

Your Global Automation Partner

TURCK

IO-Link AOI Getting Started Guide

— This page intentionally left blank —

1	General Information	4
1.1	About these instructions	4
1.2	Explanation of symbols used	4
1.2.1	Warnings	4
1.3	Contents	5
1.4	Feedback about these instructions	5
1.5	Technical support	5
2	Opening the AOI .L5K File	6
2.1	Adding the AOI to the program	8
3	TBEN-Lx-8IOL	10
3.1	Setting up the AOI	10
3.2	Instantiating the data types	12
3.3	Configuring the message instructions	15
3.4	Explanation of Data Types	19
3.4.1	TBEN_Lx_8IOL_AOI	19
3.4.2	Common_Data	19
3.4.3	IOL_Read_MSG and IOL_Write_MSG	19
3.4.4	TBEN_(I,O,C)_DATA	19
3.4.5	IOLx_DI	19
3.4.6	DXPx_IN	19
4	TBIL-M1-16DXP	21
4.1	Setting up the AOI	21
5	Turck_IOL_Generic_AOI	27
5.1	Generic AOI Data Types	27
5.1.1	Common_Data	27
5.1.2	Input_Data_Wordx and Output_Data_Wordx	27
6	Universal Device Info	28
6.1	Universal Device Info Data Types	28
6.1.1	Common_Data	28
6.1.2	Accessing the Device Information	28

1 General Information

1.1 About these instructions

The following user manual describes the setup, functions, and use of the system. It helps you to plan, design, and implement the system for its intended purpose.

Note*: Please read this manual carefully before using the system. This will prevent the risk of personal injury or damage to property or equipment. Keep this manual safe during the service life of the system. If the system is passed on, be sure to transfer this manual to the new owner as well.

1.2 Explanation of symbols used

1.2.1 Warnings

Action-related warnings are placed next to potentially dangerous work steps and are marked by graphic symbols. Each warning is initiated by a warning sign and a signal word that expresses the gravity of the danger. The warnings absolutely have to be observed:



DANGER!

DANGER indicates an immediately dangerous situation, with high risk, the death or severe injury, if not avoided.



WARNING!

WARNING indicates a potentially dangerous situation with medium risk, the death or severe injury, if not avoided.



ATTENTION!

ATTENTION indicates a situation that may lead to property damage, if it is not avoided.



NOTE

In NOTES you find tips, recommendations and important information. The notes facilitate work, provide more information on specific actions and help to avoid overtime by not following the correct procedure.

➤ **CALL TO ACTION**

This symbol identifies steps that the user has to perform.

➔ **RESULTS OF ACTION**

This symbol identifies relevant results of steps

1.3 Contents

- Adding the AOI to a Studio5000 project
- Initializing the AOI
- Explanation of AOI and data types

1.4 Feedback about these instructions

We make every effort to ensure that these instructions are as informative and as clear as possible. If you have any suggestions for improving the design or if some information is missing in the document, please send your suggestions to USTechnical.Documentation@turck.com

1.5 Technical support

For additional support, email inquiries to appsupport@turck.com, or call Application Support at 763-553-7300, Monday-Friday 8AM-5PM CST.

2 Opening the AOI .L5K File

1. Drag and drop TBEN-Lx-8IOL from Catalog File and assign IP Address. For more details on Turck Catalog Files, refer to section 8.10 of the TBEN-Lx-8IOL manual:
<https://www.turck.us/attachment/d301407.pdf>
2. When opening the .L5K file, select "From Import"

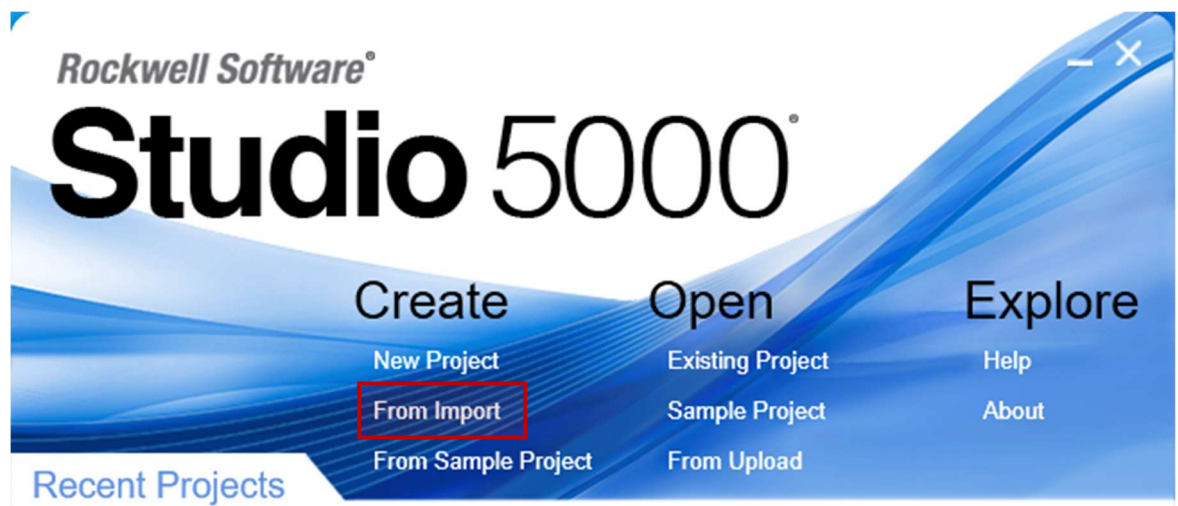


Figure 2.1 – Selecting "From Import"

