Your Global Automation Partner

Catalog Files
Startup Guide

0621A

— This page intentionally left blank —

Turck Inc. | 3000 Campus Drive, Minneapolis, MN 55441 | T +1 800 544 7769 | F +1 763 553 0708 | www.turck.com

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.

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 of death or severe injury if not avoided.

WARNING!

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

ATTENTION!

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

NOTE

In NOTES you will find tips, recommendations, and important information. The notes facilitate work, provide more information on specific actions, and help to avoid unnecessary effort caused 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

- Before You Begin
- Obtaining Catalog Files
- Importing Catalog Files into Logix Designer
- Adding I/O Stations from Catalog Project to New/Existing Project
- Setting Up Your I/O Stations
- Examples

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 **techdoc@turck.com**.

1.5 Technical support

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

2 Before You Begin

2.1 What's a Catalog File?

A Catalog File is a blank project in which we've mapped all our I/O stations as Generic Ethernet Modules. All the controller tags are named and commented for you, so it's clear what each tag does.

2.2 Why should I use them instead of EDS files?

Catalog Files serve the same purpose as EDS files, but they are far superior; they contain much more information, and they make initial commissioning and future device replacements much easier.

2.3 How do I use them?

With your existing project pulled up, open a separate instance of Logix Designer, and import the appropriate Catalog File (.L5K) to turn it into a Catalog Project (.ACD). From there, simply drag-and-drop the Turck I/O station(s) from the Catalog Project that you want to incorporate into your new/existing project.

2.4 What else do I have to do after adding the device(s) to my project?

Besides specifying IP addresses, you must also parameterize your I/O station(s) in the "Configuration" controller tags (those end with a C). By default, the settings in those Configuration tags are pushed to each I/O station every time the PLC establishes communication with them, making initial commissioning and device replacement hassle-free.

2.5 Are there any reasons I wouldn't want to use Catalog Files?

It depends on your situation, but there might be two. First, each I/O station mapped using Catalog Files consumes three CIP connections instead of one, which may be an issue in smaller PLCs depending on how many slaves you plan on connecting. Second, as with all Generic Ethernet Modules in Logix Designer, you won't have the “connection status” information offered by EDS files.

Turck Inc. | 3000 Campus Drive, Minneapolis, MN 55441 | T +1 800 544 7769 | F +1 763 553 0708 | www.turck.com

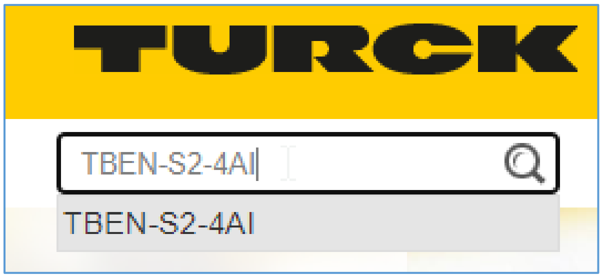
3 Obtaining Catalog Files

3.1 Overview

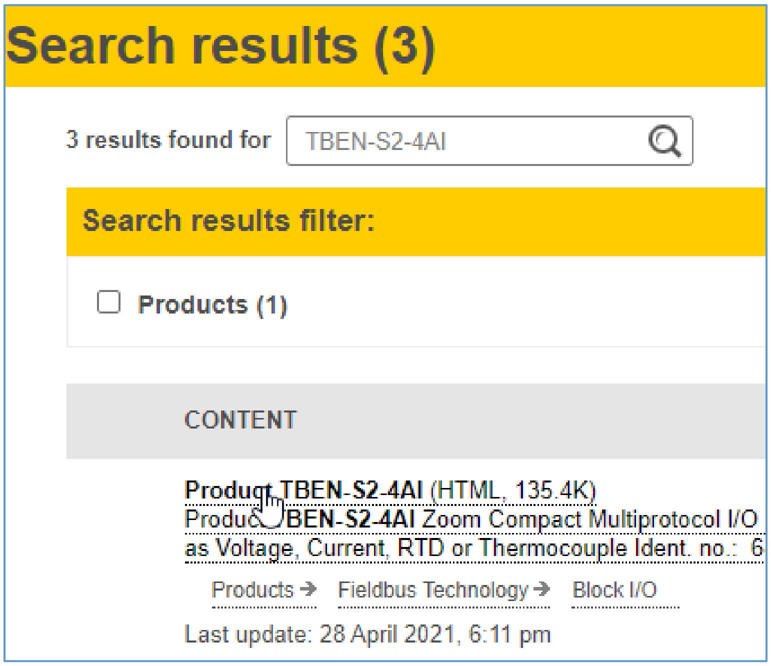
Catalog files are hosted on the Turck company website, and are available free of charge.

3.2 Steps

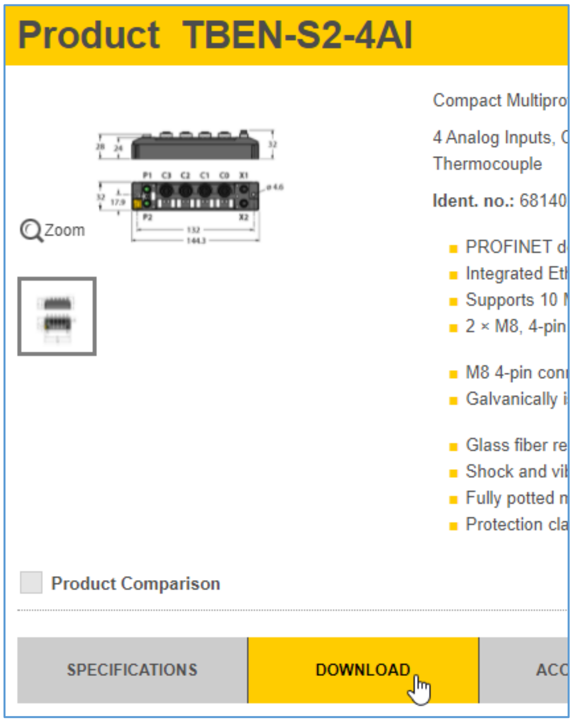
1. Go to turck.us, type in the name of your device into the search bar in the upper-right, and press “Enter.” In this guide, the device will be the TBEN-S2-4AI, a popular analog input block.



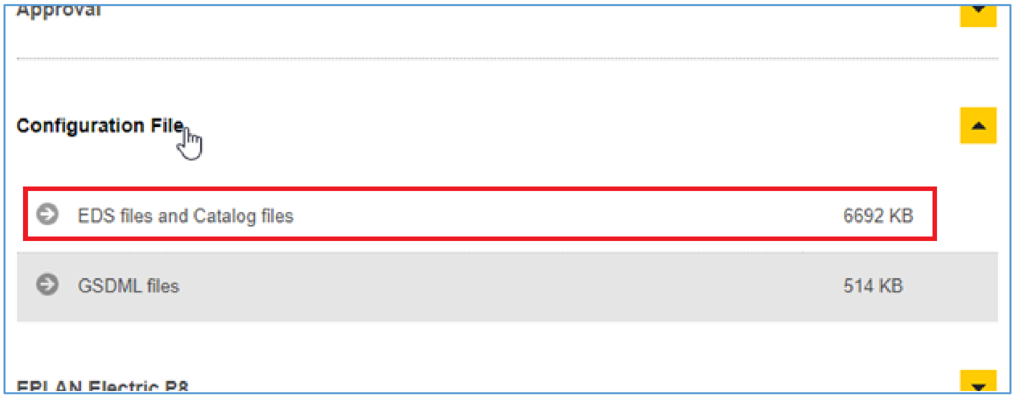
2. Select your device from the list of results.



3. That will take you to the product page of that device. From here, click the “Download” tab in the middle-center of the screen.



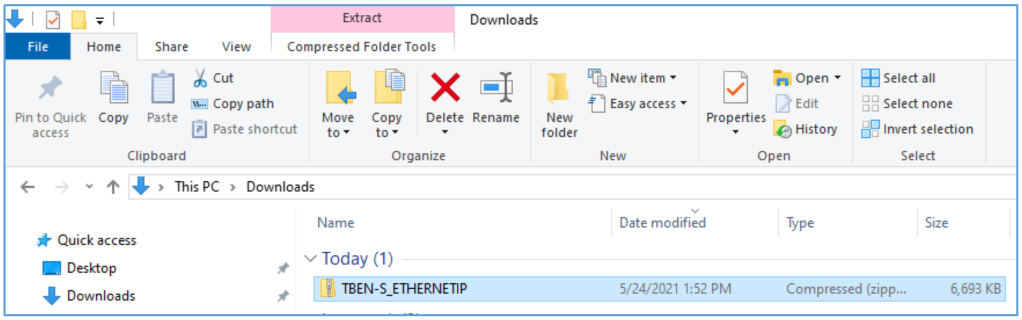
4. Scroll down until you see a “Configuration File” option. Click it, and then click “EDS files and Catalog files” to download a ZIP folder containing both the EDS files and Catalog Files for your device.



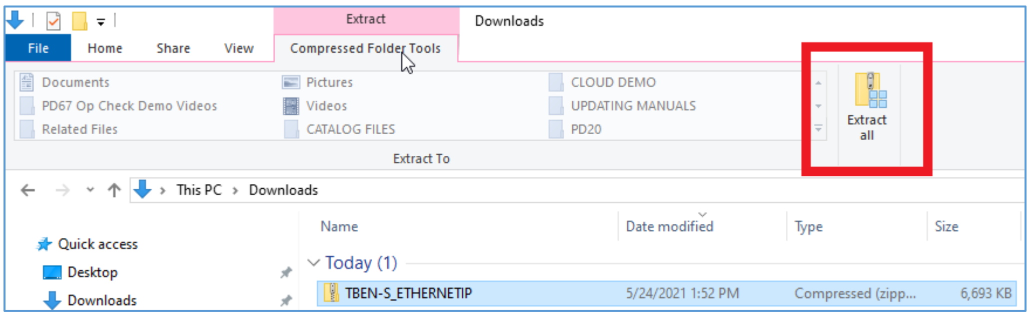
NOTE

Don’t be alarmed if the screenshots in this guide don’t exactly match what you are seeing. Filenames and folder contents may change, but the Catalog File process as a whole will not.

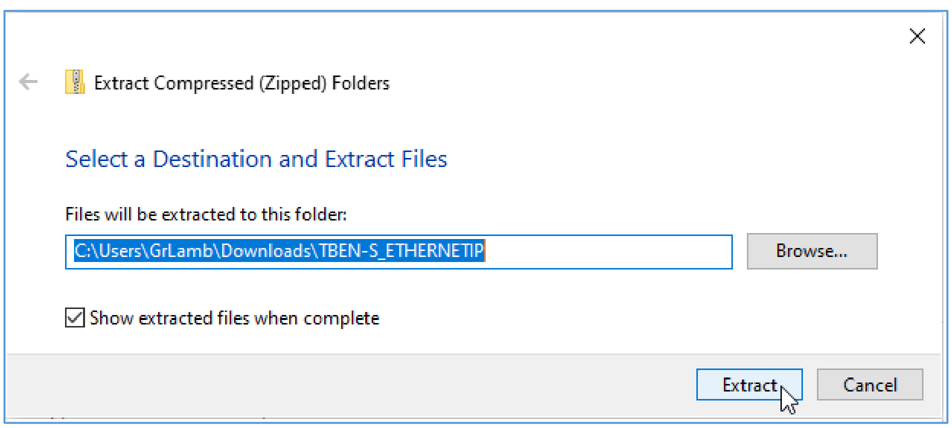
5. In Windows Explorer, open the folder to which the ZIP folder was saved, and select the ZIP folder. The default save location is your Downloads folder.



6. The ZIP folder must be unzipped for it to be useful. Click the "Compressed Folder Tools" tab in Windows Explorer, and select "Extract All."

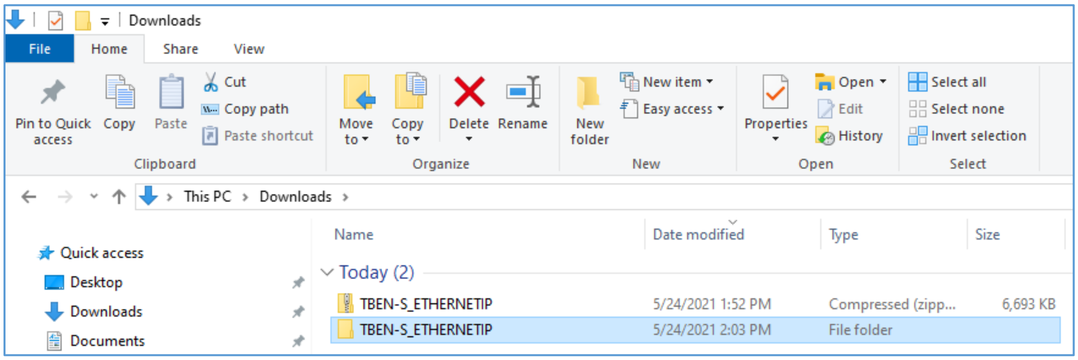


7. Once "Extract All" is clicked, a dialog box will appear asking to what location you would like the files extracted. By default, they will be extracted to a new folder in the same location as the ZIP folder. Specify an extraction location, then click "Extract."



