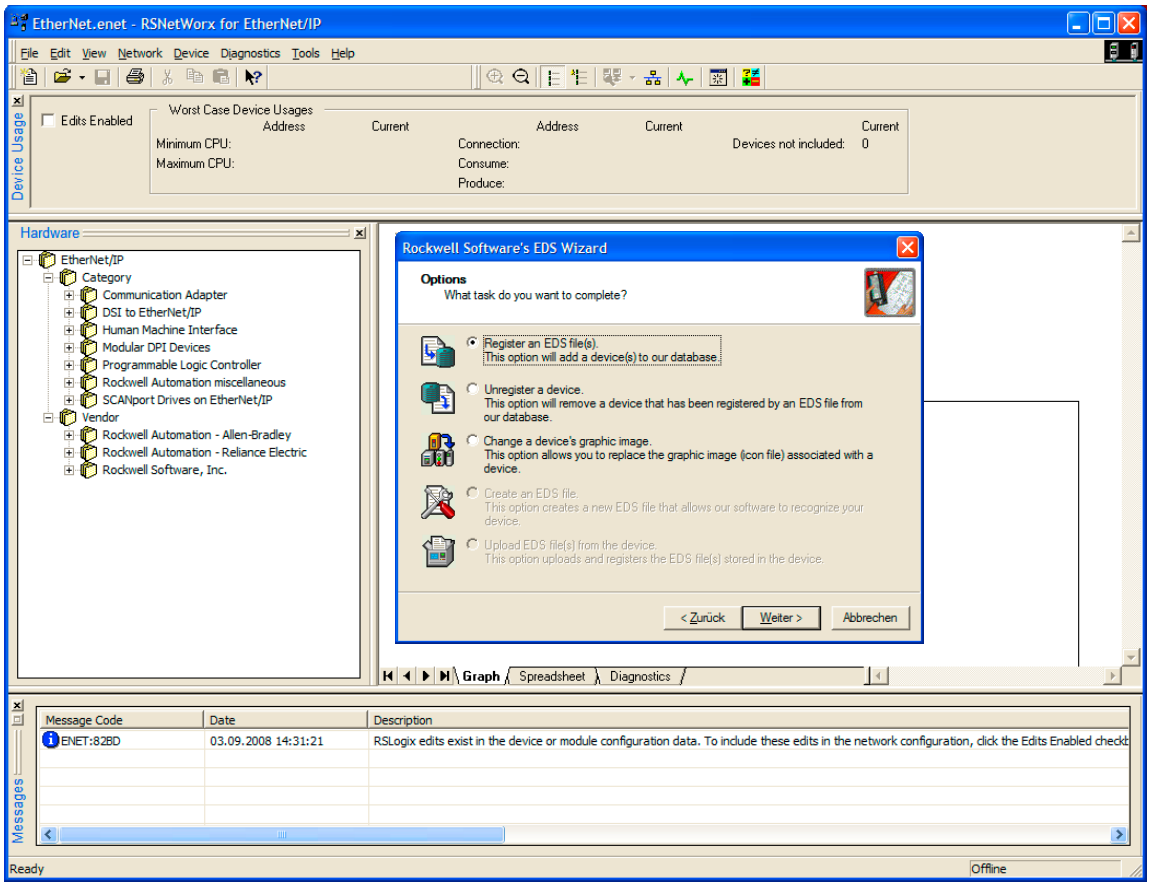
7.1.3 RSNetWorx™

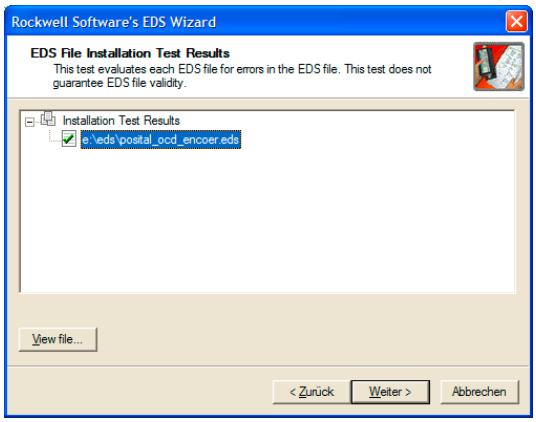
RSNetWorx™ products provide design and configuration management services for EtherNet/IP. The program defines and configures the devices on the network quickly through a simple software interface. This definition can take place offline using drag and drop operations or online by using RSLinx® to browse a EtherNet/IP network.

EDS Wizard

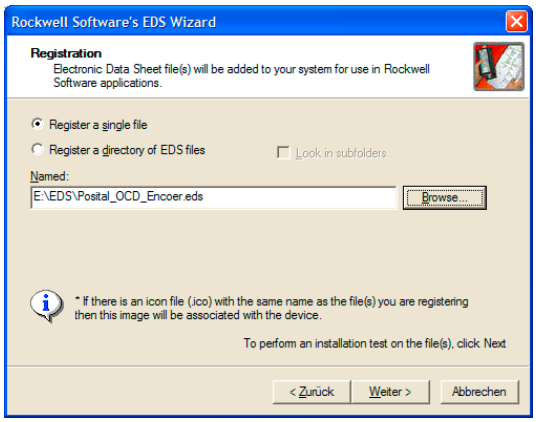
The EDS File contains information about device specific parameters as well as possible operating modes of the encoder. With this file you have a data sheet in an electronic format, which can be used to configure the device in the network, for example with RSNetWorx™ from Rockwell. In this sample the PLC uses address 192.168.0.100 and the encoder 192.100.0.252.

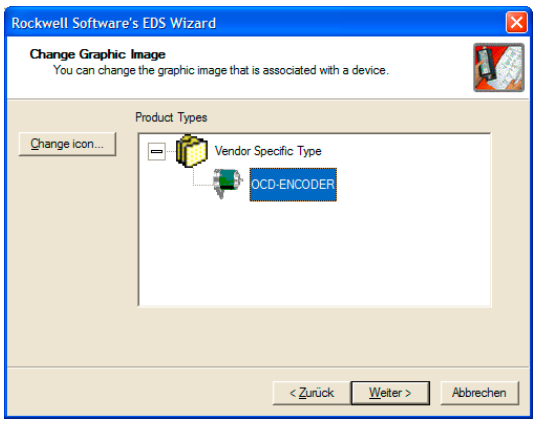
To install the EDS file the EDS Wizard has to be started, that can be done in the menu *Tools/EDS Wizard*. If the EDS Wizard is activated successfully the *Register an EDS File(s)* has to be chosen and after that the button *weiter*. In the next step the *Register a directory of EDS files* has to be chosen and with *Browse* the path of the EDS file(s). That is indicated in the next pictures.



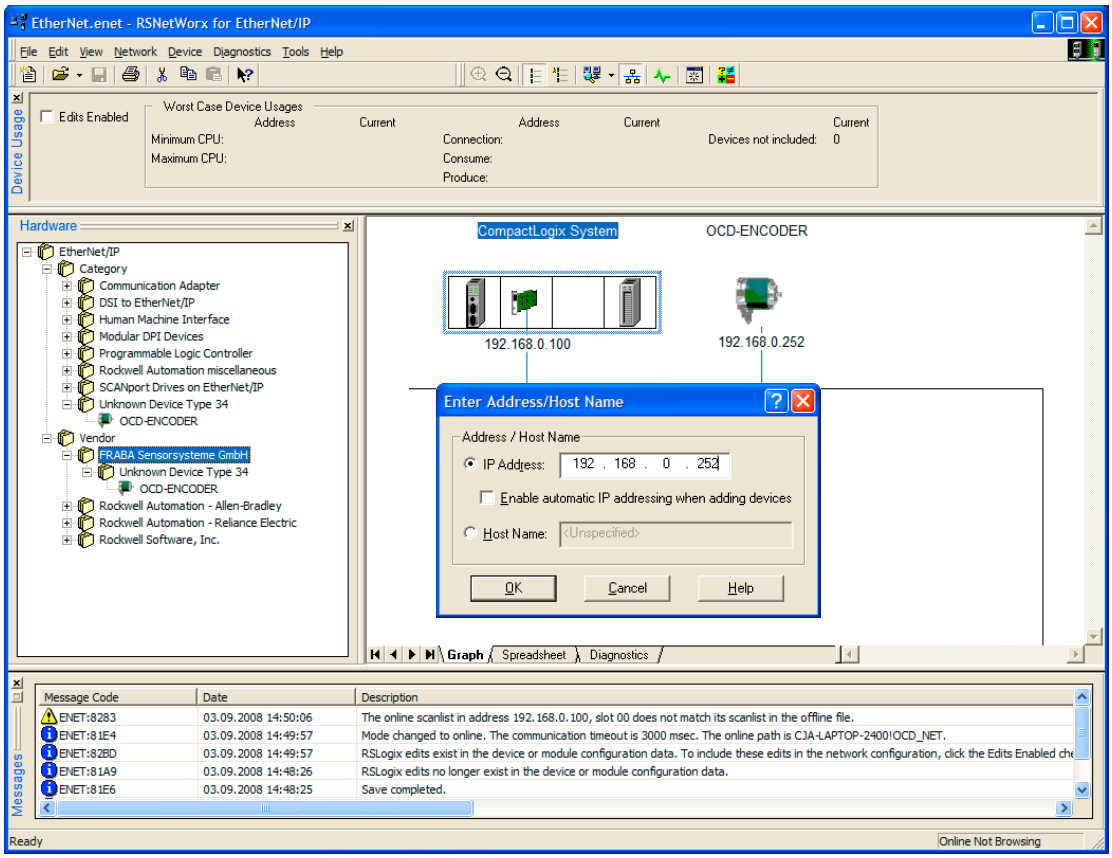


The Wizard finds all EDS files that are discarded in the choosing path and operates a test to check the EDS files on errors. In the next step pictures can be selected for the using nodes. With the button *weiter* the installation can be continued and finished.