3. Open the .L5K and save it as a new .ACD project.
4. Once the .L5K has been opened, the available AOI's can be seen

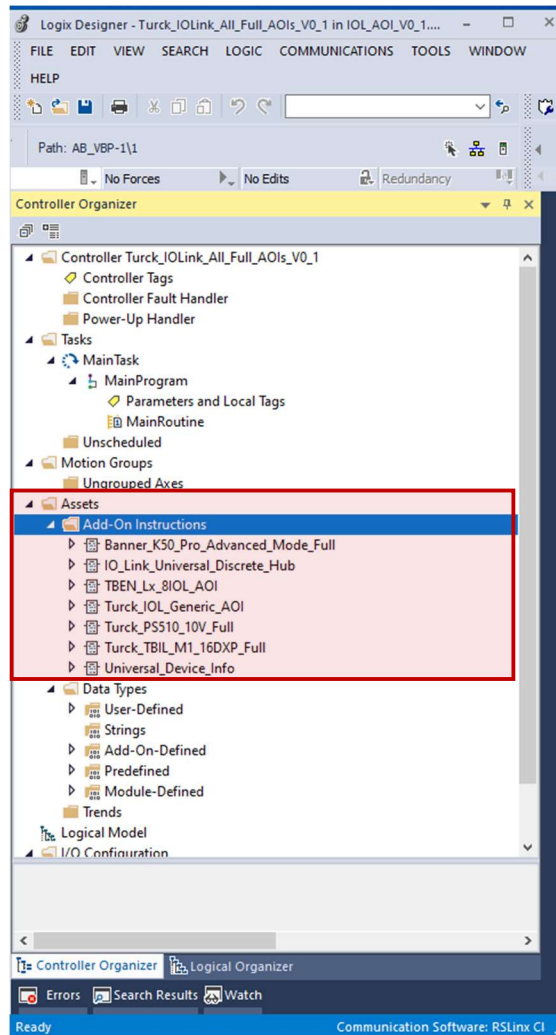


Figure 2.2 – The opened .L5K file

2.1 Adding the AOI to the program

1. The AOI's can now be dragged into the working project under the AOI section

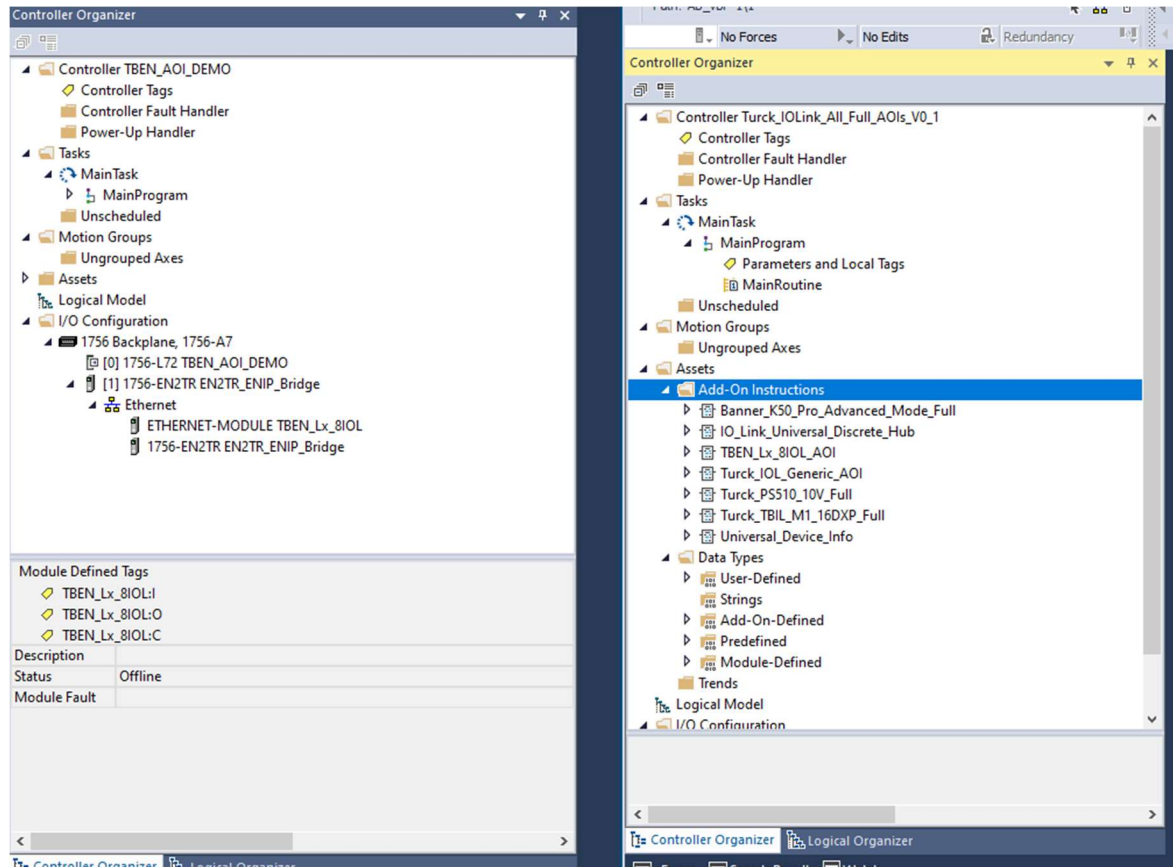


Figure 2.3 – Standard project open next to AOI File

2. Added AOI's should resemble the asset section shown in figure 2.4

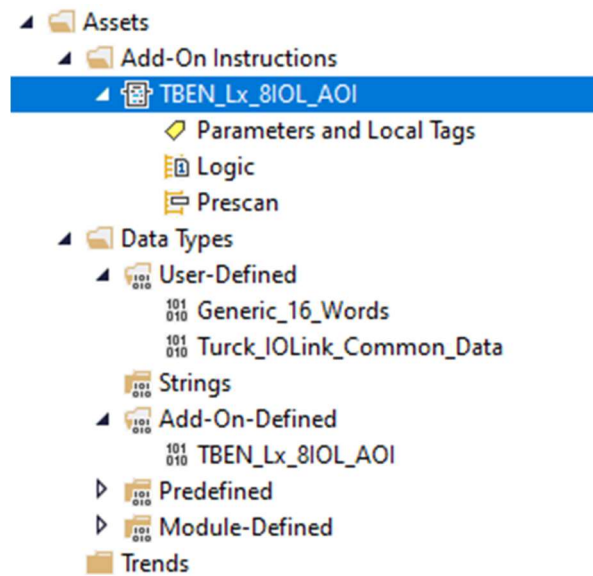


Figure 2.4 – Asset section of a project containing the TBEN-Lx-8IOL AOI

3 TBEN-Lx-8IOL

3.1 Setting up the AOI

1. Drag the AOI into the desired rung. The rung may be empty or have other items in it. For the purposes of this manual, each rung will have only AOI in question. The TBEN-Lx-8IOL AOI can be seen in figure 3.1

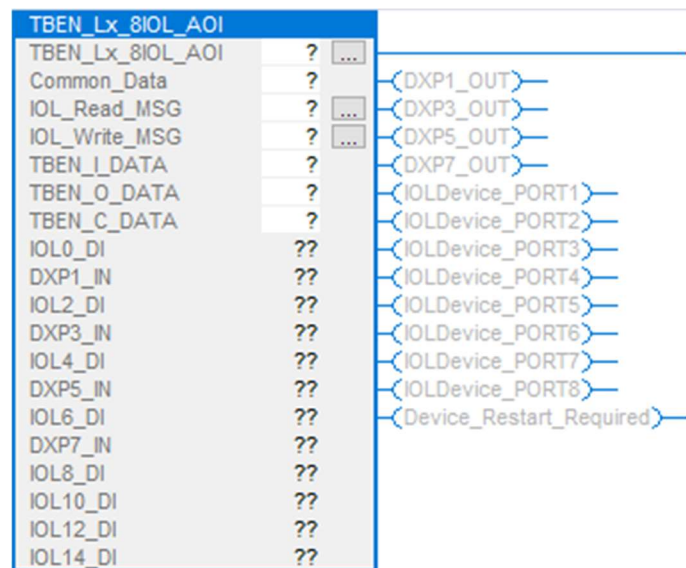


Figure 3.1 – TBEN-Lx-8IOL AOI

- To begin configuring the AOI, all of the Data types must be instantiated. To do this, enter a desired name into the blank fields containing a "?" Figure 3.2 shows an example.