Turck Inc. | 3000 Campus Drive, Minneapolis, MN 55441 | T +1 800 544 7769 | F +1 763 553 0708 | www.turck.com

8. Windows has now created a regular folder bearing the same name as the ZIP folder. The ZIP folder can be deleted if you wish.



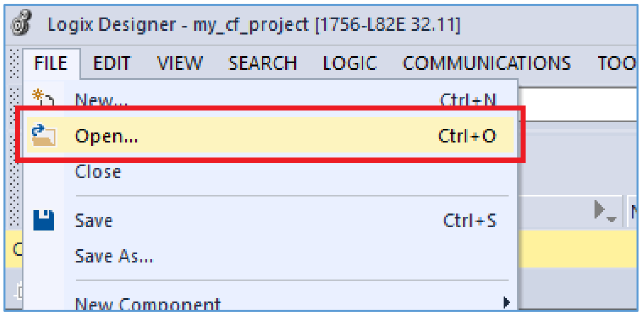
4 Importing Catalog Files into Logix Designer

4.1 Overview

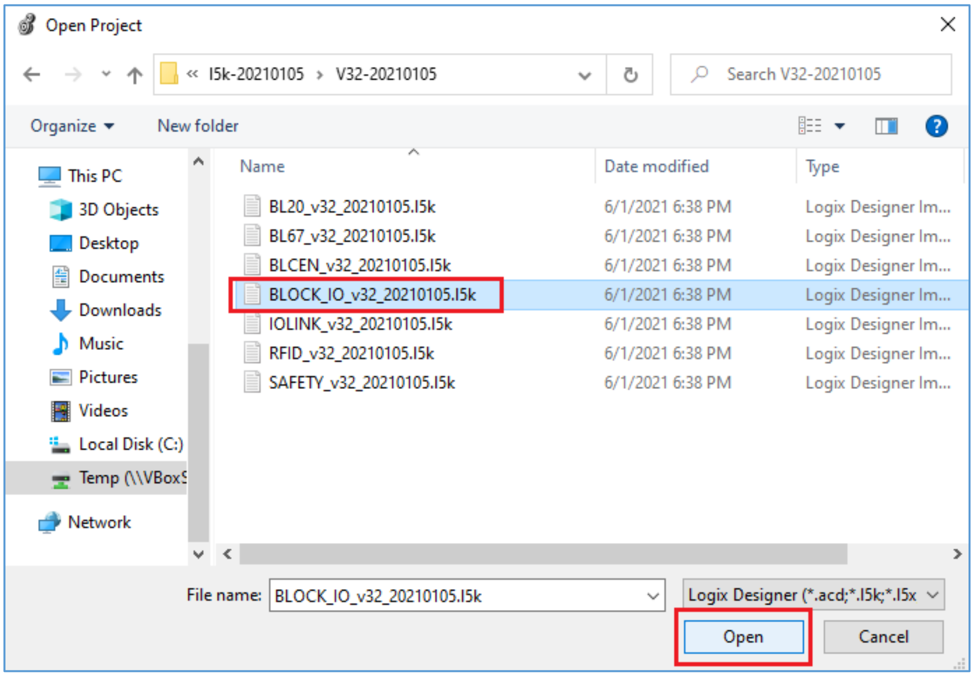
Now that you've downloaded and unzipped the Catalog Files for your I/O station(s), it's time to import the correct .L5K Catalog File for you PLC into Logix Designer, and save it as a .ACD file. In this guide, the Turck device will be the popular analog input block TBEN-S2-4AI, and the PLC will be running v32, though the process is similar for other I/O stations and PLC versions.

4.2 Steps

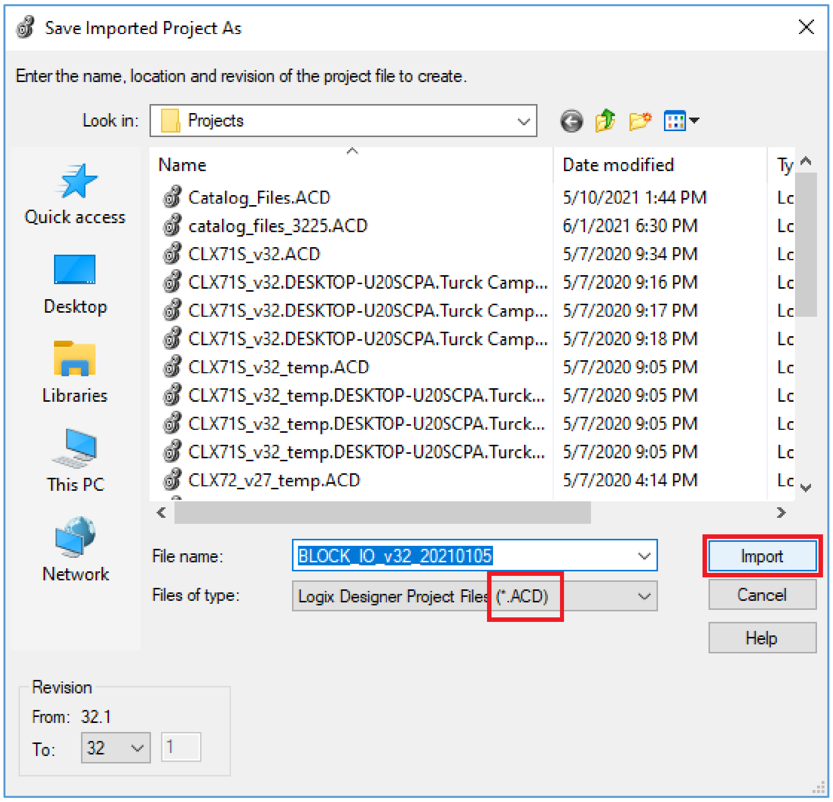
1. Open Logix Designer.
2. Select File → Open.



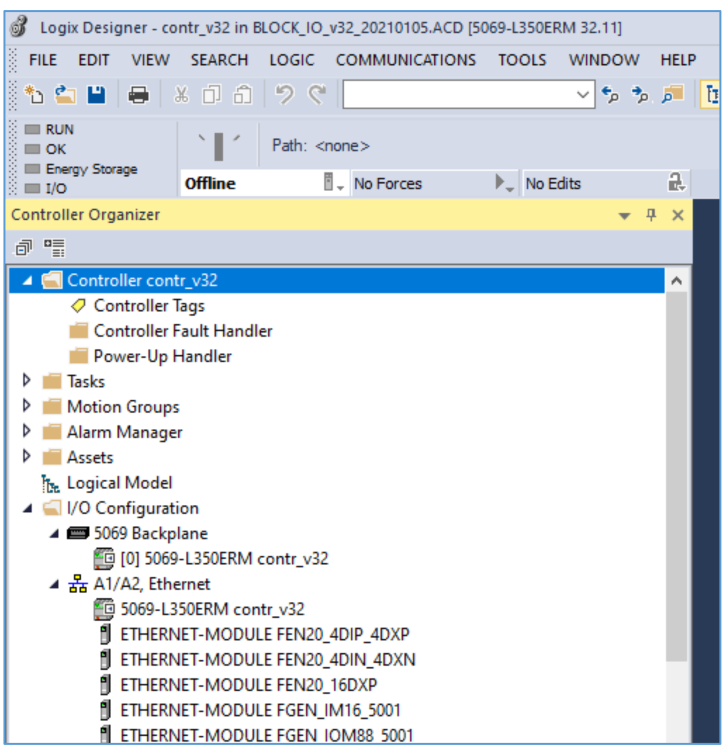
3. Navigate to the **unzipped** catalog file that applies to your I/O station(s) and your PLC, and open that file.



4. A “Save Imported Project As” window will appear. Change the name if you like, then import the .L5K Catalog File as a .ACD file.



5. Logix Designer will take a few seconds to import the Catalog File. Once it’s done, you’ll be looking at a blank project with every Turck device covered by the Catalog File mapped as a Generic Ethernet Module. From here on, this project will be called the “Catalog Project.”



5 Adding I/O Stations from Catalog Project to Your Project

5.1 Overview

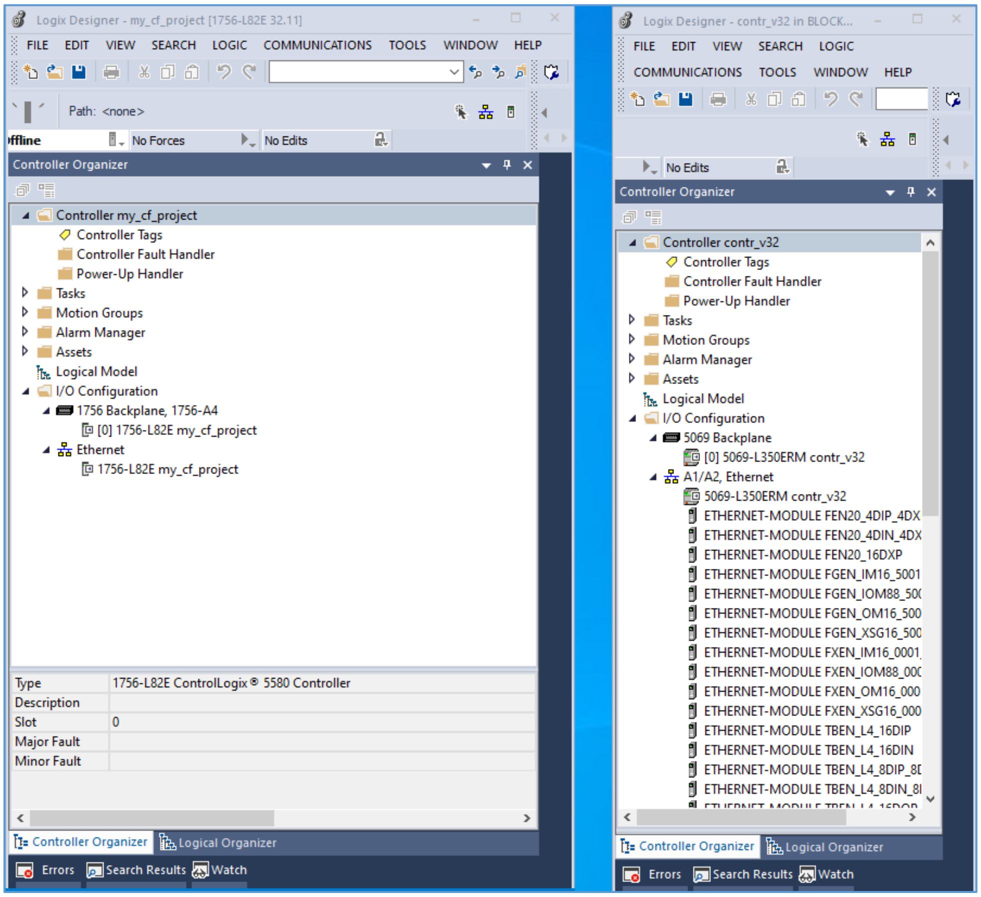
There are two main station configurations; they are Block I/O and Modular I/O. This chapter covers some steps common to both configurations, then it is divided into two sections, as the process for adding block and modular devices to a project are slightly different.

NOTE

If you haven't already imported a Catalog File (.L5K) and saved it as a Catalog Project (.ACD), see Importing Catalog Files into Logix Designer.

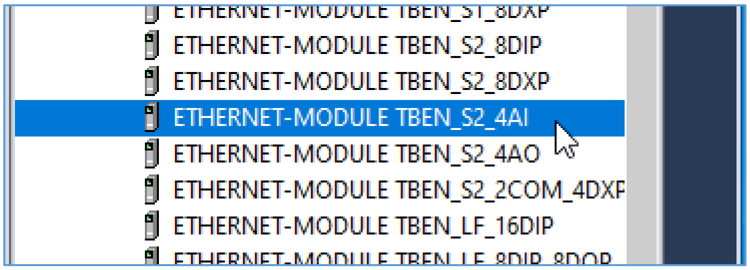
5.2 Steps

1. Open your Catalog Project if it is not already open.
2. In a new instance of Logix Designer, open the new or existing project to which you want to add a Turck I/O station from your Catalog Project. From here on, this new or existing project will be called the "target project."
3. Arrange the two instances of Logix Designer so that you can see both projects' "I/O Configuration" folders.