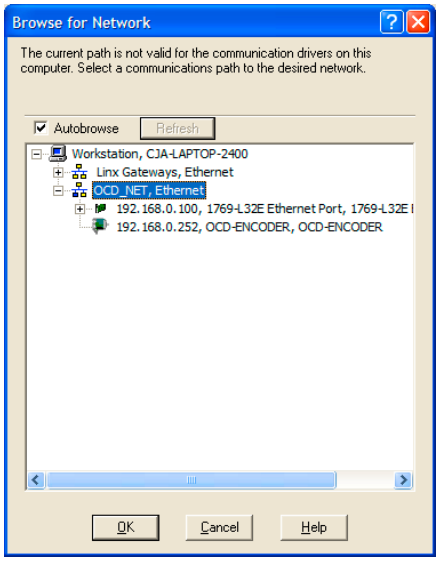




Load a saved *.enet file or start a new project. Add the devices per Drag and Drop to the net work line and set the IP-Address.



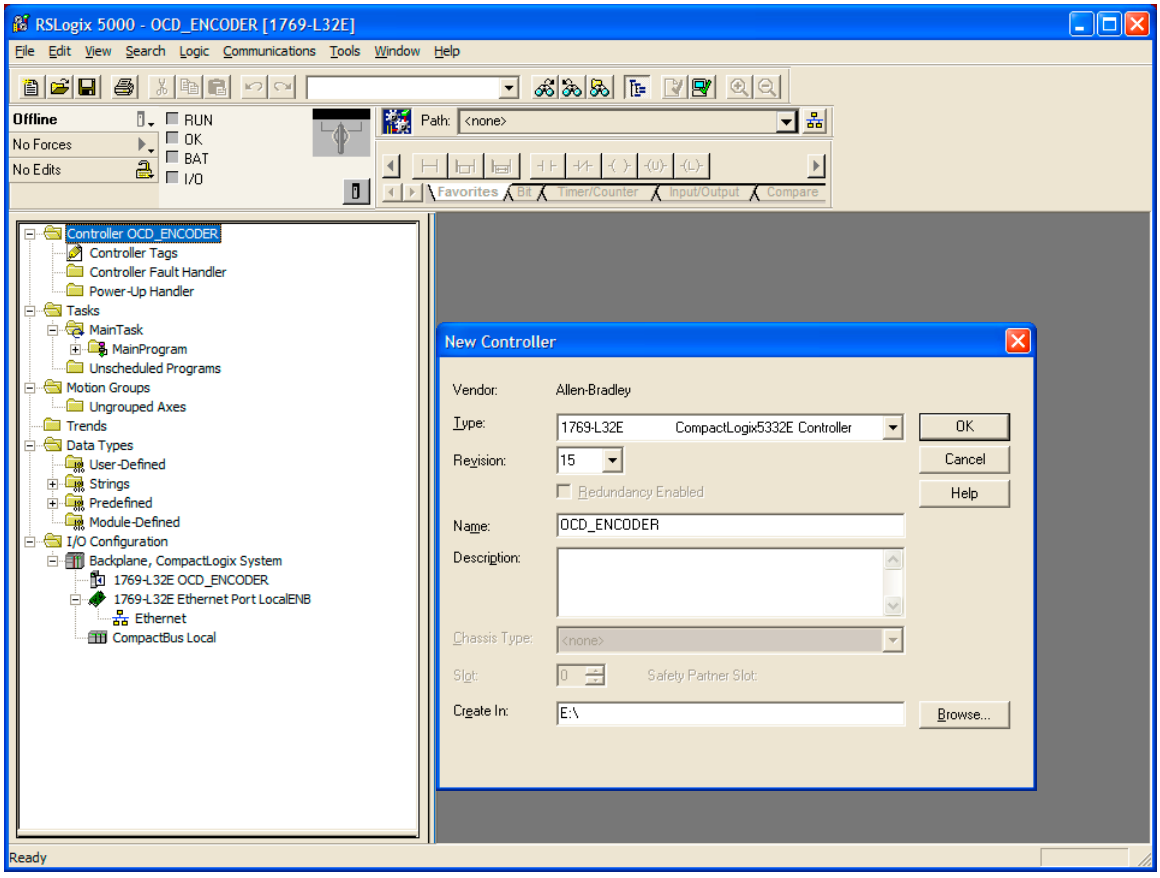
Optional browse the network with all devices with Button  or *Upload from Network*. So it is not necessary to set the IP-Address manually. For using this configuration in RSLogix save the *.enet file.



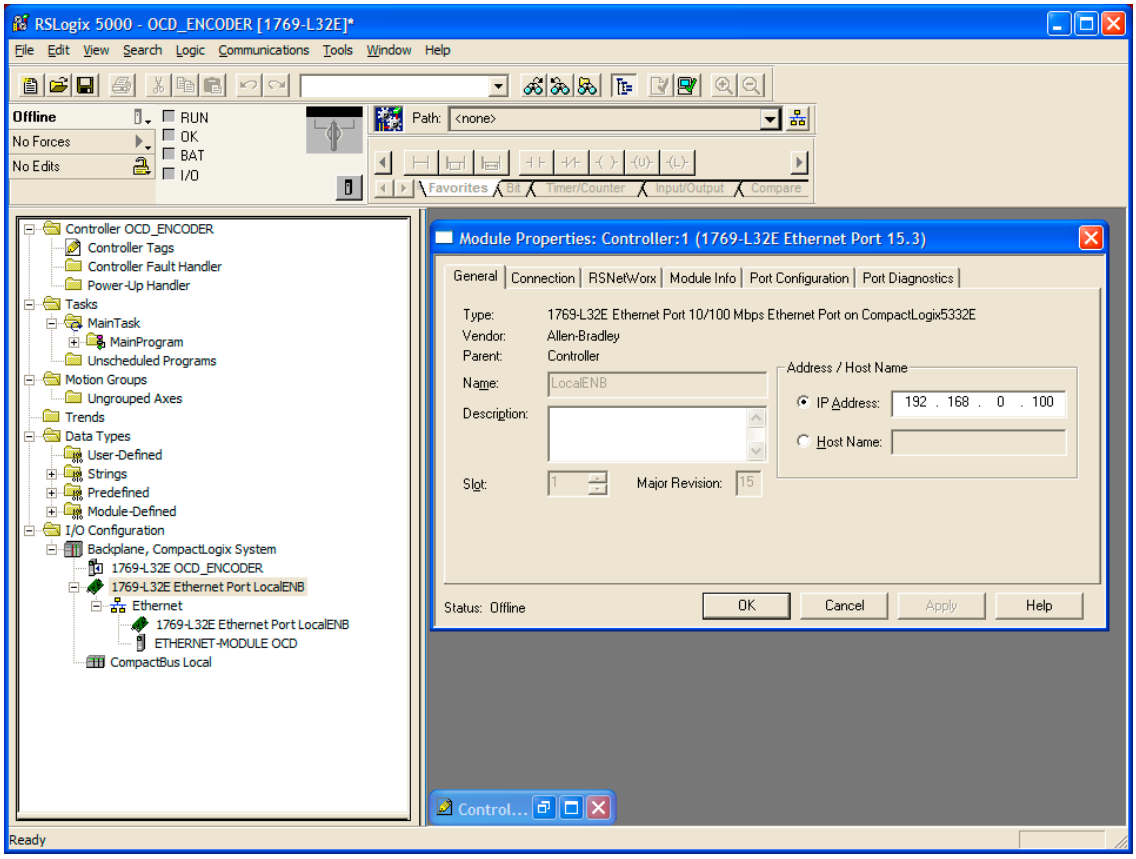
7.1.4 Configuration RSLogix 5000

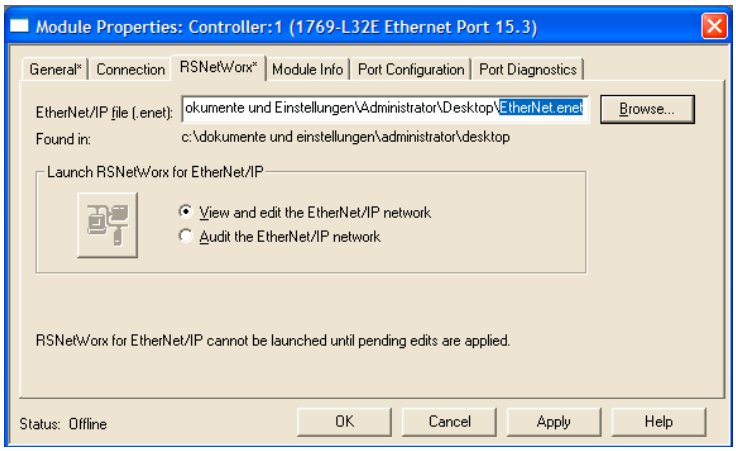
The RSLogix 5000 Series environment offers an easy-to-use, IEC61131-3 compliant interface, symbolic programming with structures and arrays, and a comprehensive instruction set that serves many types of applications. It supports relay ladder, structured text, function block diagram, and sequential function chart editors for you to develop application programs.