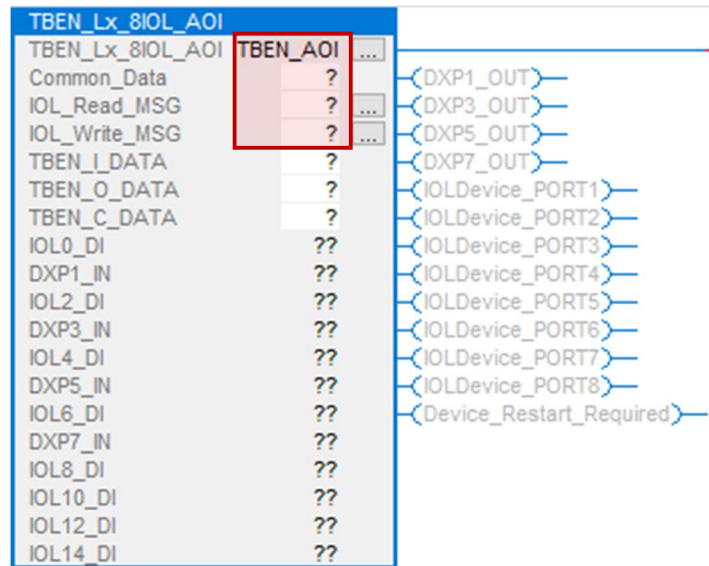


Figure 3.2 – Creating an instance of the AOI titled "TBEN_AOI"

- Repeat this step for Common_Data, IOL_Read_MSG, and IOL_Write_MSG.
- The input, output, and configuration data must all be pointed to the correct places. To do this, double click on the "?" next to the data type and use the drop-down arrow to find the corresponding data.

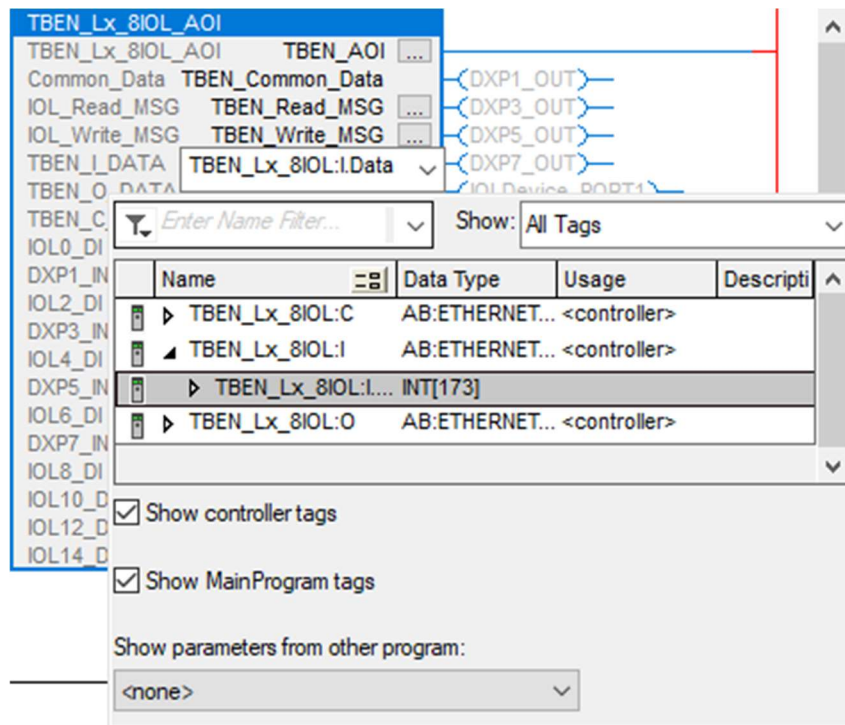


Figure 3.3 – Selecting Input, output and Configuration data

3.2 Instantiating the data types

1. After every field has been filled out, the new data types will need to be instantiated correctly. To do this, right-click on one of the new data types. In this case "TBEN_AOI" which will produce the window shown in figure 3.4. Select "New "TBEN_AOI"".

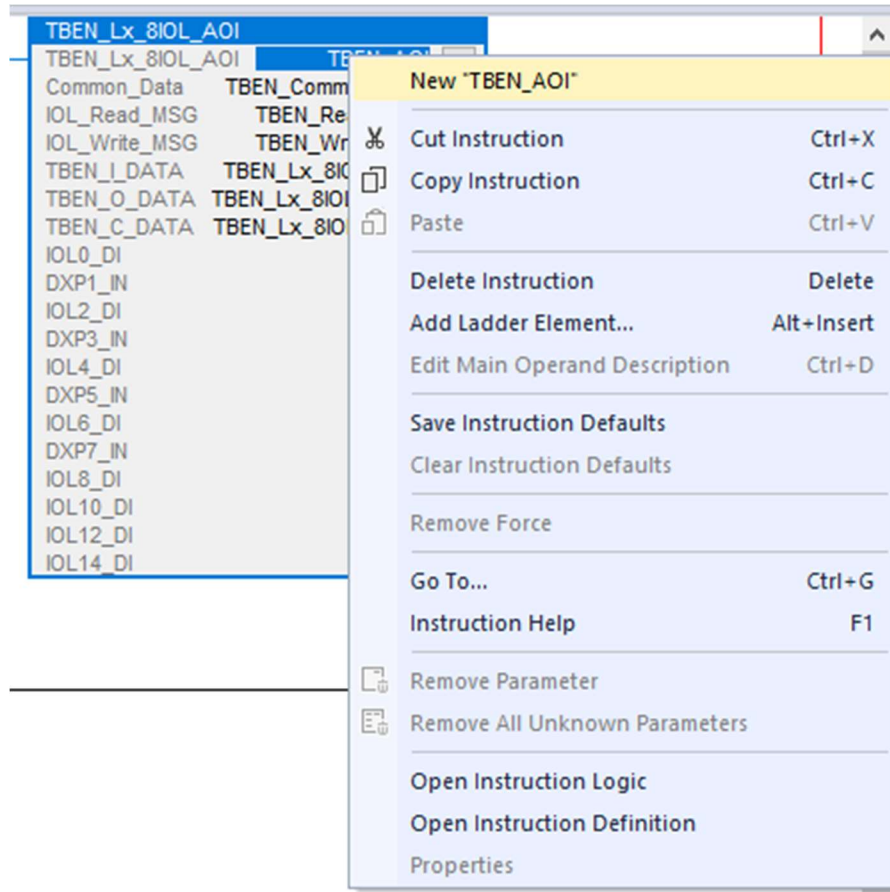


Figure 3.4 – Create an instance of the AOI

2. In the window that follows, the scope of the tag will need to be set to the program name itself. This ensures all the various features will have access to each other feature. For this example, "TBEN_AOI_DEMO" was chosen.

Figure 3.5 – Defining the new tag



NOTE

If the "Main program" scope is selected, this AOI will not function correctly as "common data" will not be accessible to other AOI's.

3. Step 3.2.2 will need to be done for “Common Data”, “Read_MSG” and “Write_MSG”

The figure displays three screenshots of the 'New Tag' dialog box, each showing the configuration for a different tag. The dialog box has a 'Name' field, a 'Description' text area, and several dropdown menus and checkboxes.

