Startup of Axioline E EtherNet/IP™ IO-Link devices on an RSLogix™ 5000 system (for experienced RSLogix™ users)

Quick start guide

UM QS EN AXL E EIP IOL RSLOGIX

PHOENIX CONTACT
INSPIRING INNOVATIONS

Quick start guide

Startup of Axioline E EtherNet/IP™ IO-Link devices on an RSLogix™ 5000 system (for experienced RSLogix™ users)

2015-07-16

Designation: UM QS EN AXL E EIP IOL RSLOGIX

Revision: 00

The document is valid for all Axioline E EtherNet/IP™ IO-Link devices.

Please observe the following notes

User group of this manual

The use of products described in this manual is oriented exclusively to:

- Qualified electricians or persons instructed by them, who are familiar with applicable standards and other regulations regarding electrical engineering and, in particular, the relevant safety concepts.
- Qualified application programmers and software engineers, who are familiar with the safety concepts of automation technology and applicable standards.

Explanation of symbols used and signal words

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety measures that follow this symbol to avoid possible injury or death.

There are three different categories of personal injury that are indicated with a signal word.

DANGER This indicates a hazardous situation which, if not avoided, will result in death or serious injury.

WARNING This indicates a hazardous situation which, if not avoided, could result in death or serious injury.

CAUTION This indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.

This symbol together with the signal word **NOTE** and the accompanying text alert the reader to a situation which may cause damage or malfunction to the device, hardware/software, or surrounding property.

This symbol and the accompanying text provide the reader with additional information or refer to detailed sources of information.

How to contact us

Internet

Up-to-date information on Phoenix Contact products and our Terms and Conditions can be found on the Internet at:
phoenixcontact.com

Make sure you always use the latest documentation.
It can be downloaded at:
phoenixcontact.net/products

Subsidiaries

If there are any problems that cannot be solved using the documentation, please contact your Phoenix Contact subsidiary.
Subsidiary contact information is available at phoenixcontact.com.

Published by

PHOENIX CONTACT GmbH & Co. KG
Flachsmarktstraße 8
32825 Blomberg
GERMANY

Should you have any suggestions or recommendations for improvement of the contents and layout of our manuals, please send your comments to:
tecdoc@phoenixcontact.com

General terms and conditions of use for technical documentation

Phoenix Contact reserves the right to alter, correct, and/or improve the technical documentation and the products described in the technical documentation at its own discretion and without giving prior notice, insofar as this is reasonable for the user. The same applies to any technical changes that serve the purpose of technical progress.

The receipt of technical documentation (in particular user documentation) does not constitute any further duty on the part of Phoenix Contact to furnish information on modifications to products and/or technical documentation. You are responsible to verify the suitability and intended use of the products in your specific application, in particular with regard to observing the applicable standards and regulations. All information made available in the technical data is supplied without any accompanying guarantee, whether expressly mentioned, implied or tacitly assumed.

In general, the provisions of the current standard Terms and Conditions of Phoenix Contact apply exclusively, in particular as concerns any warranty liability.

Phoenix Contact reserves the right to register its own intellectual property rights for the product identifications of Phoenix Contact products that are used here. Registration of such intellectual property rights by third parties is prohibited.

Other product identifications may be afforded legal protection, even where they may not be indicated as such.

Table of contents

1 Basics and example project

1.1 Information about this document

This document describes the basic procedure for accessing IO-Link device parameters in conjunction with the Axioline E EtherNet/IP™ IO-Link master and the RSLogix™ engineering system from Rockwell.

The document is aimed at experienced PC Worx users. It does not cover a complete project, just specific features associated with the use of Axioline E devices.

It is assumed the user has knowledge and experience in the operation of PCs and Windows operating systems, knowledge of the PC Worx software from Phoenix Contact, and Ethernet basics.

1.2 System requirements

1.2.1 Software requirements

Alongside the Axioline E-EtherNet/IP™ IO-Link device you need the RSLogix™ 5000 software v20.01.00 (CPR 9 SR 5).

1.2.2 Hardware requirements

Please refer to the RSLogix™ documentation for the hardware requirements.

1.2.3 Axioline E EtherNet/IP™ IO-Link devices

The document is valid for the following Axioline E EtherNet/IP™ IO-Link devices.

Designation	Order No.	ProdCode	Firmware	EDS file
AXL E EIP IOL8 DI4 M12 6P	2701496	8178	v2.1.0.0 or later	20150213_AXLEEIPIOL8DI4M12_6P_V160 or later
AXL E EIP IOL8 DI4 M12 6M	2701491	8179	v2.1.0.0 or later	20150213_AXLEEIPIOL8DI4M12_6P_V160 or later

1.3 Example network configuration

This example project consists of an Allen Bradley ControlLogix controller, the PC with the RSLogix™ 5000 software, and Axioline E EtherNet/IP™ IO-Link devices. If DLR (Device Level Rings) are to be used, there is an optional DLR supervisor-capable switch in the structure.
The example system is shown in Figure 1-1.

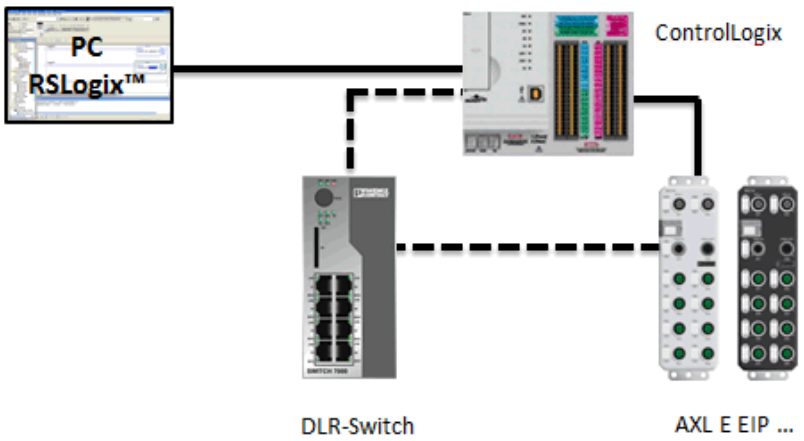


Figure 1-1 Example system

1.4 General startup steps

To startup the Axioline E EtherNet/IP™ IO-Link devices, proceed as follows:

- Install the Axioline E devices and connect the power supply.
- Create a new project in the engineering tool.
- Add the controller and the Axioline E devices to the project.
- Enter addresses/names.
- Compile the project and send it to the controller.
- Establish communication with the Axioline E device.
- Test the communication.

2 Creating an RSLogix™ project

To integrate the Axioline E EtherNet/IP™ IO-Link devices into the network, proceed according to the documentation for your Allen Bradley controller. This section only explains fundamental steps that are relevant to the Axioline E system.

It is assumed that the IP address of the controller is known and that a connection can be established. For further information, please refer to the user manual for the associated controller.

2.1 Creating/opening a project

- In RSLogix™, create a new project or open an existing project. In this example, a new project is created.
- In the menu bar, click on “File” and select “New”.
- Alternatively, you can use the “Ctrl + N” key combination.

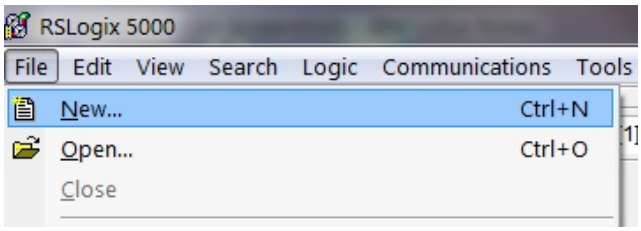


Figure 2-1 Creating a new project

- The “New Controller” window opens.

- Select your controller (1) and associated parameters (2 to 5) here.
- Click “OK” (6) to proceed.

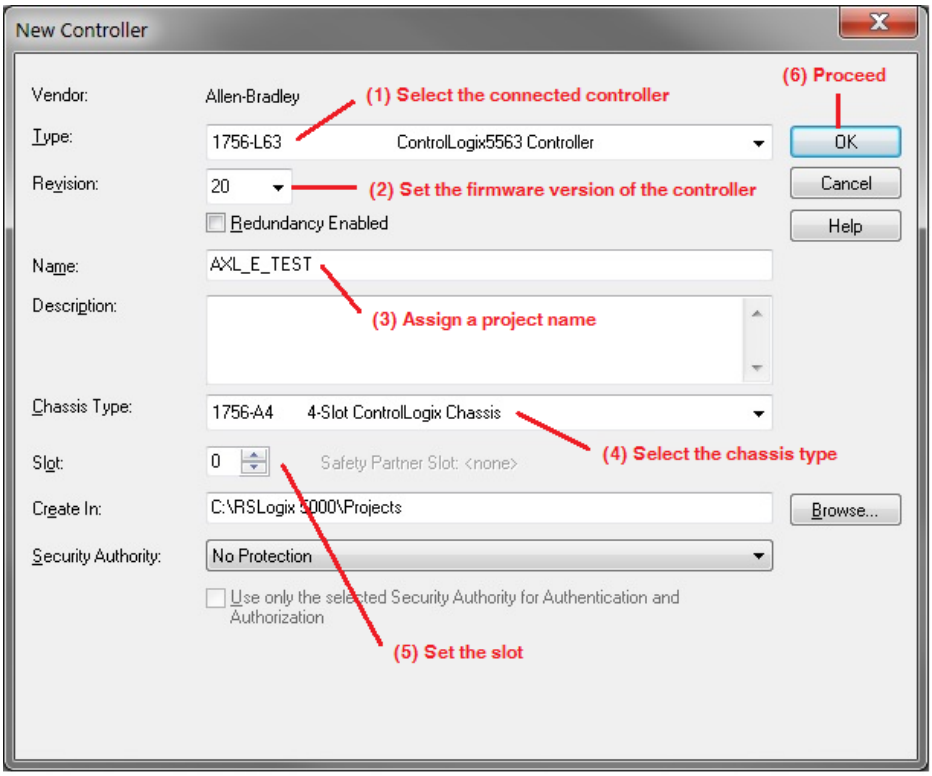


Figure 2-2 Selecting the controller and parameters

The project is created and the necessary parameters are automatically set. The project tree may look like the one below.

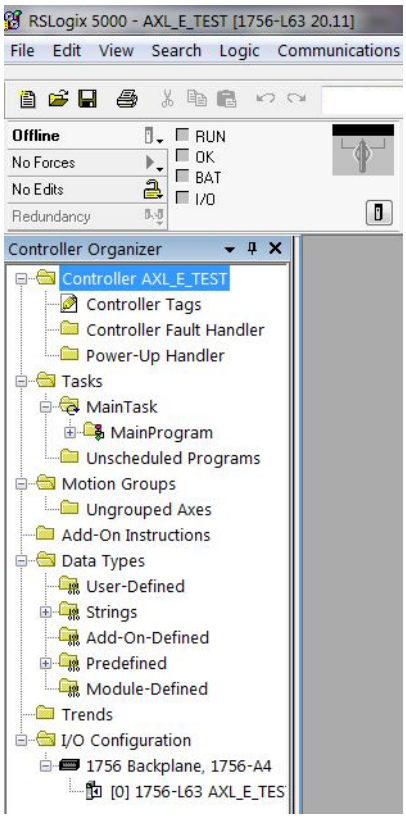


Figure 2-3 Project tree

2.2 Installing the I/O device description file

If this is the first time you have worked with Axioline E devices in your engineering system, you need to install the necessary device description files.

Make sure you are using the latest device description files. These are available on the Internet at www.phoenixcontact.net/catalog.

Make sure that the name of the downloaded device description file is the same as the name displayed in the Download area. If the file name differs following download (e.g., after downloading with Mozilla Firefox), rename the file.

- In the “Tools” menu, open the menu item “EDS Hardware Installation Tool”.

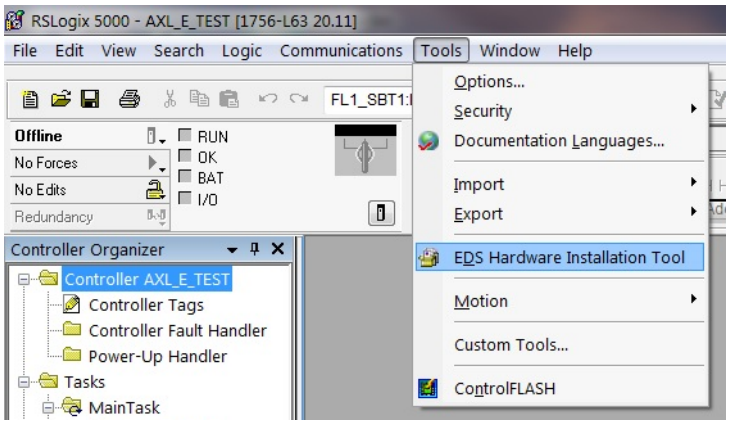


Figure 2-4 EDS Hardware Installation Tool

- This opens a new window entitled “Rockwell Automation’s EDS Wizard”.
- Select the “Register an EDS file(s)” (1) option.
- Click “Next” (2) to proceed.

Figure 2-5 Selecting “Register an EDS file(s)”.

- In the next window, select the “Register a single file” (1) option if you only want to install a single EDS file.
- Alternatively, select “Register a directory of EDS files” (1) to install a folder containing several EDS files.
- Enter the folder (2) where the files are located.
- Click “Next” (3) to proceed.

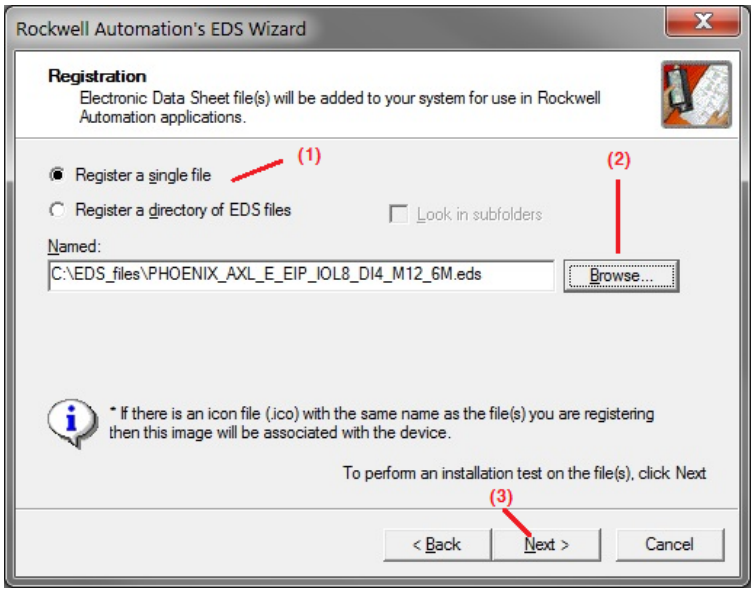


Figure 2-6 Selecting EDS files

- Follow the subsequent steps until the file(s) is/are successfully installed.
- Close the wizard by clicking “Finish”.

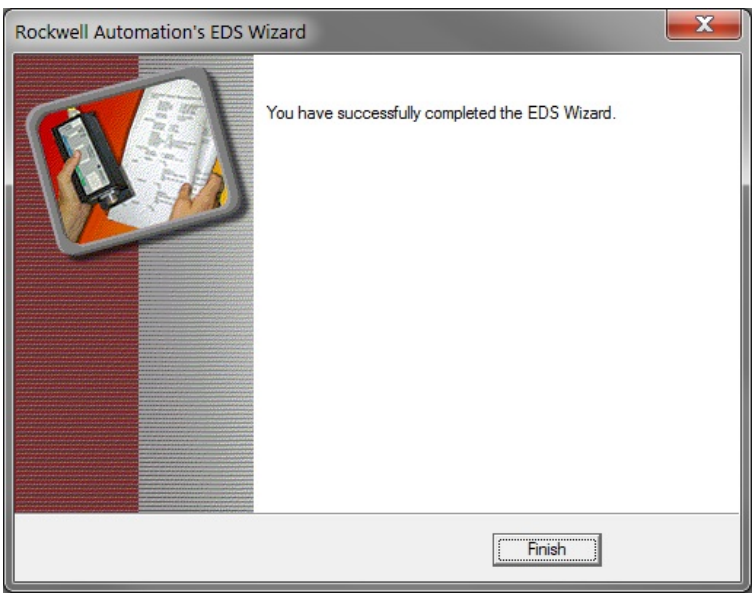


Figure 2-7 Installing EDS files

2.3 Establishing a connection to the controller

The RSLinx® tool enables you to create the EtherNet/IP™ structures required for RSLogix™ via an automatic network scan. To do so, a connection to the controller must be established.

- Open the “RSLinx®” tool.
- Select the menu item “Configure Drivers…” in the “Communications” menu.

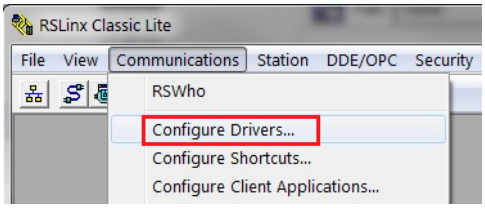


Figure 2-8 Selecting the “Configure Drivers...” menu item.

- In the new window, select “EtherNet/IP Driver” from the “Available Driver Types” drop-down menu.
- Then click “Add New...”.

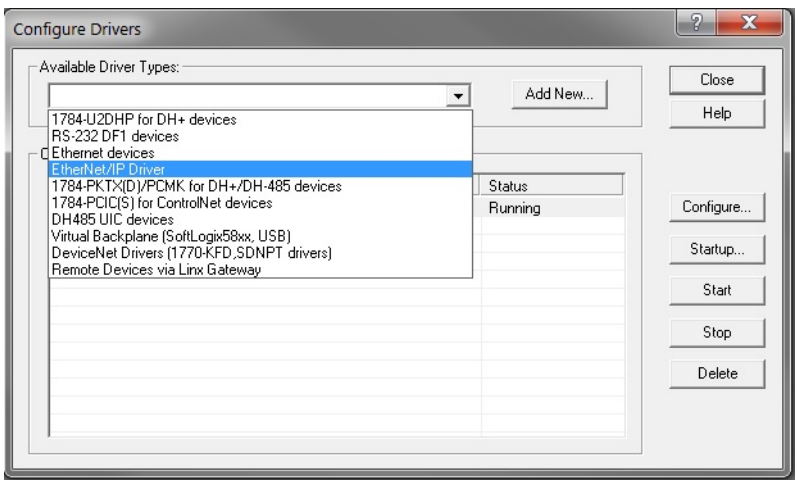


Figure 2-9 Selecting “Available Driver Types”

- In the new window, enter a name for the new connection, e.g., “Rockwell_EIP”.
- Click “OK” to proceed.

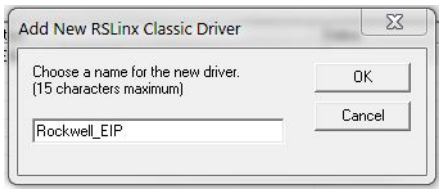


Figure 2-10 Entering a name for the new connection

- In the next window, select a network adapter.
- Click “OK” to proceed.

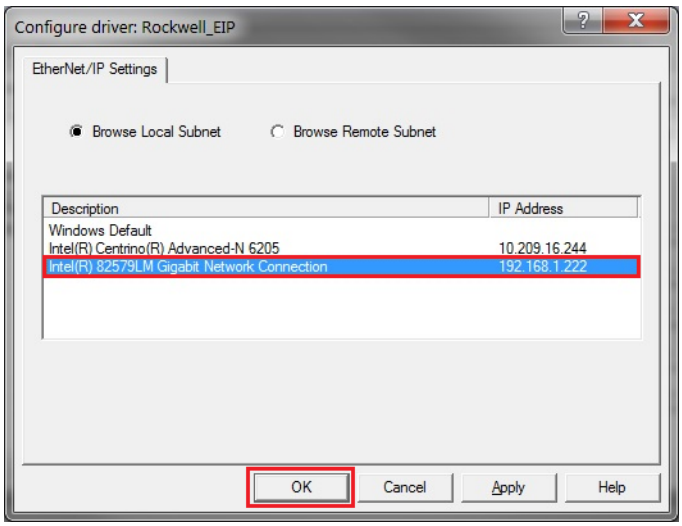


Figure 2-11 Selecting the network adapter

- In the “Configured Drivers” field, you will now be able to see the newly created connection.
- Close the window by clicking “Close”.

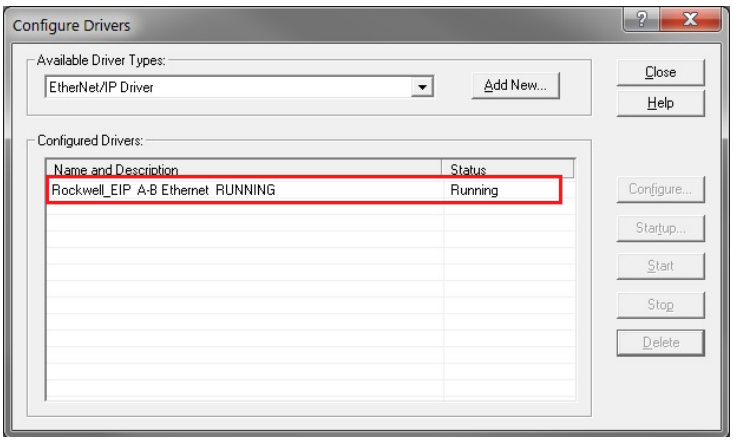


Figure 2-12 New Connection

- You can read all connected devices on the set network adapter by going to the “Communications” menu and selecting “RSWho”.