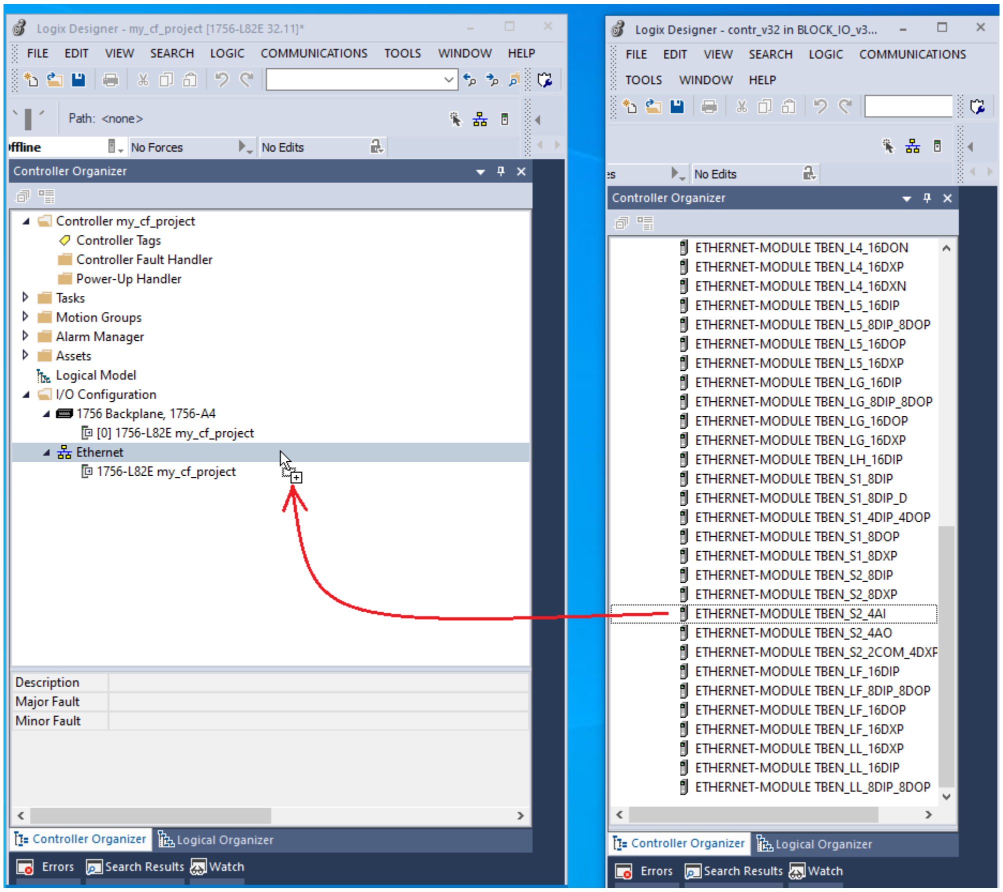


5.2.1 Block I/O Stations

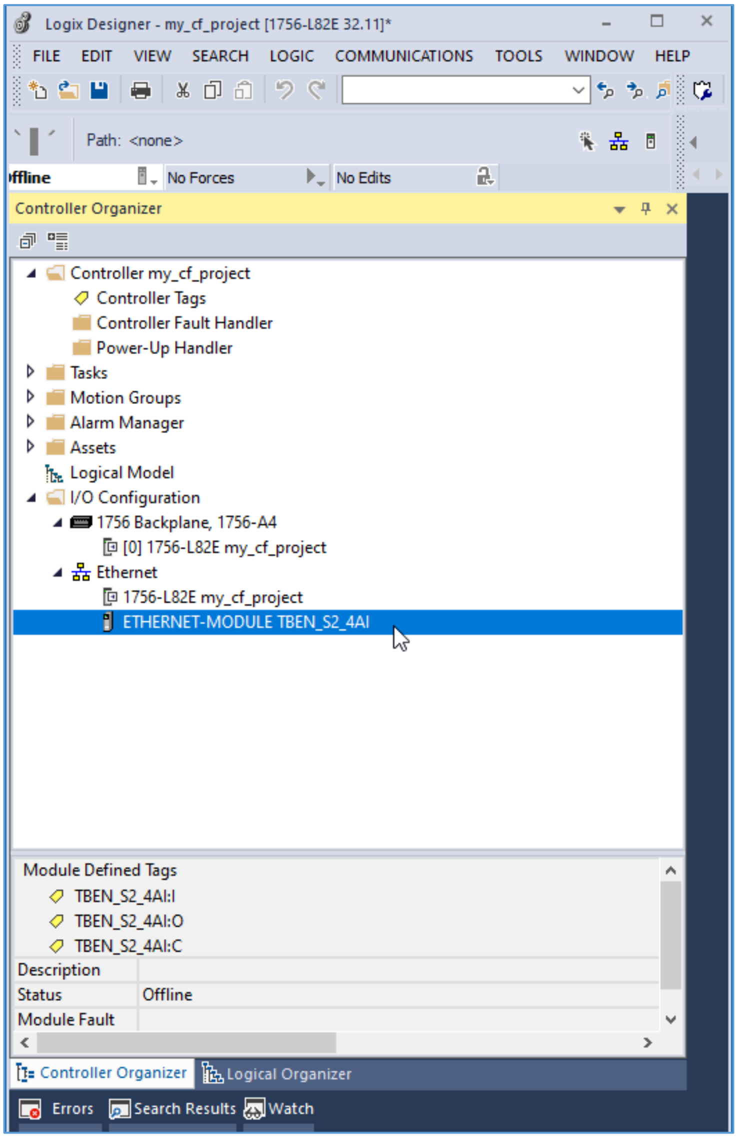
1. Find the device you want in the Catalog Project's "I/O Configuration" folder.



2. Click and drag it from the Catalog Project to the "Ethernet" subfolder in the target project's "I/O Configuration" folder.

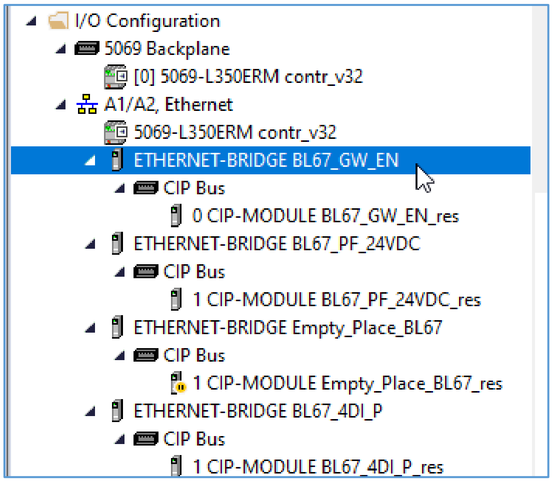


3. Your device is now mapped in the target project.



5.2.2 Modular I/O Stations

1. Find the ETHERNET-BRIDGE that contains your gateway's name in the Catalog Project's "I/O Configuration" folder. In this guide, the gateway will be a BL67-GW-EN-4F, with one BL67-8XSG-P and one BL67-2AI2AO-V/I I/O module.

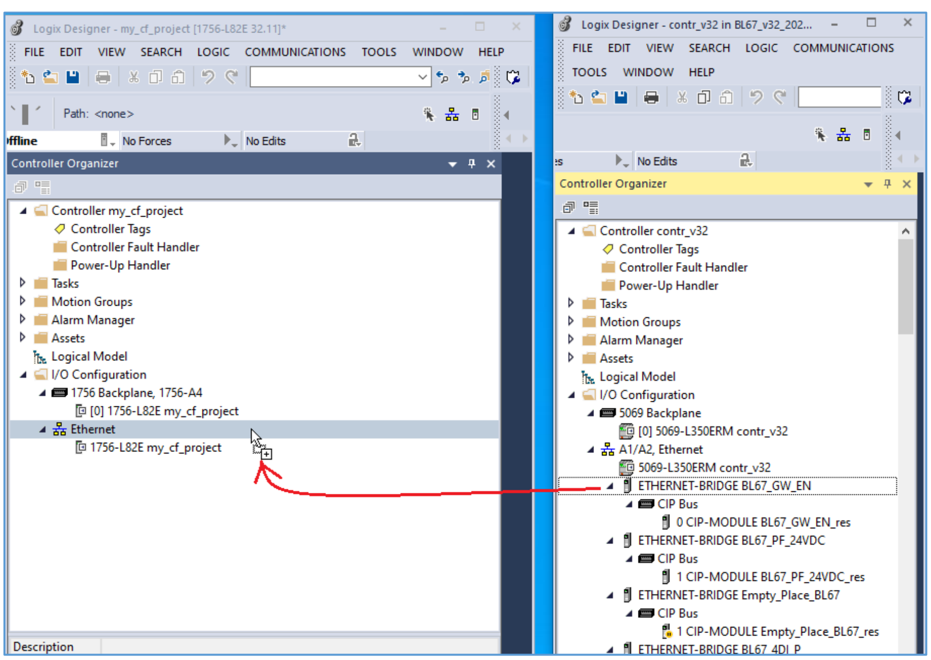


NOTE

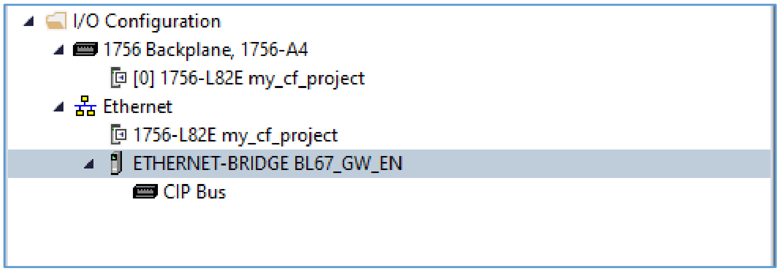
You can tell gateways from I/O modules in the Catalog Project by the fact that the gateways have a slot number of zero. Note that some Catalog Projects might only have one gateway listed; if that's the case, use the ETHERNET-BRIDGE associated with it, even if it doesn't match your gateway's part number. This is allowable because all that mapping via Generic Ethernet Modules is doing is reserving a known number of bytes for a station, and that number is usually the same across different gateways in the same family.

---

2. Click and drag the ETHERNET-BRIDGE from the Catalog Project into the "Ethernet" subfolder in the target project's "I/O Configuration" folder.



3. The ETHERNET-BRIDGE device is now added to the target project. The “CIP bus” device that was automatically created below the ETHERNET-BRIDGE device is where I/O modules are added.

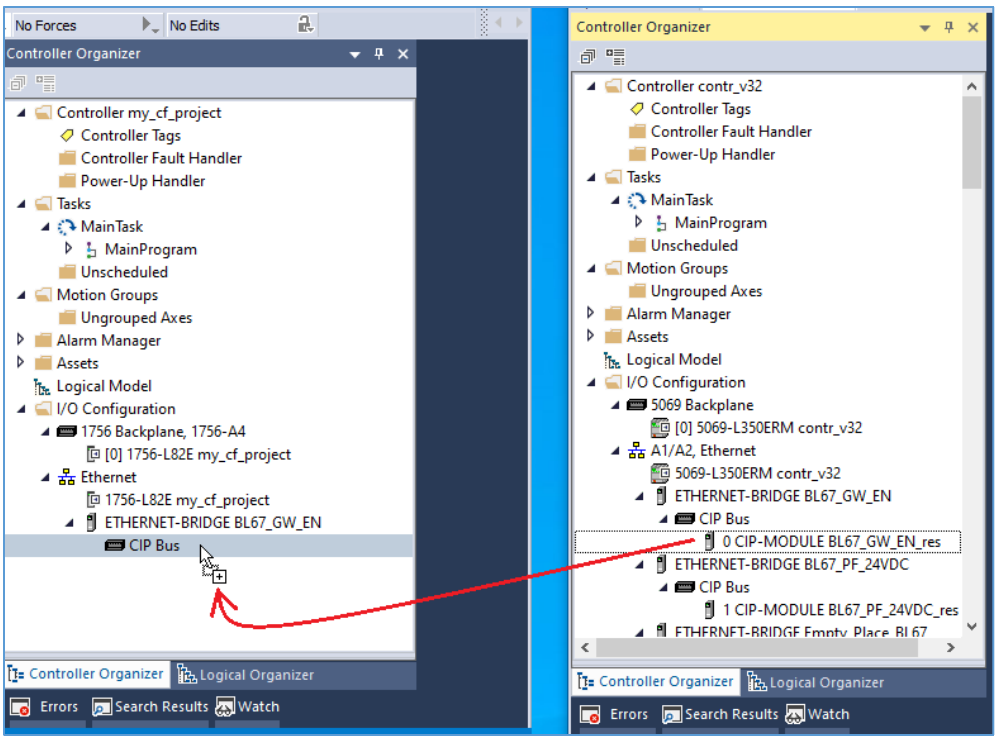


NOTE

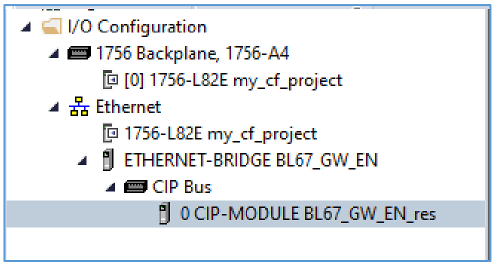
The gateway is considered “module 0” for all Turck’s modular I/O platforms, so that must be the first “I/O module” dragged over to the CIP Bus in the target project. Gateways don’t have any I/O channels, but they do contain some diagnostic information pertaining to the gateway itself and the status of the entire modular I/O rack.