Top Left Screenshot (TBEN_Common_Data):

- Name: TBEN_Common_Data
- Description: (empty)
- Usage: <controller>
- Type: Base
- Alias For: (empty)
- Data Type: Turck_IOLink_Common_Data
- Parameter Connection: (empty)
- Scope: @ TBEN_AOI_DEMO
- External Access: Read/Write
- Style: (empty)
- Constant: ☐
- Sequencing: ☐
- Open Configuration: ☐
- Open Parameter Connections: ☐

Top Right Screenshot (TBEN_Read_MSG):

- Name: TBEN_Read_MSG
- Description: (empty)
- Usage: <controller>
- Type: Base
- Alias For: (empty)
- Data Type: MESSAGE
- Parameter Connection: (empty)
- Scope: @ TBEN_AOI_DEMO
- External Access: Read/Write
- Style: (empty)
- Constant: ☐
- Sequencing: ☐
- Open MESSAGE Configuration: ☐
- Open Parameter Connections: ☐

Bottom Screenshot (TBEN_Write_MSG):

- Name: TBEN_Write_MSG
- Description: (empty)
- Usage: <controller>
- Type: Base
- Alias For: (empty)
- Data Type: MESSAGE
- Parameter Connection: (empty)
- Scope: @ TBEN_AOI_DEMO
- External Access: Read/Write
- Style: (empty)
- Constant: ☐
- Sequencing: ☐
- Open MESSAGE Configuration: ☐
- Open Parameter Connections: ☐

Figure 3.6 – Setting Common, Read, Write parameters

3.3 Configuring the message instructions

1. The read and write message instructions must be configured beyond instantiating the data type. To do this, select the (...) button, which will present the "Message Configuration" screen

Message Configuration - TBEN_Read_MSG

Configuration* Communication Tag

Message Type: CIP Generic

Service Type: Custom

Source Element:

Source Length: 0 (Bytes)

Service Code: 0 (Hex) Class: 0 (Hex)

Destination Element:

Instance: 0 Attribute: 0 (Hex)

New Tag...

☐ Enable
 ☐ Enable Waiting
 ☐ Start
 ☒ Done
 Done Length: 224

☐ Error Code: Extended Error Code: ☐ Timed Out

Error Path: TBEN_Lx_8IOL

Error Text:

OK Cancel Apply Help

Figure 3.7 – Message configuration screen for TBEN_Read_MSG

- For the read message, Service code will need to be set to “4B” and class will need to be set to “67”. Instance and Attribute can remain “0”. Source Element will need to be set to the common data sub-type “SendData”. In the case of this example, that data pointer is “TBEN_Common_Data.SendData”. Source length should be set to “3” and Destination Element should be set to the common data sub-type “ReceiveData”, or in the case of this example, “TBEN_Common_Data.ReceiveData”

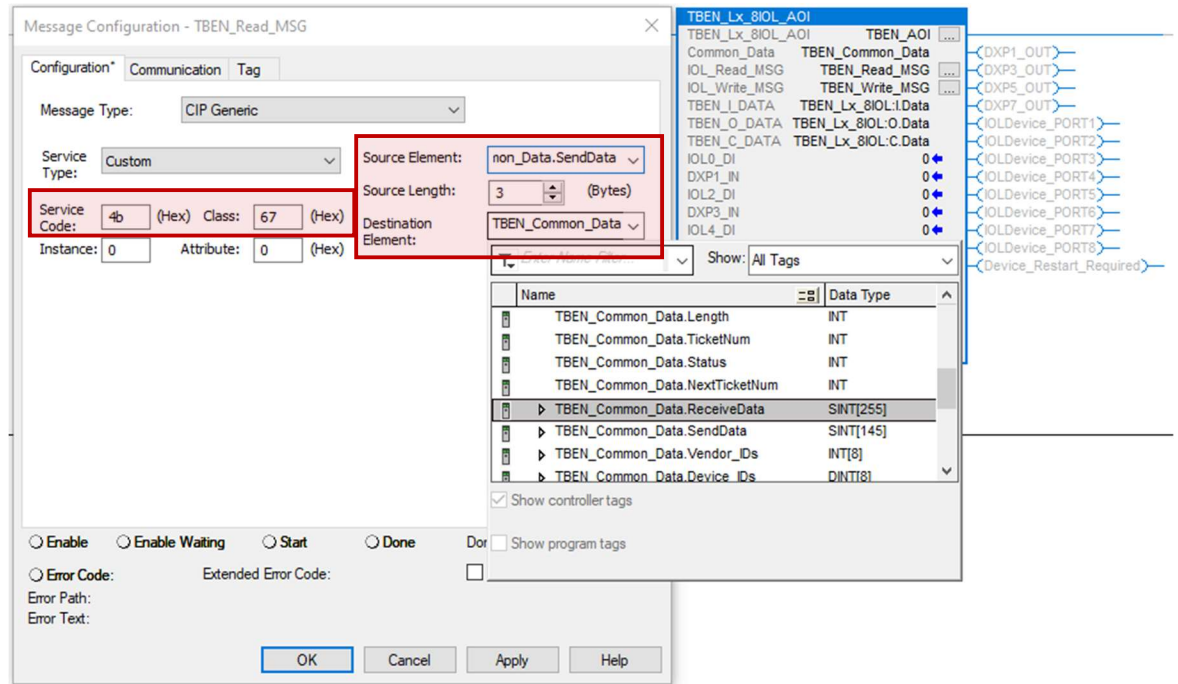


Figure 3.8 – Configuring the read message instruction

- Once the Configuration tab has been completed, the communication tab will need to be handled as well. By selecting "Browse" next to the path, the TBEN-8IOL in the network may be pointed to.