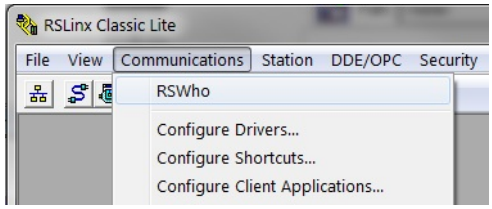


Figure 2-13 Reading connected devices

- In the network structure, select the connection you created before (e.g., Rockwell_EIP).

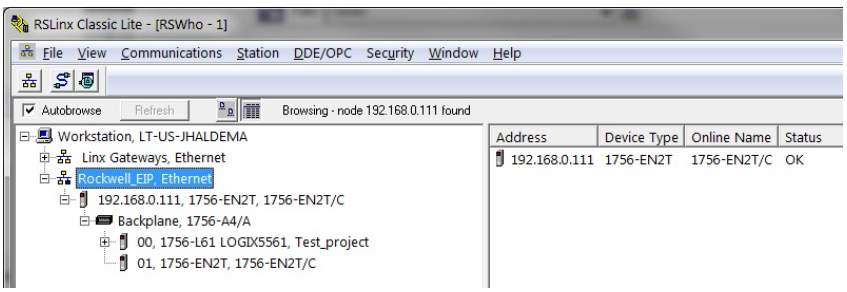


Figure 2-14 Selecting a connection

If the entries are not automatically updated, check the “Autobrowse” checkbox.

- Now switch to the RSLogix™ engineering tool.
- Click the “Active Modules” button

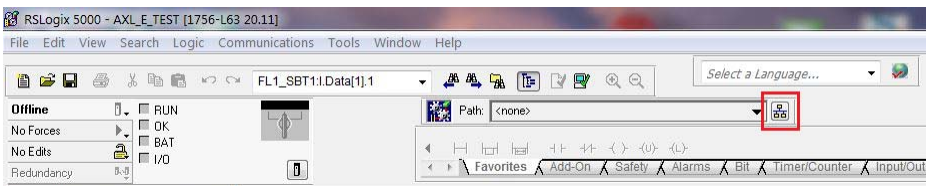


Figure 2-15 Selecting “Active Modules”

- In the new window, select your controller (1), in this case “Workstation > Rockwell_EIP > Backplane > 00, 1756-L73 LOGIX5573”.
- Then click “Go Online” (2).

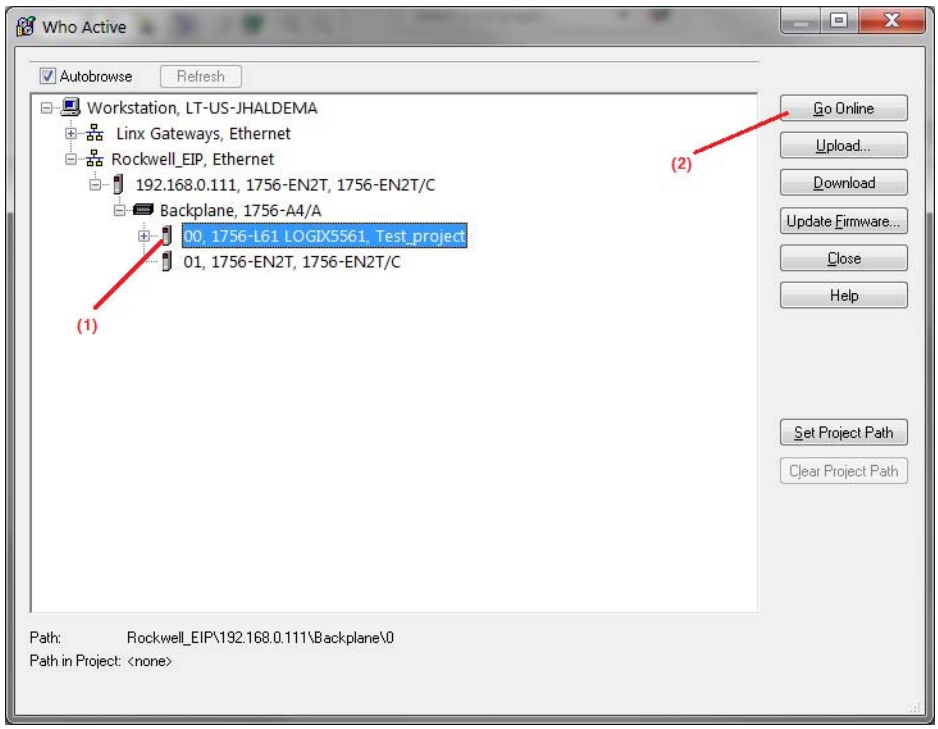


Figure 2-16 Selecting the controller

If the connection to the controller has been successfully established, the “Run Mode” and “Controller OK” indicators switch to green

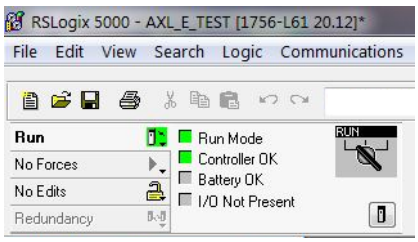


Figure 2-17 Green “Run Mode” and “Controller OK” indicators

2.4 Inserting the Ethernet/IP™ bridge module in the bus configuration

From RSLogix™ version 20 or later, you can insert the connected devices directly into the project in online mode via the discover functionality (Discover Modules...).

- Right-click the backplane “1756-ControlBus” and select “Discover Modules...”.

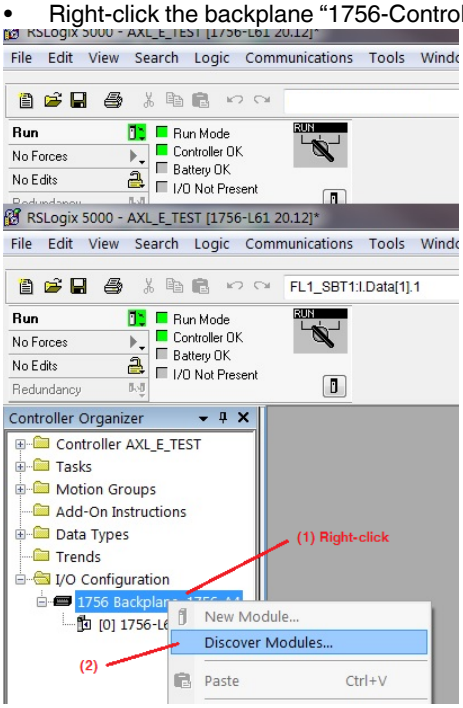


Figure 2-18 Selecting “Discover Modules...”

- In the new window, click “Create” next to your EtherNet/IP™ module, in this case 1756-EN2TR.

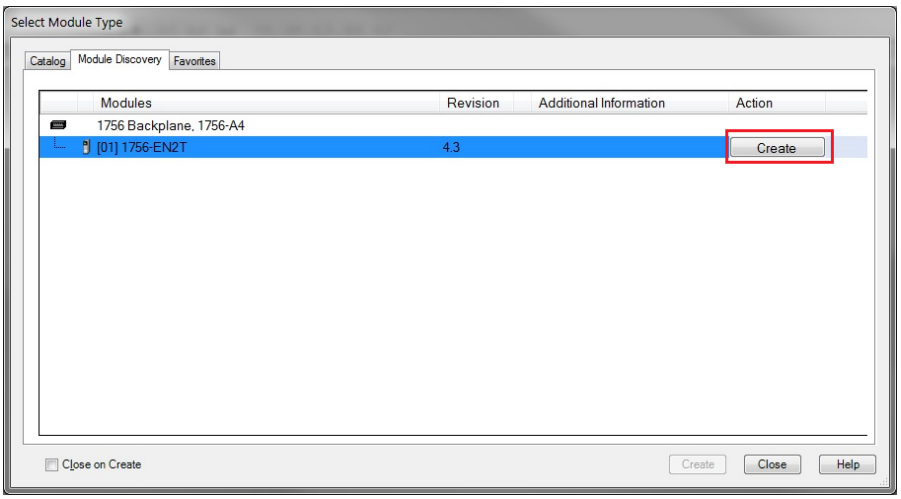


Figure 2-19 “Select Module Type” window

- In the following window give the module a name (1), e.g., EN2TR, and set additional required parameters (2, 3).
- Click “OK” (4) to proceed.

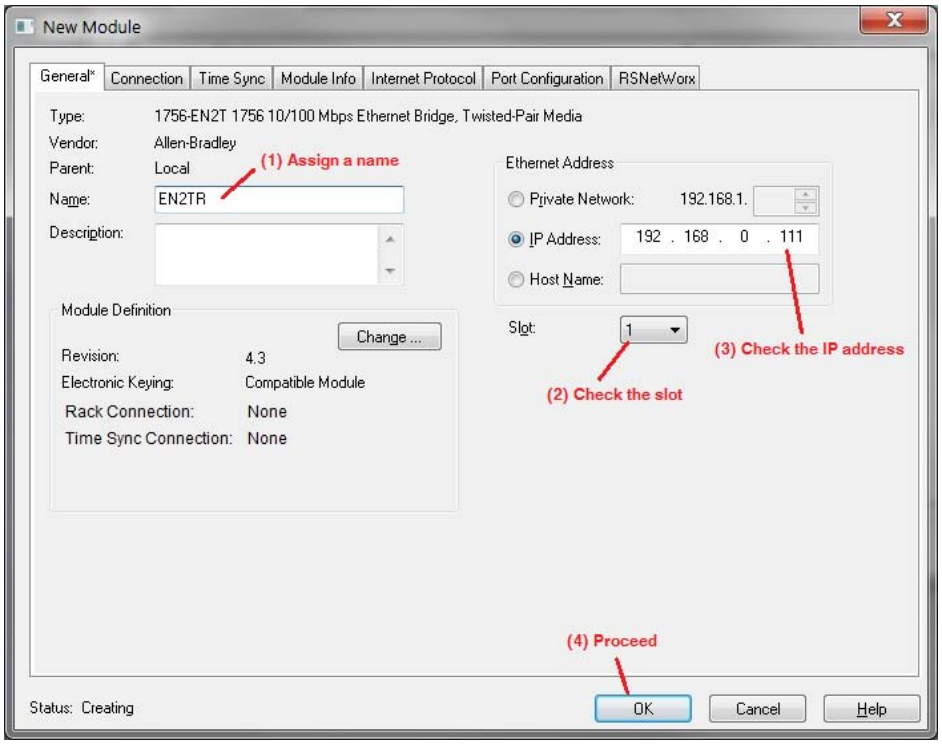


Figure 2-20 Assigning a name and setting the parameters

A warning message appears, telling you that the controller is in online mode.

- As long as there are no executable programs on the controller, you can continue by clicking “Yes”.

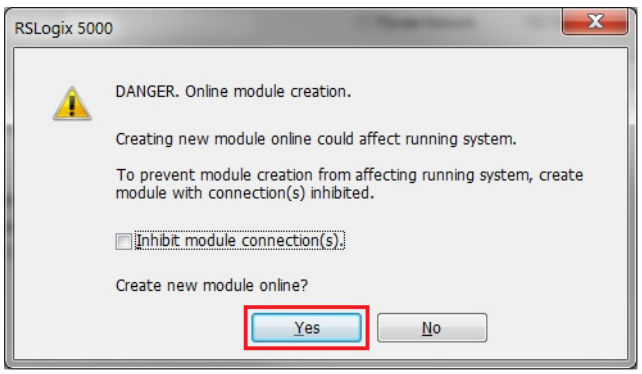


Figure 2-21 Warning message

Make sure that there are no executable programs and that the controller is not operated in a running environment, otherwise this can result in problems.

The module you have just inserted is now visible in your project structure under the “I/O Configuration” folder.

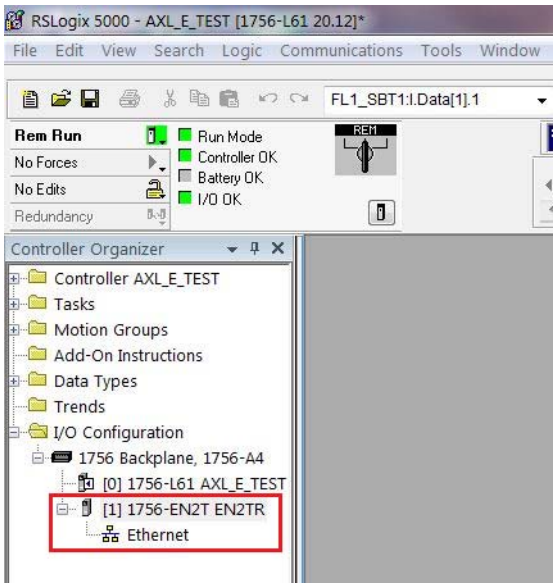


Figure 2-22 Project structure

2.5 Inserting an I/O device into the bus configuration

There are two ways of adding I/O devices to the project using RSLogix™:

– **In online mode:** by reading the network with the “Discover Modules...” function.
– **In offline mode**: by adding the device manually from the module catalog

In order for an I/O device to be inserted into the bus configuration, it needs an IP address. This can be set, for example, via the BOOTP/DHCP server tool from Rockwell.

To do so, follow the instructions in the section “Setting the Axioline E device > Assigning the IP address”.

2.5.1 Inserting an I/O device into the bus configuration online

- Right-click (1) and select “I/O Configuration > 1756-ControlBus > [1] 1756-EN2TR > Ethernet > Discover Modules...” (2).

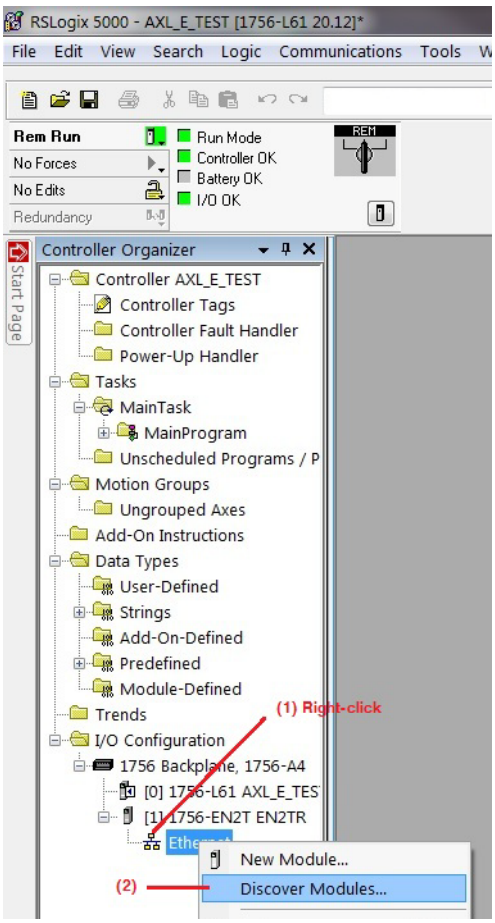


Figure 2-23 Searching for modules online

In the new window you can see all the devices found on the network, in this case only the device AXL E EIP IOL8 DI4 M12 6P has been found.

- Click “Create” next to the device you want to insert.

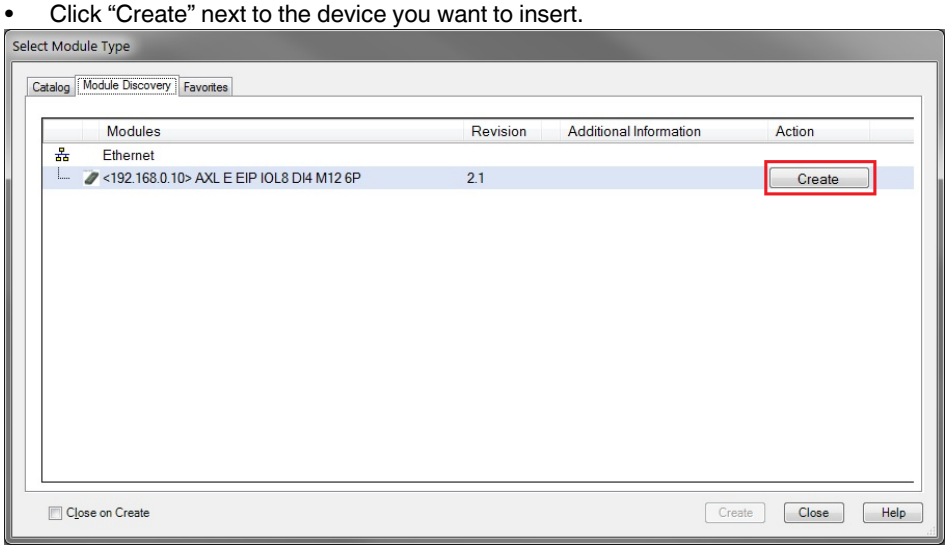


Figure 2-24 Inserting a device

- In the window that appears, give the device a name, e.g., “AXL_E_EIP_IOL8_6P” (1).
- Check the IP address (2).
- Then click on “Change” (3) in the “Module Definition” field.

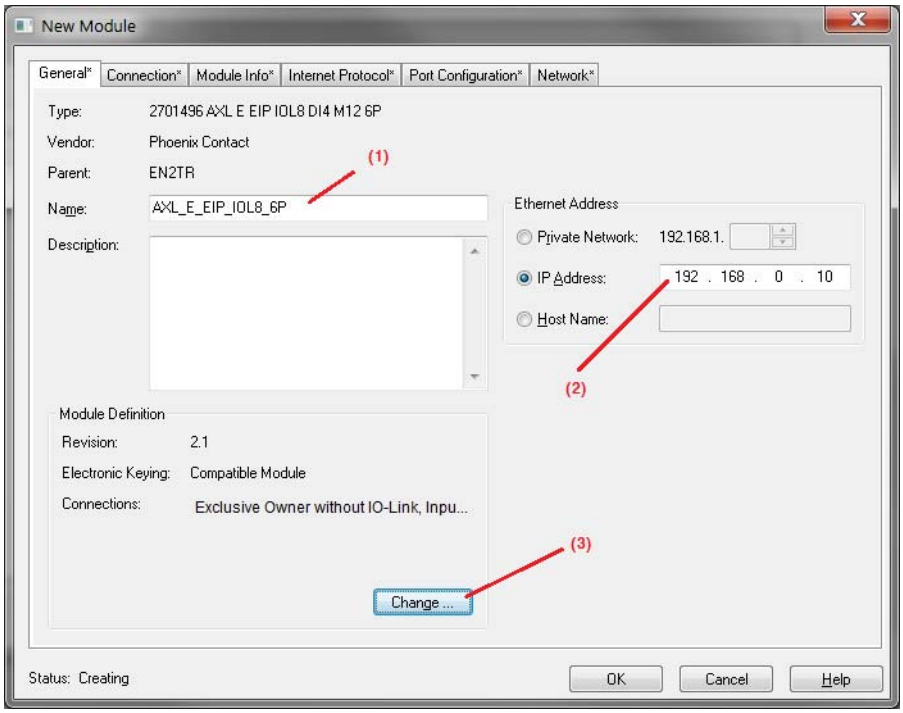


Figure 2-25 Entering the device name and checking the IP address

- In the new window, set the connection type, e.g., “Exclusive Owner with IO-Link I/O” (1) if you want to both send and receive IO-Link process data.
- Click “OK” (2) to proceed.

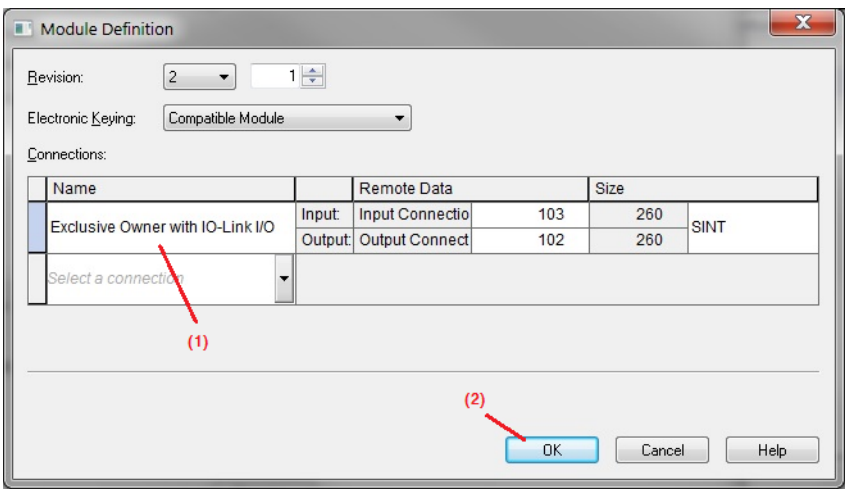


Figure 2-26 Setting the connection type

The differences between the individual connection types can be found in the associated device-specific data sheet.

- Respond to the following warning message by clicking “Yes”.