---

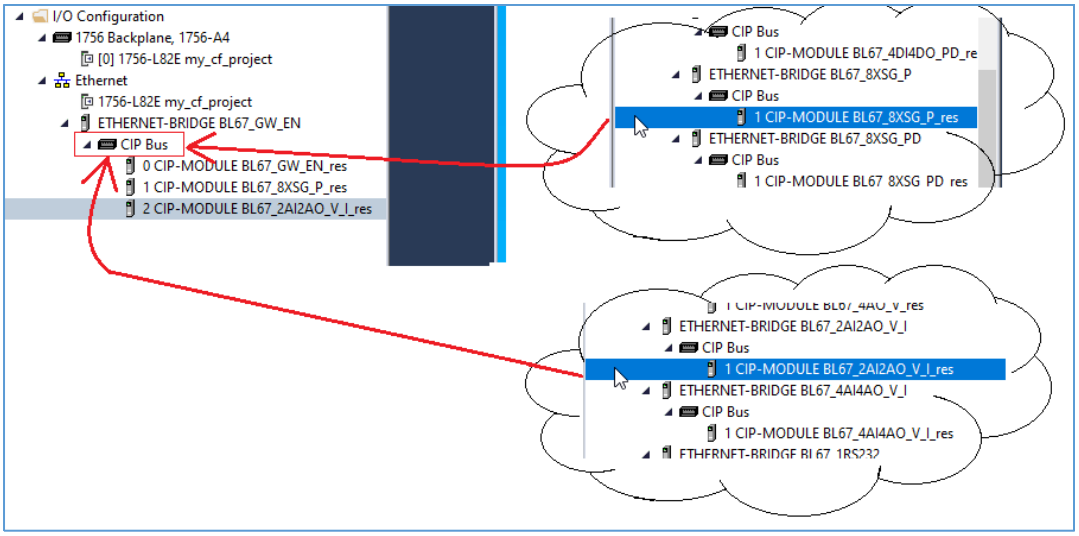
4. Drag over the gateway device from the Catalog Project that was directly under the ETHERNET-BRIDGE device you chose.



5. The gateway is now added to the target project.



6. Drag over the I/O modules (named “1 CIP-Module xxxx”) in your modular I/O rack from the Catalog Project to the CIP Bus in the target project. In this guide, we will add one 8XSG-P and one 2AI2AO-V/I module to our BL67-GW-EN-4F gateway.



WARNING!

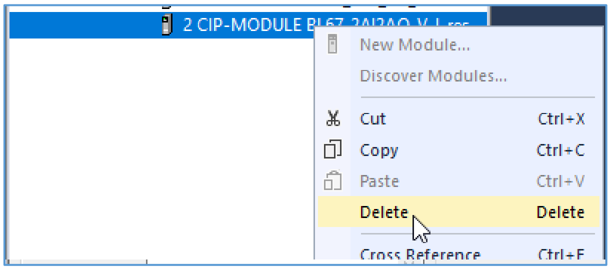
The order of the modules in the target project *must* match the order of the modules in the real world.

NOTE

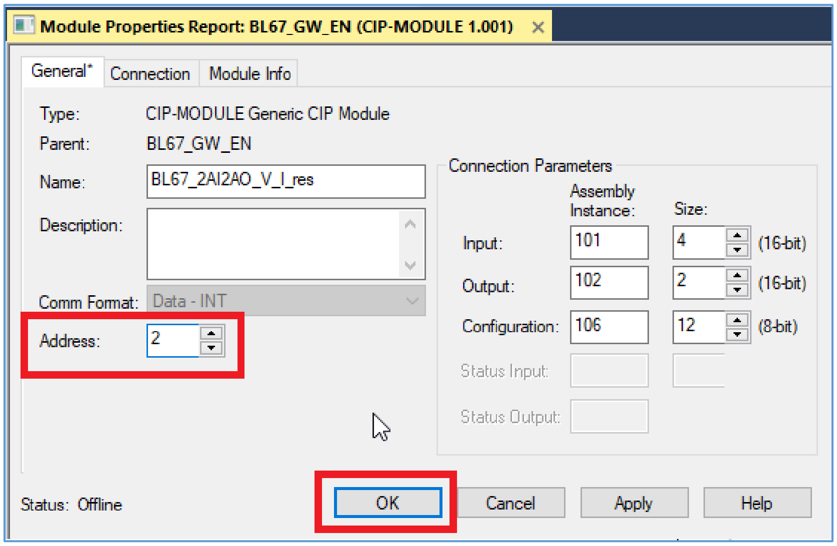
You will see warnings appear after you add more than one I/O module. That is normal. Logix Designer is simply alerting you that it’s automatically updating the module numbers as you drag them over.

---

7. You are free to delete and reorder I/O modules as needed. To delete an I/O module, right-click it, and select Delete.



To reorder an I/O module, right-click the I/O module, select "Properties," change its address, and click OK.



WARNING!

If you delete an I/O module from the middle of the rack, you must either delete all the modules proceeding it and drag over new ones from the Catalog Project, or reorder all the I/O modules proceeding the one you deleted.

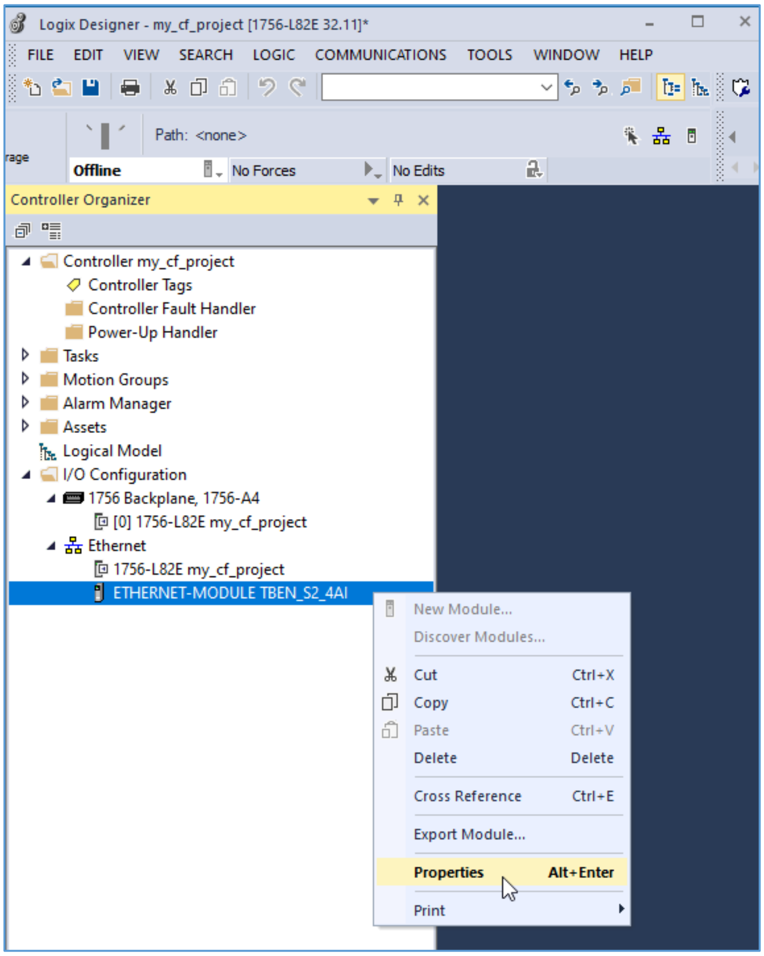
6 Setting Up Your I/O Stations

6.1 Overview

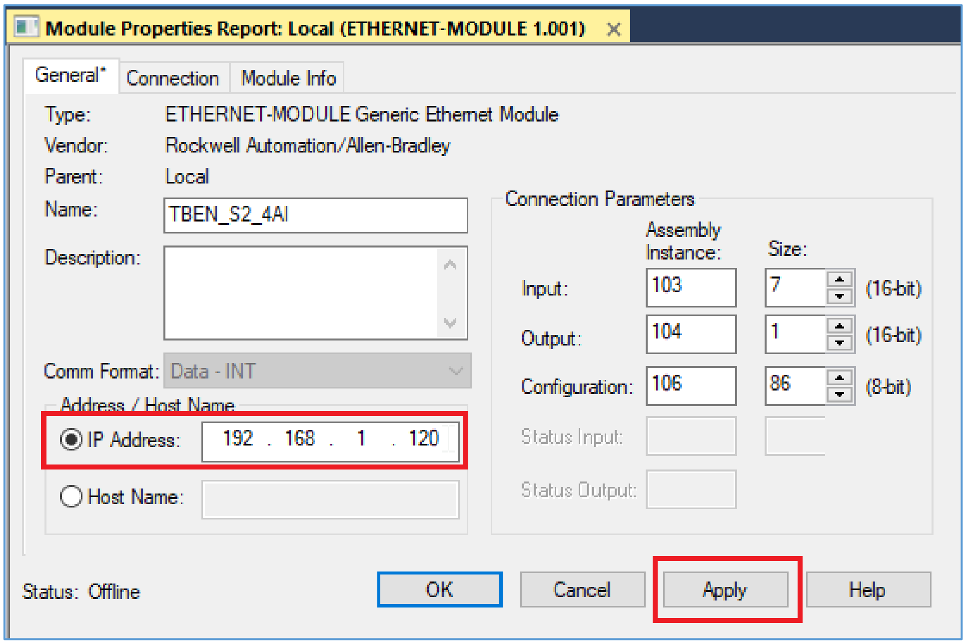
Now that your I/O station(s) are in your project, it's time for configuration. There are a few configuration steps common to all devices. Once those are covered, this guide will go through a few configuration examples for a few specific types of I/O.

6.2 General Configuration

1. To declare a device's expected IP address, right-click the device, then click "Properties."



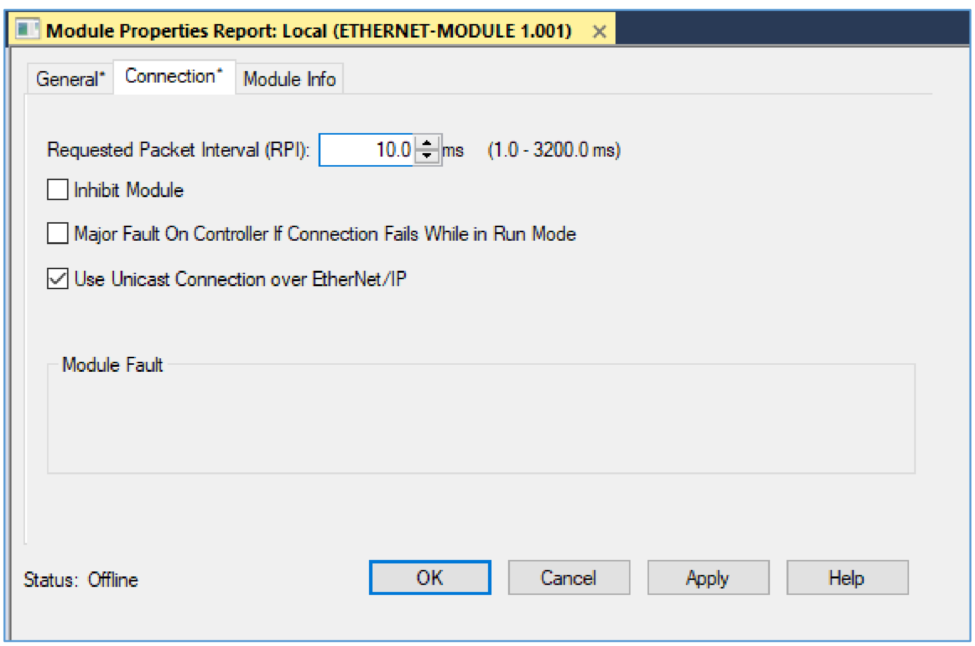
2. With the Properties window open, click the "IP Address" radio button under "Address / Host Name" in the General tab. Type in your device's expected IP address, and click "Apply."



NOTE

You may also change your device's Name in the Properties tab, but this is not required.