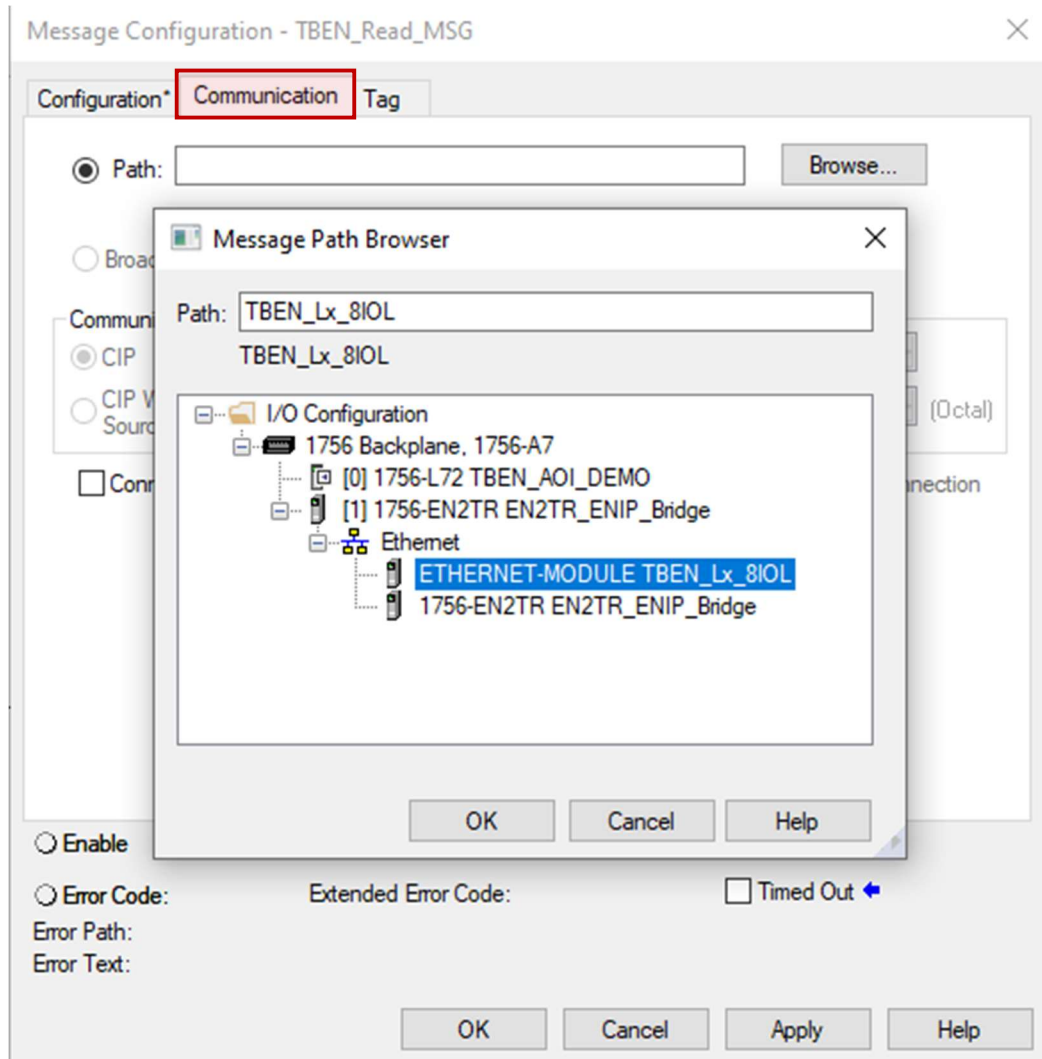


Figure 3.9 – Selecting the Read Message path

- Step 3.3.2 will need to be repeated for Write_MSG as well. In the case of the Write Message, Source Code will be "4C", Class will be 67. The Source Length will be 3. Source element will be TBEN_Common_Data.SendData. Destination Element will again be TBEN_Common_Data.ReceiveData

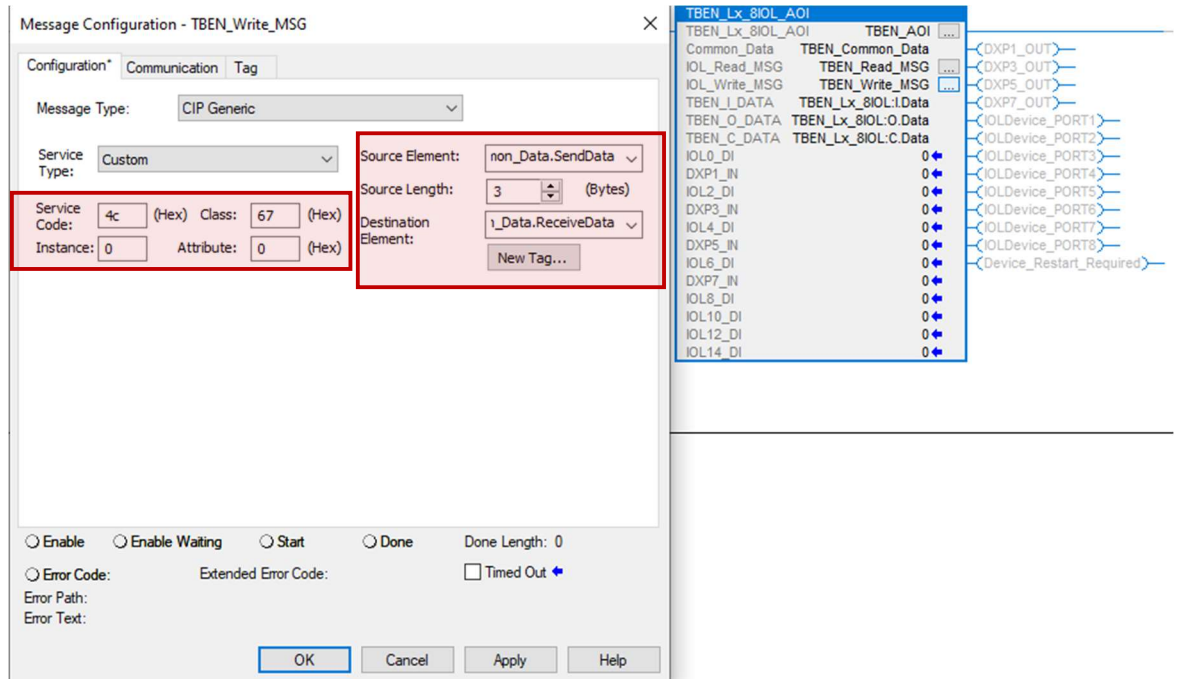


Figure 3.10 – Configuring the write message instruction

- With these steps completed, the next step is to download to the PLC and set to run. This should produce no errors

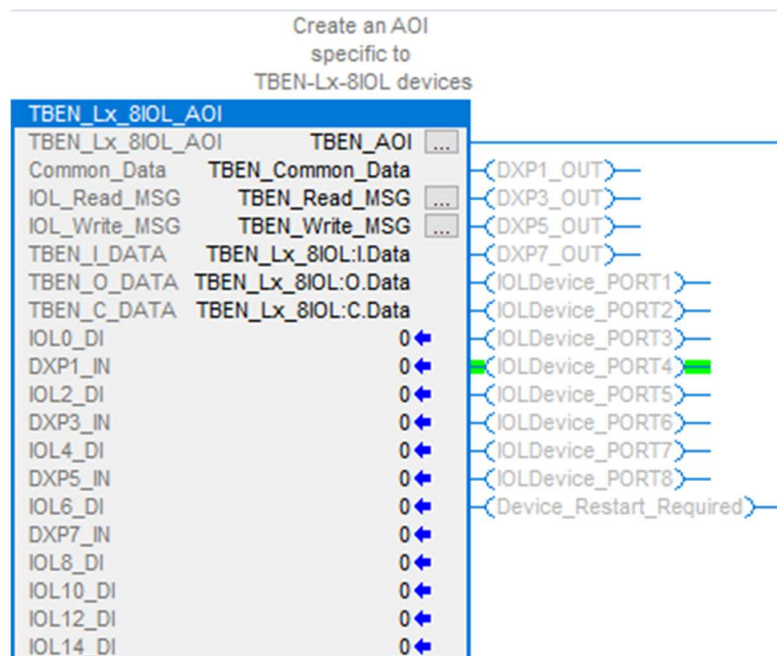


Figure 3.11 – Active connection to a TBEN-Lx-8IOL with a sensor connected to port 4

3.4 Explanation of Data Types

3.4.1 TBEN_Lx_8IOL_AOI

- This AOI is designed to allow other IO-Link AOIs to integrate with the Turck TBEN_Lx_8IOL master



NOTE

Each new TBEN_Lx_8IOL master will require it's own AOI.