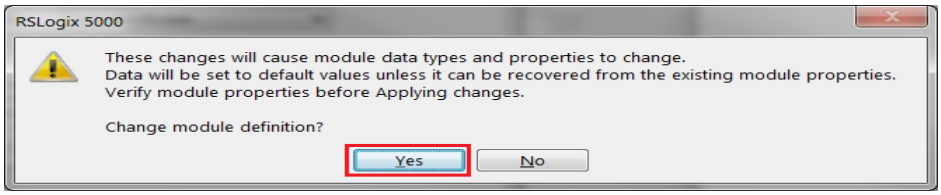


Figure 2-27 Warning message

A warning message appears, telling you that the controller is in online mode.

- As long as there are no executable programs on the controller, you can continue by clicking “Yes”.

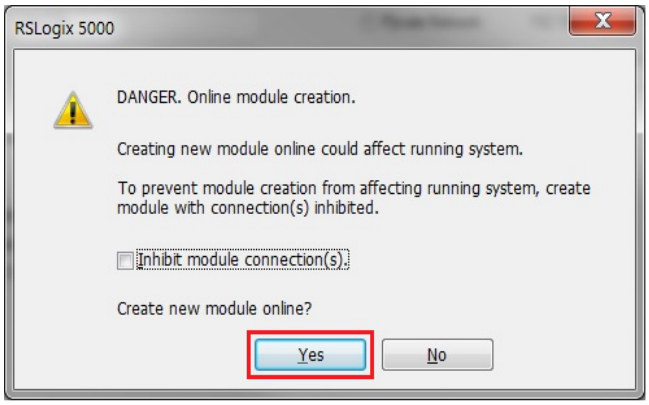


Figure 2-28 Warning message

Make sure that there are no executable programs and that the controller is not operated in a running environment, otherwise this can result in problems.

- Close the “Select Module Type” window by clicking “Close”.

- The device you have just inserted is now visible in your project structure under the “I/O Configuration” folder.

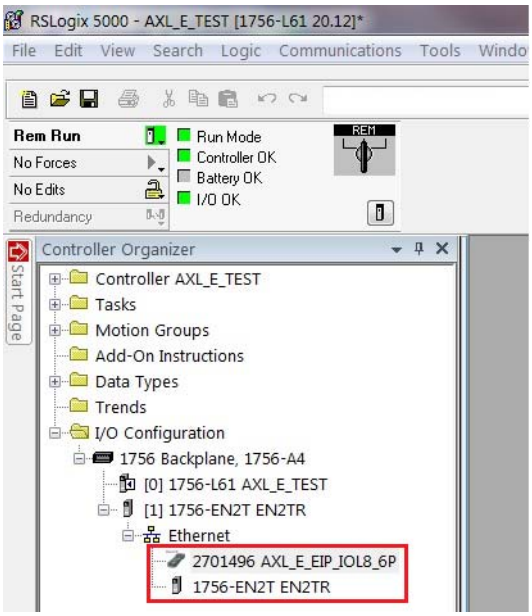


Figure 2-29 Device in the I/O Configuration folder

The “I/O OK” indicator in the RSLogix™ status bar changes to green, as soon as the controller has successfully established connection with the I/O device.

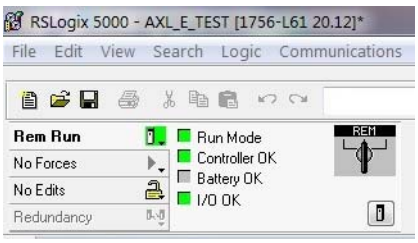


Figure 2-30 Green “I/O OK” indicator

2.5.2 Inserting an I/O device into the bus configuration offline

- Right-click (1) and select “I/O Configuration > 1756-ControlBus > [1] 1756-EN2TR > Ethernet > New Module...” (2).

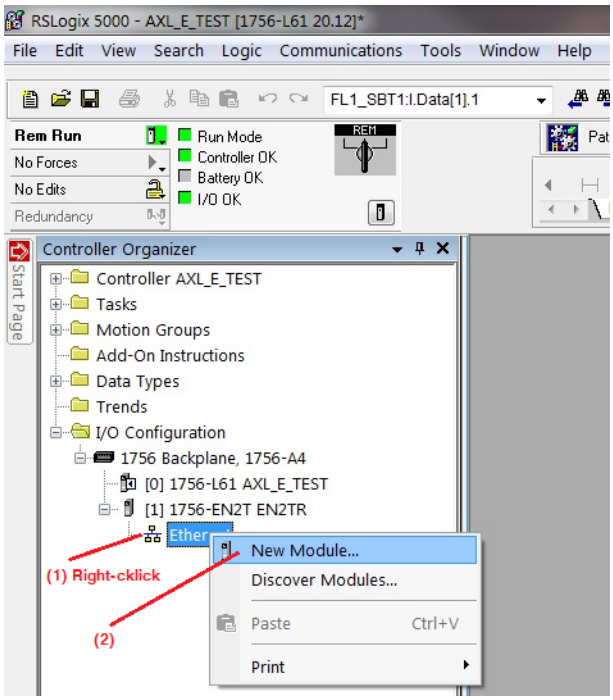


Figure 2-31 Searching for a device offline

The “Select Module Type” window opens with the entire RSLogix™ catalog.

- Enter “AXL E” (1) in the search screen to narrow down the search to Axioline E devices.
- Select the corresponding device (2).
- Click “Create” (3) to proceed.

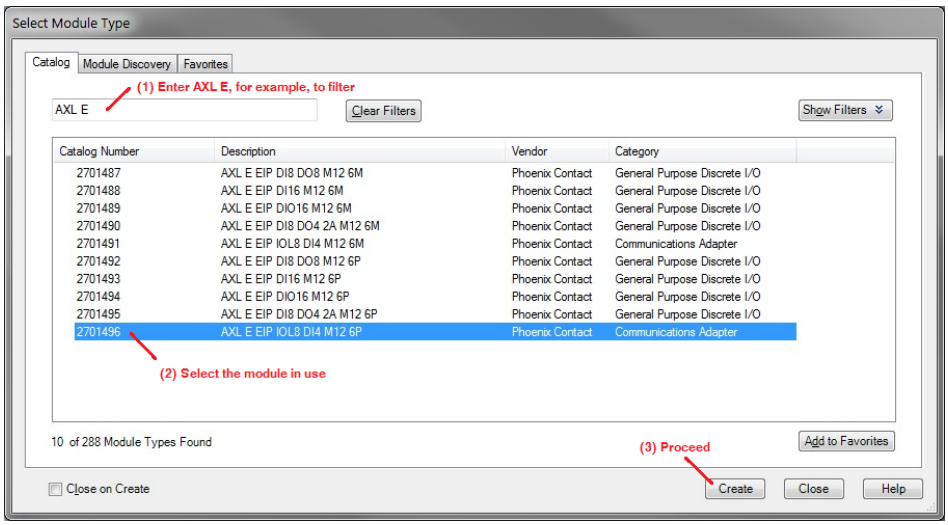


Figure 2-32 Selecting the device

- In the window that appears, give the device a name, e.g., “AXL_E_EIP_IOL8_6P” (1).
- Check the IP address (2).
- Click on “Change” (3) in the “Module Definition” field.

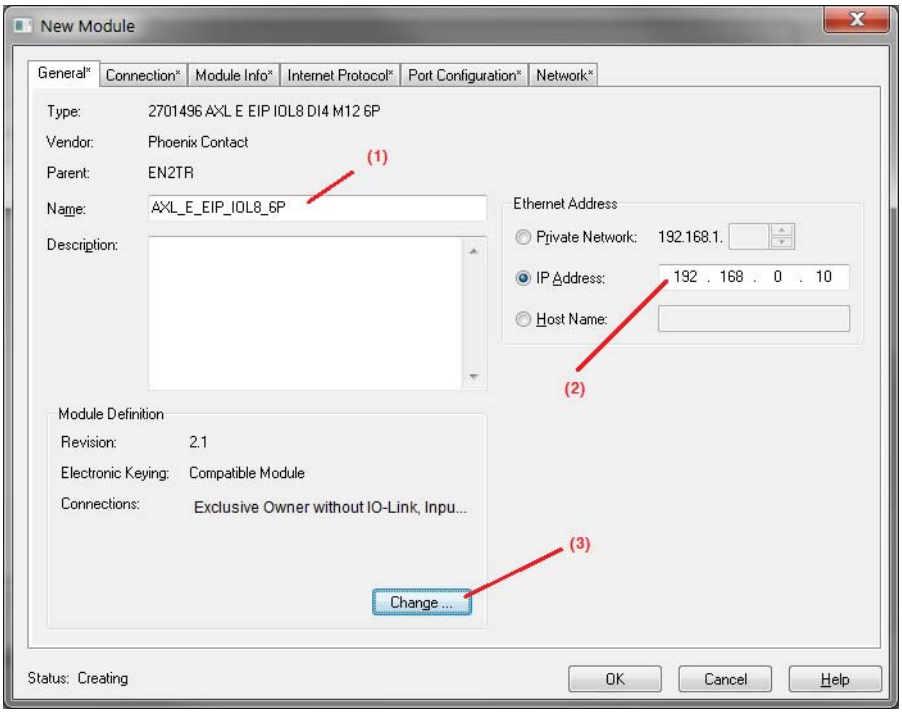


Figure 2-33 Entering the device name and setting the IP address

- In the new window, set the connection type, e.g., “Exclusive Owner with IO-Link I/O” (1) if you want to both send and receive IO-Link process data.
- Click “OK” (2) to proceed.

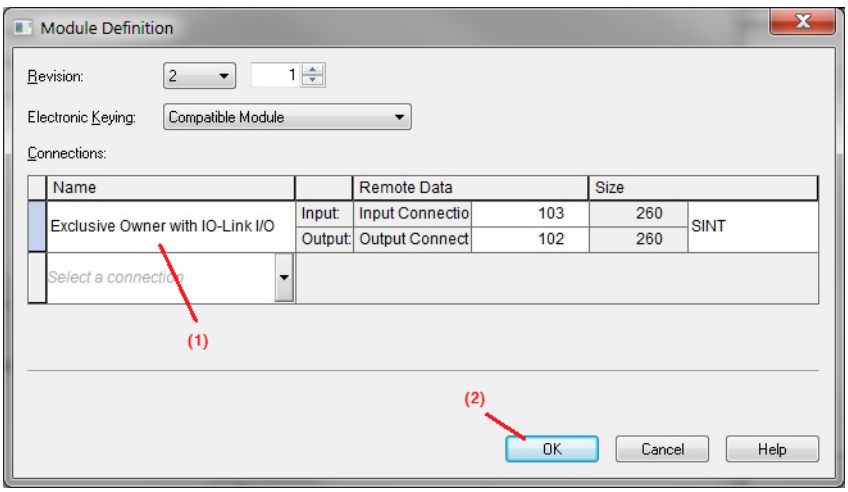


Figure 2-34 Setting the connection type

The differences between the individual connection types can be found in the associated device-specific data sheet.

- Respond to the following warning message by clicking “Yes”.

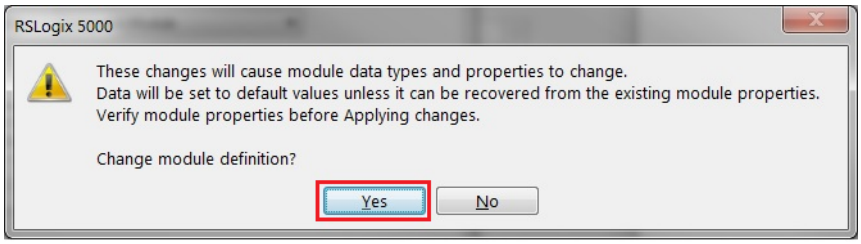


Figure 2-35 Warning message

Close the “Select Module Type” window by clicking “Close”.

- The device you have just inserted is now visible in your project structure under the “I/O Configuration” folder.

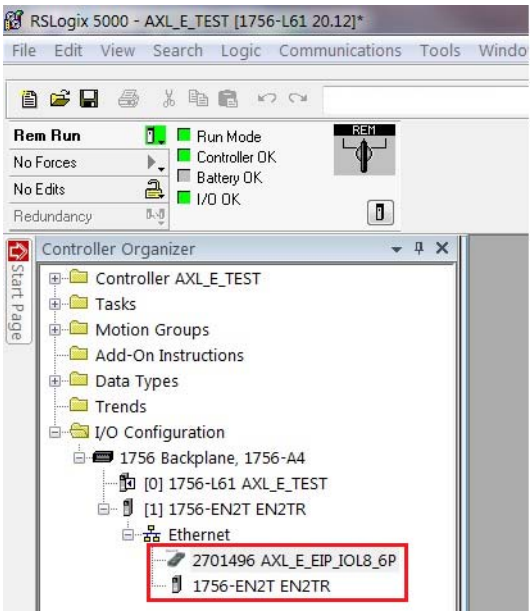


Figure 2-36 Device in the I/O Configuration folder

3 Setting Axioline E EtherNet/IP™ IO-Link devices

3.1 Assigning the IP address

The Axioline E EtherNet/IP™ IO-Link device supports several methods for assigning the IP address.

– BootP
– DHCP
– Rotary coding switch

With the aid of an example, this document describes how to assign an IP address via the Rockwell BOOTP/DHCP server tool. This tool is included in the RSLogix™ software package.

It is assumed that the Axioline E EtherNet/IP™ IO-Link device has its factory settings, i.e., the rotary coding switch is set to position 00 and the device does not have an IP address.

- Open the “BOOTP/DHCP Server x.x” tool.
- In the “Request History?” field, you can see all the received BootP and/or DHCP requests.
- Double-click on the entry with the corresponding MAC address.

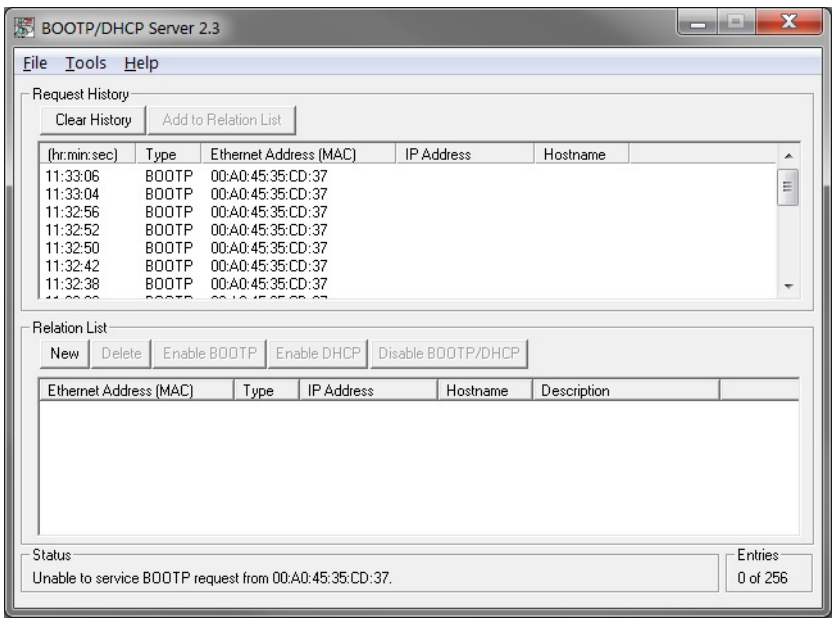


Figure 3-1 Double-clicking the MAC address

Phoenix Contact devices always have the MAC address 00:A0:45:XX:XX:XX.

- In the new window, give the device an IP address. This must be in the same network as the controller.
- Click “OK” to proceed.

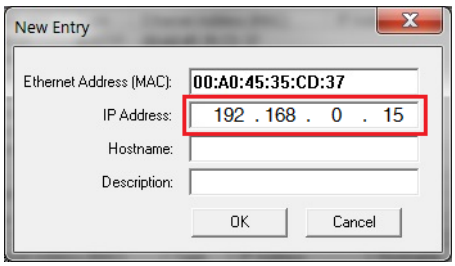


Figure 3-2 Assigning the IP address

A new entry is created in the “Relation List” field. A new entry which contains the assigned IP address appears in the “Request History” field.

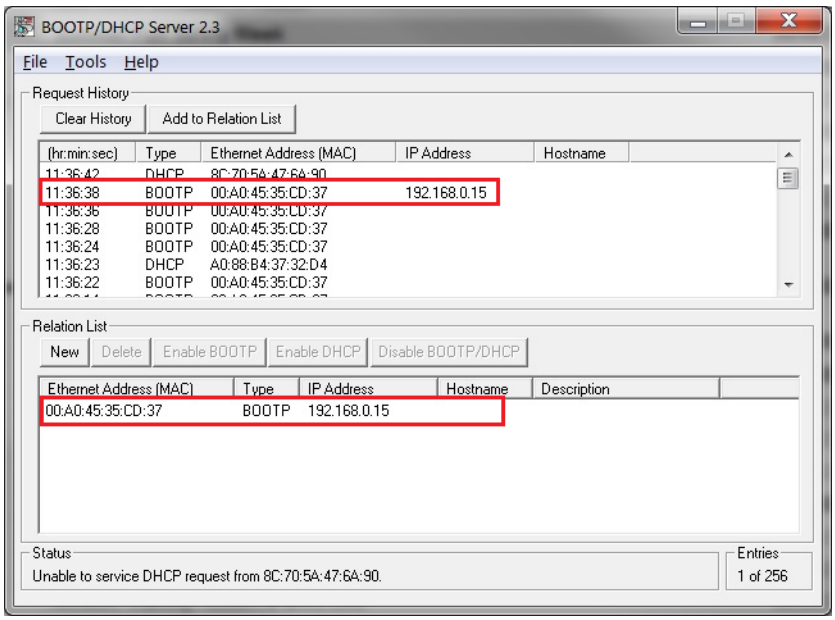


Figure 3-3 New entries in the “Request History” and “Relation List” fields

You can see whether the IP address assignment has been successful or not by the status of the device LEDs. The MOD LED changes to permanent green and the NET LED flashes green.

Once the IP address has been successfully assigned, switch the BootP mechanism off via the “Disable BOOTP/DHCP” button. This prevents the IP address from being changed accidentally following a power reset.

3.2 Setting Request Packet Interval (RPI)

You can change the RPI time for the Axioline E EtherNet/IP™ IO-Link devices. The smallest RPI time is 1 ms.

To set the RPI time perform the following steps:

- Double-click the Axioline E EtherNet/IP™ IO-Link device in your I/O configuration.
- Then select the “Connection” tab.

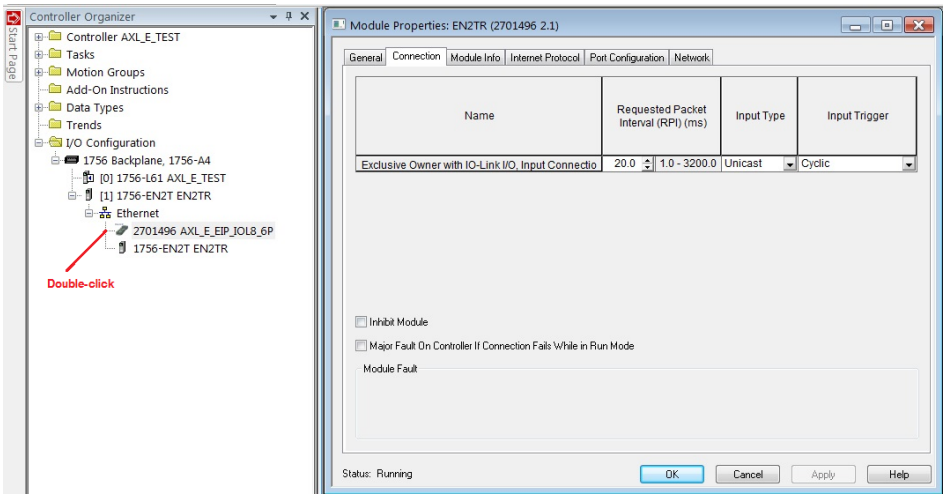


Figure 3-4 Changing the RPI time

- In the new window, set the required RPI time (1).
- Click “Apply” (2) to proceed.

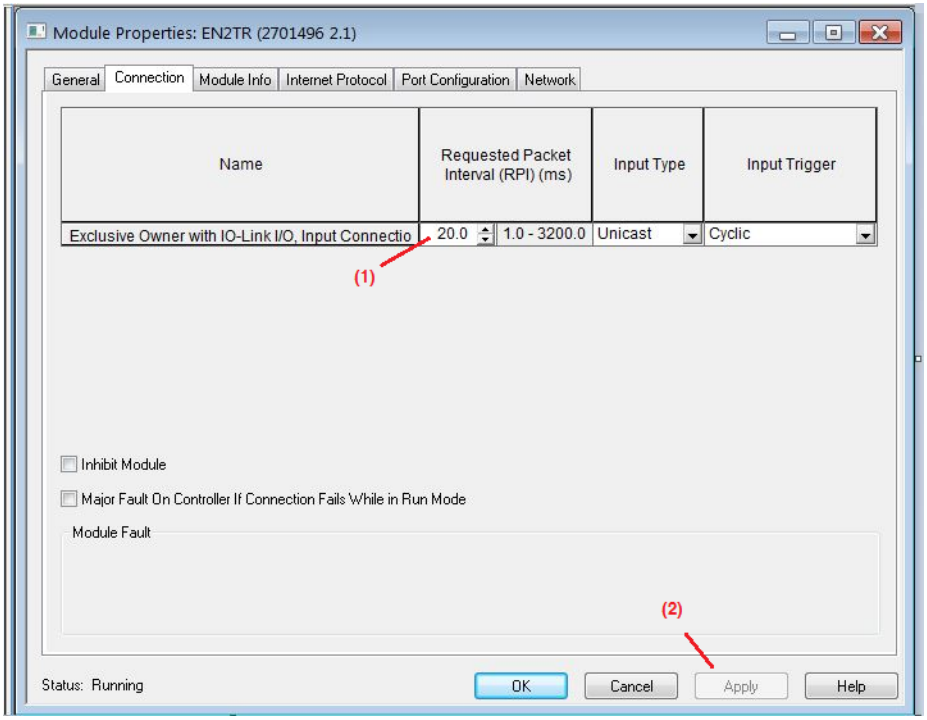


Figure 3-5 Setting the RPI time

The Axioline E EtherNet/IP™ devices support a minimum cycle time of 1 ms (RPI).