In the first step load a configuration or add a new controller and input a name. In this sample is used the CompactLogix5332E.

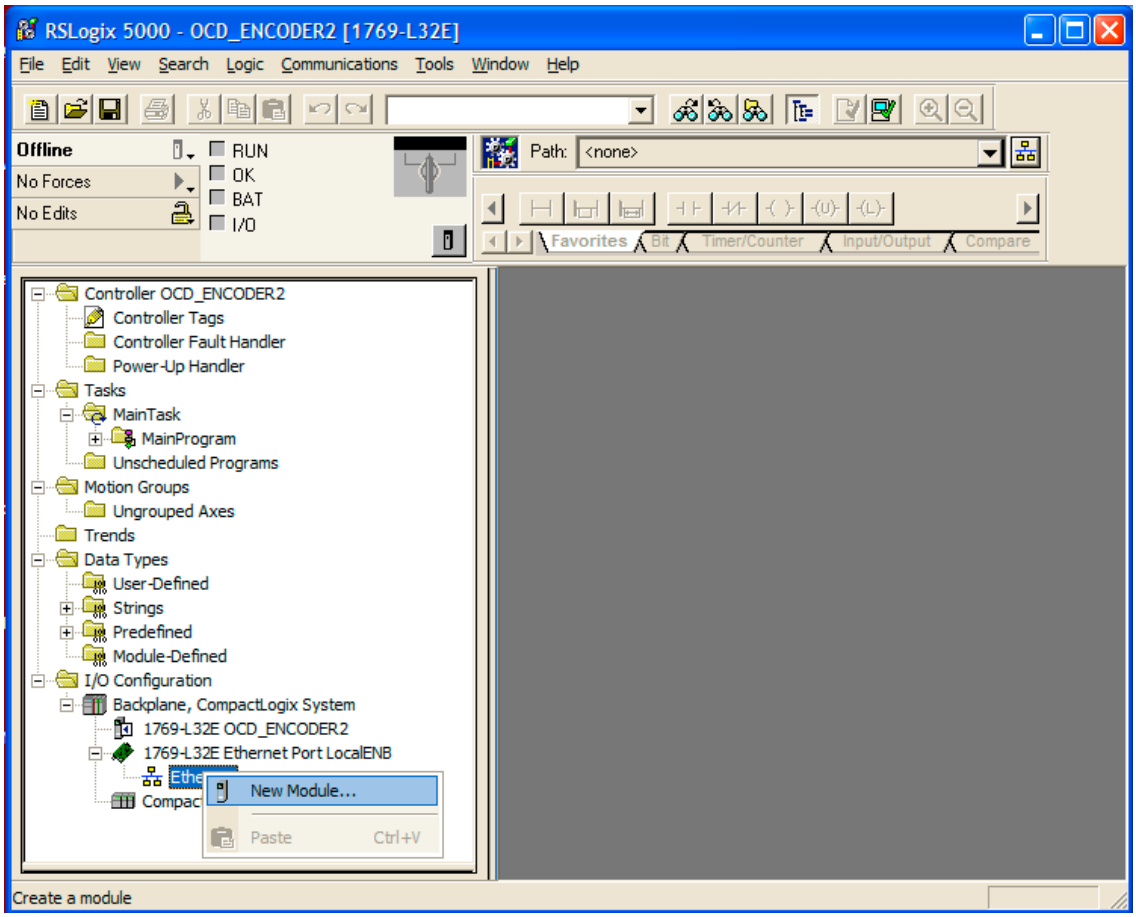


Start the configuration of the controller or load the *.enep file in the module properties of tab RSNetWorx™ that was created with RSNetWorx™.

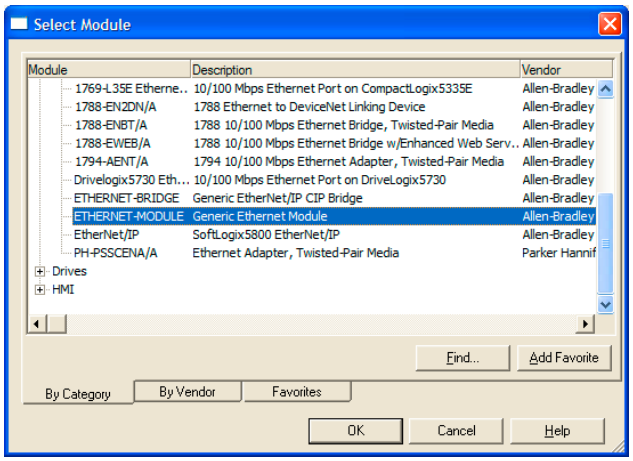




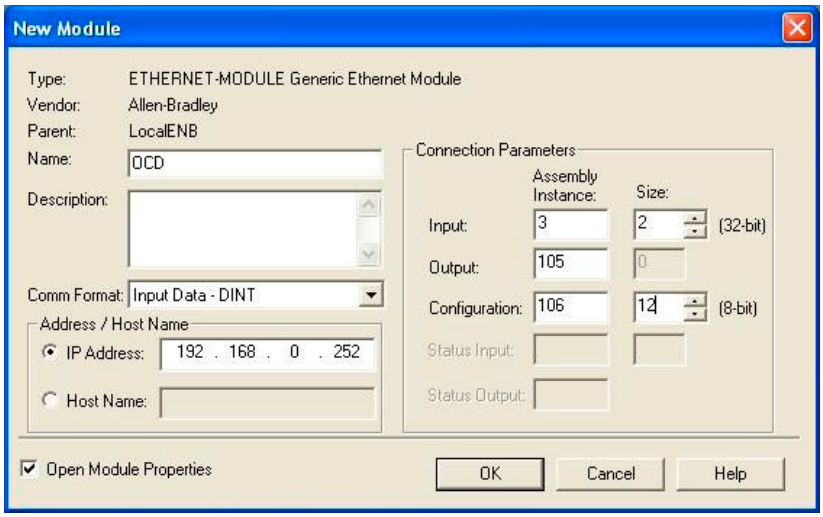
Select the network in the I/O Configuration and add New Module.



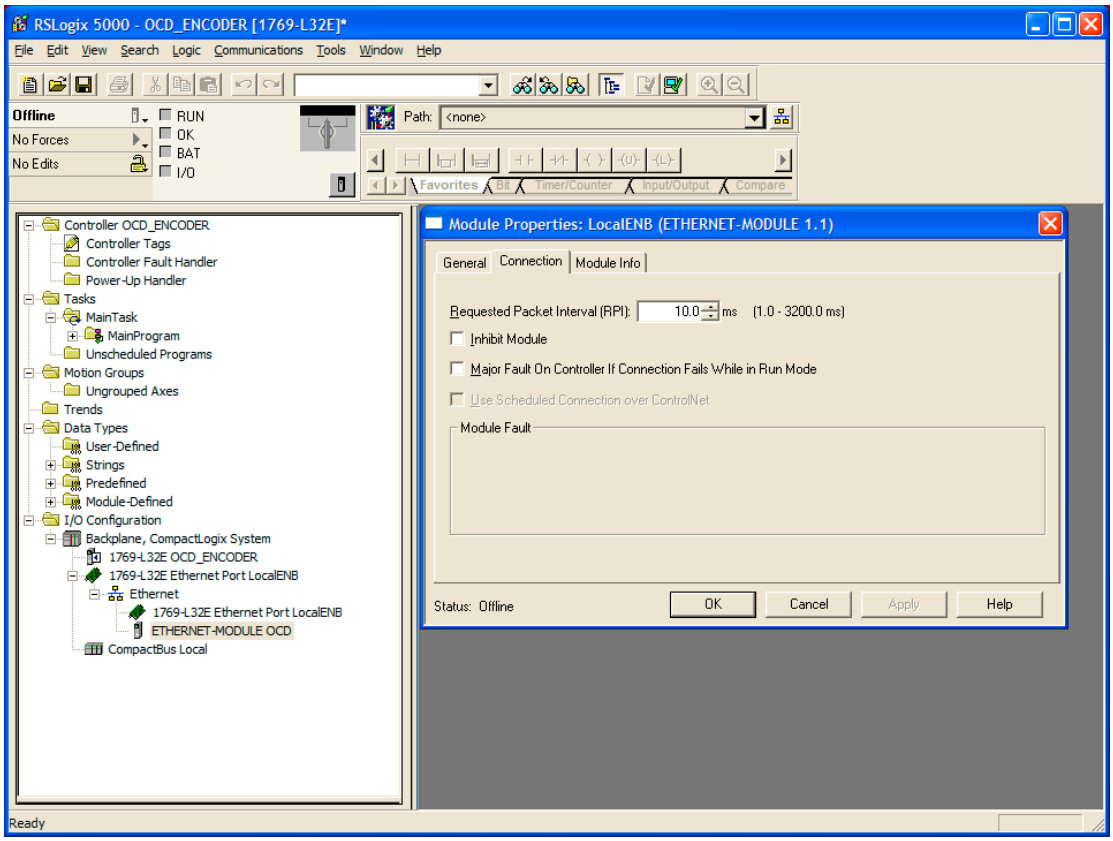
For using a Generic Device select the Generic Ethernet Module. Some PLC's support Encoder Devices too. Please check that the matching EDS file complies to the configuration of the encoder. The device type is programmable.