3.4.2 Common_Data

- Common_Data is used to pass data between different AOIs and the TBEN_Lx_8IOL master
- The example in 4.1 shows how to share Common_Data between AOIs
- Common_Data should never be modified by the user as it is designed to pass data between AOIs

3.4.3 IOL_Read_MSG and IOL_Write_MSG

- IOL_Read_MSG and IOL_Write_MSG are the data types that will function as the read and write message instructions being sent by the IO-Link master to the relevant IO-Link slave
- These message instructions are what allows the TBEN AOI to send a receive IO-Link calls
- The user does not need to interact with these message instructions directly as they are designed to be shared between different IO-Link AOIs and the TBEN master AOI

3.4.4 TBEN_(I,O,C)_DATA

- TBEN_I_Data is the process input data from the TBEN IO-Link master
- TBEN_O_Data is the process output data from the TBEN IO-Link master
- TBEN_C_Data is the configuration data from the TBEN IO-Link master

3.4.5 IOLx_DI

- Each of the C/Q (IO-Link) pins on a TBEN_Lx_8IOL may be configured as a PNP digital input pin. If the IO-Link pin is configured this way and that pin becomes energized, the corresponding IOLx_DI pin will show that input.

3.4.6 DXPx_IN

- Each class A port of the TBEN_Lx_8IOL contains a DXP pin. When that pin is energized, the corresponding DXPx_IN will show that input.

3.5 Controller Explanation

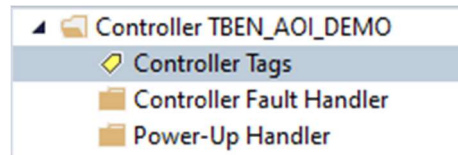


Figure 3.12 – Controller Tags Location

3.5.1 TBEN_Lx_8IOL_AOI

Tag Name	Tag Description
TBEN_AOI.EnableIn	Internal parameter used by the AOI. Should be left at 1
TBEN_AOI.EnableOut	Internal parameter used by the AOI. Should be left at 1
TBEN_AOI.DXPx_OUT	0 – DXP output x is OFF 1 – DXP output x is ON
TBEN_AOI.IOLDevice_PORTx	0 – No device connected in port x 1 – Device connected at port x
TBEN_AOI.IOLx_DI	0 – No input active on DI configured IOL port x 1 – Input active on DI configured IOL port x
TBEN_AOI.DXPx_IN	0 – No input active on DXP pin x 1 – Input is active on DXP pin x
TBEN_AOI.PORTx_Mode_IOL_or_DI	0 – Port x is in IO-Link without validation mode 1 – Port x is in DI mode Note: changing this requires a device restart
TBEN_AOI.DisableOutput_DXPx	0 – Enable output on DXP pin on port x 1 – Disable output on DXP pin on port x
TBEN_AOI.Device_Restart_Required	0 – No restart of device required 1 – Device reconnect is required. Disconnect and reconnect the device
TBEN_AOI.DeviceStatus	Reserved for future use

Table 1 – Controller Tag Explanation

4 TBIL-M1-16DXP

4.1 Setting up the AOI

- See section 2 for a walkthrough of setting up the .L5K
- 1. Create a rung for the AOI. This block will need to be instantiated by creating a name for the block in the top most “?”