3.3 Configuring the IO-Link port

A port configuration which configures the operating mode for the relevant IO-Link port is required for the Axioline E EtherNet/IP™ IO-Link devices.

The options are as follows:

00_{dec}	Deactivated
01_{dec}	Digital input (DI)
02_{dec}	Digital output (DO)
03_{dec}	DI with IO-Link
04_{dec}	IO-Link (IOL)

In the default state all ports work in “digital input” mode.

For further information about the individual operating modes, please refer to the device-specific data sheet.

To set the operating mode perform the following steps:

- Switch to the configuration assembly of the IO-Link master in use by double-clicking on “Controller project name > Controller Tags”.

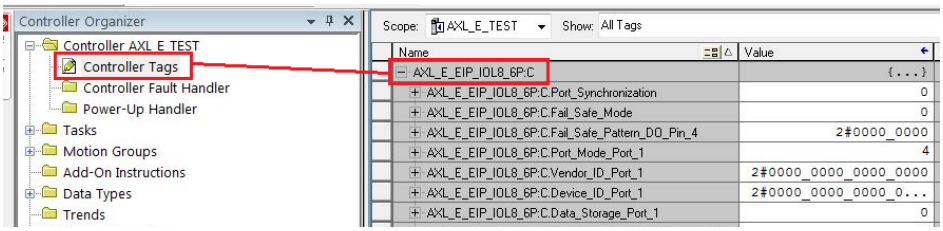


Figure 3-6 Switching to the configuration assembly

The entry “device name:C.Port_Mode_Port_X” exists for each port.

- Set the required operating mode here, e.g., 4_{dec} for the IO-Link mode.

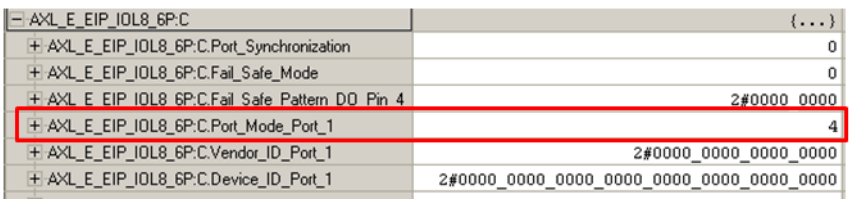


Figure 3-7 Setting the operating mode

3.4 Setting the IO-Link device identification

The Axioline E EtherNet/IP™ IO-Link devices support recognition and calibration of connected IO-Link devices. The vendor ID and the device ID can be set for each port.

As soon as the contents of the vendor ID and/or device ID do not equal “0”, the “Type Compatible” IO-Link inspection level is activated. It is only when both the parameterized vendor ID and device ID match the IDs that are read (in the device) that communication is established with the device (COM state bit of the corresponding port = 1), otherwise this is rejected (red IO-Link LED on).

To set the device ID perform the following steps:

- Switch to the configuration assembly of the IO-Link master in use by double-clicking on “Controller project name > Controller Tags”.

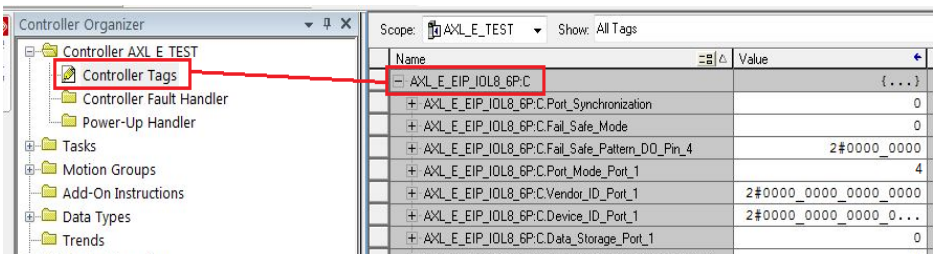


Figure 3-8 Switching to the configuration assembly

The entries “device name:C.Vendor_ID_Port_X” and “device name:C.Device_ID_Port_X” exist for each port.

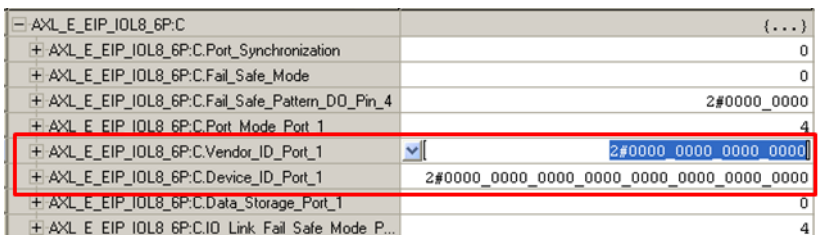
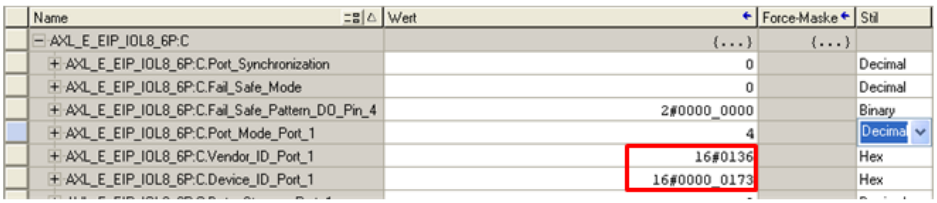


Figure 3-9 Entries “device name:C.Vendor_ID_Port_X” and “device name:C.Device_ID_Port_X”.

In order for the set vendor ID and device ID to be evaluated, the port must operate in IO-Link mode (04d).

Using the example of a capacitive sensor KQ6005 from ifm, the following settings are required: vendor ID 310d (01 36 h) and device ID 371d (00 01 73 h).



Name	Wert	Force-Maske	Stil
AXL_E_EIP_IOL8_6P:C	{...}	{...}	
AXL_E_EIP_IOL8_6P:C.Port_Synchronization	0		Decimal
AXL_E_EIP_IOL8_6P:C.Fail_Safe_Mode	0		Decimal
AXL_E_EIP_IOL8_6P:C.Fail_Safe_Pattern_DO_Pin_4	2#0000_0000		Binary
AXL_E_EIP_IOL8_6P:C.Port_Mode_Port_1	4		Decimal
AXL_E_EIP_IOL8_6P:C.Vendor_ID_Port_1	16#0136		Hex
AXL_E_EIP_IOL8_6P:C.Device_ID_Port_1	16#0000_0173		Hex

Figure 3-10 Vendor ID and device ID settings

Although the device ID only consists of 3 bytes, the Axioline E EtherNet/IP™ IO-Link device has 4 bytes. Better EDS support is thereby achieved, as only standard data types are used.

Always complete this byte with zeros.

3.5 Setting the IO-Link data storage mechanism (parameter server)

The Axioline E EtherNet/IP™ IO-Link devices support the data storage mechanism, which enables parameters to be exchanged between the master and device.

Connected devices must support at least IO-Link specification v1.1 in order to use the function.

The options are as follows:

0000_{hex}	Deactivated
0001_{hex}	Download only
0002_{hex}	Upload only
0003_{hex}	Download/upload allowed
0004_{hex}	Cleared

For further information about the individual functions, please refer to the device-specific data sheet.

- To set the data storage mechanism perform the following steps:
- Switch to the configuration assembly of the IO-Link master in use by double-clicking on “Controller project name > Controller Tags”.

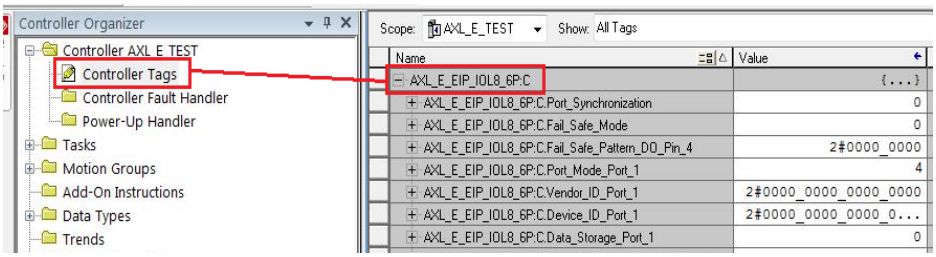


Figure 3-11 Switching to the configuration assembly

The entry “device name:C.Data_Storage_Port_X” exists for each port.

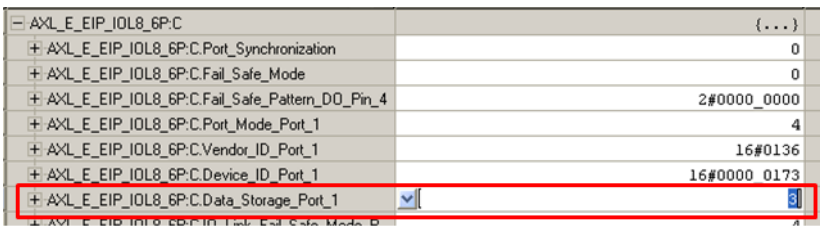


Figure 3-12 Entry “device name:C.Data_Storage_Port_X”

In order for the data storage mechanism to work, you should carry out the following steps:

Use of “Upload only” > “Download only”

1 Connect a fully parameterized IO-Link device to the appropriate port.
2 Configure the port to “Upload only”.
3 Download the program and switch the controller to online mode.
   – The parameters are now read by the device and stored in the master on the port.
4 The device can now be swapped with a new device.
5 Re-parameterize the port to “Download only”.
6 Download the program and switch the controller to online mode.
7 The parameters are transferred as soon as the master is started and the device has been detected.

Use of “Download/Upload allowed”

1 Connect a fully parameterized IO-Link device to the appropriate port.
2 Configure the port to “Download/Upload allowed”.
3 Download the program and switch the controller to online mode.
   – The parameters are now read by the device and stored in the master on the port.
4 The device can now be swapped with a new device.
   – The parameters are written to the device from the master.

3.6 Reading IO-Link device events (diagnostics)

The Axioline E EtherNet/IP™ IO-Link devices support IO-Link events, which represent the diagnostic data from IO-Link devices.

So that the IO-Link event data can be mapped in the process data, specific connection types must be set for the device, these are:

– Exclusive Owner with IO-Link I/O + Status + Events
– Input Only with IO-Link input + Status + Events
– Listen Only with IO-Link input + Status + Events

For further information about the individual functions and the exact structure of the process data, please refer to the device-specific data sheet.

To set the connection type perform the following steps:

- Double-click the Axioline E EtherNet/IP™ IO-Link device (1) to open the settings.
- Then click on “Change” (2) in the “Module Definition” field.

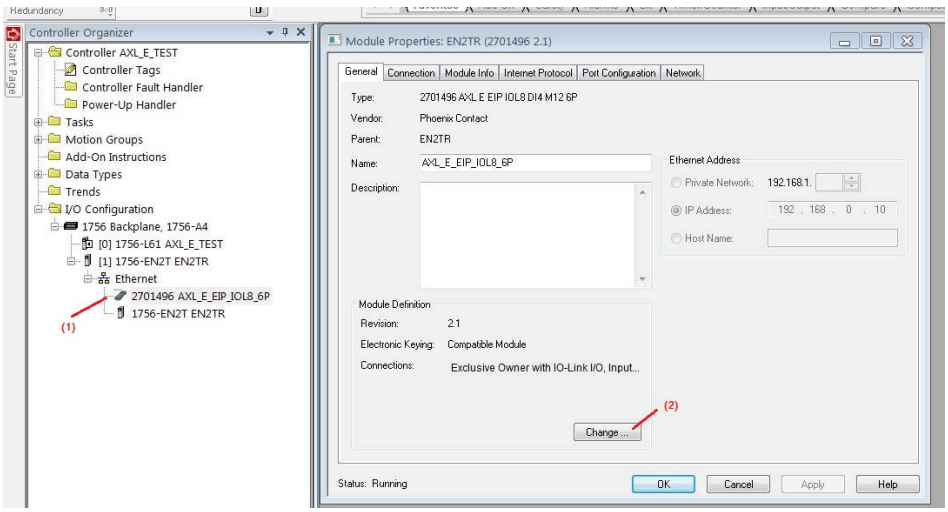


Figure 3-13 Setting the connection type

- In the new window, select the required connection type with "... + Events" (1).
- Click "OK" (2) to proceed.

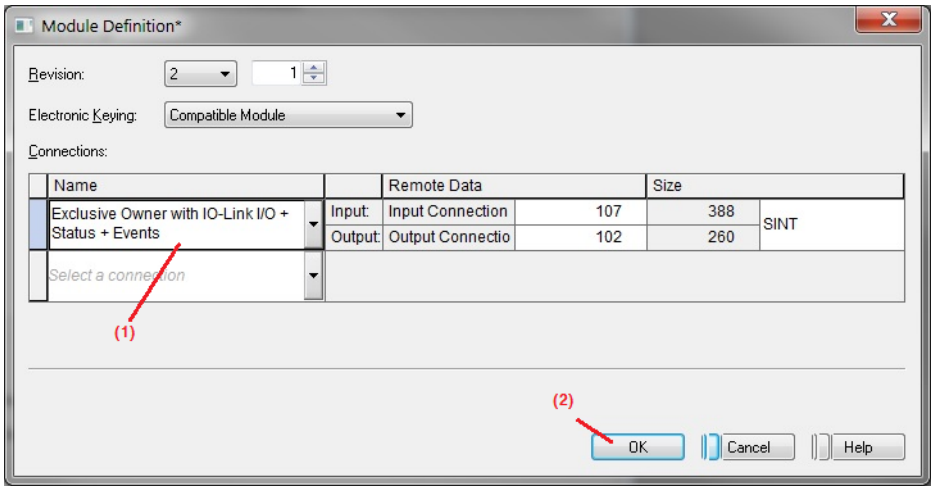
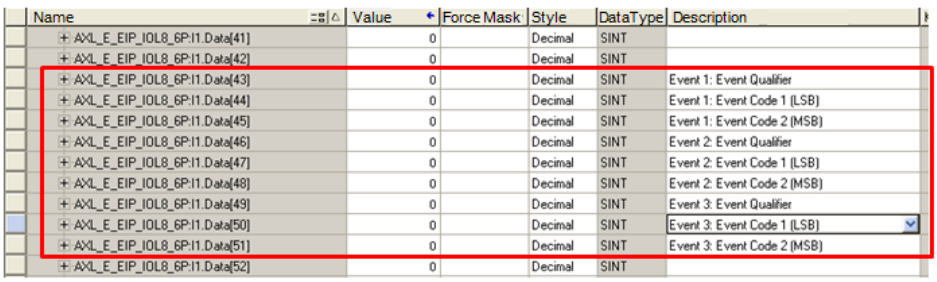


Figure 3-14 Selecting the connection type

The event data is mapped, for example for port 1, with Exclusive Owner connection type, to bytes 43 … 51 (starting at byte 0).



Name	Value	Force Mask	Style	DataType	Description
AXL_E_EIP_IOL8_6P:I1.Data[41]	0		Decimal	SINT	
AXL_E_EIP_IOL8_6P:I1.Data[42]	0		Decimal	SINT	
AXL_E_EIP_IOL8_6P:I1.Data[43]	0		Decimal	SINT	Event 1: Event Qualifier
AXL_E_EIP_IOL8_6P:I1.Data[44]	0		Decimal	SINT	Event 1: Event Code 1 (LSB)
AXL_E_EIP_IOL8_6P:I1.Data[45]	0		Decimal	SINT	Event 1: Event Code 2 (MSB)
AXL_E_EIP_IOL8_6P:I1.Data[46]	0		Decimal	SINT	Event 2: Event Qualifier
AXL_E_EIP_IOL8_6P:I1.Data[47]	0		Decimal	SINT	Event 2: Event Code 1 (LSB)
AXL_E_EIP_IOL8_6P:I1.Data[48]	0		Decimal	SINT	Event 2: Event Code 2 (MSB)
AXL_E_EIP_IOL8_6P:I1.Data[49]	0		Decimal	SINT	Event 3: Event Qualifier
AXL_E_EIP_IOL8_6P:I1.Data[50]	0		Decimal	SINT	Event 3: Event Code 1 (LSB)
AXL_E_EIP_IOL8_6P:I1.Data[51]	0		Decimal	SINT	Event 3: Event Code 2 (MSB)
AXL_E_EIP_IOL8_6P:I1.Data[52]	0		Decimal	SINT	

Figure 3-15 Event data

3.7 Setting substitute value behavior (fail safe)

If EtherNet/IP™ communication fails, all device outputs are set to the parameterized substitute values.

3.7.1 Ports in “digital output” mode

If there are ports with “digital output (02d)” mode in your configuration, the substitute value behavior can be set via the following parameters.

Fail Safe Mode

– Sets the global behavior for all ports

Fail Safe Pattern DO Pin 4

– The behavior for each port can be set using the substitute value pattern.
– This only applies when “Substitute value pattern (03d)” is parameterized in the previous parameter.

To set the substitute value behavior perform the following steps:

- Switch to the configuration assembly of the IO-Link master in use by double-clicking on “Controller project name > Controller Tags”.

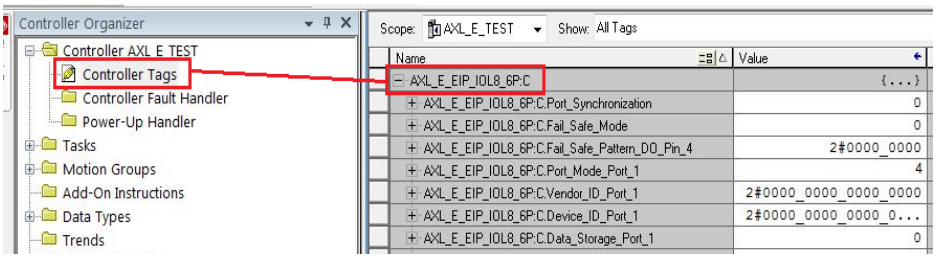


Figure 3-16 Switching to the configuration assembly

- Set both parameters.

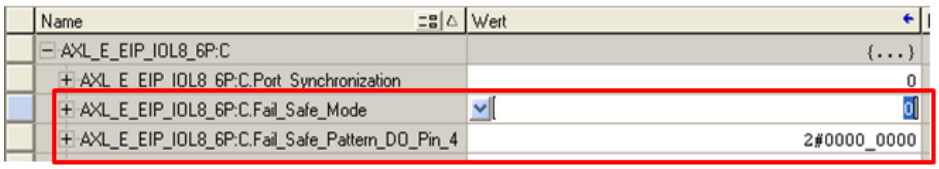


Figure 3-17 Parameter setting

Example: setting the substitute value pattern

Port 2, 5, and 8 work in “digital output” mode and should be set in the event of an error.

Port	1	2	3	4	5	6	7	8
Bit	0	1	2	3	4	5	6	7
Substitute value	0	1	0	0	1	0	0	1

Set the following (binary is selected as the style):

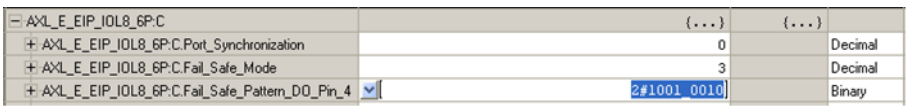


Figure 3-18 Setting substitute values

3.7.2 Port in the “IO-Link” mode

If there are ports with IO-Link (04d) mode in your configuration, the substitute value behavior can be set per port.

In the default state, “IO-Link master command (04d)” is set for the ports. The “IO-Link master command” option enables the use of IO-Link-specific mechanisms for valid/invalid OUT process data. Therefore the IO-Link device itself determines which substitute value behavior should be used.

Alternatively, the Axioline E EtherNet/IP™ IO-Link device can provide the substitute behavior. The following options are available.

00_{hex}	Reset (all bits to “0”)
01_{hex}	Set (all bits to “1”)
02_{hex}	Hold last value

To set the substitute value behavior perform the following steps:

- Switch to the configuration assembly of the IO-Link master in use by double-clicking on “Controller project name > Controller Tags”.

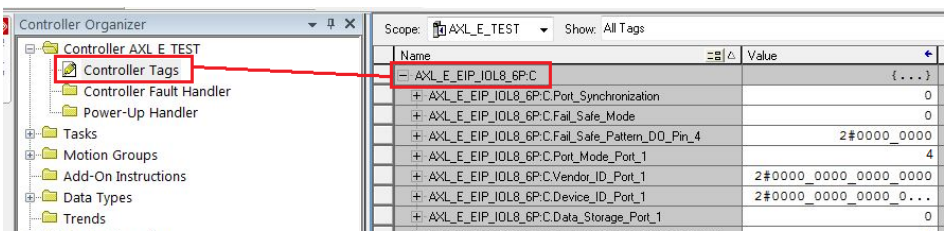


Figure 3-19 Switching to the configuration assembly

The entry “device name:C.IO_Link_Fail_Safe_Mode_Port_X” exists for each port.