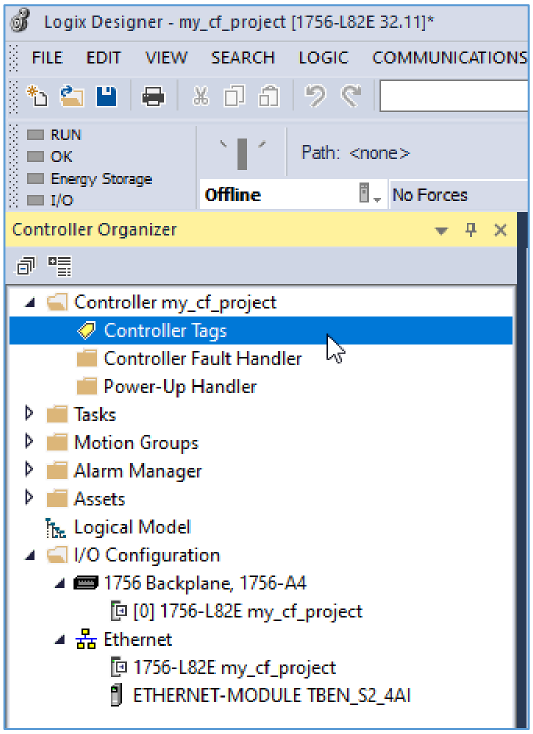
3. If you wish to change your device's connection properties, such as RPI, click the Connection tab in the Properties window. Make any changes you want, then click "Apply."



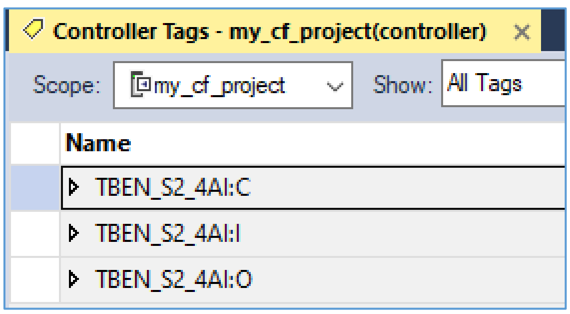
4. When you are finished changing your device's properties, click "OK" to close the window.

Turck Inc. | 3000 Campus Drive, Minneapolis, MN 55441 | T +1 800 544 7769 | F +1 763 553 0708 | www.turck.com

5. It's now time to choose parameters for each station and its I/O channel(s). Start by double-clicking "Controller Tags" to open the Controller Tags window.



6. You'll see three groups of tags: C (configuration), I (input), and O (output).



NOTE

The "Description" column in the Controller Tags window tells you what each tag does. You may have to expand the groups of tags for the information to appear, and you may have to widen the Description column to see all the text in each cell.

---

7. From here, configuration becomes specific to the type(s) of I/O that you're using. The next section contains some examples.

6.3 Example – Digital

Digital I/O is by far the easiest type of I/O to configure because of its simplicity. This example covers the parameterization of a couple channels of a TBEN-LL-16DXP. The process is similar for all other digital I/O.

1. Expand the Configuration tags so that you see text in the Description column.

Controller Tags - my_cf_project(controller)

Scope: my_cf_project Show: All Tags

Name	Value	Forc	Style	Data T	Description
TBEN_LL_16DXP:I	{...}	{...}		AB:ET...	
TBEN_LL_16DXP:O	{...}	{...}		AB:ET...	
TBEN_LL_16DXP:C	{...}	{...}		AB:ET...	
TBEN_LL_16DXP:C.Data	{...}	{...}	Hex	SINT[...	
TBEN_LL_16DXP:C.Data[0]	16#00		Hex	SINT	res
TBEN_LL_16DXP:C.Data[1]	16#00		Hex	SINT	res
TBEN_LL_16DXP:C.Data[2]	16#00		Hex	SINT	res
TBEN_LL_16DXP:C.Data[3]	16#00		Hex	SINT	res
TBEN_LL_16DXP:C.Data[4]	16#00		Hex	SINT	res
TBEN_LL_16DXP:C.Data[5]	16#00		Hex	SINT	res
TBEN_LL_16DXP:C.Data[6]	16#00		Hex	SINT	res
TBEN_LL_16DXP:C.Data[7]	16#00		Hex	SINT	res
TBEN_LL_16DXP:C.Data[8]	16#00		Hex	SINT	res
TBEN_LL_16DXP:C.Data[9]	16#00		Hex	SINT	Quick Connect, Eth Custom Setup
TBEN_LL_16DXP:C.Data[10]	16#00		Hex	SINT	Digital In/Out 0 - Invert digital input
TBEN_LL_16DXP:C.Data[11]	16#00		Hex	SINT	Digital In/Out 1 - Invert digital input
TBEN_LL_16DXP:C.Data[12]	16#00		Hex	SINT	Digital In/Out 2 - Invert digital input
TBEN_LL_16DXP:C.Data[13]	16#00		Hex	SINT	Digital In/Out 3 - Invert digital input
TBEN_LL_16DXP:C.Data[14]	16#00		Hex	SINT	Digital In/Out 4 - Invert digital input
TBEN_LL_16DXP:C.Data[15]	16#00		Hex	SINT	Digital In/Out 5 - Invert digital input

6.3.1 Inverted Input

In this example, we will tell the TBEN to automatically invert the signal coming in on channel 0, i.e. a true signal is reported as false, and a false signal is reported as true. This is useful if a program was written expecting a normally-closed (N.C.) sensor but the sensor in the application is normally-open (N.O.), or if a N.O. sensor is being swapped out for a N.C. sensor due to part availability.

1. Expand the "Invert digital input" tag, and enter a 1. That corresponds with an inverted input.

TBEN_LL_16DXP:C.Data[8]	16#00		Hex	SINT	res
TBEN_LL_16DXP:C.Data[9]	16#00		Hex	SINT	Quick Connect, Eth Custom Setup
TBEN_LL_16DXP:C.Data[10]	16#01		Hex	SINT	Digital In/Out 0 - Invert digital input
TBEN_LL_16DXP:C.Data[10].0	1		Decimal	BOOL	Digital In/Out 0 - Invert digital input (ENUM bit0): 0=no, 1=yes
TBEN_LL_16DXP:C.Data[10].1	0		Decimal	BOOL	res
TBEN_LL_16DXP:C.Data[10].2	0		Decimal	BOOL	res
TBEN_LL_16DXP:C.Data[10].3	0		Decimal	BOOL	res
TBEN_LL_16DXP:C.Data[10].4	0		Decimal	BOOL	res
TBEN_LL_16DXP:C.Data[10].5	0		Decimal	BOOL	res
TBEN_LL_16DXP:C.Data[10].6	0		Decimal	BOOL	res
TBEN_LL_16DXP:C.Data[10].7	0		Decimal	BOOL	res
TBEN_LL_16DXP:C.Data[11]	16#00		Hex	SINT	Digital In/Out 1 - Invert digital input

6.3.2 Activate Output

In this example, we will tell the TBEN to remove the ability of channel 1 to act as an output. By default, all channels have the ability to act as inputs and outputs.

1. Expand the “Activate output” tag, and enter a 0. That corresponds with a deactivated output.

▸ TBEN_LL_16DXP:C.Data[42]	16#01		Hex	SINT	Digital In/Out 0 - Activate output
◢ TBEN_LL_16DXP:C.Data[43]	16#00		Hex	SINT	Digital In/Out 1 - Activate output
TBEN_LL_16DXP:C.Data[43].0	0		Decimal	BOOL	Digital In/Out 1 - Activate output (ENUM bit0) 0=no, 1=yes
TBEN_LL_16DXP:C.Data[43].1	0		Decimal	BOOL	res
TBEN_LL_16DXP:C.Data[43].2	0		Decimal	BOOL	res
TBEN_LL_16DXP:C.Data[43].3	0		Decimal	BOOL	res
TBEN_LL_16DXP:C.Data[43].4	0		Decimal	BOOL	res
TBEN_LL_16DXP:C.Data[43].5	0		Decimal	BOOL	res
TBEN_LL_16DXP:C.Data[43].6	0		Decimal	BOOL	res
TBEN_LL_16DXP:C.Data[43].7	0		Decimal	BOOL	res
▸ TBEN_LL_16DXP:C.Data[44]	16#01		Hex	SINT	Digital In/Out 2 - Activate output

6.4 Example – Analog

Many of Turck's analog I/O stations are compatible with multiple types or ranges of signals. This example covers setting up the four ports of a TBEN-S2-4AI to accept current, voltage, RTD, thermocouple, and resistance (i.e. potentiometer or thermistor) inputs. The process is similar for all other analog I/O.

6.4.1 Current

In this example, we will configure port 0 of a TBEN-S2-4AI to accept a 2-wire 4…20mA input. The process is similar for other wirings and ranges.

2. Expand the "Operation mode" tag, and make sure it reads 0-1-0-0 from top to bottom. That corresponds with a current input.

▸ TBEN_S2_4AI:C.Data[9]	16#00	Hex	SINT	Quick Connect, Eth Custom Setup
◢ TBEN_S2_4AI:C.Data[10]	16#02	Hex	SINT	Analog input 0 - Operation mode
TBEN_S2_4AI:C.Data[10].0	0	Decimal	BOOL	Analog input 0 - Operation mode (ENUM bit0): 0000=thermocouple, 0001=voltage
TBEN_S2_4AI:C.Data[10].1	1	Decimal	BOOL	Analog input 0 - Operation mode (ENUM bit1): 0010=current
TBEN_S2_4AI:C.Data[10].2	0	Decimal	BOOL	Analog input 0 - Operation mode (ENUM bit2): 0011=resistance
TBEN_S2_4AI:C.Data[10].3	0	Decimal	BOOL	Analog input 0 - Operation mode (ENUM bit3): 0100=RTD
TBEN_S2_4AI:C.Data[10].4	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[10].5	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[10].6	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[10].7	0	Decimal	BOOL	res
▸ TBEN_S2_4AI:C.Data[11]	16#00	Hex	SINT	Analog input 0 - Thermocouple type

3. Expand the "Current range" tag, and make sure it reads 0-0 from top to bottom. That corresponds with a range of 4…20mA.

▸ TBEN_S2_4AI:C.Data[14]	16#00	Hex	SINT	Analog input 0 - Voltage wiring type
◢ TBEN_S2_4AI:C.Data[15]	16#00	Hex	SINT	Analog input 0 - Current range
TBEN_S2_4AI:C.Data[15].0	0	Decimal	BOOL	Analog input 0 - Current range (ENUM bit0) 00=4...20 mA, 01=0...20 mA
TBEN_S2_4AI:C.Data[15].1	0	Decimal	BOOL	Analog input 0 - Current range (ENUM bit1): 10=-20...20 mA
TBEN_S2_4AI:C.Data[15].2	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[15].3	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[15].4	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[15].5	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[15].6	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[15].7	0	Decimal	BOOL	res
▸ TBEN_S2_4AI:C.Data[16]	16#02	Hex	SINT	Analog input 0 - Current wiring type

4. Expand the "Current wiring type" tag, and make sure it reads 0-1 from top to bottom. That corresponds with a range of 4…20mA.

▸ TBEN_S2_4AI:C.Data[15]	16#00	Hex	SINT	Analog input 0 - Current range
◢ TBEN_S2_4AI:C.Data[16]	16#02	Hex	SINT	Analog input 0 - Current wiring type
TBEN_S2_4AI:C.Data[16].0	0	Decimal	BOOL	Analog input 0 - Current wiring type (ENUM bit0): 00=differential, 01=single ended
TBEN_S2_4AI:C.Data[16].1	1	Decimal	BOOL	Analog input 0 - Current wiring type (ENUM bit1): 10=differential without ground
TBEN_S2_4AI:C.Data[16].2	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[16].3	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[16].4	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[16].5	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[16].6	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[16].7	0	Decimal	BOOL	res
▸ TBEN_S2_4AI:C.Data[17]	16#00	Hex	SINT	Analog input 0 - Resistance wiring type

5. The current input is now configured. There are other channel-specific features, like filtering or deactivating diagnostics, that you may configure if you wish.

6.4.2 Voltage

In this example, we will configure port 1 of a TBEN-S2-4AI to accept a 3-wire 0…5V input. The process is similar for other wirings and ranges.

1. Expand the “Operation mode” tag, and make sure it reads 1-0-0-0 from top to bottom. That corresponds with a voltage input.