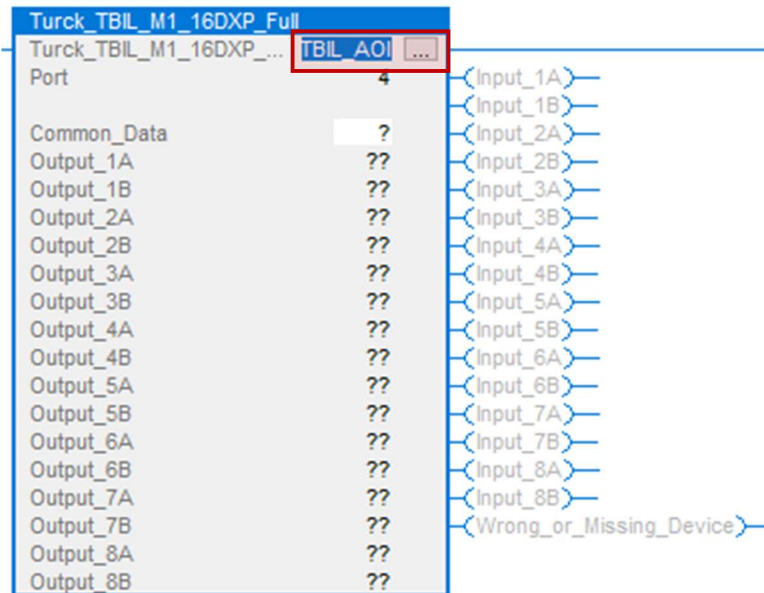


Figure 4.1 – Name the new instance of the TBIL AOI

2. With this name filled out right-click on the new name and select “New ...” in the menu that appears

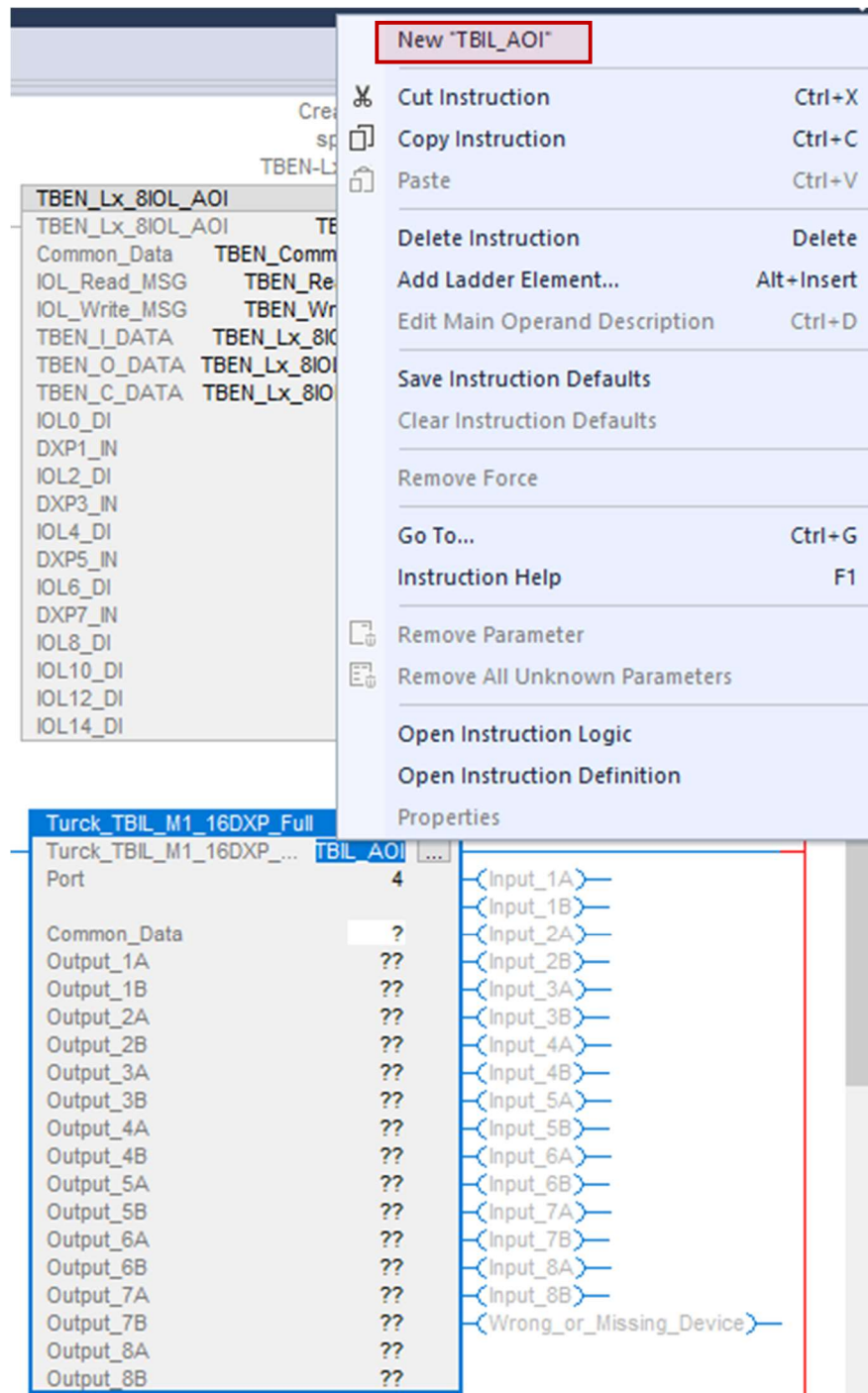


Figure 4.2 – Create a new instance of this TBIL AOI

3. This new tag will need to be set to the scope of the entire project

The 'New Tag' dialog box is shown with the following parameters:

- Name:** TBIL_AOI
- Description:** (Empty text area)
- Usage:** <controller>
- Type:** Base
- Alias For:** (Empty)
- Data Type:** Turck_TBIL_M1_16DXP_Full
- Parameter Connection:** (Empty)
- Scope:** TBEN_AOI_DEMO
- External Access:** Read/Write
- Style:** (Empty)
- ☐ Constant
- ☐ Sequencing
- ☐ Open Configuration
- ☐ Open Parameter Connections

Figure 4.3 – Setting the parameters of the new AOI tag

4. The “Port” selection should be filled out with the port containing the TBIL. In the case of this example, this is port 4.

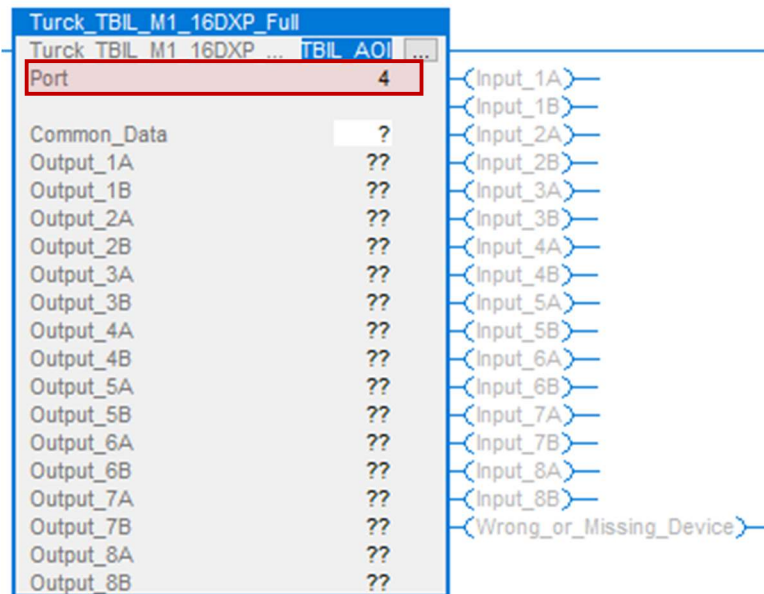


Figure 4.4 – Enter the port number



NOTE

Port count begins at 1. C0 = Port 1, C1 = Port 2... C7 = Port 8. These will follow the labels on the TBEN-Lx-8IOL

5. Add the common data from the TBEN AOI to the common data of the TBIL AOI. The common data will need to be shared between all AOIs accessing the same IOL master. To do this, select the common data "?". Within the dropdown produced, select the common data used in the TBEN AOI