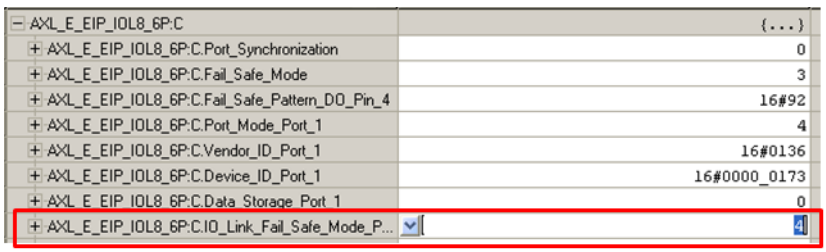


Figure 3-20 Entry “device name:C.IO_Link_Fail_Safe_Mode_Port_X”.

3.8 Setting the Device Level Ring (DLR)

The Axioline E EtherNet/IP™ IO-Link device supports the role of a DLR node in a DLR network.

The DLR protocol supports the 1-ring topology; multiple rings or overlapping rings are not possible. It is possible, when using suitable switches, to connect a redundant ring or operate multiple, restricted rings. As such, DLR protocol information may not leave the individual ring and appear in other rings.

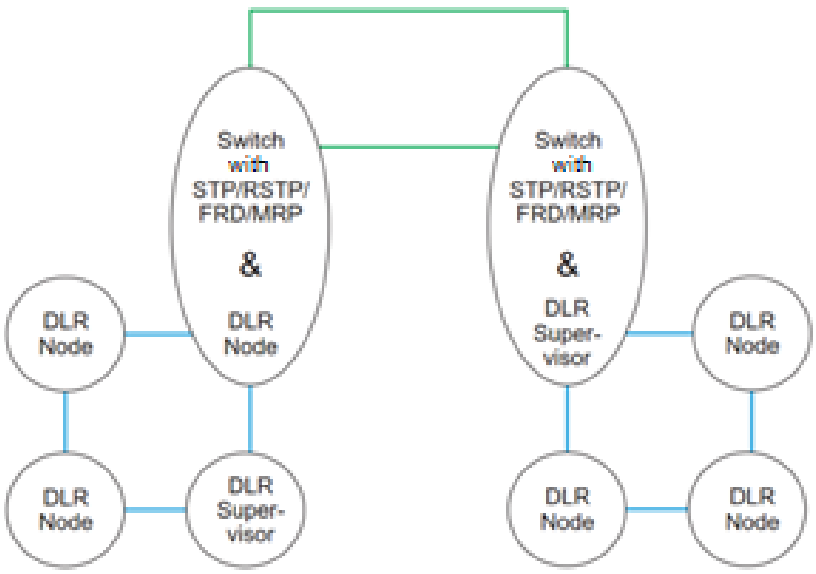


Figure 3-21 Redundant coupling of two DLR rings

- To configure the Device Level Ring protocol (DLR), use a DLR supervisor-capable switch, e.g., FL Switch 7008-EIP (Order No. 2701418) from Phoenix Contact or a corresponding controller.
- Configure your Ethernet/IP™ switch as DLR supervisor.
- To do so, follow the documentation for your device.

As an example, here you can see the settings on the FL SWITCH 7008-EIP:

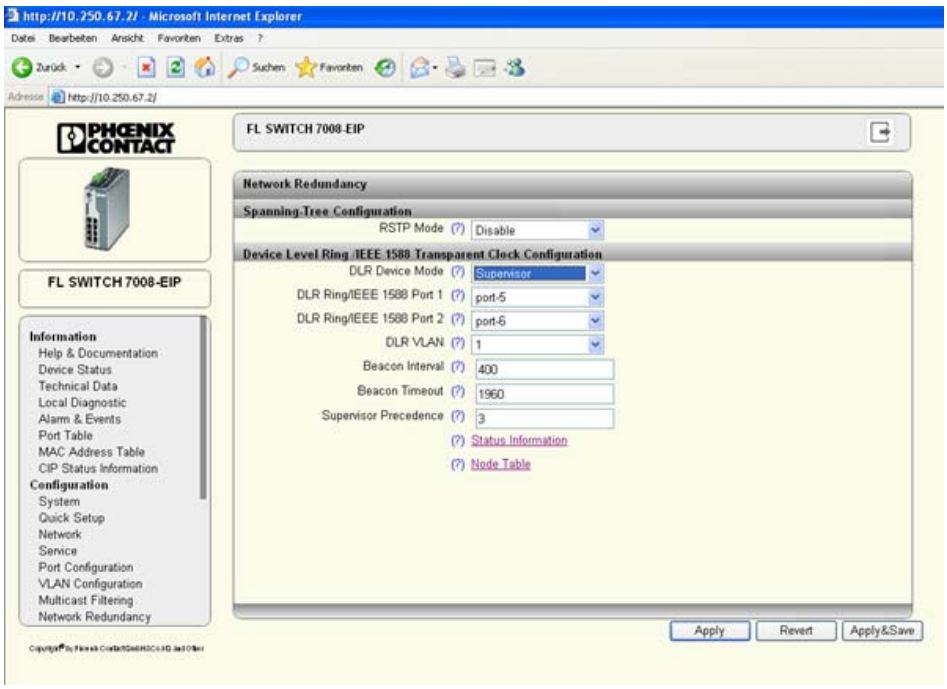


Figure 3-22 Settings on the FL SWITCH 7008-EIP

You do not need to make any further settings on the Axioline E EtherNet/IP™ IO-Link device, as all the necessary parameters, such as beacon interval, are specified by the DLR supervisor.

In RSLogix™ you can check the network topology and the status.

Perform the following steps:

- Double-click the Axioline E EtherNet/IP™ IO-Link device to open the settings.

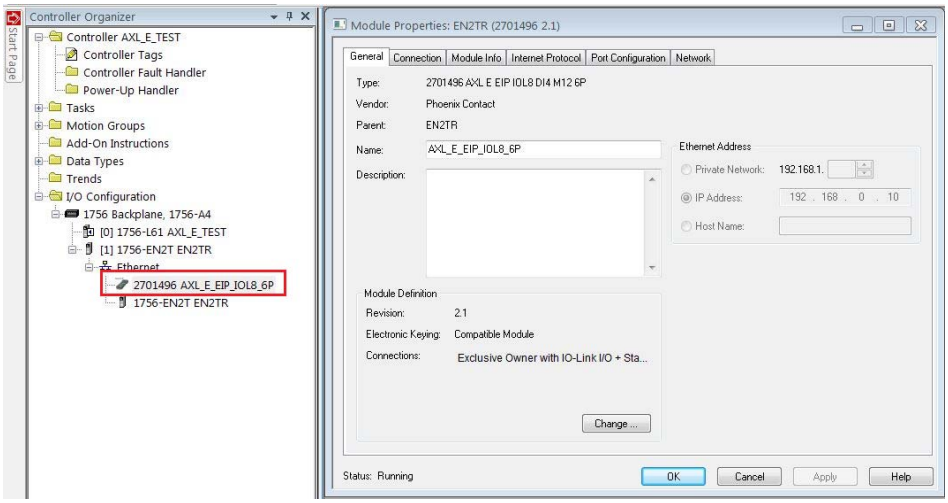


Figure 3-23 Opening the settings

- Then select the “Network” tab.
- You can see all the necessary status information in a new window in the top left corner.

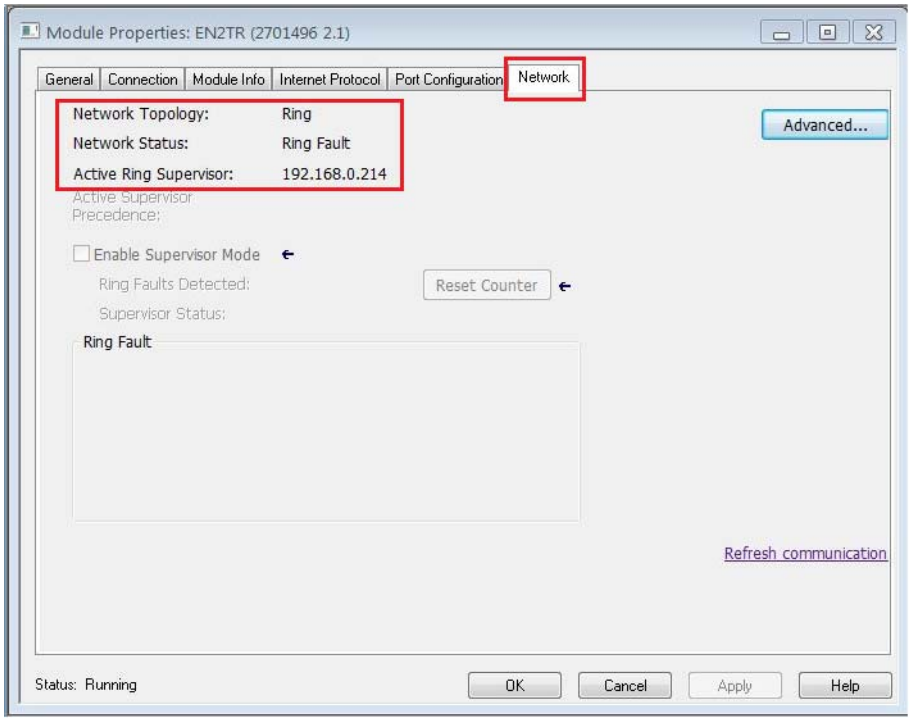


Figure 3-24 Status information

3.9 Using Quick Connect

In principle, the Quick Connection function can be activated/deactivated in a variety of ways. You can basically use any device, PLC or tool that can send CIP messages for this.

As an example, this is demonstrated below with the RSLogix5000 PLC program.

Perform the following steps:

- Select the “New Tag…” (2) option by right-clicking the “Controller Tags” (1) element in the project tree.

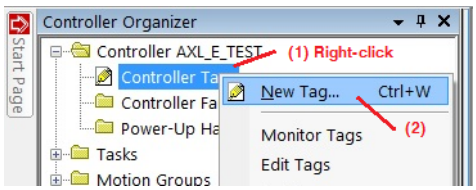


Figure 3-25 Selecting the “New Tag...” option

- Then enter a name and select the “MESSAGE” data type under the “Data Type” parameter.
- Then click “Create” to add the tag to the controller tags.

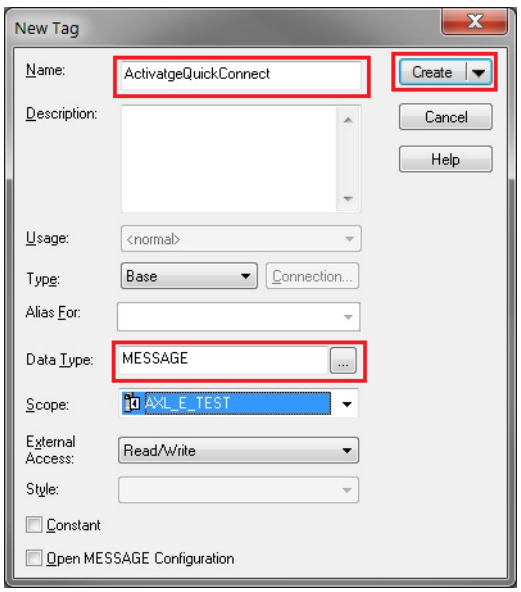


Figure 3-26 “New Tag” window, MESSAGE data type

One more tag is needed to define the data for the CIP message. Only one byte is needed to activate “Quick Connect”. Therefore “SINT” is selected as the data type.

- To do this, repeat the previous steps.

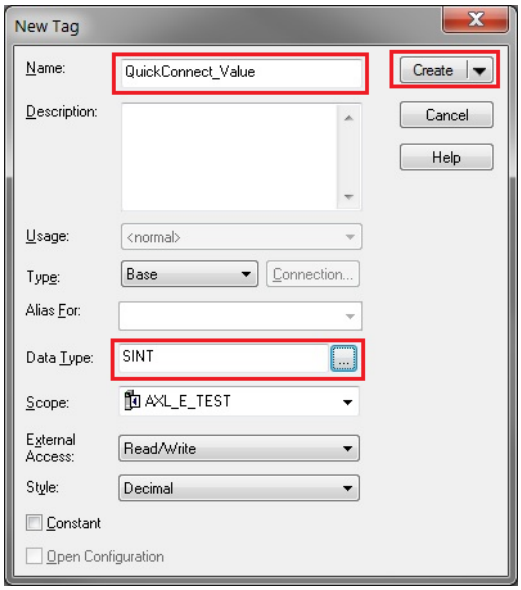


Figure 3-27 “New Tag” window, SINT data type

The tag value indicates whether Quick Connect is activated or deactivated (0 = deactivated, 1= activated).

The two new tags are now displayed with the controller tags.

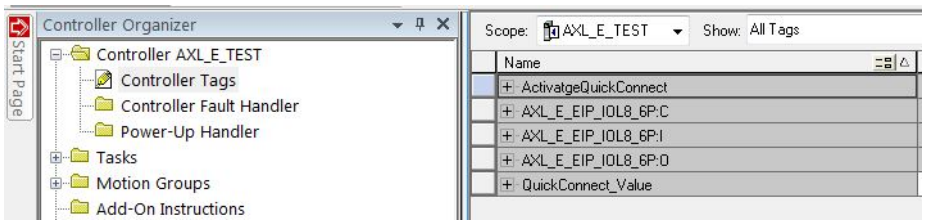


Figure 3-28 New tags displayed

- Navigate to the “MainRoutine” of your program.

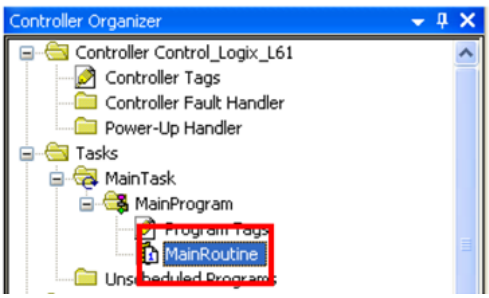


Figure 3-29 “MainRoutine”

- Then insert a “Message Instruction” (MSG) from the Input/Output element group or an empty rung into your program.

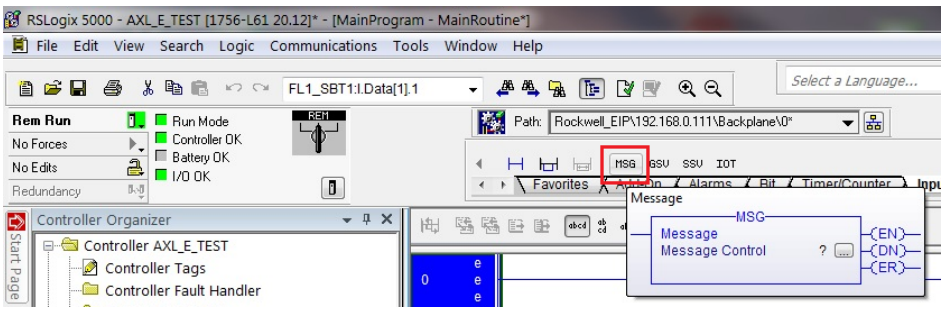


Figure 3-30 Inserting a Message Instruction

- Double-click on the question mark to assign the previously created “ActivateQuickConnect” MESSAGE tag to the newly created “Message Instruction”.

Figure 3-31 Assigning the MESSAGE tag to the newly created “Message Instruction”.

- Configure the CIP message via the small button next to the “Message Instruction”.

Figure 3-32 Configuring the CIP message

- Configure the “Message Instruction” as follows.

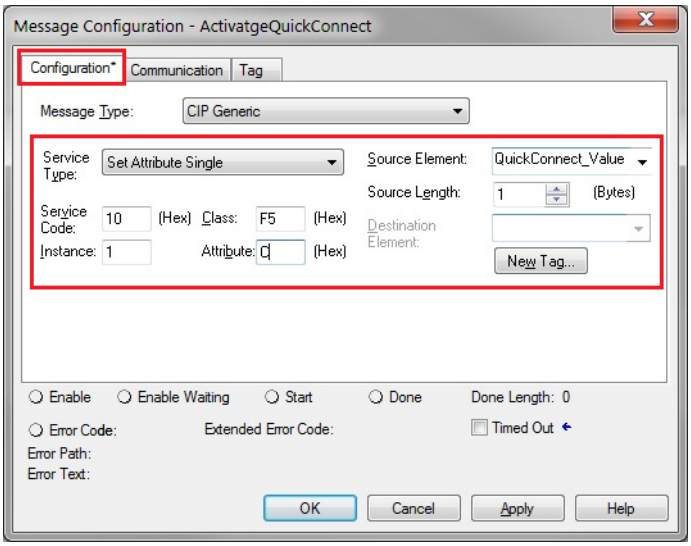


Figure 3-33 Configuring the “Message Instruction”

- Finally, set the communication path for the “Message Instruction”.
- To do so, click the “Communication” tab.
- Select the Axioline E EtherNet/IP™ IO-Link device via the “Browse” button.

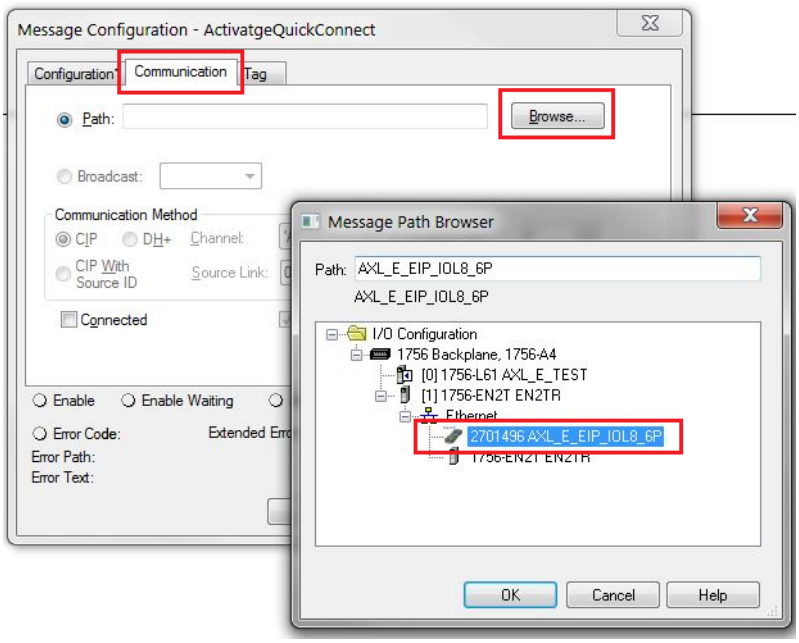


Figure 3-34 Selecting the Axioline E EtherNet/IP™ IO-Link device

For additional information, please refer to the documentation for RSLogix™.

4 Starting up RSLogix™

4.1 Creating variables

The RSLogix™ engineering tool automatically creates controller tags for an inserted I/O device. These tags help enable direct access to input and output data, even online.

Therefore, this document does not cover the creation of further variables for reading/writing data.

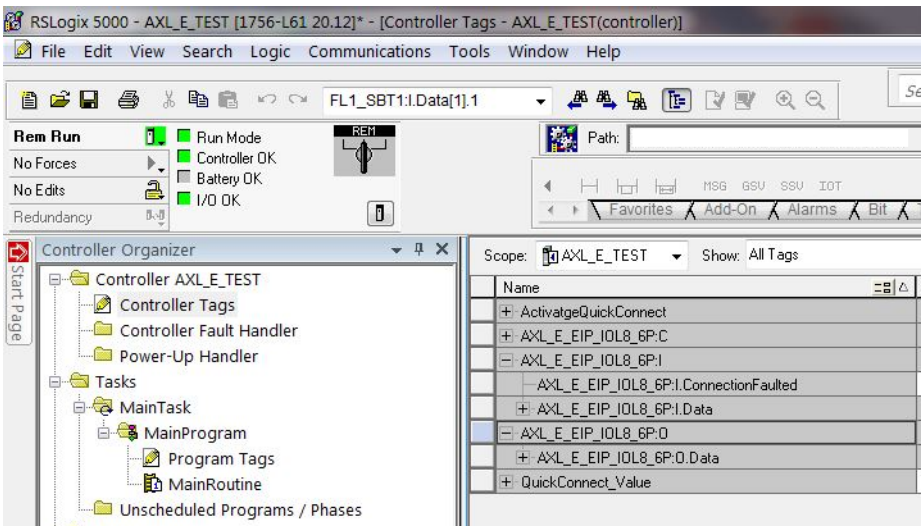


Figure 4-1 Controller tags

For additional information, please refer to the documentation for RSLogix™.

4.2 Compiling and building a project

In order for the created project to be sent to the controller, it must not contain any errors.

- Carry out the following steps to check your project for possible errors:
- Verify the created project for errors by clicking “Verify Controller”.