▸ TBEN_S2_4AI:C.Data[27]	16#00	Hex	SINT	res
◢ TBEN_S2_4AI:C.Data[28]	16#01	Hex	SINT	Analog input 1 - Operation mode
TBEN_S2_4AI:C.Data[28].0	1	Decimal	BOOL	Analog input 1 - Operation mode (ENUM bit0): 0000=thermocouple, 0001=voltage
TBEN_S2_4AI:C.Data[28].1	0	Decimal	BOOL	Analog input 1 - Operation mode (ENUM bit1): 0010=current
TBEN_S2_4AI:C.Data[28].2	0	Decimal	BOOL	Analog input 1 - Operation mode (ENUM bit2): 0011=resistance
TBEN_S2_4AI:C.Data[28].3	0	Decimal	BOOL	Analog input 1 - Operation mode (ENUM bit3): 0100=RTD
TBEN_S2_4AI:C.Data[28].4	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[28].5	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[28].6	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[28].7	0	Decimal	BOOL	res
▸ TBEN_S2_4AI:C.Data[29]	16#00	Hex	SINT	Analog input 1 - Thermocouple type…

2. Expand the “Voltage range” tag, and make sure it reads 1-1-0-0 from top to bottom. That corresponds with a range of 0…5V.

▸ TBEN_S2_4AI:C.Data[30]	16#00	Hex	SINT	Analog input 1 - Thermocouple cold junc. config.
◢ TBEN_S2_4AI:C.Data[31]	16#03	Hex	SINT	Analog input 1 - Voltage range
TBEN_S2_4AI:C.Data[31].0	1	Decimal	BOOL	Analog input 1 - Voltage range (ENUM bit0): 0000=-10…10 V, 0001=0…10 V, 0010=2…10 V
TBEN_S2_4AI:C.Data[31].1	1	Decimal	BOOL	Analog input 1 - Voltage range (ENUM bit1): 0011=0…5 V, 0100=1…5 V
TBEN_S2_4AI:C.Data[31].2	0	Decimal	BOOL	Analog input 1 - Voltage range (ENUM bit2): 0101=-1…1 V, 0110=-500…500 mV
TBEN_S2_4AI:C.Data[31].3	0	Decimal	BOOL	Analog input 1 - Voltage range (ENUM bit3): 0111=-100…100 mV, 1000=-50…50 mV
TBEN_S2_4AI:C.Data[31].4	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[31].5	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[31].6	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[31].7	0	Decimal	BOOL	res
▸ TBEN_S2_4AI:C.Data[32]	16#01	Hex	SINT	Analog input 1 - Voltage wiring type…

3. Expand the “Voltage wiring type” tag, and make sure it reads 1-0 from top to bottom. That corresponds with a 3-wire input (also known as single ended).

▸ TBEN_S2_4AI:C.Data[31]	16#03	Hex	SINT	Analog input 1 - Voltage range
◢ TBEN_S2_4AI:C.Data[32]	16#01	Hex	SINT	Analog input 1 - Voltage wiring type
TBEN_S2_4AI:C.Data[32].0	1	Decimal	BOOL	Analog input 1 - Voltage wiring type (ENUM bit0): 00=differential, 01=single ended
TBEN_S2_4AI:C.Data[32].1	0	Decimal	BOOL	Analog input 1 - Voltage wiring type (ENUM bit1): 10=differential without ground
TBEN_S2_4AI:C.Data[32].2	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[32].3	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[32].4	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[32].5	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[32].6	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[32].7	0	Decimal	BOOL	res

4. The voltage input is now configured. There are other channel-specific features, like filtering or deactivating diagnostics, that you may configure if you wish.

6.4.3 RTD

In this example, we will configure port 2 of a TBEN-S2-4AI to accept a 3-wire Pt1000 RTD input with a temperature range of -200…+850°C. The process is similar for other wirings, ranges, and RTD types.

1. Expand the “Operation mode” tag, and make sure it reads 0-0-1-0 from top to bottom. That corresponds with an RTD input.

▸ TBEN_S2_4AI:C.Data[44]	16#00		Hex	SINT	Analog input 1 - Mains suppression
▸ TBEN_S2_4AI:C.Data[45]	16#00		Hex	SINT	res
◢ TBEN_S2_4AI:C.Data[46]	16#04		Hex	SINT	Analog input 2 - Operation mode
TBEN_S2_4AI:C.Data[46].0	0		Decimal	BOOL	Analog input 2 - Operation mode (ENUM bit0): 0000=thermocouple, 0001=voltage
TBEN_S2_4AI:C.Data[46].1	0		Decimal	BOOL	Analog input 2 - Operation mode (ENUM bit1): 0010=current
TBEN_S2_4AI:C.Data[46].2	1		Decimal	BOOL	Analog input 2 - Operation mode (ENUM bit2): 0011=resistance
TBEN_S2_4AI:C.Data[46].3	0		Decimal	BOOL	Analog input 2 - Operation mode (ENUM bit3): 0100=RTD
TBEN_S2_4AI:C.Data[46].4	0		Decimal	BOOL	res
TBEN_S2_4AI:C.Data[46].5	0		Decimal	BOOL	res
TBEN_S2_4AI:C.Data[46].6	0		Decimal	BOOL	res
TBEN_S2_4AI:C.Data[46].7	0		Decimal	BOOL	res
▸ TBEN_S2_4AI:C.Data[47]	16#00		Hex	SINT	Analog input 2 - Thermocouple type

2. Expand the “RTD type” tag, and make sure it reads 0-0-0-1 from top to bottom. That corresponds with a Pt1000 RTD and a temperature range of -200…+850°C. RTDs are a special case in that there are too many valid values to include them all in the Description column, so the manual must be consulted.

▸ TBEN_S2_4AI:C.Data[54]	16#00		Hex	SINT	Analog input 2 - Resistance range
◢ TBEN_S2_4AI:C.Data[55]	16#08		Hex	SINT	Analog input 2 - RTD type
TBEN_S2_4AI:C.Data[55].0	0		Decimal	BOOL	For codes - refer to user manual
TBEN_S2_4AI:C.Data[55].1	0		Decimal	BOOL	For codes - refer to user manual
TBEN_S2_4AI:C.Data[55].2	0		Decimal	BOOL	For codes - refer to user manual
TBEN_S2_4AI:C.Data[55].3	1		Decimal	BOOL	For codes - refer to user manual
TBEN_S2_4AI:C.Data[55].4	0		Decimal	BOOL	For codes - refer to user manual
TBEN_S2_4AI:C.Data[55].5	0		Decimal	BOOL	For codes - refer to user manual
TBEN_S2_4AI:C.Data[55].6	0		Decimal	BOOL	For codes - refer to user manual
TBEN_S2_4AI:C.Data[55].7	0		Decimal	BOOL	For codes - refer to user manual
▸ TBEN_S2_4AI:C.Data[56]	16#01		Hex	SINT	Analog input 2 - RTD wiring type

Table 7-2: parameters

Parameter name	Abbrev.	Value	Description
RTD type	RTDT	0000 0000 = Pt100, -200…850 °C, -328…1562 °F **A**	
		0000 0001 = Pt100, -200…150 °C, -328…302 °F	
		0000 0010 = Ni100, -60…250 °C, -76…482 °F	
		0000 0011 = Ni100, -60…150 °C, -76…302 °F	
		0000 0100 = Pt200, -200…850 °C, -328…1562 °F	
		0000 0101 = Pt200, -200…150 °C, -328…302 °F	
		0000 0110 = Pt500, -200…850 °C, -328…1562 °F	
		0000 0111 = Pt500, -200…150 °C, -328…302 °F	
		0000 1000 = Pt1000, -200…850 °C, -328…1562 °F	
		0000 1001 = Pt1000, -200…150 °C, -328…302 °F	
		0000 1010 = Ni1000, -60…250 °C, -76…482 °F	
		0000 1011 = Ni1000, -60…150 °C, -76…302 °F	

3. Expand the "RTD wiring type" tag, and make sure it reads 1-0 from top to bottom. That corresponds with a 3-wire RTD.

▸ TBEN_S2_4AI:C.Data[55]	16#00	Hex	SINT	Analog input 2 - RTD type
◢ TBEN_S2_4AI:C.Data[56]	16#01	Hex	SINT	Analog input 2 - RTD wiring type
TBEN_S2_4AI:C.Data[56].0	1	Decimal	BOOL	Analog input 2 - RTD wiring type (ENUM bit0): 00=2-wire, 01=3-wire
TBEN_S2_4AI:C.Data[56].1	0	Decimal	BOOL	Analog input 2 - RTD wiring type (ENUM bit1): 10=4-wire
TBEN_S2_4AI:C.Data[56].2	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[56].3	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[56].4	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[56].5	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[56].6	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[56].7	0	Decimal	BOOL	res
▸ TBEN_S2_4AI:C.Data[57]	16#00	Hex	SINT	Analog input 2 - Data representation

4. The RTD input is now configured. There are other channel-specific features, like filtering or deactivating diagnostics, that you may activate if you wish.

6.4.4 Thermocouple

In this example, we will configure port 3 of a TBEN-S2-4AI to accept a type-J thermocouple input with Pt100 cold-junction compensation and a temperature range of -210…+1200°C. The process is similar for other thermocouple configurations.

1. Expand the "Operation mode" tag, and make sure it reads 0-0-0-0 from top to bottom. That corresponds with a thermocouple input.

▸ TBEN_S2_4AI:C.Data[63]	16#00	Hex	SINT	res
◢ TBEN_S2_4AI:C.Data[64]	16#00	Hex	SINT	Analog input 3 - Operation mode
TBEN_S2_4AI:C.Data[64].0	0	Decimal	BOOL	Analog input 3 - Operation mode (ENUM bit0): 0000=thermocouple, 0001=voltage
TBEN_S2_4AI:C.Data[64].1	0	Decimal	BOOL	Analog input 3 - Operation mode (ENUM bit1): 0010=current
TBEN_S2_4AI:C.Data[64].2	0	Decimal	BOOL	Analog input 3 - Operation mode (ENUM bit2): 0011=resistance
TBEN_S2_4AI:C.Data[64].3	0	Decimal	BOOL	Analog input 3 - Operation mode (ENUM bit3): 0100=RTD
TBEN_S2_4AI:C.Data[64].4	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[64].5	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[64].6	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[64].7	0	Decimal	BOOL	res
▸ TBEN_S2_4AI:C.Data[65]	16#00	Hex	SINT	Analog input 3 - Thermocouple type

2. Expand the "Thermocouple type" tag, and make sure it reads 1-1-0-0 from top to bottom. That corresponds with a type-J thermocouple and a temperature range of -210…+1200°C.

▸ TBEN_S2_4AI:C.Data[64]	16#00	Hex	SINT	Analog input 3 - Operation mode
◢ TBEN_S2_4AI:C.Data[65]	16#03	Hex	SINT	Analog input 3 - Thermocouple type
TBEN_S2_4AI:C.Data[65].0	1	Decimal	BOOL	Analog input 3 - Thermocouple type (ENUM bit0): 0000=type K, -270...1370 C,...
TBEN_S2_4AI:C.Data[65].1	1	Decimal	BOOL	Analog input 3 - Thermocouple type (ENUM bit1): 0011=type J, -210...1200 C ...
TBEN_S2_4AI:C.Data[65].2	0	Decimal	BOOL	Analog input 3 - Thermocouple type (ENUM bit2): 0110=type S, -50...1768 C, -...
TBEN_S2_4AI:C.Data[65].3	0	Decimal	BOOL	Analog input 3 - Thermocouple type (ENUM bit3): 1000=type C, 0...2315 C, 32...
TBEN_S2_4AI:C.Data[65].4	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[65].5	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[65].6	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[65].7	0	Decimal	BOOL	res
▸ TBEN_S2_4AI:C.Data[66]	16#00	Hex	SINT	Analog input 3 - Thermocouple cold junc. config.

3. Expand the "Thermocouple cold junc. config." tag, and make sure it reads 1-0-0-0 from top to bottom. That corresponds with cold-junction compensation from a PT100 RTD

TBEN_S2_4AI:C.Data[65]	16#03	Hex	SINT	Analog input 3 - Thermocouple type
TBEN_S2_4AI:C.Data[66]	16#01	Hex	SINT	Analog input 3 - Thermocouple cold junc. config.
TBEN_S2_4AI:C.Data[66].0	1	Decimal	BOOL	Analog input 3 - Thermocouple cold junc. config. (ENUM bit0): 0000=PT1000
TBEN_S2_4AI:C.Data[66].1	0	Decimal	BOOL	Analog input 3 - Thermocouple cold junc. config. (ENUM bit1): 0001=PT100
TBEN_S2_4AI:C.Data[66].2	0	Decimal	BOOL	Analog input 3 - Thermocouple cold junc. config. (ENUM bit2): 0010=cold junction from channel 0
TBEN_S2_4AI:C.Data[66].3	0	Decimal	BOOL	Analog input 3 - Thermocouple cold junc. config. (ENUM bit3): 0011=none
TBEN_S2_4AI:C.Data[66].4	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[66].5	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[66].6	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[66].7	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[67]	16#00	Hex	SINT	Analog input 3 - Voltage range

4. The thermocouple input with cold-junction compensation is now configured. There are other channel-specific features, like filtering or deactivating diagnostics, that you may activate if you wish.