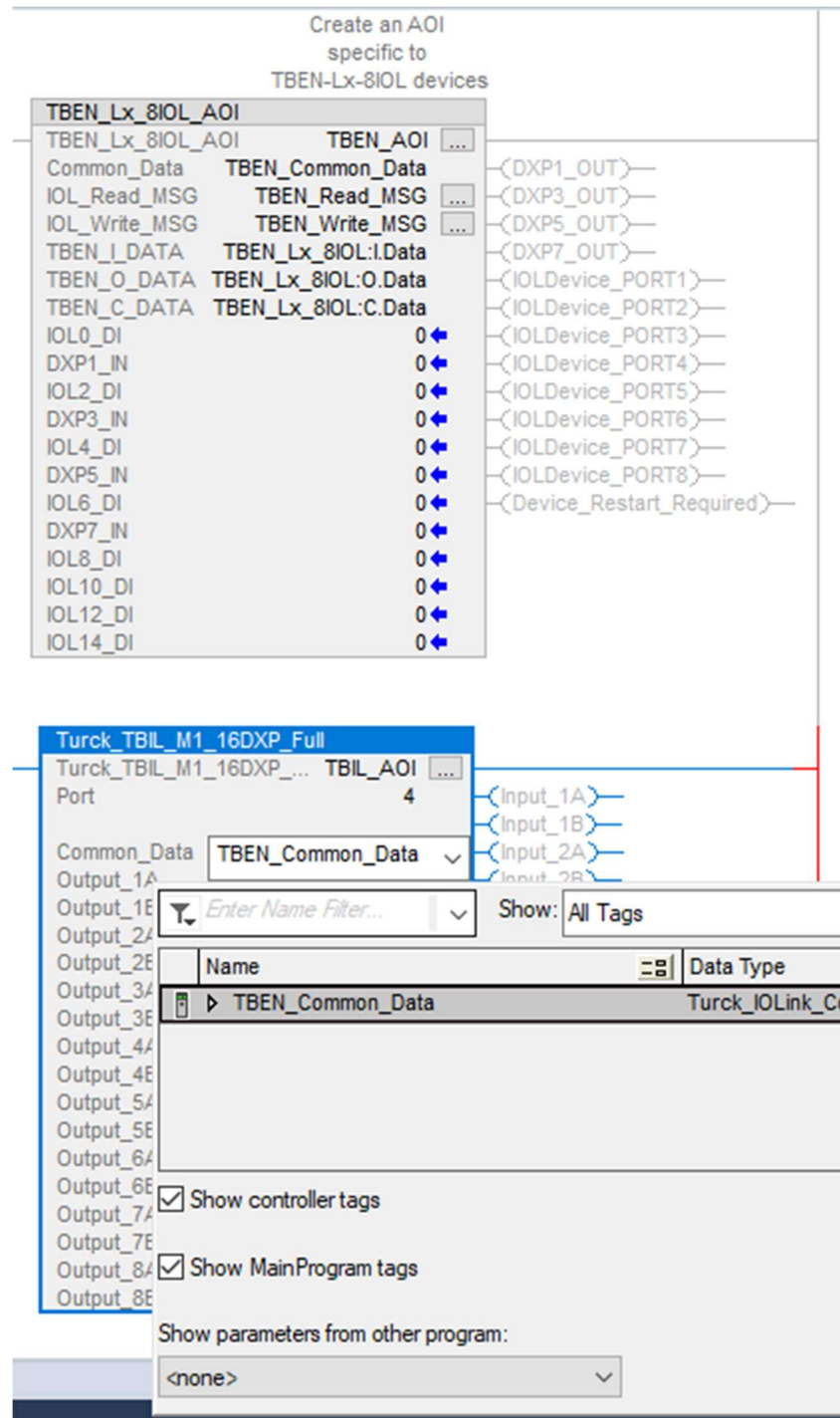


Figure 4.5 – Set the common data

6. With these steps completed, the next step is to download to the PLC and set to run. This should produce no errors

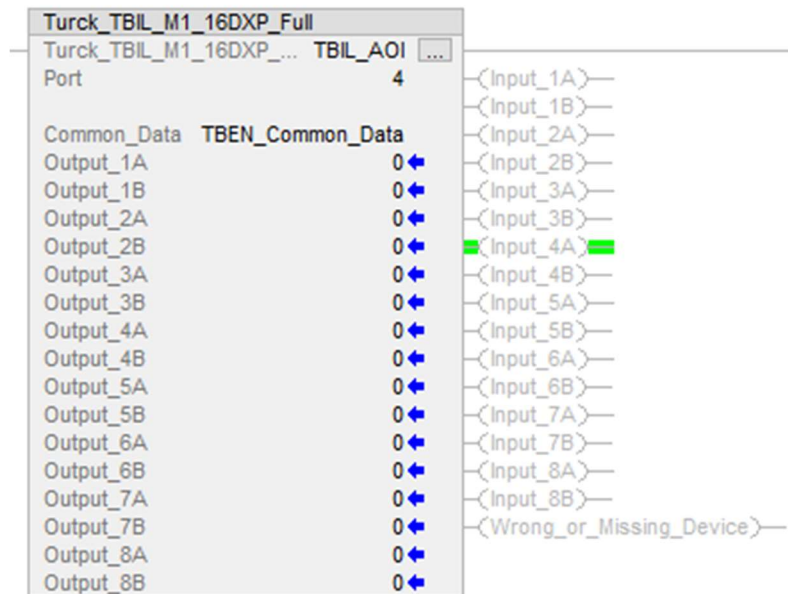


Figure 4.6 – Successfully created TBIL AOI

- Each new AOI will go through the same process. New tags will need to be instantiated, common data will need to be shared between all IOL slaves on the same master. The port will need to be set to the connected port.

5 Turck_IOL_Generic_AOI

5.1 Generic AOI Data Types

- See section 2 for a walkthrough of setting up the .L5K and section 4 for setting up the AOI.
- The generic AOI is designed to make data processing easier for all IO-Link sensors



NOTE

This AOI is generic to all sensors. Not just Turck sensors.

Turck_IOL_Generic_AOI	
Turck_IOL_Generic_AO...	GEN_AOI ...
Port	4
Common_Data TBEN_Common_Data	
Input_Data_Word0	64
Input_Data_Word1	0
Input_Data_Word2	0
Input_Data_Word3	0
Input_Data_Word4	0
Input_Data_Word5	0
Input_Data_Word6	0
Input_Data_Word7	0
Output_Data_Word0	0
Output_Data_Word1	0
Output_Data_Word2	0
Output_Data_Word3	0
Output_Data_Word4	0
Output_Data_Word5	0
Output_Data_Word6	0
Output_Data_Word7	0

Figure 5.1 – Turck_IOL_Generic_AOI

5.1.1 Common_Data

- Common_Data is used to pass data between the generic IO-Link AOI and the TBEN_Lx_8IOL master
- The example in 4.1 shows how to share Common_Data between AOIs
- Common_Data should never be modified by the user as it is designed to pass data between AOIs

5.1.2 Input_Data_Wordx and Output_Data_Wordx

- The input and output data words shown make it easier to map data into new AOIs.
- The data here is parsed such that it becomes readable without parameterizing the TBEN master

6 Universal Device Info

6.1 Universal Device Info Data Types

- See section 2 for a walkthrough of setting up the .L5K and section 4 for setting up the AOI.
- The Universal Device Info AOI is designed to read in device information from a sensor connected to a Turck TBEN IO-Link master.

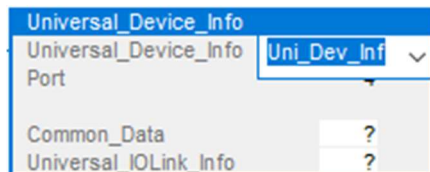


Figure 6.1 – Universal Device Info AOI

6.1.1 Common_Data

- Common_Data is used to pass data between the generic IO-Link AOI and the TBEN_Lx_8IOL master
- The example in 4.1 shows how to share Common_Data between AOIs
- Common_Data should never be modified by the user as it is designed to pass data between AOIs

6.1.2 Accessing the Device Information

- To access the device information, enter the tag definition for “Universal_IOLink_Info”

Port4_Uni_Info	{...}	{...}	Universal_IOLink_Info	
▶ Port4_Uni_Info.Vendor_Name	'TURCK'	{...}	STRING	Index 16
▶ Port4_Uni_Info.Vendor_Text	'www.turck.com'	{...}	STRING	Index 17
▶ Port4_Uni_Info.Product_Name	'TBIL-M1-16DXP'	{...}	STRING	Index 18
▶ Port4_Uni_Info.Product_ID	'6814102'	{...}	STRING	Index 19
▶ Port4_Uni_Info.Product_Text	'I/O-Hub'	{...}	STRING	Index 20
▶ Port4_Uni_Info.Serial_Number	'0000128329-00110'	{...}	STRING	Index 21
▶ Port4_Uni_Info.Hardware_Revision	''	{...}	STRING	Index 22
▶ Port4_Uni_Info.Firmware_Revision	'1.5.2.0'	{...}	STRING	Index 23
▶ Port4_Uni_Info.Application_Specifi...	' '	{...}	STRING	Index 24
▶ Port4_Uni_Info.Function_Tag	''	{...}	STRING	Index 25
▶ Port4_Uni_Info.Location_Tag	''	{...}	STRING	Index 26
▶ Port4_Uni_Info.Application_Specifi...	''	{...}	STRING	

Figure 6.2 – Example of device information acquired by the Universal Device Info AOI

1.

TURCK sells its products through Authorized Distributors. These distributors provide our customers with technical support, service and local stock. TURCK distributors are located nationwide – Including all major metropolitan marketing areas
For Application Assistance or for the location of your nearest TURCK distributor, call:
1-800-544-7769

Specifications in this manual are subject to change without notice. TURCK also reserves the right to make modifications and makes no guarantee of the accuracy of the information contained herein.

Your Global Automation Partner

TURCK



28 subsidiaries and over
60 representations worldwide!

www.turck.com