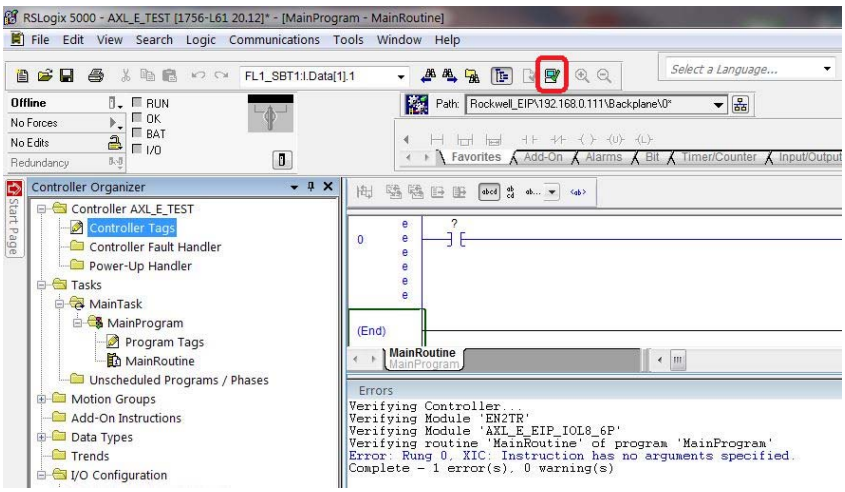


Figure 4-2 Checking the program for errors

- Correct any errors or warnings that arise so that the project can be transferred to the controller.
- Save the project by clicking the “Save” button (alternatively press CTRL+S or the menu “File > Save”).

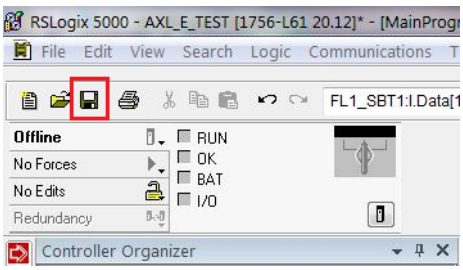


Figure 4-3 Saving the project

For additional information, please refer to the documentation for RSLogix™.

4.3 Sending the project to the controller and starting it

It is assumed that you have already created a project in RSLogix™ and verified it. Both the controller and the I/O device must have a valid
IP address.

- Carry out the following steps to send your project to the controller:
- In the status bar, click the “Offline” field and select the “Go Online” option.

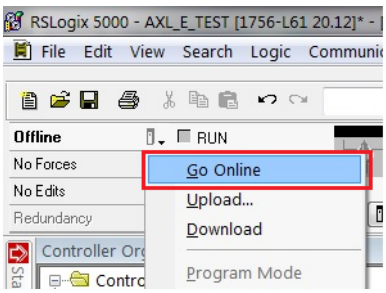


Figure 4-4 Go Online

- In the new window, click “Download”.

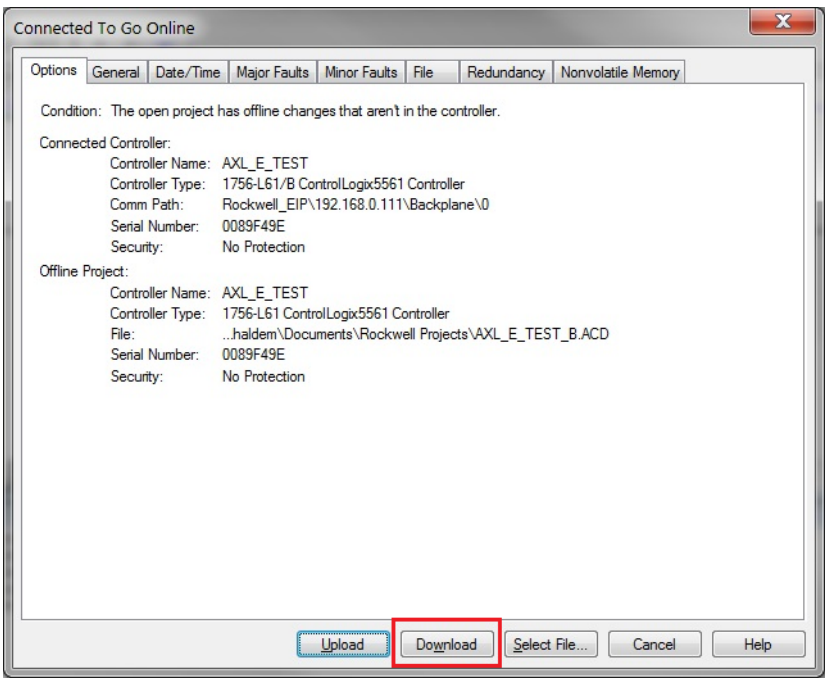


Figure 4-5 Downloading the project to the controller

- Respond to the following warning message by clicking “Download”.

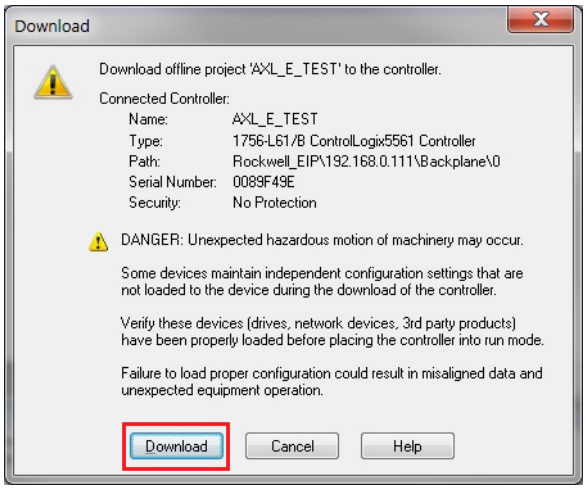


Figure 4-6 Warning message

Your controller's switch must be in REM or PROG mode, otherwise it is not possible to download the project.

As soon as the project has been successfully transmitted to the controller, the indicators in the status bar change to green.

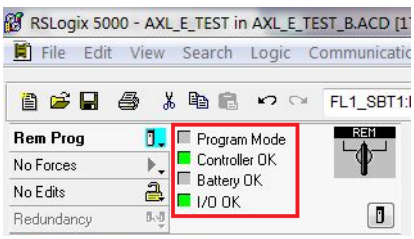


Figure 4-7 Green indicators in the status bar

If the controller is switched to RUN mode, the top indicator switches from program mode to RUN mode.

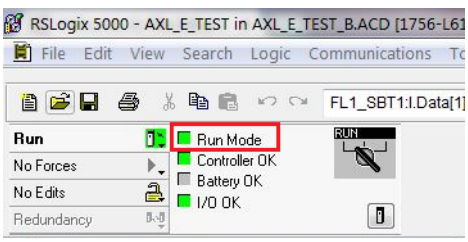


Figure 4-8 Green RUN mode indicator

4.4 Testing the communication

An executable project is not created. Instead, the automatically created controller tags are simply used to demonstrate how to read inputs and write outputs.

4.4.1 Reading digital inputs

On the Axioline E EtherNet/IP™ IO-Link device, the process data of the hard-wired inputs and also the IO-Link ports in “digital input (01d)” operating state is in byte 2 and byte 3 of the input process data.

Perform the following steps:

- Verify your project and switch your controller to RUN mode.

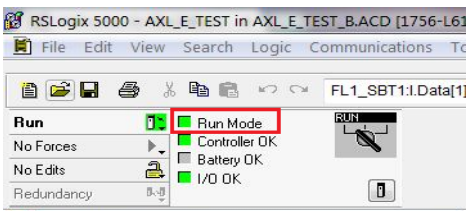


Figure 4-9 Controller in RUN mode

- Switch to the input assembly of the IO-Link master in use by double-clicking on “Controller project name > Controller Tags”.

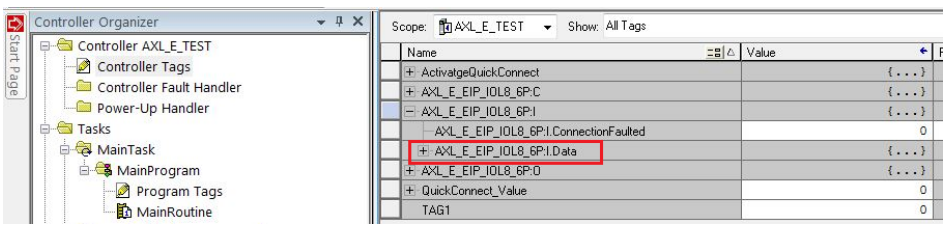


Figure 4-10 Switching to the input assembly

In byte 2, the DI state of the IO-Link ports in the “digital input (01d) operating state is shown.
In byte 3, the DI state of the hard-wired inputs of the type A ports is shown.

+ AXL_E_EIP_IOL8_6P:C	{...}	{...}		_0232:27...	
− AXL_E_EIP_IOL8_6P:I1	{...}	{...}		_0232:27...	
AXL_E_EIP_IOL8_6P:I1.ConnectionFaulted	0		Decimal	BOOL	
− AXL_E_EIP_IOL8_6P:I1.Data	{...}	{...}	Decimal	SINT[388]	
+ AXL_E_EIP_IOL8_6P:I1.Data[0]	0		Decimal	SINT	COM State
+ AXL_E_EIP_IOL8_6P:I1.Data[1]	0		Decimal	SINT	PD Valid State
− AXL_E_EIP_IOL8_6P:I1.Data[2]	4		Decimal	SINT	DI State of Pin 4
AXL_E_EIP_IOL8_6P:I1.Data[2].0	0		Decimal	BOOL	DI State of Pin 4: Port 1 (X01)
AXL_E_EIP_IOL8_6P:I1.Data[2].1	0		Decimal	BOOL	DI State of Pin 4: Port 2 (X02)
AXL_E_EIP_IOL8_6P:I1.Data[2].2	1		Decimal	BOOL	DI State of Pin 4: Port 3 (X03)
AXL_E_EIP_IOL8_6P:I1.Data[2].3	0		Decimal	BOOL	DI State of Pin 4: Port 4 (X04)
AXL_E_EIP_IOL8_6P:I1.Data[2].4	0		Decimal	BOOL	DI State of Pin 4: Port 5 (X05)
AXL_E_EIP_IOL8_6P:I1.Data[2].5	0		Decimal	BOOL	DI State of Pin 4: Port 6 (X06)
AXL_E_EIP_IOL8_6P:I1.Data[2].6	0		Decimal	BOOL	DI State of Pin 4: Port 7 (X07)
AXL_E_EIP_IOL8_6P:I1.Data[2].7	0		Decimal	BOOL	DI State of Pin 4: Port 8 (X08)
− AXL_E_EIP_IOL8_6P:I1.Data[3]	8		Decimal	SINT	DI State of Pin 2
AXL_E_EIP_IOL8_6P:I1.Data[3].0	0		Decimal	BOOL	DI State Pin 2: Port 1 (X01)
AXL_E_EIP_IOL8_6P:I1.Data[3].1	0		Decimal	BOOL	DI State Pin 2: Port 2 (X02)
AXL_E_EIP_IOL8_6P:I1.Data[3].2	0		Decimal	BOOL	DI State Pin 2: Port 3 (X03)
AXL_E_EIP_IOL8_6P:I1.Data[3].3	1		Decimal	BOOL	DI State Pin 2: Port 4 (X04)
AXL_E_EIP_IOL8_6P:I1.Data[3].4	0		Decimal	BOOL	Reserved
AXL_E_EIP_IOL8_6P:I1.Data[3].5	0		Decimal	BOOL	Reserved
AXL_E_EIP_IOL8_6P:I1.Data[3].6	0		Decimal	BOOL	Reserved
AXL_E_EIP_IOL8_6P:I1.Data[3].7	0		Decimal	BOOL	Reserved

Figure 4-11 Example: digital inputs on IO-Link port 3 and hard-wired input on port 4

4.4.2 Writing digital outputs

On the Axioline E EtherNet/IP™ IO-Link device, the process data of the IO-Link ports in “digital output (02d)” operating state is in byte 2 of the output process data.

Perform the following steps:

- Verify your project and switch your controller to RUN mode.

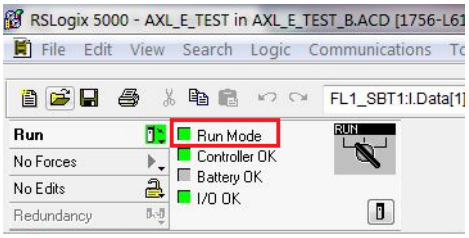


Figure 4-12 Controller in RUN mode

- Switch to the output assembly of the IO-Link master in use by double-clicking on “Controller project name > Controller Tags”.

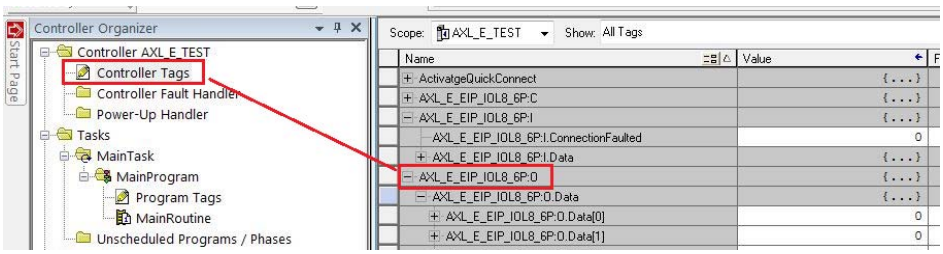


Figure 4-13 Switching to the output assembly

- Set the required output, provided the port is is “digital output (02d)” mode, via the corresponding bit.

+ AXL_E_EIP_IOL8_6P:C	{...}	{...}		_0232:27...	
+ AXL_E_EIP_IOL8_6P:I1	{...}	{...}		_0232:27...	
− AXL_E_EIP_IOL8_6P:O1	{...}	{...}		_0232:27...	
− AXL_E_EIP_IOL8_6P:O1.Data	{...}	{...}	Decimal	SINT[260]	
+ AXL_E_EIP_IOL8_6P:O1.Data[0]	0		Decimal	SINT	COM Control
+ AXL_E_EIP_IOL8_6P:O1.Data[1]	0		Decimal	SINT	Restart (Reboot)
− AXL_E_EIP_IOL8_6P:O1.Data[2]	-32		Decimal	SINT	DO state of Pin 4 (C/Q)
AXL_E_EIP_IOL8_6P:O1.Data[2].0	0		Decimal	BOOL	DO state of Pin 4 (C/Q): Port 1 (X01)
AXL_E_EIP_IOL8_6P:O1.Data[2].1	0		Decimal	BOOL	DO state of Pin 4 (C/Q): Port 2 (X02)
AXL_E_EIP_IOL8_6P:O1.Data[2].2	0		Decimal	BOOL	DO state of Pin 4 (C/Q): Port 3 (X03)
AXL_E_EIP_IOL8_6P:O1.Data[2].3	0		Decimal	BOOL	DO state of Pin 4 (C/Q): Port 4 (X04)
AXL_E_EIP_IOL8_6P:O1.Data[2].4	0		Decimal	BOOL	DO state of Pin 4 (C/Q): Port 5 (X05)
AXL_E_EIP_IOL8_6P:O1.Data[2].5	1		Decimal	BOOL	DO state of Pin 4 (C/Q): Port 6 (X06)
AXL_E_EIP_IOL8_6P:O1.Data[2].6	1		Decimal	BOOL	DO state of Pin 4 (C/Q): Port 7 (X07)
AXL_E_EIP_IOL8_6P:O1.Data[2].7	1		Decimal	BOOL	DO state of Pin 4 (C/Q): Port 8 (X08)
+ AXL_E_EIP_IOL8_6P:O1.Data[3]	0		Decimal	SINT	Reserved

Figure 4-14 Example: digital outputs on the IO-Link ports 6 to 8 set

4.4.3 Reading/writing IO-Link process data

The automatic controller tags can be used to read the process data of the connected IO-Link devices or to transmit data.

The size of the process data varies depending on your connection configuration. 32 bytes of process data are available per port, in addition certain status information can be shown.
The differences between the individual connection types can be found in the device-specific data sheet.

Make sure that you have set a connection type with IO-Link data for the Axioline E EtherNet/IP™ IO-Link device, e.g., “Exclusive Owner with IO-Link I/O”.

- Verify your project and switch your controller to RUN mode.

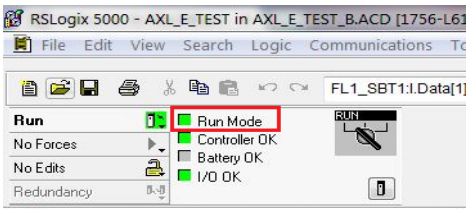


Figure 4-15 Controller in RUN mode

- Switch to the input or output assembly of the IO-Link master in use by double-clicking on “Controller project name > Controller Tags”.

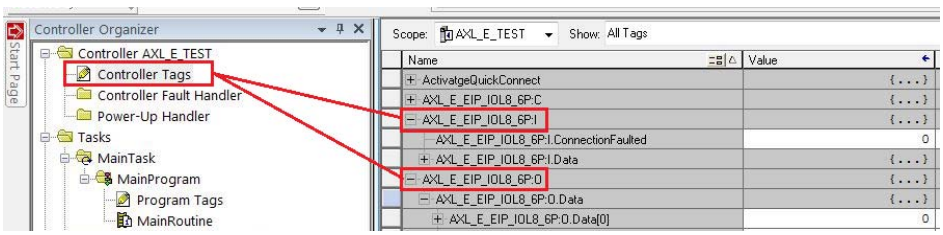
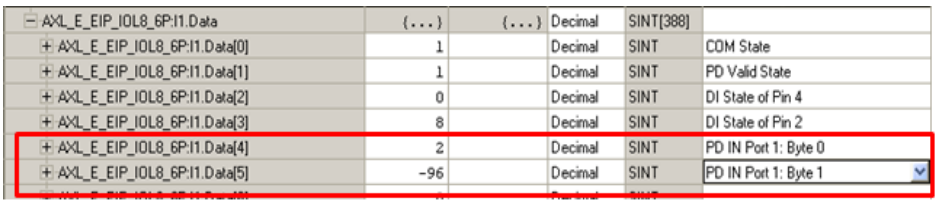


Figure 4-16 Switching to the input or output assembly

The IO-Link process data always begins from byte 4 onwards, both for the input and output process data, since the previous 4 bytes contain the status and controller bytes.



AXL_E_EIP_IOL8_6P:I1.Data	{...}	{...}	Decimal	SINT[388]	
AXL_E_EIP_IOL8_6P:I1.Data[0]	1		Decimal	SINT	COM State
AXL_E_EIP_IOL8_6P:I1.Data[1]	1		Decimal	SINT	PD Valid State
AXL_E_EIP_IOL8_6P:I1.Data[2]	0		Decimal	SINT	DI State of Pin 4
AXL_E_EIP_IOL8_6P:I1.Data[3]	8		Decimal	SINT	DI State of Pin 2
AXL_E_EIP_IOL8_6P:I1.Data[4]	2		Decimal	SINT	PD IN Port 1: Byte 0
AXL_E_EIP_IOL8_6P:I1.Data[5]	-96		Decimal	SINT	PD IN Port 1: Byte 1

Figure 4-17 Example: connected device occupies 2 bytes of input process data

Ethernet/IP™ uses the little endian format (Intel) while IO-Link uses the big endian format (Motorola). No byte swapping is carried out for IO-Link process data!

5 Reading/writing IO-Link device parameters

5.1 Basics

The Axioline E EtherNet/IP™ IO-Link device provides a manufacturer-specific IO-Link CIP object (80_{hex}) which enables access to the device parameters of IO-Link devices.

In addition to the class instance 0, the object also has instance 1, which addresses the IO-Link master.

The corresponding IO-Link master port is addressed with the aid of the instance attribute number. A precise description can be found in the device-specific data sheet.

Certain services can be sent to the IO-Link master via Message Instruction (MSG). The IO-Link master supports the following services:

Common services

Service code		Class	Instance	Service name
dec	hex			
14	0E	Yes	No	Get_Attribute_Single
75	4B	No	Yes	Read_ISDU
78	4C	No	Yes	Write_ISDU

The $4B_{hex}$ (Read_ISDU) and $4C_{hex}$ (Write_ISDU) services enable IO-Link device parameters to be read or written.

To do so, the IO-Link protocol uses ISDU objects (Index Service Data Unit). These have an index and a sub index. For a description of the ISDU objects supported by a device, please refer to the specific data sheet of the IO-Link device.

The IO-Link protocol uses ISDU objects (Index Service Data Unit) for acyclic data. Not all IO-Link devices have ISDU objects.

5.2 Requirements

You should have an executable project and observe the following points:

– The EDS file of the Axioline E EtherNet/IP™ IO-Link device is installed in RSLogix™.
– A connection can be established with the controller and the Axioline E EtherNet/IP™ IO-Link device.
– Custom data types and tags are created for the read and write services according to the IO-Link master data sheet.

5.3 Creating custom data types

A source and destination element is required for the Message Instruction.