6.4.5 Resistance (i.e. potentiometers, thermistors)

In this example, we will configure port 3 of a TBEN-S2-4AI to accept a 3-wire potentiometer with a range of 0…2000 Ohms. The process is similar for other resistance configurations.

1. Expand the "Operation mode" tag, and make sure it reads 1-1-0-0 from top to bottom. That corresponds with a resistance input.

TBEN_S2_4AI:C.Data[63]	16#00	Hex	SINT	res
TBEN_S2_4AI:C.Data[64]	16#03	Hex	SINT	Analog input 3 - Operation mode
TBEN_S2_4AI:C.Data[64].0	1	Decimal	BOOL	Analog input 3 - Operation mode (ENUM bit0): 0000=thermocouple, 0001=voltage
TBEN_S2_4AI:C.Data[64].1	1	Decimal	BOOL	Analog input 3 - Operation mode (ENUM bit1): 0010=current
TBEN_S2_4AI:C.Data[64].2	0	Decimal	BOOL	Analog input 3 - Operation mode (ENUM bit2): 0011=resistance
TBEN_S2_4AI:C.Data[64].3	0	Decimal	BOOL	Analog input 3 - Operation mode (ENUM bit3): 0100=RTD
TBEN_S2_4AI:C.Data[64].4	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[64].5	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[64].6	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[64].7	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[65]	16#03	Hex	SINT	Analog input 3 - Thermocouple type

2. Expand the "Resistance wiring type" tag, and make sure it reads 0-1 from top to bottom. That corresponds with a 3-wire setup for our potentiometer (leg 1, leg 2, and a wiper).

TBEN_S2_4AI:C.Data[70]	16#00	Hex	SINT	Analog input 3 - Current wiring type
TBEN_S2_4AI:C.Data[71]	16#01	Hex	SINT	Analog input 3 - Resistance wiring type
TBEN_S2_4AI:C.Data[71].0	1	Decimal	BOOL	Analog input 3 - Resistance wiring type (ENUM bit0): 00=2-wire, 01=3-wire
TBEN_S2_4AI:C.Data[71].1	0	Decimal	BOOL	Analog input 3 - Resistance wiring type (ENUM bit1): 10=4-wire
TBEN_S2_4AI:C.Data[71].2	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[71].3	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[71].4	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[71].5	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[71].6	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[71].7	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[72]	16#02	Hex	SINT	Analog input 3 - Resistance range

3. Expand the “Resistance range” tag, and make sure it reads 0-1 from top to bottom. That corresponds with a range of 0…2000 Ohms.

▶ TBEN_S2_4AI:C.Data[71]	16#02	Hex	SINT	Analog input 3 - Resistance wiring type
◢ TBEN_S2_4AI:C.Data[72]	16#02	Hex	SINT	Analog input 3 - Resistance range
TBEN_S2_4AI:C.Data[72].0	0	Decimal	BOOL	Analog input 3 - Resistance range (ENUM bit0): 00=0...100 Ohm, 01=0...400 Ohm
TBEN_S2_4AI:C.Data[72].1	1	Decimal	BOOL	Analog input 3 - Resistance range (ENUM bit1): 10=0...2000 Ohm, 11=0...4000 Ohm
TBEN_S2_4AI:C.Data[72].2	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[72].3	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[72].4	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[72].5	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[72].6	0	Decimal	BOOL	res
TBEN_S2_4AI:C.Data[72].7	0	Decimal	BOOL	res
▶ TBEN_S2_4AI:C.Data[73]	16#00	Hex	SINT	Analog input 3 - RTD type

4. The resistance input is now configured. There are other channel-specific features, like filtering or deactivating diagnostics, that you may activate if you wish.

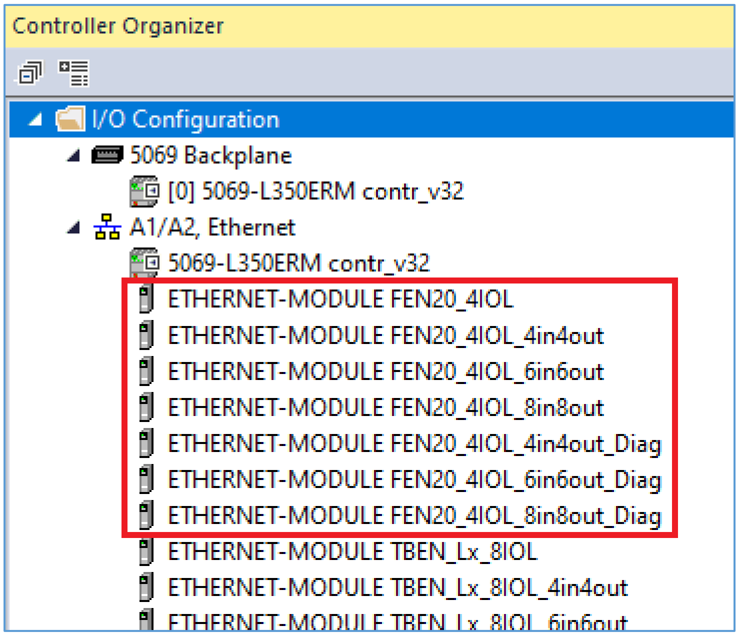
Turck Inc. | 3000 Campus Drive, Minneapolis, MN 55441 | T +1 800 544 7769 | F +1 763 553 0708 | www.turck.com

6.6 Example – IO-Link (IOL)

Parameterizing IO-Link (IOL) channels is more complex, but the effort put in now will make programming and device replacement much easier. This guide will cover which IOL station to choose from the Catalog Project, then setting up a TBIL-M1-16DXP I/O hub as a slave on channel 0 of an FEN20-4IOL.

6.6.1 Choosing an IOL Station from the Catalog Project

IOL stations are unique in that there are a handful of options for each model number.



Although they are meant for the same physical station, the options differ in input and output size. By default, each IOL channel on the IOL station has 16 words of input and output. That can quickly become a problem of required space in your PLC, but if you don't need all 16 input/output words or IOL diagnostics, this problem is avoidable by choosing an option that has fewer of its IOL words mapped. The table below shows the difference between your options.

Station Suffix	IOL input words per channel	IOL output words per channel
none	16	16
_4in4out	2	2
_6in6out	3	3
_8in8out	4	4
_4in4out_Diag	2 + 1 diagnostic word	2
_6in6out_Diag	3 + 1 diagnostic word	3
_8in8out_Diag	4 + 1 diagnostic word	4

Turck Inc. | 3000 Campus Drive, Minneapolis, MN 55441 | T +1 800 544 7769 | F +1 763 553 0708 | www.turck.com

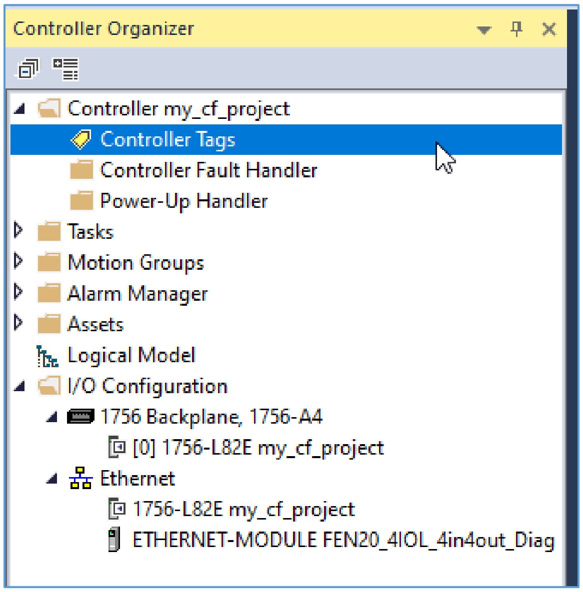
NOTE

The “diagnostic” word referenced above contains Booleans (fifteen for each channel) that become true if the wrong device is connected, the IOL slave is not working correctly, or maintenance is required, among other things.

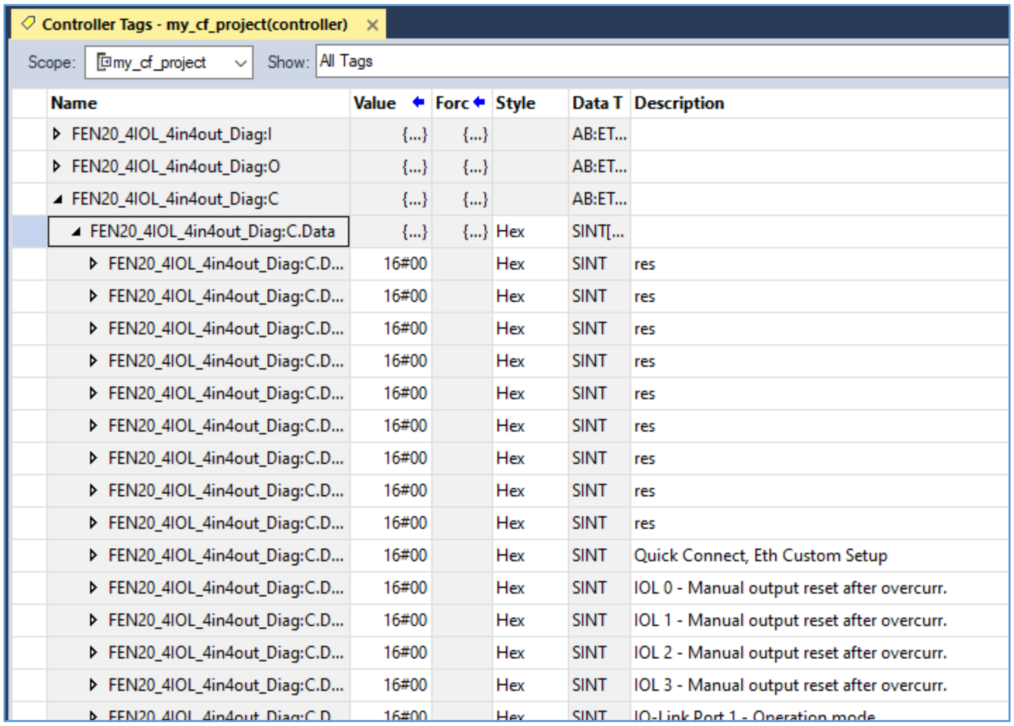
6.6.2 IOL Slave with Inputs, Outputs, and Device Validation

In this example, a TBIL-M1-16DXP will be added to channel 0 of an FEN20-4IOL with device validation active. The station from the Catalog Project chosen for this example is the “FEN20_4IOL_4in4out_Diag,” as the TBIL-M1-16DXP has one word of input, one word of output, and the diagnostic information is desired in this case.

1. Start by double-clicking “Controller Tags” to open the Controller Tags window.



2. Expand the Configuration tags so that you see text in the Description column.

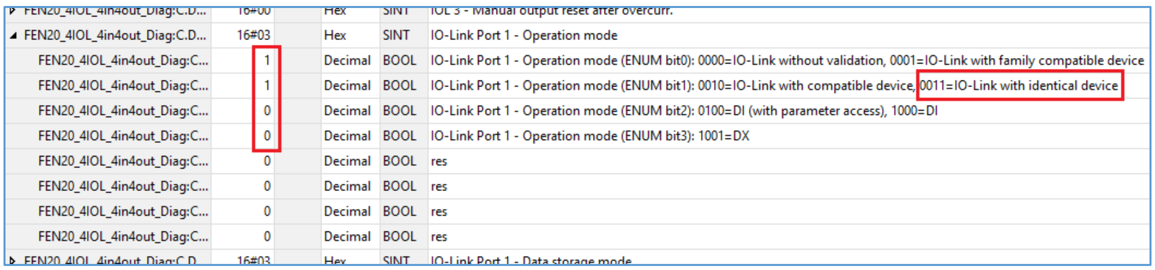


Controller Tags - my_cf_project(controller)