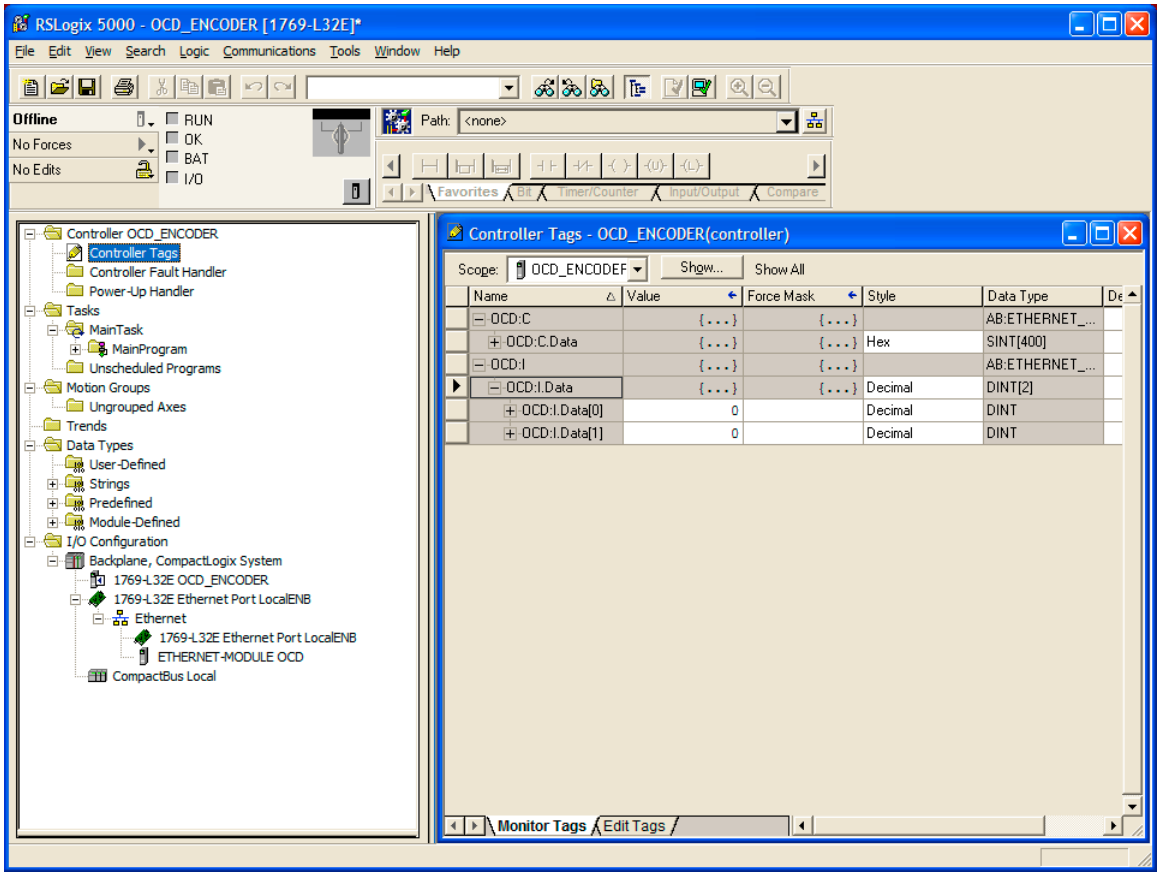
Set the Connection Parameters according the following figure.

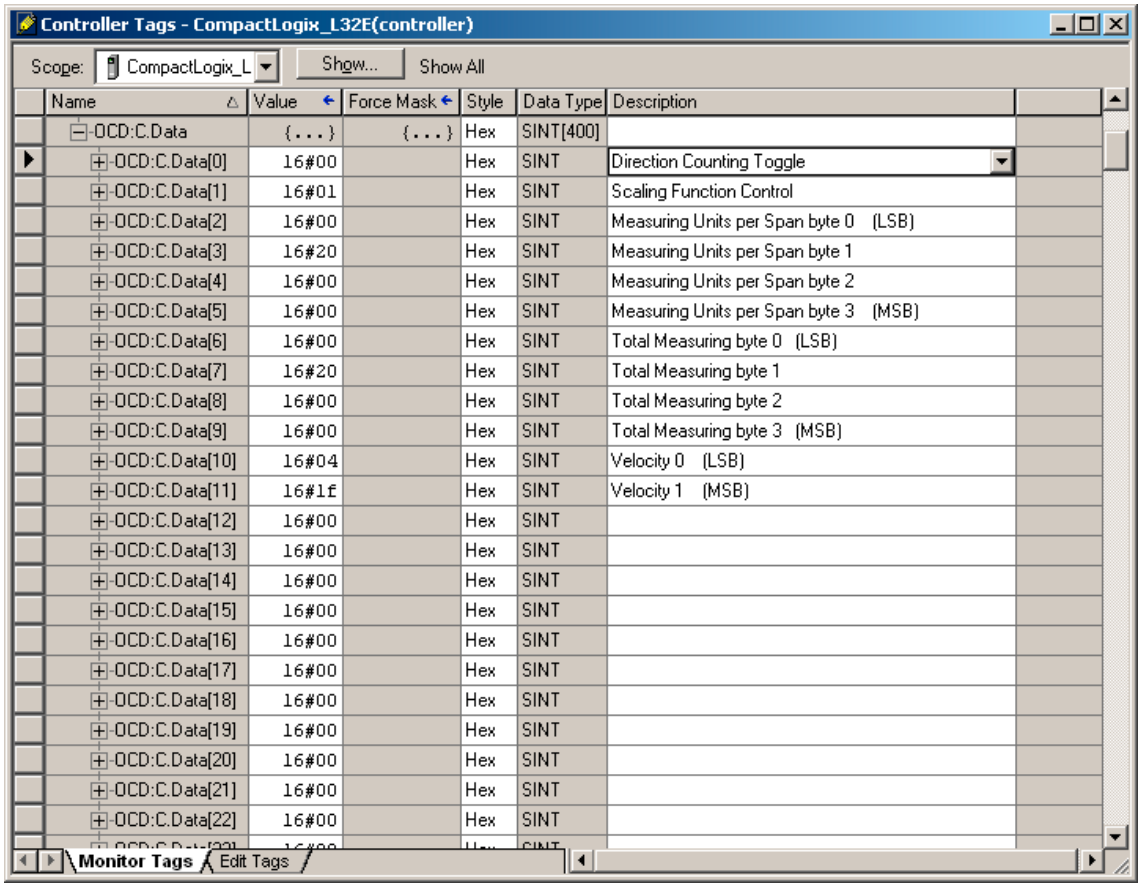


Set the cycle time.