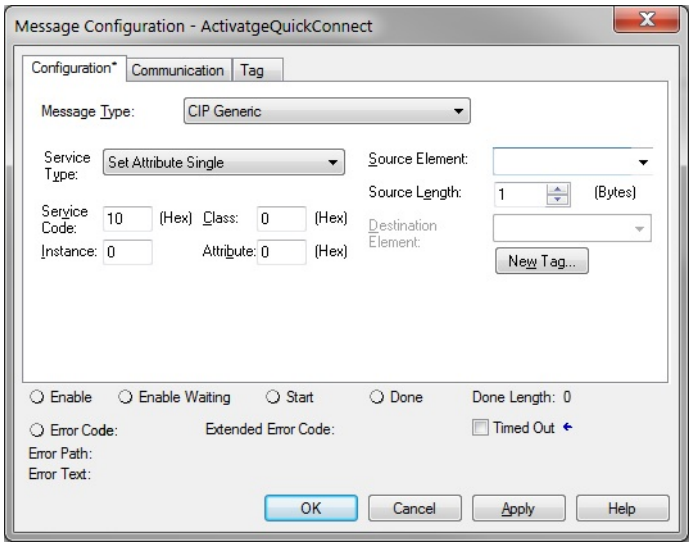


Figure 5-1 Message configuration

These elements must have a structure which corresponds to the IO-Link master data sheet.

5.3.1 Creating an ISDU_READ data type

To read device parameters, two new data types are required. The first is the ISDU_READ data type, in which the index and sub index of the object to be read is specified and the second is the ISDU_DATA data type, in which the read data is stored.

You can find a description of the ISDU_DATA data type in the following section.

Create the ISDU_READ data type as follows:

- Select the “New Data Type…” (2) option by right-clicking (1) the “Data Types” folder in the project tree.

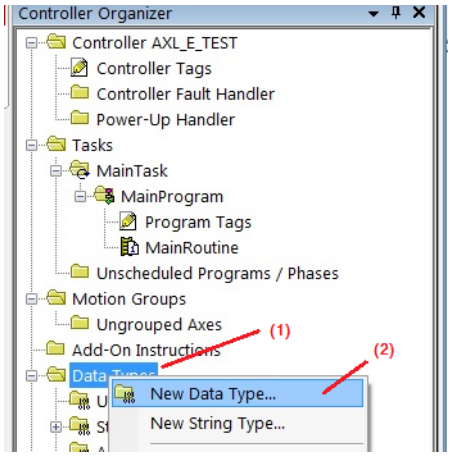


Figure 5-2 Selecting “New Data Type...”

- In the window that opens, specify the name (1) and the structure (2) of the new data type.

The new data type may look something like this:

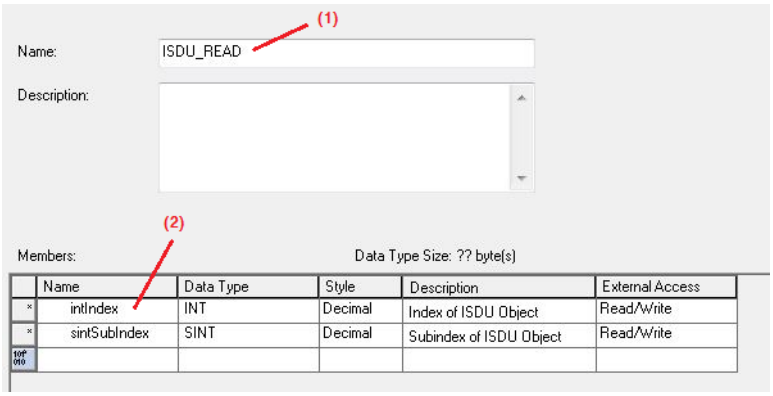


Figure 5-3 Specifying the name and structure of the new data type

For a description of the structure, please refer to the IO-Link master data sheet. It is important that you adhere to the structure and the length of the elements corresponds exactly to the data sheet specifications.

5.3.2 Creating an ISDU_DATA data type

It is advisable to have your own data type for storing read data and also for any error messages for the master or devices.

With IO-Link, a maximum of 232 bytes of parameter data can be transmitted. However, the new structure should be at least as large as the data size of the ISDU object to be read, otherwise the service will be terminated with a corresponding error message.

Create the ISDU_READ data type as follows:

- Select the “New Data Type…” (2) option by right-clicking the “Data Types” (1) folder in the project tree.

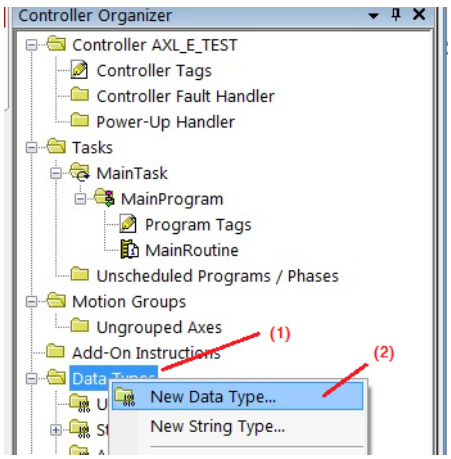


Figure 5-4 Selecting “New Data Type...”

- In the window that opens, specify the name (1) and the structure (2) of the new data type.

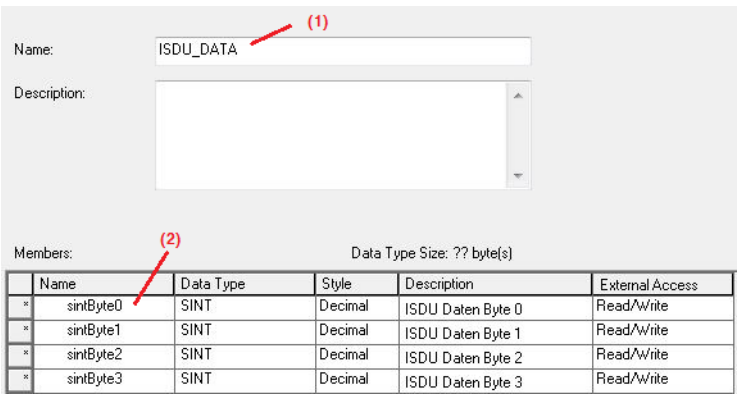


Figure 5-5 Specifying the name and structure of the new data type

5.3.3 Creating an ISDU_WRITE data type

Another new data type is required to read device parameters, since in addition to the index and sub index of the object to be described, the data must also be provided.

Create the ISDU_WRITE data type as follows:

- Select the “New Data Type…” (2) option by right-clicking the “Data Types” (1) folder in the project tree.

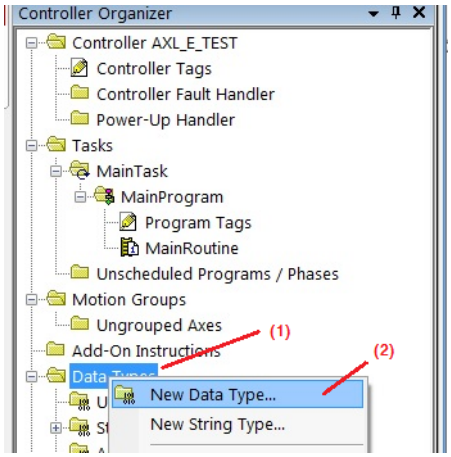
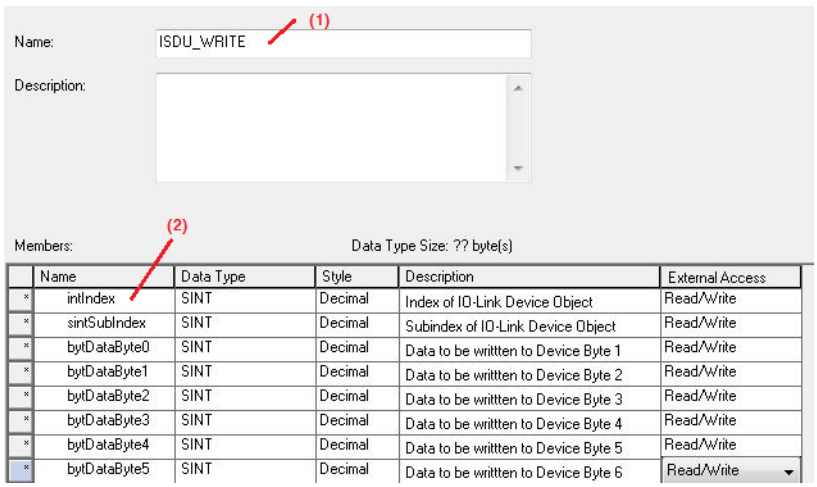


Figure 5-6 Selecting “New Data Type...”

In the window that opens, specify the name (1) and the structure (2) of the new data type. The new data type may look something like this.



Name	Data Type	Style	Description	External Access
intIndex	SINT	Decimal	Index of IO-Link Device Object	Read/Write
sintSubIndex	SINT	Decimal	Subindex of IO-Link Device Object	Read/Write
bytDataByte0	SINT	Decimal	Data to be writtten to Device Byte 1	Read/Write
bytDataByte1	SINT	Decimal	Data to be writtten to Device Byte 2	Read/Write
bytDataByte2	SINT	Decimal	Data to be writtten to Device Byte 3	Read/Write
bytDataByte3	SINT	Decimal	Data to be writtten to Device Byte 4	Read/Write
bytDataByte4	SINT	Decimal	Data to be writtten to Device Byte 5	Read/Write
bytDataByte5	SINT	Decimal	Data to be writtten to Device Byte 6	Read/Write

Figure 5-7 Specifying the name and structure of the new data type

For a description of the structure, please refer to the IO-Link master data sheet. It is important that you adhere to the structure and the length of the elements corresponds exactly to the data sheet specifications.

5.4 Creating tags

Once the customized data types in the previous section have been created, tags should be created for them.

To do so, perform the following steps:

- Select the “New Tag…” (2) option by right-clicking the “Controller Tags” (1) element in the project tree.

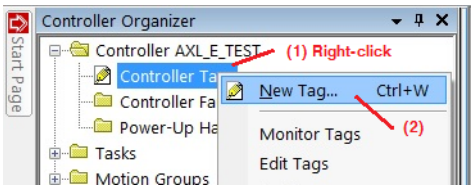


Figure 5-8 Selecting “New Tag...”

- Then assign a name (1).
- Under the “Data Type” parameter, select the newly defined data type (2).
- Click “Create” (3) to proceed.

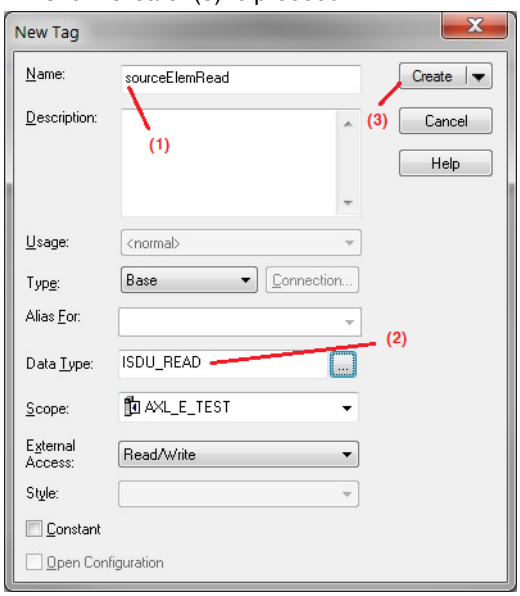


Figure 5-9 Assigning a name and setting the data type

These steps should be repeated for all created data types so that, in total, four new tags are created:
1 x tag for ISDU_READ, 1 x tag for ISDU_WRITE, 2 x tags for ISDU_DATA.

5.5 Creating a Message Instruction

As described above, device parameters can be read or written via Message Instructions (MSG). It is advisable to create two Message Instructions, because the service codes and the structure of the source elements differ.

The following notes describe the procedure for adding a Message Instruction into your own program.

- Navigate to the MainRoutine of your program.
- Then insert a “Message Instruction” (MSG) from the Input/Output tab into your program.

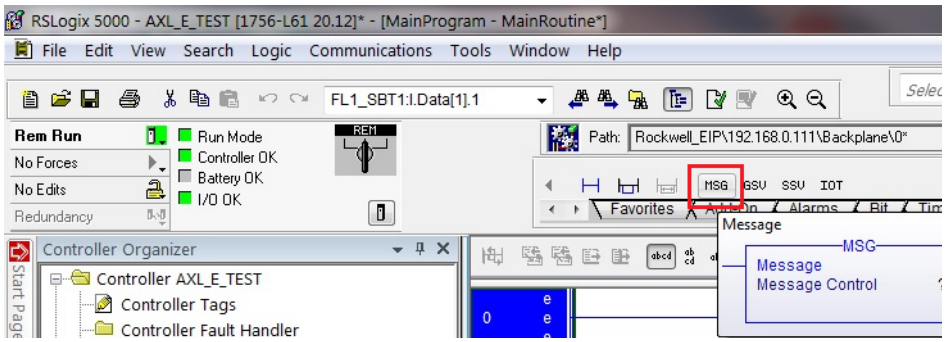


Figure 5-10 Inserting a Message Instruction

- Give the Message Instruction a name, e.g., msgReadISDU.

Figure 5-11 Assigning a name

- Then right-click on the name (1).
- Select the “New “msgReadISDU”” (2) option.

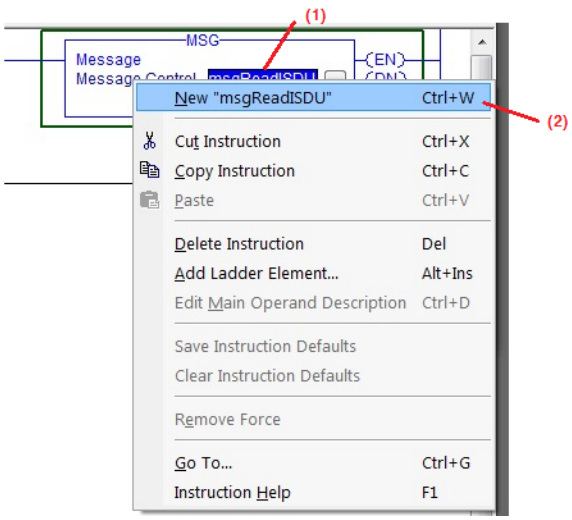


Figure 5-12 Selecting the “New “msgReadISDU”” option.

- A new window opens. Define the parameters for the new tag here.
- Adjust the parameters.
- Then click “Create” to create a new tag for the MSG.

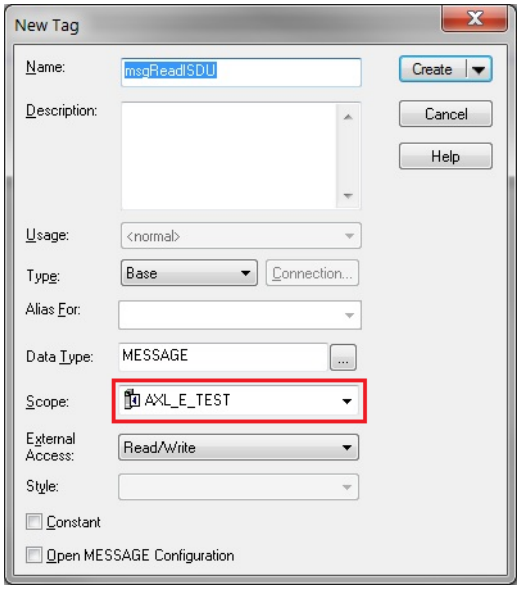


Figure 5-13 Defining the tag parameters

Make sure that your controller scope is selected in the “Scope” field, otherwise it will not be possible to create the tag.

- Then open the “configuration field” of the Message Instruction, by clicking the “...” button.

Figure 5-14 Opening the configuration field

- Configure the “Message Instruction” according to the specifications in the IO-Link master data sheet.

The following image shows an example configuration for a Message Instruction for reading device parameters.

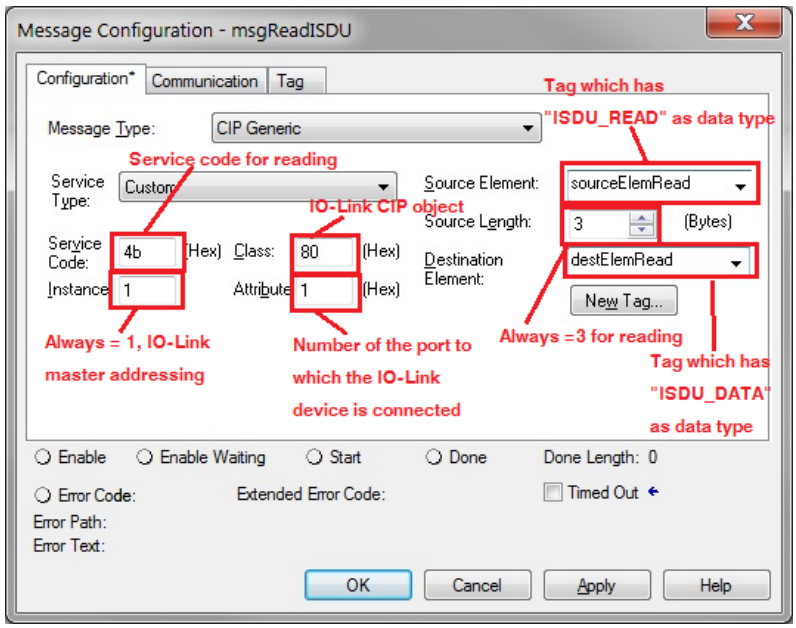


Figure 5-15 Example configuration for a Message Instruction (reading)

- The following image shows an example configuration for a Message Instruction for writing device parameters.

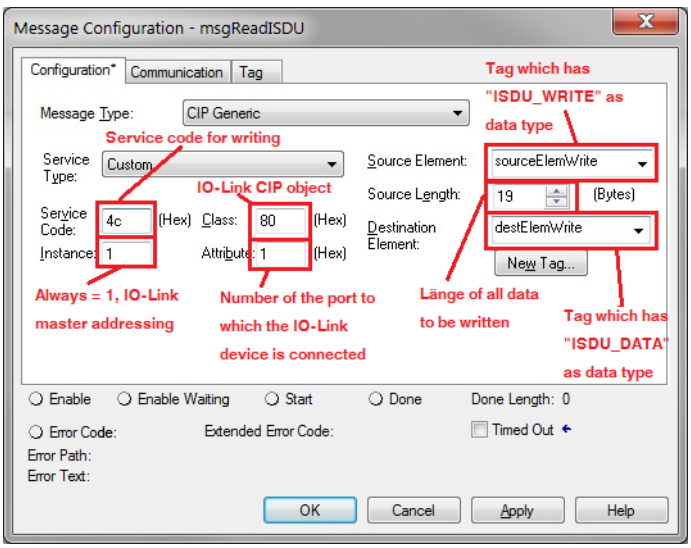


Figure 5-16 Example configuration for a Message Instruction (writing)

- Finally, set the communication path for the “Message Instruction”.
- To do so, click the “Communication” (1) tab.
- Select the Axioline E EtherNet/IP™ IO-Link device (3) via the “Browse” (2) button.

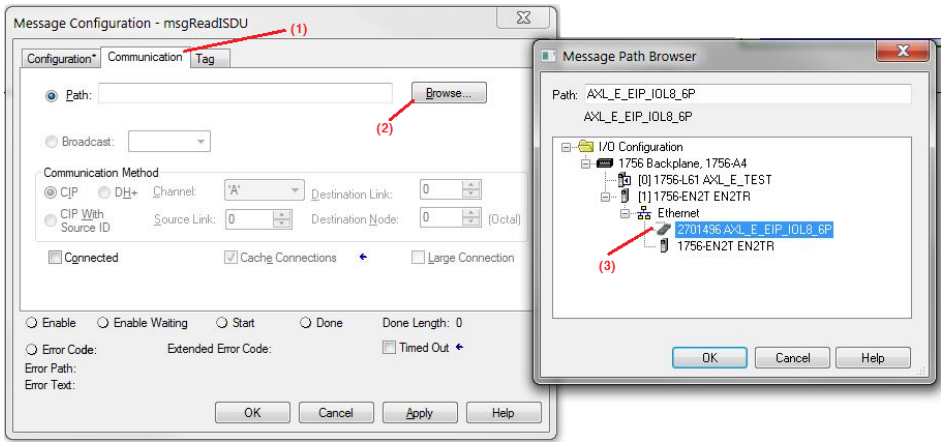


Figure 5-17 Setting the communication path

Error codes

If the service could not be successfully carried out, the destination element contains additional error codes from the master or the device.

The structure is as follows:

Name	Data type	Value/description	
IO-Link master error	UINT	1	Service not available
		2	Port blocked
		3	Timeout
		4	Invalid index
		5	Invalid sub index
		6	Wrong port
		7	Wrong port function
		8	Invalid length
		9	ISDU not supported
IO-Link device error	USINT	See specific device data sheet	
IO-Link device additional error	USINT	See specific device data sheet	

5.6 Reading IO-Link device parameters

This example uses the IO-Link device KQ6005 (capacitive sensor) from ifm.

– The sensor is connected to port 1 of the IO-Link master.
– “Exclusive Owner with IO-Link I/O” is set as the connection type.
– The ISDU object “Product ID” (19_{dec}) is to be read.

To do so, perform the following steps:

- Switch to the configuration assembly of the IO-Link master in use.