Scope: my_cf_project Show: All Tags

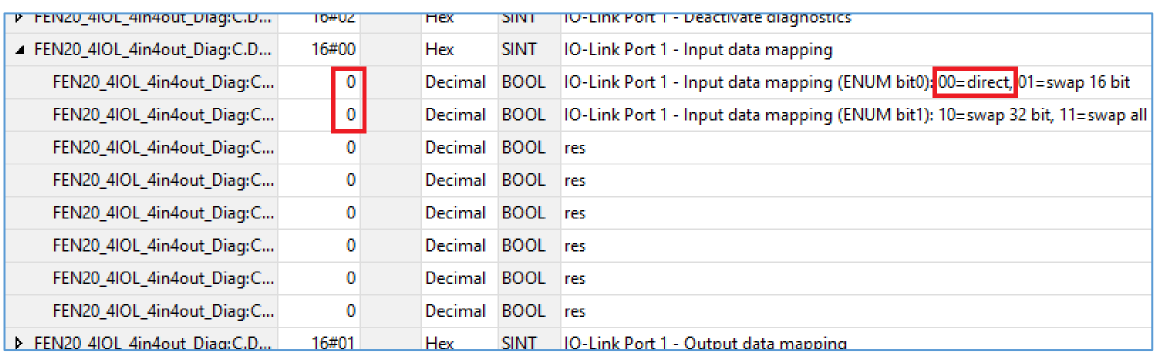
Name	Value	Forc	Style	Data T	Description
FEN20_4IOL_4in4out_Diag:I	{...}	{...}		AB:ET...	
FEN20_4IOL_4in4out_Diag:O	{...}	{...}		AB:ET...	
FEN20_4IOL_4in4out_Diag:C	{...}	{...}		AB:ET...	
FEN20_4IOL_4in4out_Diag:C.Data	{...}	{...}	Hex	SINT[...	
FEN20_4IOL_4in4out_Diag:C.D...	16#00		Hex	SINT	res
FEN20_4IOL_4in4out_Diag:C.D...	16#00		Hex	SINT	res
FEN20_4IOL_4in4out_Diag:C.D...	16#00		Hex	SINT	res
FEN20_4IOL_4in4out_Diag:C.D...	16#00		Hex	SINT	res
FEN20_4IOL_4in4out_Diag:C.D...	16#00		Hex	SINT	res
FEN20_4IOL_4in4out_Diag:C.D...	16#00		Hex	SINT	res
FEN20_4IOL_4in4out_Diag:C.D...	16#00		Hex	SINT	res
FEN20_4IOL_4in4out_Diag:C.D...	16#00		Hex	SINT	res
FEN20_4IOL_4in4out_Diag:C.D...	16#00		Hex	SINT	res
FEN20_4IOL_4in4out_Diag:C.D...	16#00		Hex	SINT	Quick Connect, Eth Custom Setup
FEN20_4IOL_4in4out_Diag:C.D...	16#00		Hex	SINT	IOL 0 - Manual output reset after overcurr.
FEN20_4IOL_4in4out_Diag:C.D...	16#00		Hex	SINT	IOL 1 - Manual output reset after overcurr.
FEN20_4IOL_4in4out_Diag:C.D...	16#00		Hex	SINT	IOL 2 - Manual output reset after overcurr.
FEN20_4IOL_4in4out_Diag:C.D...	16#00		Hex	SINT	IOL 3 - Manual output reset after overcurr.
FEN20_4IOL_4in4out_Diag:C.D...	16#00		Hex	SINT	IO-Link Port 1 - Operation mode

3. Expand the "Operation mode" tag, and make sure it reads 1-1-0-0 from top to bottom. That corresponds with the IOL master only accepting an identical device, that is, only a TBIL-M1-16DXP.



FEN20_4IOL_4in4out_Diag:C.D...	16#00	Hex	SINT	IOL 3 - Manual output reset after overcurr.
FEN20_4IOL_4in4out_Diag:C.D...	16#03	Hex	SINT	IO-Link Port 1 - Operation mode
FEN20_4IOL_4in4out_Diag:C...	1	Decimal	BOOL	IO-Link Port 1 - Operation mode (ENUM bit0): 0000=IO-Link without validation, 0001=IO-Link with family compatible device
FEN20_4IOL_4in4out_Diag:C...	1	Decimal	BOOL	IO-Link Port 1 - Operation mode (ENUM bit1): 0010=IO-Link with compatible device, 0011=IO-Link with identical device
FEN20_4IOL_4in4out_Diag:C...	0	Decimal	BOOL	IO-Link Port 1 - Operation mode (ENUM bit2): 0100=DI (with parameter access), 1000=DI
FEN20_4IOL_4in4out_Diag:C...	0	Decimal	BOOL	IO-Link Port 1 - Operation mode (ENUM bit3): 1001=DX
FEN20_4IOL_4in4out_Diag:C...	0	Decimal	BOOL	res
FEN20_4IOL_4in4out_Diag:C...	0	Decimal	BOOL	res
FEN20_4IOL_4in4out_Diag:C...	0	Decimal	BOOL	res
FEN20_4IOL_4in4out_Diag:C...	0	Decimal	BOOL	res
FEN20_4IOL_4in4out_Diag:C.D...	16#03	Hex	SINT	IO-Link Port 1 - Data storage mode

4. Expand the "Input data mapping" tag, and make sure it reads 0-0 from top to bottom. That corresponds with the IOL master leaving input byte and bit order as they are. Some slaves require swapping byte order, bit order, or both.



FEN20_4IOL_4in4out_Diag:C.D...	16#02	Hex	SINT	IO-Link Port 1 - Deactivate diagnostics
FEN20_4IOL_4in4out_Diag:C.D...	16#00	Hex	SINT	IO-Link Port 1 - Input data mapping
FEN20_4IOL_4in4out_Diag:C...	0	Decimal	BOOL	IO-Link Port 1 - Input data mapping (ENUM bit0): 00=direct, 01=swap 16 bit
FEN20_4IOL_4in4out_Diag:C...	0	Decimal	BOOL	IO-Link Port 1 - Input data mapping (ENUM bit1): 10=swap 32 bit, 11=swap all
FEN20_4IOL_4in4out_Diag:C...	0	Decimal	BOOL	res
FEN20_4IOL_4in4out_Diag:C...	0	Decimal	BOOL	res
FEN20_4IOL_4in4out_Diag:C...	0	Decimal	BOOL	res
FEN20_4IOL_4in4out_Diag:C...	0	Decimal	BOOL	res
FEN20_4IOL_4in4out_Diag:C...	0	Decimal	BOOL	res
FEN20_4IOL_4in4out_Diag:C...	0	Decimal	BOOL	res
FEN20_4IOL_4in4out_Diag:C.D...	16#01	Hex	SINT	IO-Link Port 1 - Output data mapping

5. Expand the “Output data mapping” tag, and make sure it reads 0-0 from top to bottom. That corresponds with the IOL master leaving input byte and bit order as they are. Some slaves require swapping byte order, bit order, or both.

▸ FEN20_4IOL_4in4out_Diag:C.D...	16#00	Hex	SINT	IO-Link Port 1 - Input data mapping
◢ FEN20_4IOL_4in4out_Diag:C.D...	16#00	Hex	SINT	IO-Link Port 1 - Output data mapping
FEN20_4IOL_4in4out_Diag:C...	0	Decimal	BOOL	IO-Link Port 1 - Output data mapping (ENUM bit0): 00=direct, 01=swap 16 bit
FEN20_4IOL_4in4out_Diag:C...	0	Decimal	BOOL	IO-Link Port 1 - Output data mapping (ENUM bit1): 10=swap 32 bit, 11=swap all
FEN20_4IOL_4in4out_Diag:C...	0	Decimal	BOOL	res
FEN20_4IOL_4in4out_Diag:C...	0	Decimal	BOOL	res
FEN20_4IOL_4in4out_Diag:C...	0	Decimal	BOOL	res
FEN20_4IOL_4in4out_Diag:C...	0	Decimal	BOOL	res
FEN20_4IOL_4in4out_Diag:C...	0	Decimal	BOOL	res
FEN20_4IOL_4in4out_Diag:C...	0	Decimal	BOOL	res
▸ FEN20_4IOL_4in4out_Diag:C.D...	16#00	Hex	SINT	IO-Link Port 1 - Vendor ID

6. So far, the IOL master knows to only accept a certain IOL slave, but it does not know the vendor ID and device ID against which to check. Those values are found in the TBIL-M1-16DXP manual:

Parameter	Content
Vendor ID	317 (0x13D)
Device ID	1979139 (0x1E3303)

7. Starting with the vendor ID, enter the hex value found in the manual. It will often be split into multiple tags, and order matters.

▸ FEN20_4IOL_4in4out_Diag:C.D...	16#00	Hex	SINT	IO-Link Port 1 - Output data mapping
▸ FEN20_4IOL_4in4out_Diag:C.D...	16#3d	Hex	SINT	IO-Link Port 1 - Vendor ID
▸ FEN20_4IOL_4in4out_Diag:C.D...	16#01	Hex	SINT	IO-Link Port 1 - Vendor ID
▸ FEN20_4IOL_4in4out_Diag:C.D...	16#00	Hex	SINT	IO-Link Port 1 - Device ID

8. Next, enter the device ID found in the manual. Much like the vendor ID, it will often be split into multiple tags, and order matters.

▸ FEN20_4IOL_4in4out_Diag:C.D...	16#01	Hex	SINT	IO-Link Port 1 - Vendor ID
▸ FEN20_4IOL_4in4out_Diag:C.D...	16#03	Hex	SINT	IO-Link Port 1 - Device ID
▸ FEN20_4IOL_4in4out_Diag:C.D...	16#33	Hex	SINT	IO-Link Port 1 - Device ID
▸ FEN20_4IOL_4in4out_Diag:C.D...	16#1e	Hex	SINT	IO-Link Port 1 - Device ID
▸ FEN20_4IOL_4in4out_Diag:C.D...	16#00	Hex	SINT	IO-Link Port 1 - Device ID
▸ FEN20_4IOL_4in4out_Diag:C.D...	16#00	Hex	SINT	IO-Link Port 2 - Operation mode

Now that the FEN20 knows which IOL slave to expect, if connection is ever lost to the TBIL or if a slave other than a TBIL-M1-16DXP is connected, the “Wrong or missing device” bit in the channel 0 diagnostic word will become true.

FEN20_4IOL_4in4out_Diag:I.Da...	0	Decimal	INT	Overcurrent supply VAUX1 Ch0,Overcurrent supply VAUX1 Ch1,O
FEN20_4IOL_4in4out_Diag:I.Da...	0	Decimal	INT	IOL 0 - Port parameterization error,Wrong or missing device,Data
FEN20_4IOL_4in4out_Diag:I....	0	Decimal	BOOL	res
FEN20_4IOL_4in4out_Diag:I....	0	Decimal	BOOL	IOL 0 - Port parameterization error (ENUM bit0): 0=-, 1=active
FEN20_4IOL_4in4out_Diag:I....	0	Decimal	BOOL	IOL 0 - Wrong or missing device (ENUM bit0): 0=-, 1=active
FEN20_4IOL_4in4out_Diag:I....	0	Decimal	BOOL	IOL 0 - Data storage error (ENUM bit0): 0=-, 1=active
FEN20_4IOL_4in4out_Diag:I....	0	Decimal	BOOL	IOL 0 - Hardware error (ENUM bit0): 0=-, 1=active
FEN20_4IOL_4in4out_Diag:I....	0	Decimal	BOOL	IOL 0 - Process input data invalid (ENUM bit0): 0=-, 1=active
FEN20_4IOL_4in4out_Diag:I....	0	Decimal	BOOL	IOL 0 - Maintenance events (ENUM bit0): 0=-, 1=active
FEN20_4IOL_4in4out_Diag:I....	0	Decimal	BOOL	IOL 0 - Out of spec. error (ENUM bit0): 0=-, 1=active
FEN20_4IOL_4in4out_Diag:I....	0	Decimal	BOOL	IOL 0 - Parameterization error (ENUM bit0): 0=-, 1=active
FEN20_4IOL_4in4out_Diag:I....	0	Decimal	BOOL	IOL 0 - Overtemperature (ENUM bit0): 0=-, 1=active
FEN20_4IOL_4in4out_Diag:I....	0	Decimal	BOOL	IOL 0 - Lower limit value underrun (ENUM bit0): 0=-, 1=active
FEN20_4IOL_4in4out_Diag:I....	0	Decimal	BOOL	IOL 0 - Upper limit value exceeded (ENUM bit0): 0=-, 1=active
FEN20_4IOL_4in4out_Diag:I....	0	Decimal	BOOL	IOL 0 - Undervoltage (ENUM bit0): 0=-, 1=active
FEN20_4IOL_4in4out_Diag:I....	0	Decimal	BOOL	IOL 0 - Overvoltage (ENUM bit0): 0=-, 1=active
FEN20_4IOL_4in4out_Diag:I....	0	Decimal	BOOL	IOL 0 - Overload (ENUM bit0): 0=-, 1=active
FEN20_4IOL_4in4out_Diag:I....	0	Decimal	BOOL	IOL 0 - Common error (ENUM bit0): 0=-, 1=active
FEN20_4IOL_4in4out_Diag:I.Da...	0	Decimal	INT	IOL 1 - Port parameterization error,Wrong or missing device,Data

The IOL channel is now parameterized to accept an IOL slave with inputs, outputs, and device validation. There are other parameters, like quick startup or cycle time, that can be changed in much the same way if the application requires it.

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.

Turck Inc. | 3000 Campus Drive, Minneapolis, MN 55441 | T +1 800 544 7769 | F +1 763 553 0708 | www.turck.com

Your Global Automation Partner

TURCK

28 subsidiaries and over
60 representations worldwide!

www.turck.com