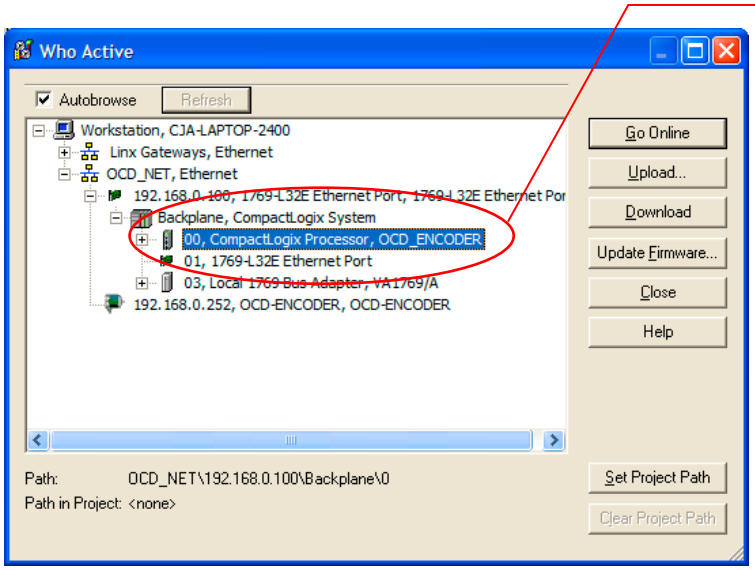
To read or write data use *Logic* - *Monitor Tags*



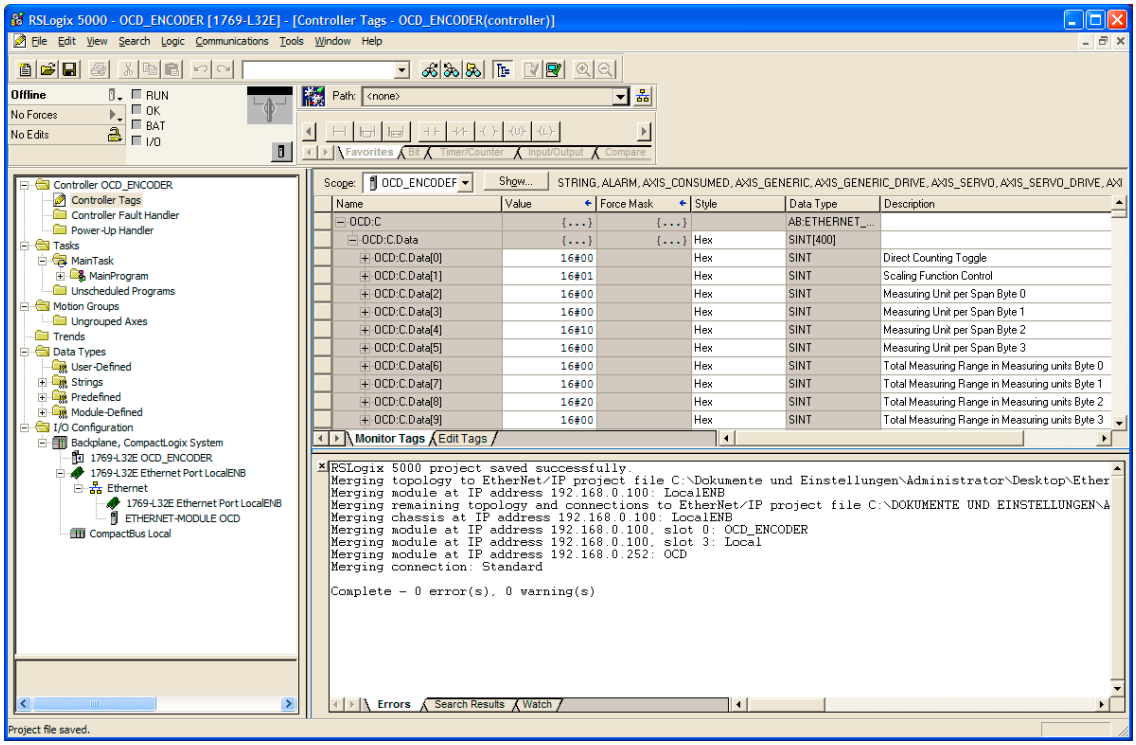


Name	Value	Force Mask	Style	Data Type	Description
OCD:C.Data	{...}	{...}	Hex	SINT[400]	
OCD:C.Data[0]	16#00		Hex	SINT	Direction Counting Toggle
OCD:C.Data[1]	16#01		Hex	SINT	Scaling Function Control
OCD:C.Data[2]	16#00		Hex	SINT	Measuring Units per Span byte 0 (LSB)
OCD:C.Data[3]	16#20		Hex	SINT	Measuring Units per Span byte 1
OCD:C.Data[4]	16#00		Hex	SINT	Measuring Units per Span byte 2
OCD:C.Data[5]	16#00		Hex	SINT	Measuring Units per Span byte 3 (MSB)
OCD:C.Data[6]	16#00		Hex	SINT	Total Measuring byte 0 (LSB)
OCD:C.Data[7]	16#20		Hex	SINT	Total Measuring byte 1
OCD:C.Data[8]	16#00		Hex	SINT	Total Measuring byte 2
OCD:C.Data[9]	16#00		Hex	SINT	Total Measuring byte 3 (MSB)
OCD:C.Data[10]	16#04		Hex	SINT	Velocity 0 (LSB)
OCD:C.Data[11]	16#1f		Hex	SINT	Velocity 1 (MSB)
OCD:C.Data[12]	16#00		Hex	SINT	
OCD:C.Data[13]	16#00		Hex	SINT	
OCD:C.Data[14]	16#00		Hex	SINT	
OCD:C.Data[15]	16#00		Hex	SINT	
OCD:C.Data[16]	16#00		Hex	SINT	
OCD:C.Data[17]	16#00		Hex	SINT	
OCD:C.Data[18]	16#00		Hex	SINT	
OCD:C.Data[19]	16#00		Hex	SINT	
OCD:C.Data[20]	16#00		Hex	SINT	
OCD:C.Data[21]	16#00		Hex	SINT	
OCD:C.Data[22]	16#00		Hex	SINT	

- If the value is 00 then the standard configuration will be used
- If the Paramter are out of range the maximum value of the encoder will be used as parameter
- To change parameters open *Communication Who Active*, *Go Offline*, *File Save* , select controller, *Download* , Run
- These parameter can set by a standard EtherNet/IP scanner tool too.

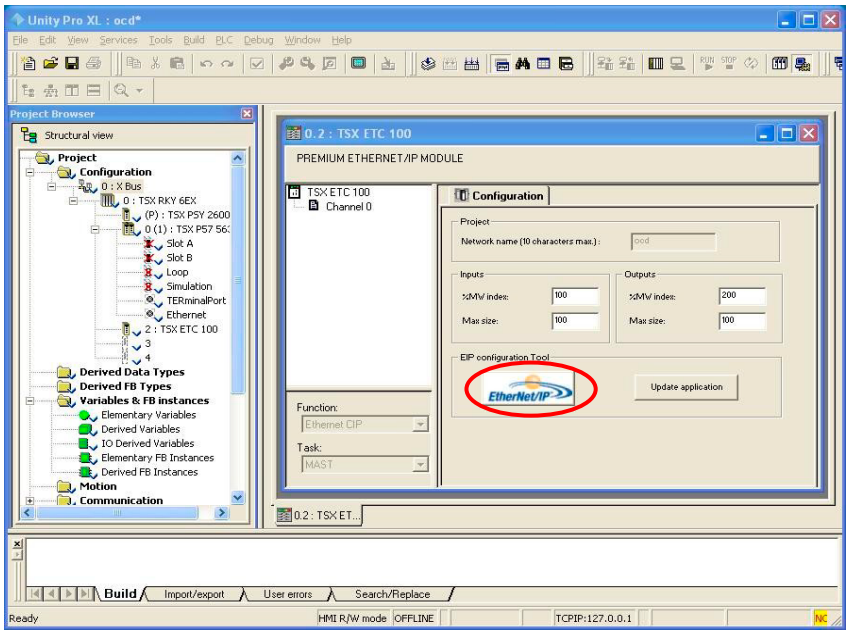


If everything is running then, in the “Errors tab” the message 0 error(s) should appear.



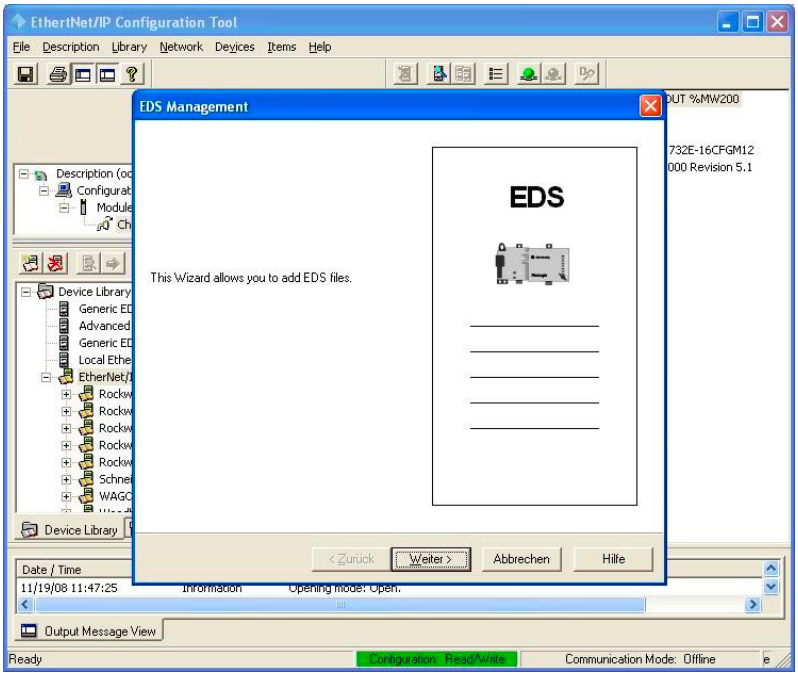
7.2 Schneider configuration tools

In the software tool Unity it is possible to configure the parameters of the encoders. EDS file help to change the parameters on an easy way. Select the EtherNet/IP module and start the EtherNet/IP configuration tool.