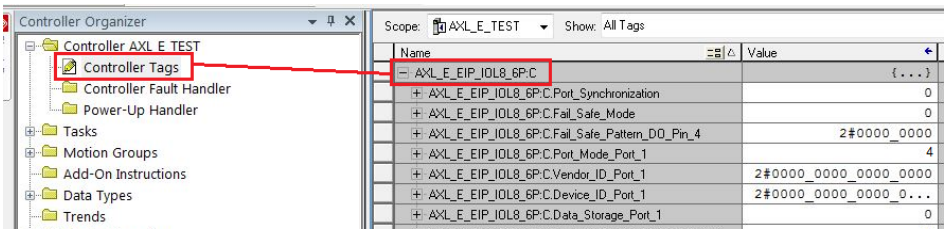


Figure 5-18 Switching to the configuration assembly

- Reconfigure the port in use, to which the actual device is connected, to “IO-Link” mode, this corresponds to value 4 in parameter “…:Port_Mode_Port_X”.

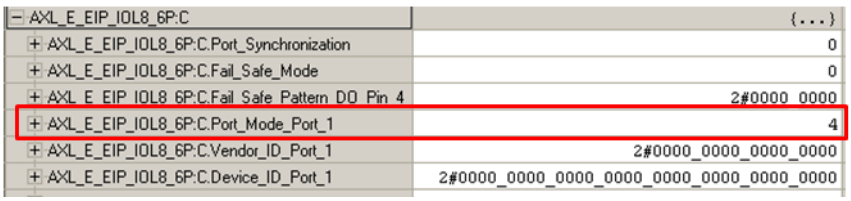


Figure 5-19 Reconfiguring the port

In this example, the tags “sourcElemRead” (with “ISDU_READ” data type) and “destElem-Read” (with “ISDU_DATA” data type) have been used.

The index and sub index of the object to be read are now specified in the “sourcElemRead” tag. In this case, the value 19 for the index and 0 for the sub index.

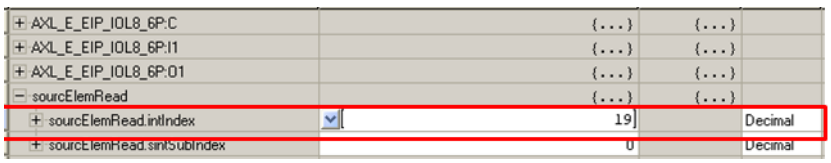


Figure 5-20 Value 19 for index and value 0 for sub index

The “destElemRead” tag does not need any more parameters, as the data is stored here. You simply need to ensure that the “ISDU_DATA” data type is at least as large as the size of the device ISDU object to be read. In this case, the object has a maximum size of 32 bytes.

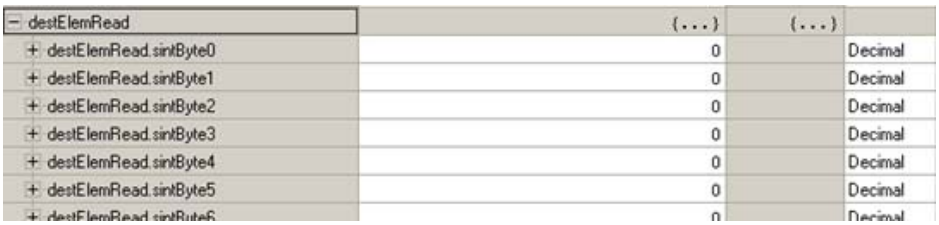


Figure 5-21 “destElemRead” tag

- Switch to the MainRoutine of your program.
- Insert a Message Instruction and an additional element, e.g., “Examine if closed” from the “Bit” element group in order to be able to toggle the Message Instruction. In this example, the name “msgReadISDU” is used for the Message Instruction and “bToggleRead” is used for the ToggleBit.

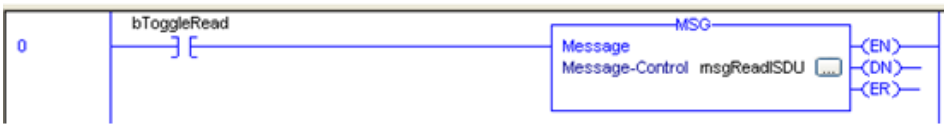


Figure 5-22 Message Instruction and ToggleBit example

The configuration of the Message Instruction looks like this:

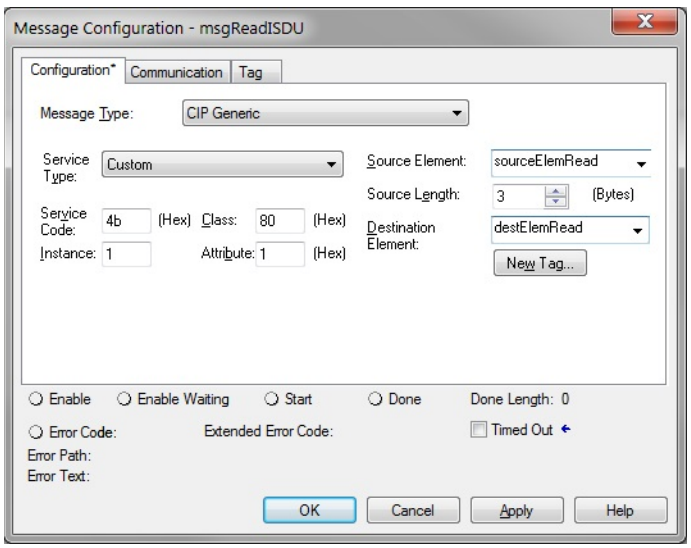


Figure 5-23 Message Instruction configuration

- Then load the program into the controller (1).
- Switch to online mode (2).

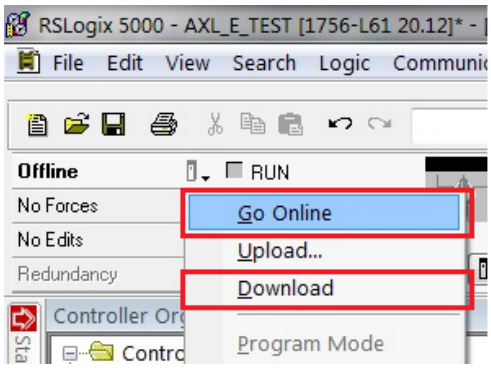


Figure 5-24 Loading the program into the controller and activating online mode

- Now right-click the ToggleBit.
- Select the “Toggle Bit” option. The Message Instruction is executed.

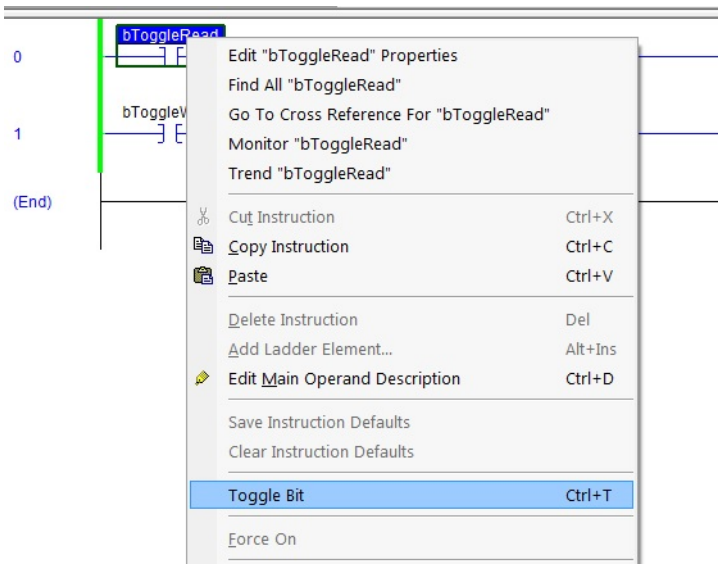


Figure 5-25 Executing a Message Instruction

If the Message Instruction has been executed successfully, there is nothing at the “ER” output.

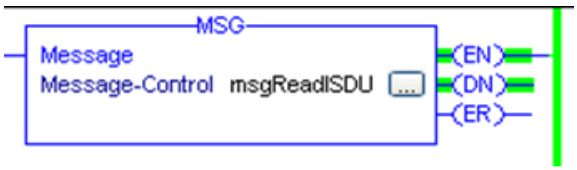


Figure 5-26 Message Instruction successfully executed.

- You can now evaluate the read data in the created “destElemRead” tag.
- To do so, switch to the “Monitor Tags” area by double-clicking on “Controller Tags” in the project tree.

The image below shows the read product ID KQ6005. This corresponds to the connected IO-Link device.

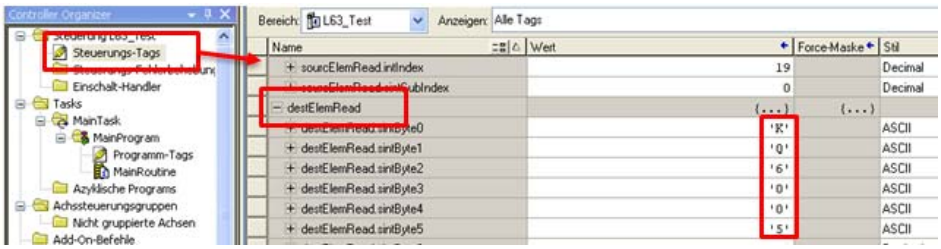


Figure 5-27 Product ID KQ6005

5.7 Writing IO-Link device parameters

This example uses the IO-Link device KQ6005 (capacitive sensor).

– The sensor is connected to port 1 of the IO-Link master.
– “Exclusive Owner with IO-Link I/O” is set as the connection type.

The ISDU object “User-specific marking” (24_{dec})“ should be written.

To do so, perform the following steps:

- Switch to the configuration assembly of the IO-Link master in use.

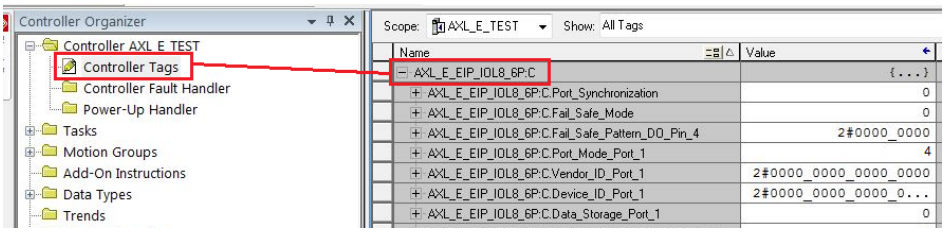


Figure 5-28 Switching to the configuration assembly

- Reconfigure the port in use, to which the actual device is connected, to “IO-Link” mode, this corresponds to value 4 in parameter “…:Port_Mode_Port_X”.

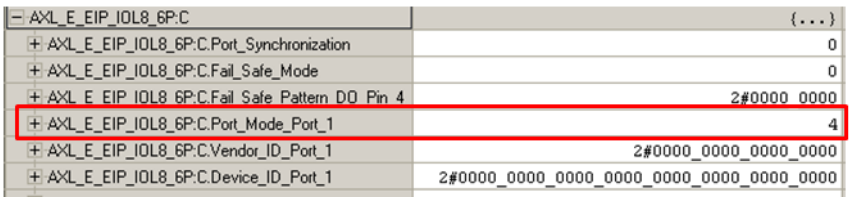


Figure 5-29 Reconfiguring the port

In this example, the tags “sourcElemWrite” (with “ISDU_WRITE” data type) and “destElemWrite” (with “ISDU_DATA” data type) are used.

The index and sub index of the object to be written and the data to be written are specified in the “sourcElemWrite” tag. In this case, the value 24 for the index and 0 for the sub index.

The first ten bytes, which can be seen on the following image, should be written (in ASCII: TEST_WRITE).

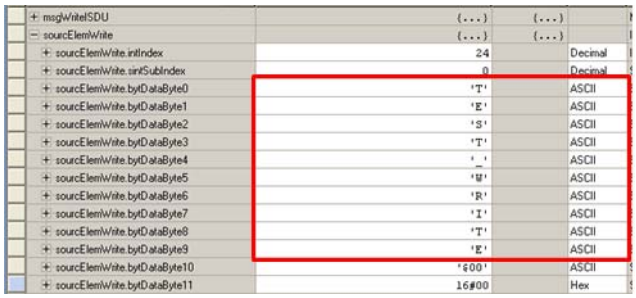


Figure 5-30 The first ten bytes

The “destElemWrite” tag does not need any more parameters, as the data is only stored here in the event of an error.

- Next, switch to the destElemWrite of the program and insert a Message Instruction and an additional element, e.g., “Examine if closed” from the “Bit” element group in order to be able to toggle the Message Instruction.

In this example, the name “msgWriteISDU” is used for the Message Instruction and “bToggleWrite” is used for the ToggleBit.

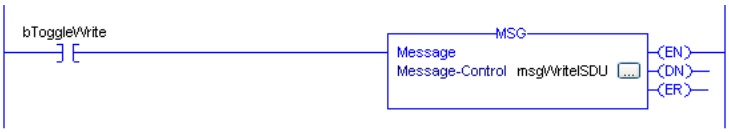


Figure 5-31 Message Instruction and ToggleBit example

The configuration of the Message Instruction looks like this:

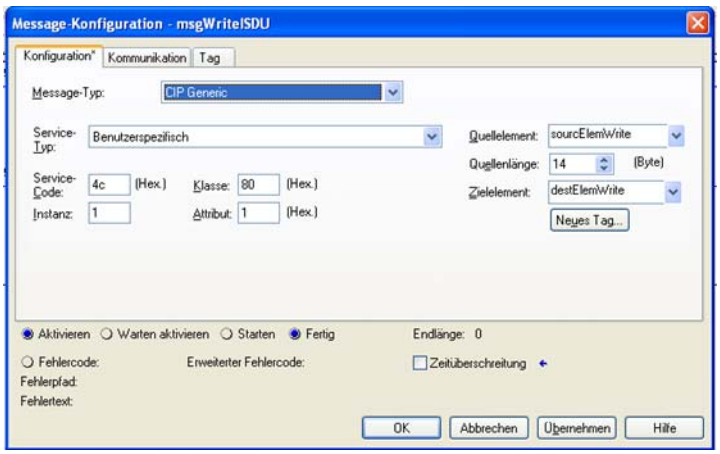


Figure 5-32 Message Instruction configuration

- Then load the program into the controller (1).
- Switch to online mode (2).

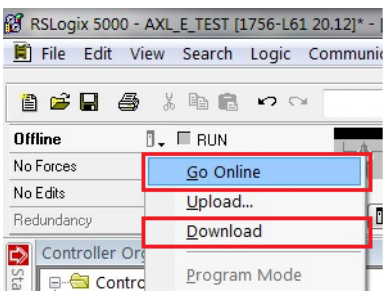


Figure 5-33 Loading the program into the controller and activating online mode

- Now right-click the ToggleBit.
- Select the “Toggle Bit” option. The Message Instruction is executed.

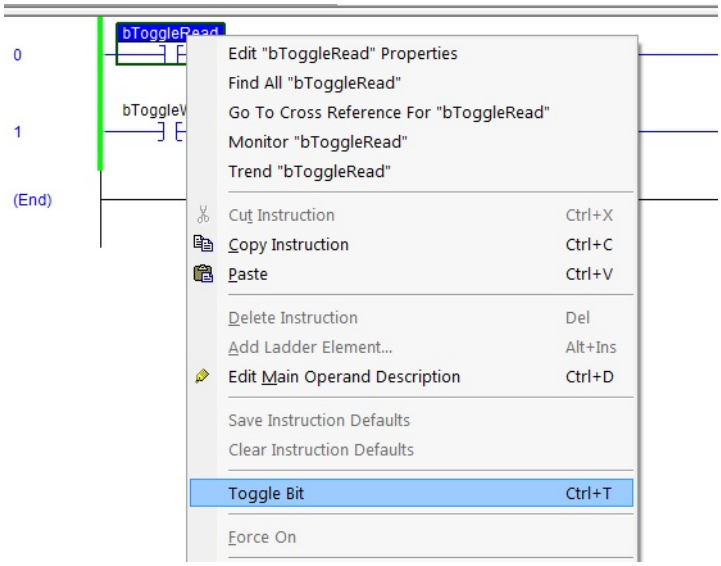


Figure 5-34 Executing a Message Instruction

If the Message Instruction has been executed successfully, there is nothing at the “ER” output.

Figure 5-35 Message Instruction successfully executed.

- Finally, you can check whether the data has actually been written to the device via another read access.
- To do so, you should follow the steps in the example described above.

On the following image you can see that the “destElemRead” tag contains exactly the written data.

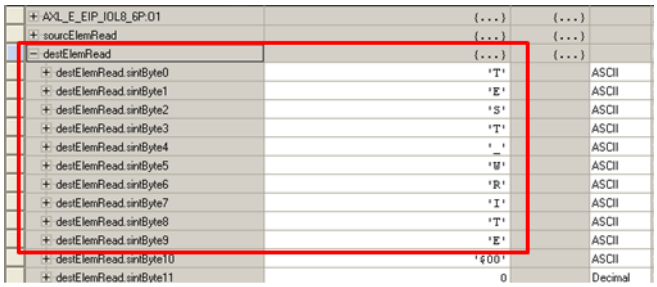


Figure 5-36 “destElemRead” tag

SCATTERGOOD & JOHNSON LTD
ELECTRICAL ENGINEERING & FLUID CONTROL DISTRIBUTORS
Est.1899

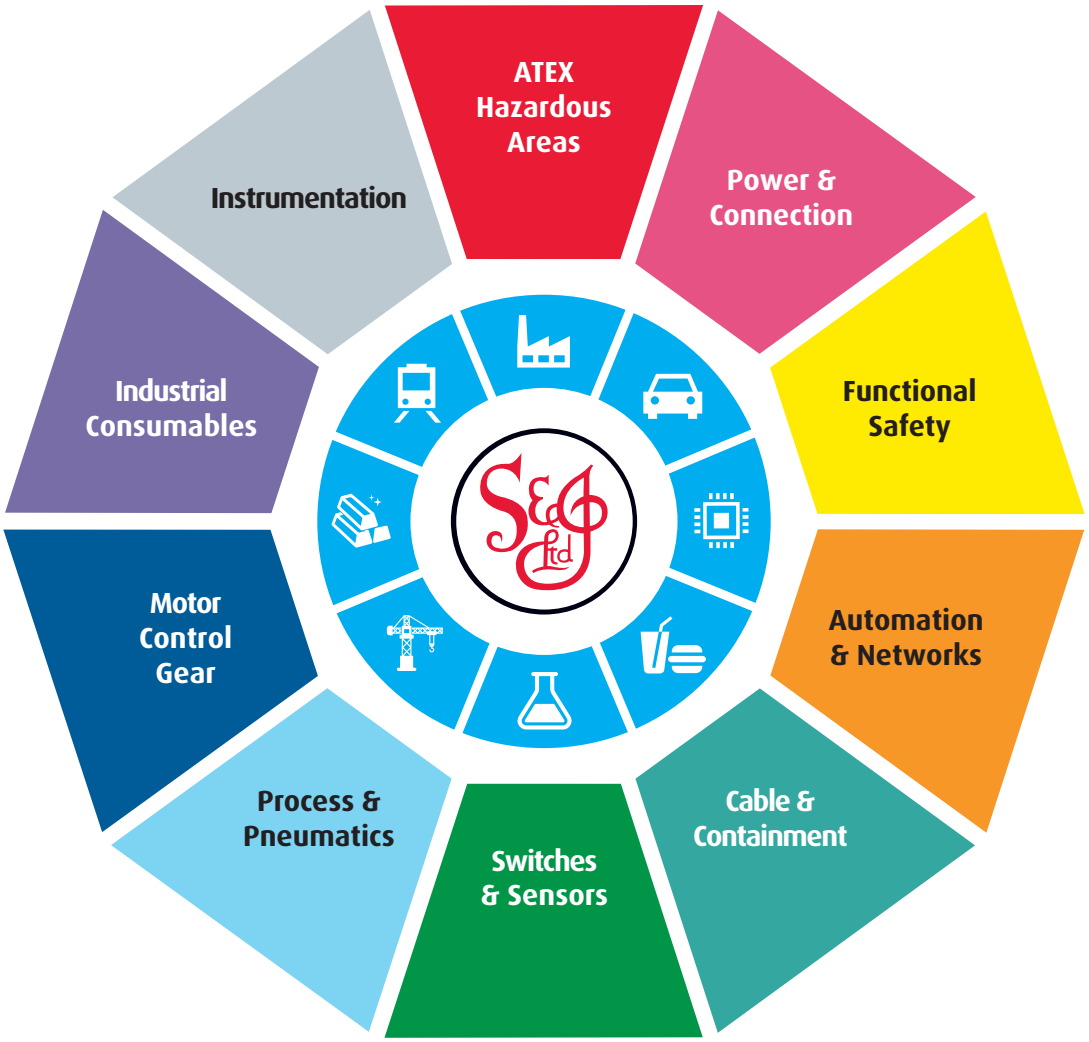
At Scattergood & Johnson Ltd, we pride ourselves on being a technical distributor to specialist industries.

Working with a range of quality product suppliers across a number of specialist markets, we are not your average 'box shifter' - we are your technical and supply chain partner.

We fully support every product we sell - for free! Our internal team and external sales engineers can answer any product or application question, no matter the complexity.

Backing up this technical ability is a range of 50,000+ products available from stock for nationwide next day delivery (same day if required!), or you can collect what you need from any of our trade counters around the UK.

Select your specialist interest below to learn more about how we can help.



Online, In Branch and On the Road - Scattergood & Johnson Ltd, there when you need us.

www.scatts.co.uk