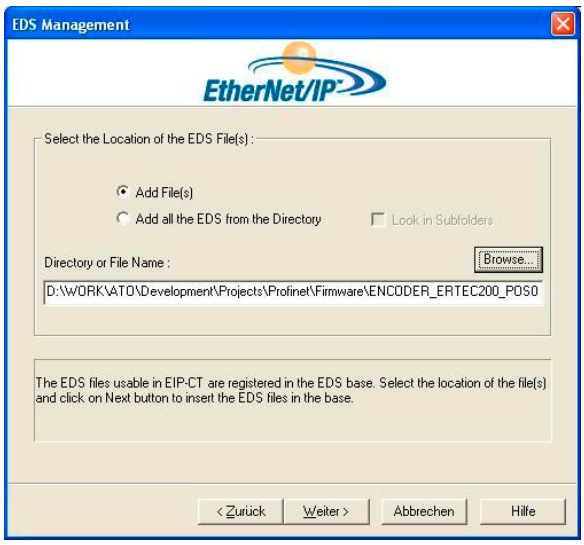


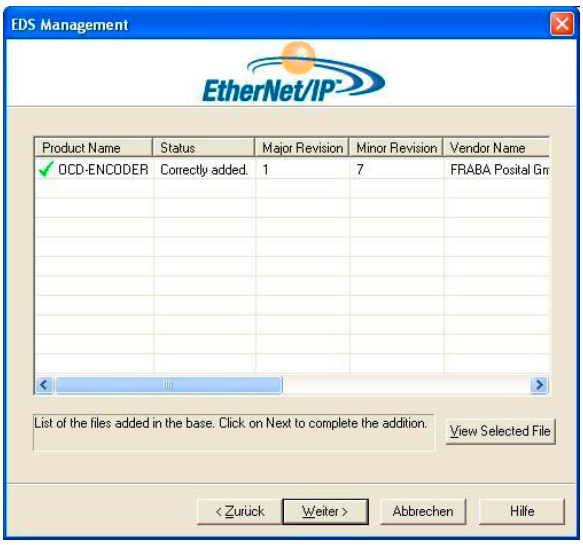
7.2.1 Setting configuration

In the first time it is necessary to install the EDS-File with the wizard.

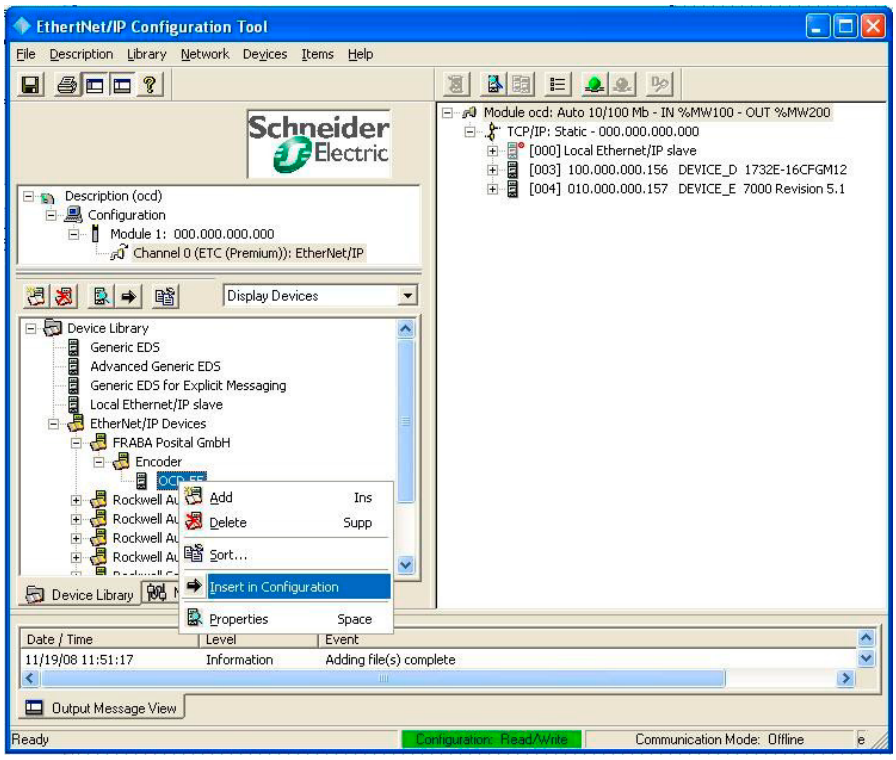


Select the EDS-File, available on our web side, and follow the wizard to the end.

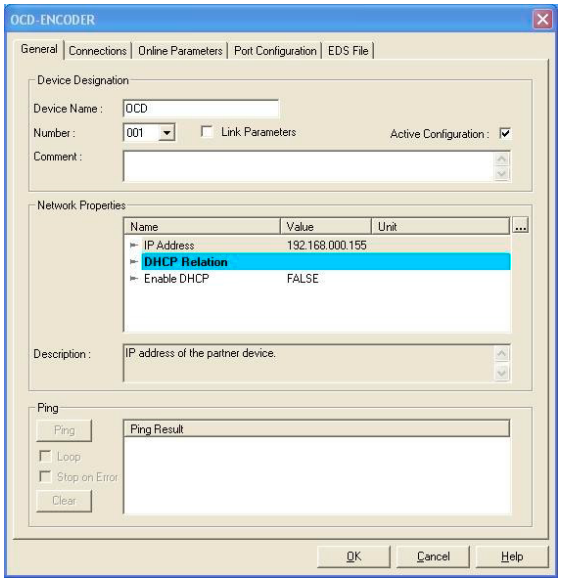


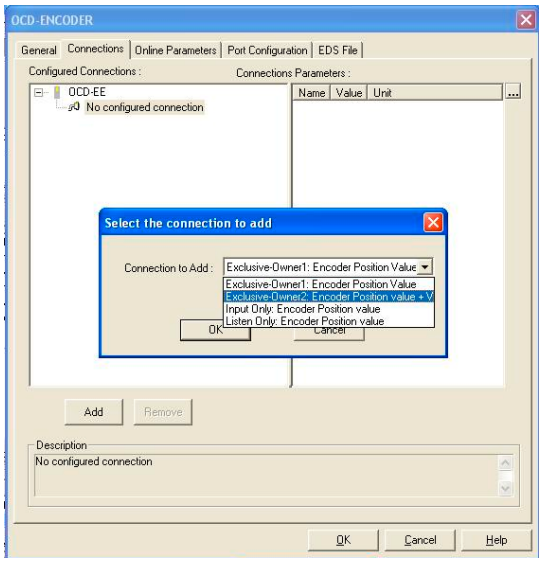


Select the encoder in the Device Library and *Insert in Configuration* (menu opens on right button click of the mouse).

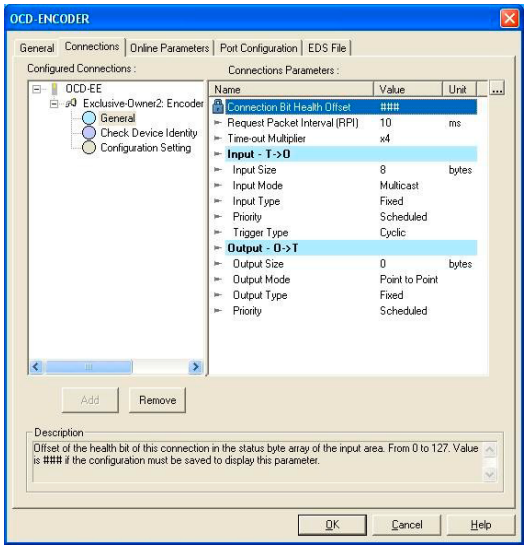


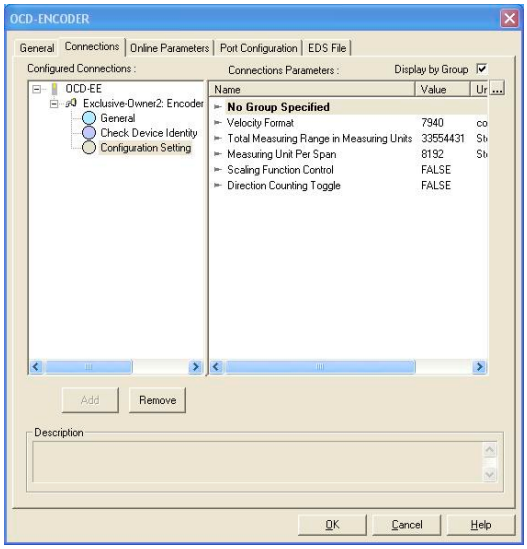
Set the IP-Adress of the encoder in the *General*-Tab. Add a connection for reading the position value or the position value and velocity. More details about the different connections are available in chapter 2.





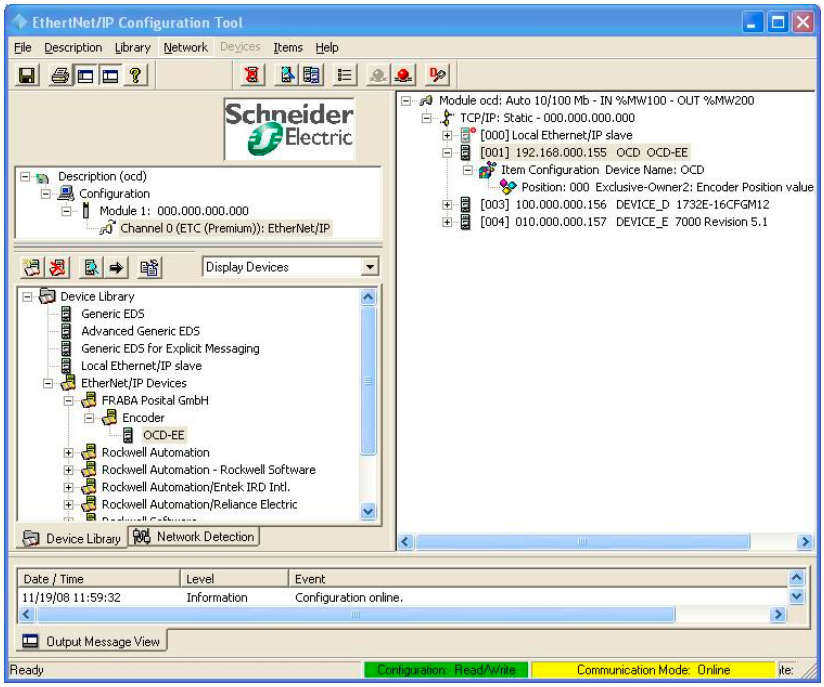
In tab *Connections* under *General* can be checked the cycle time (RPI), the input and output configuration. Under *Configuration* *Setting* are the offline parameters available, that will be used after the PLC goes in the Run state.

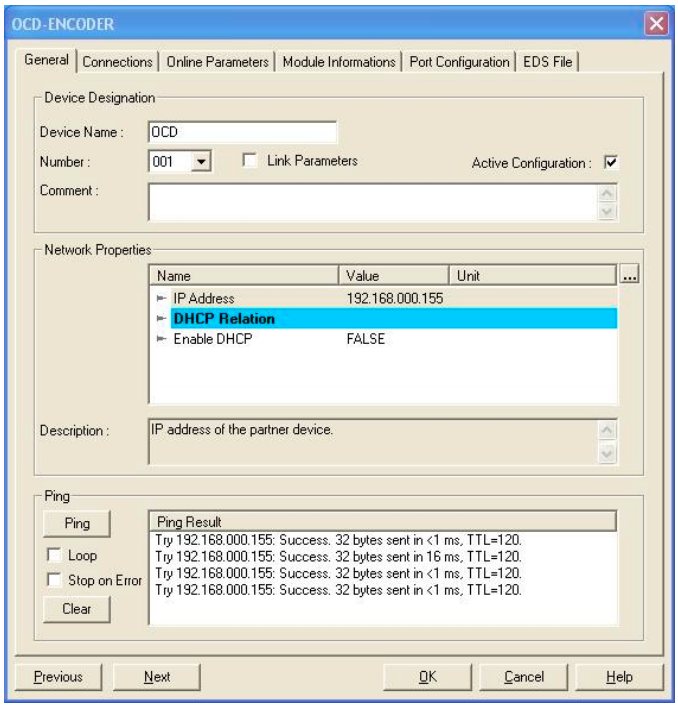




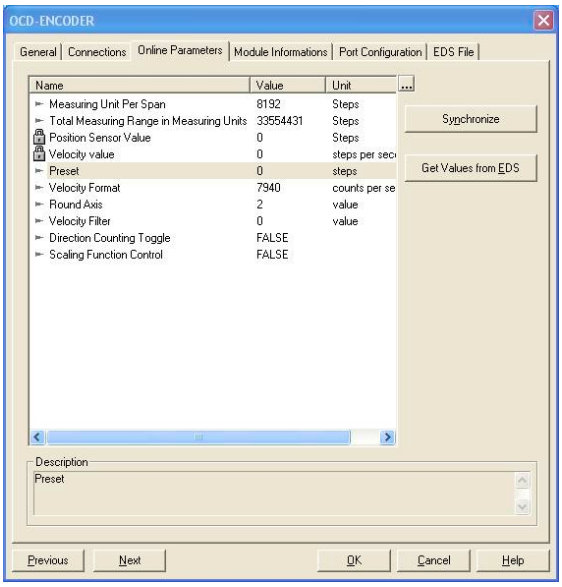
7.2.2 Online configuration

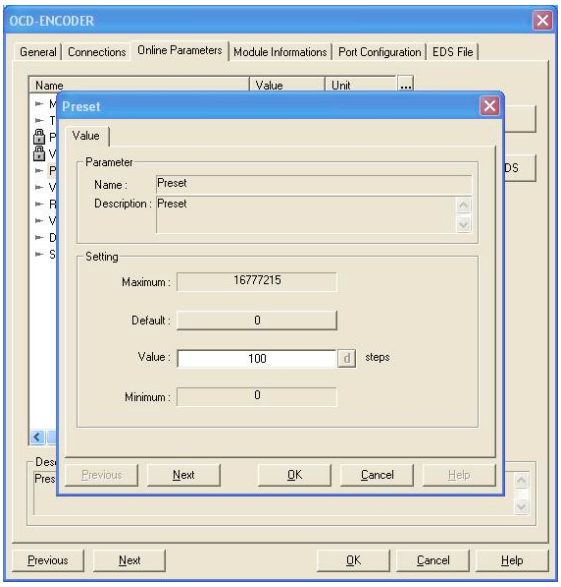
If the last steps were successful the encoder can go in the status online. In the configuration window in tab General it is possible to test the encoder connection with sending Ping commands to the encoder.

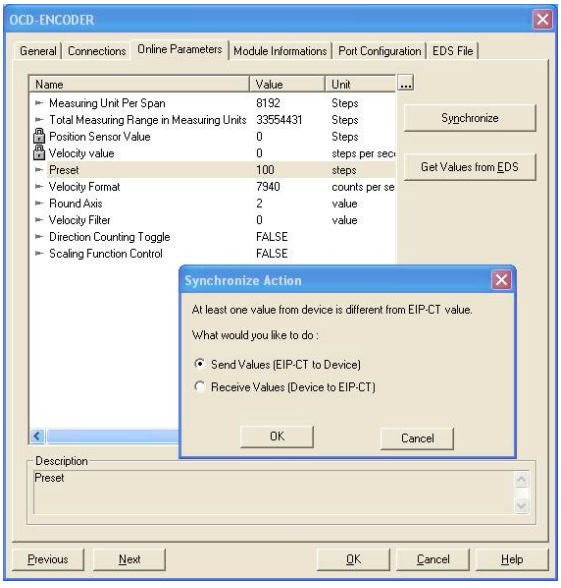




In tab *Online Parameters* is it possible to change the parameters. In a separate window it is possible to use the wished parameter. There are the minimum, maximum and default values available. With the *Synchronize* button it is possible to send the parameters to the encoder oder read them out from the encoder.





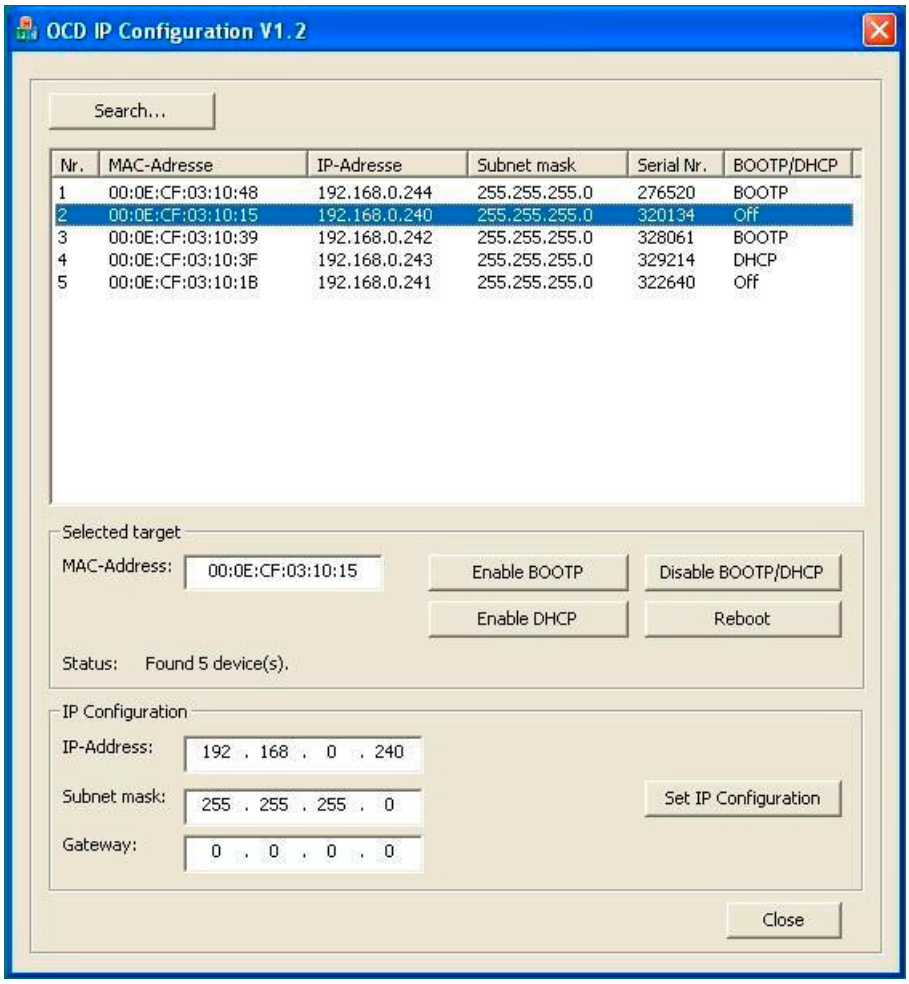


7.3 BOOTP/DHCP and IP configuration tool

For EtherNet/IP encoders it is necessary to know the IP-Address. On our website there is a free special tool available that allows scanning the complete network segment for MAC-Addresses of encoders. Our tool will find the devices with deactivated BOOTP and DHCP too.
Connect all devices, turn on the power supply and push the "*Search…*" button. Select the encoder and push the button for the needed Functionality. After changing the status it is necessary to push the button "*Search…*" to get the actual status of the encoders.

With this tool is it possible to change the IP-Adress, the Subnet and Gateway too.

This program use UDP Port 4000. **Attention: Check that the firewall don't block this port!**


OCD IP Configuration V1.2
Search...
Nr. MAC-Adresse IP-Adresse Subnet mask Serial Nr. BOOTP/DHCP
1 00:0E:CF:03:10:48 192.168.0.244 255.255.255.0 276520 BOOTP
2 00:0E:CF:03:10:15 192.168.0.240 255.255.255.0 320134 Off
3 00:0E:CF:03:10:39 192.168.0.242 255.255.255.0 328061 BOOTP
4 00:0E:CF:03:10:3F 192.168.0.243 255.255.255.0 329214 DHCP
5 00:0E:CF:03:10:1B 192.168.0.241 255.255.255.0 322640 Off
Selected target
MAC-Address: 00:0E:CF:03:10:15
Enable BOOTP
Disable BOOTP/DHCP
Enable DHCP
Reboot
Status: Found 5 device(s).
IP Configuration
IP-Address: 192 . 168 . 0 . 240
Subnet mask: 255 . 255 . 255 . 0
Gateway: 0 . 0 . 0 . 0
Set IP Configuration
Close

8 FAQ

8.1 Problem: IP Address unknown and BOOTP/DHCP is deactivated
Solution:
Download the tool to read out the IP-Address based on the MAC-Address from our web site: http://www.hohner-elektrotechnik.de

8.2 Problem: Replace a rotary encoder in the machine and the controller cannot start the application. Additional the Stat LED is flashing with 4 Hz
Solution: Start the BOOTP/DHCP server to set the IP-Address and deactivate BOOTP and DHCP. See chapter 7.1.1

8.3 Problem: The BOOTP/DHCP Configuration Tool doesn't find the encoders.
Solution: Check that TCP Port 4000 is not blocked from a firewall.

8.4 Problem: Cannot deactivate BOOT/P or DHCP
Background: Firewall and/or WLAN block communication.
Solutions:
1. Firewall must not block Port 4000 and 5000
2. Deactivate WLAN and all addional network cards.

8.5 Problem: Error message "Invalid identifier"
Solution: Check that the major revision of the encoder use the same number as the EDS-File. Check the website.

8.6 Problem: IP Address unknown and BOOTP/DHCP is deactivated
Solution: Download the tool to read out the IP-Address based on the MAC-Address from our web site.

8.7 Problem: Stat LED is flashing with 4 Hz
Background: After replacing of a rotary encoder in the machine the controller cannot start the application. Additional the Stat LED is flashing with 4 Hz
Solution: Start the BOOTP/DHCP server to set the IP-Address and deactivate BOOTP and DHCP.

8.8 Problem: After Power-up the programmed parameter were lost
Solution: Use the save command to save all programmed parameters in the non volatile memory (NVM). Only Preset is saved automatically in the NVM.

8.9 Problem: Parameter from Configuration tool i.e. RSLogix overwrite saved values of the encoder
Answer: Yes, that is according the encoder profile correct. It could be changed according the new Profibus functionality of parameterization

8.10 Problem: Additional transmisson of LLDP frames

Answer: According the used stack version will transmit additional ~1% LLDP frames. This should be not a problem with the total network traffic.

8.11 Problem: Problems with Configuration Tool
Answer: Check if the Version 1.4 is in use!

8.12 Problem: How many encoders can work with one Rockwell PLC?
Answer: One encoder raise only a traffic of 100 Ethernet-Paket/s. The PLCs can manage 6000-20000 Ethernet-Pakets/s. Rockwell has got two tools:
- EtherNet/IP Capacity Tool
- Integrated Architecture Builder (overdressed for analyzis)

9 Glossar

Term	Explanation
10Base-T	Transmission line with 10 Mbit data transmission rate
100Base-T	Transmission line with 100 Mbit data transmission rate
Autocrossing	Allow to use straight or crossover wiring
Autonegotiation	Is an Ethernet procedure by which two connected devices choose common transmission parameters, such as speed and duplex mode
Baudrate	Transmission rate; it display the transmission bits per second
Big Endian	Variables will use Byte 0 as Low and last Byte as High
Binary	Numeric system with value 0 or 1.
BootP	A UDP network protocol used by a network client to obtain its IP address automatically
CAT5	Terminations for transmission rates up to 100 Mbit.
CIP	**C**ontrol and **I**nformation **P**rotocol
DHCP	**D**ynamic **H**ost **C**onfiguration **P**rotocol is a protocol used by networked devices (clients) to obtain the parameters necessary for operation in an Internet Protocol network. This protocol reduces system administration workload, allowing devices to be added to the network with little or no manual configuration.
EIP	**E**theNet/**IP**
EMC	**E**lectro**m**agnetic **c**ompatibility, there are rules to verifying devices.
ENIP	**E**ther**N**et/**IP**
Ethernet	Ethernet is a computer network technology based on frames.
Explicit Messages	Communication between i.e. a Ethernet scanner and encoder

Term	Explanation
Fast Ethernet	Transmission technology with 100 Mbit transmission rate.
Flash	Internal memory, saved data will be available after power down.
Implicit Messaging	IO Connection: communication between controller and device

IP-Address	Allow a logic addressing from computer in a network.
IP-Protocol	The **I**nternet **P**rotocol is widespread in computer networks. It is the implementation of the internet layer of the TCP/IP-model
MAC Address	Worldwide explicit address of a device. The encoder uses three MAC Adresses: one for internal interface and two for the ports.
Mbit	Transmission rate or baud rate, million bits per second
OCD	Acronym: **O**PTO**C**O**D**E, name of an encoder series
OSI-Model	The **O**pen **S**ystem **I**nterconnection reference model is a open layer model for the organization of a communication.
Scanner	Program to send Explicit Messages to the encoder
Switch	A switch is an electronic device to connect computers e.g. network segments in a local network. Unlike a hub, a switch uses stacks to avoid network collisions.
TCP	The **T**ransmission **C**ontrol **P**rotocol is a connection orientated transmission protocol, in a network.
UDP	**U**ser **D**atagram **P**rotocol is utilized to send data that does not need to be transferred in a reliable way.

We do not assume responsibility for technical inaccuracies or omissions. Specifications are subject to change without notice.

13 Revision index

Revision	Date	Revision
First release	5.6.2008	1.0
Delete section device type	2.7.2009	1.7
Delete save single attributes, Cycle of write parameters from 5,000 Mio -> 100.000, 15-30V -> 10-30V, Preset info to save in non volatile memory, add Port 4000 FAQ	1.9.2009	1.8
Change Cycle of write parameters from 100,000 to 5 Mio	14.1.2010	1.9
Add table Control Bits	17.11.2010	1.10
Add information about filter definition, velocity, Unicast Update FAQ Delete technical data and drawings	10.1.2013	1.11

hohner

Elektrotechnik Werne

Hohner Elektrotechnik GmbH
Gewerbehof 1 · 59368 Werne
Telefon 02389 - 9878-0 · Telefax 02389 - 9878-27
info@hohner-elektrotechnik.de · www.hohner-elektrotechnik.de