docs

ATOM / Fieldbus

Author: Control Concepts, Inc.

Date: 2025-10-30

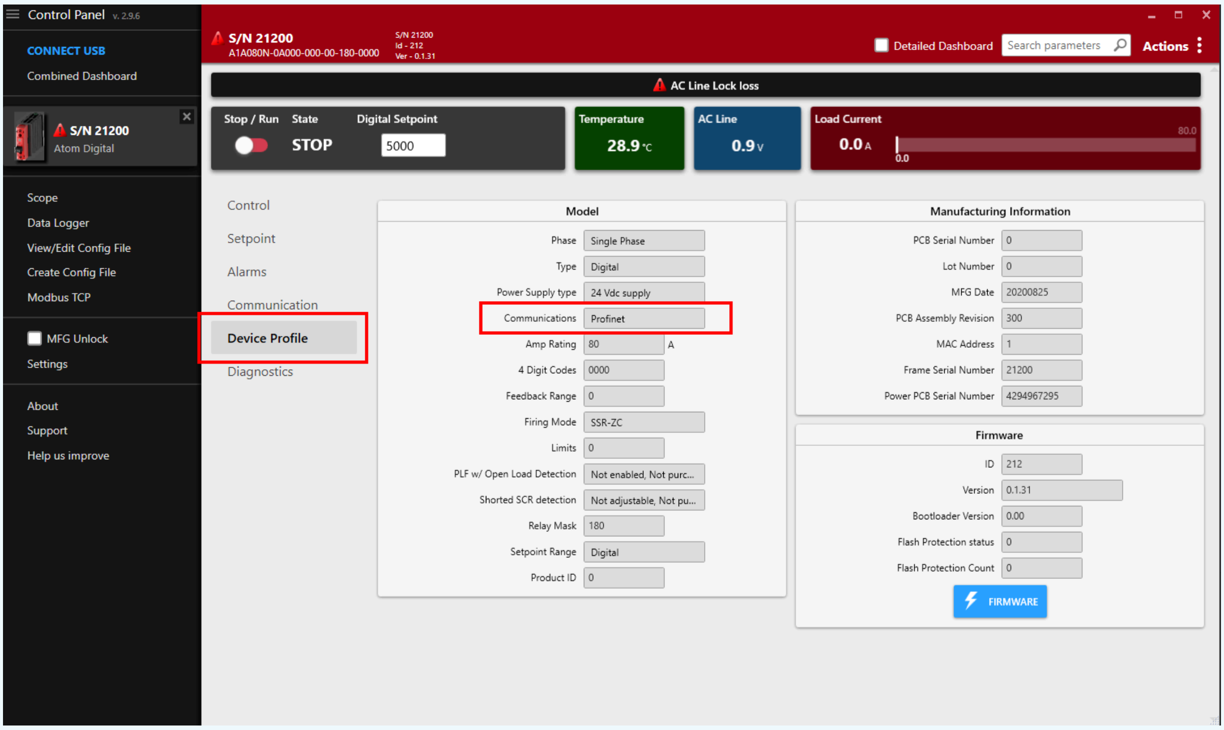
Contents

ATOM / Fieldbus / Overview

ATOM implements the most popular industry standard fieldbus protocols to allow you to connect and integrate ATOM into your power control systems quickly and easily. ATOM can be configured with EtherNet/IP, ModbusTCP, PROFINET, or EtherCAT

INFO

Scan the QR code on your ATOM product label or use this utility to determine which fieldbus your ATOM is configured for. You can also check the **Device profile** tab in Control Panel:

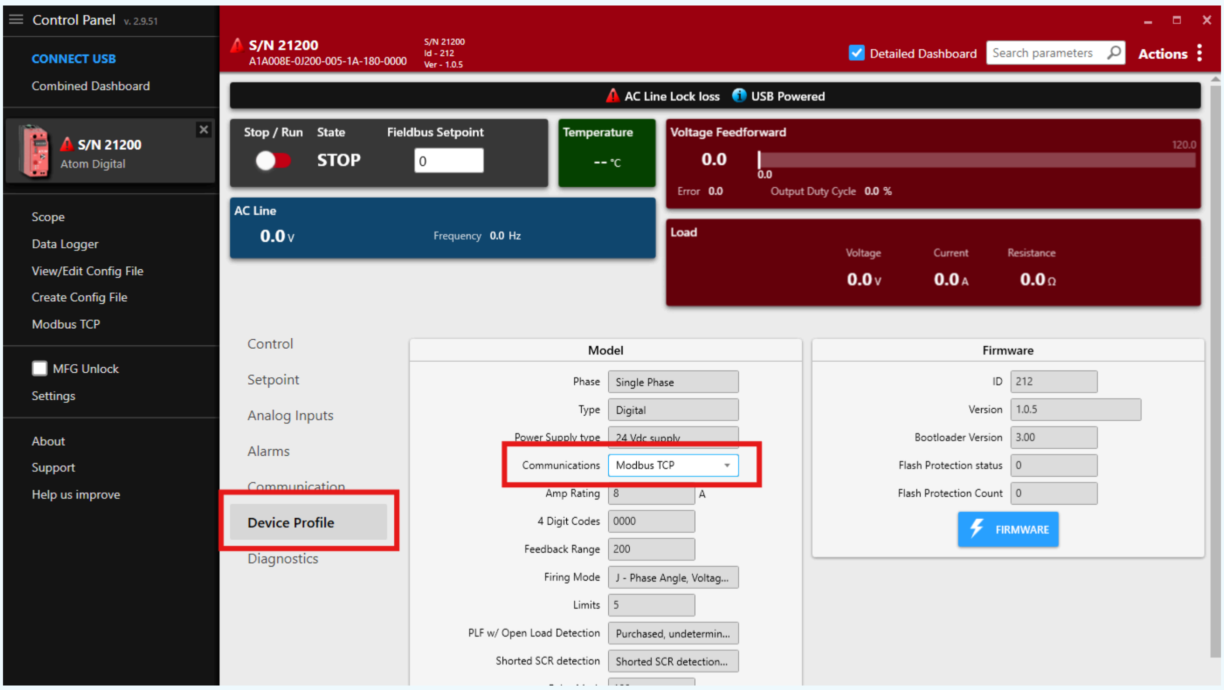


INFO

ATOM can be ordered in the following configurations:

- Single phase (`1PH`)
  - Analog - Fieldbus not supported
  - Digital - ModbusTCP, EtherNet/IP, or Profinet
  - EtherCAT - EtherCAT only
- Three phase (`3PH`)
  - Analog - Fieldbus not supported
  - Digital - ModbusTCP, EtherNet/IP, or Profinet

ATOM *digital* units can be configured for **ModbusTCP**, **EtherNet/IP**, or **Profinet** in software. Change the active fieldbus protocol in the **Device profile** tab in Control Panel. Changes to the active fieldbus take effect immediately — no reboot required.



INFO

Learning resources

https://www.plcacademy.com

ATOM / Fieldbus / EtherNet/IP / Overview

INFO

ATOM is ODVA EtherNet/IP CT19 Conformant.

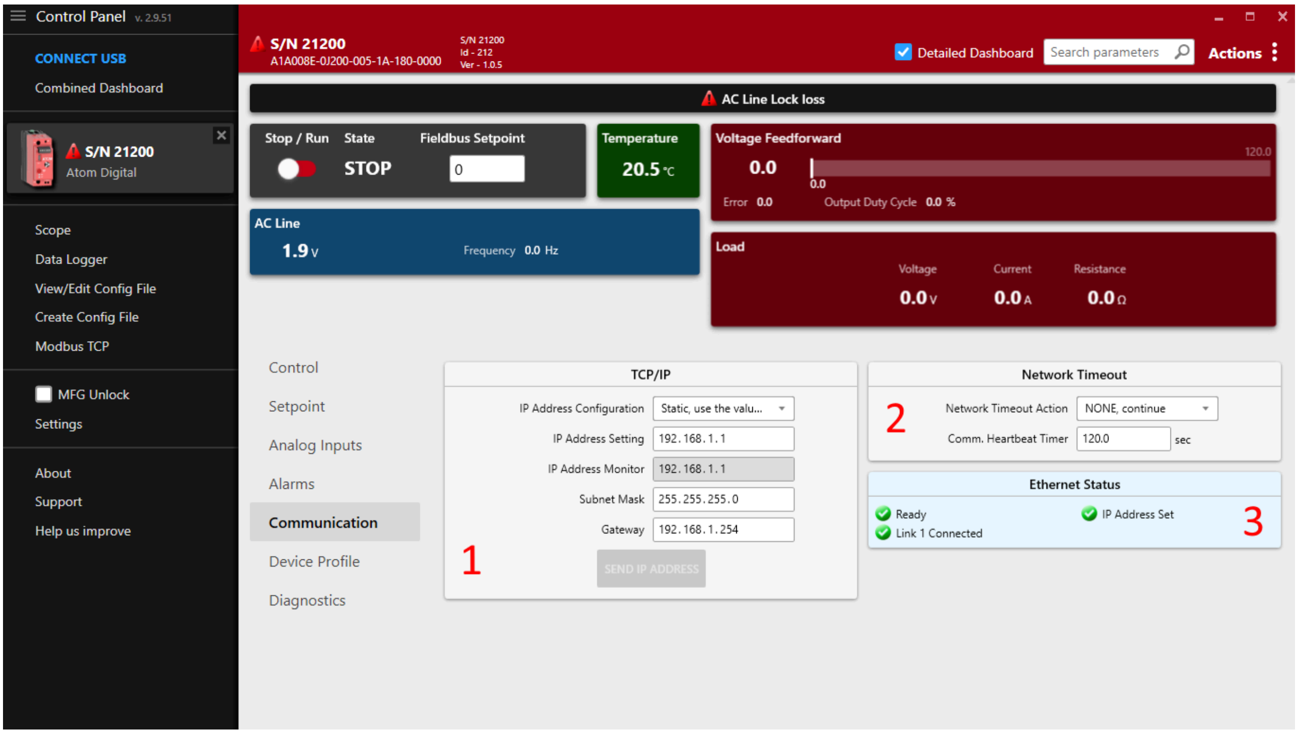
- Statement of Conformance
- Declaration of Conformance
- Passing Test Report
- ODVA listing

EDS

INFO

Download the EDS file for ATOM **here**.

Control Panel Communication Settings



Some communication settings can be configured in the **Communication** tab in Control Panel.

- Section `1`: TCP/IP settings
  - **IP Address Configuration**
    - `Static`: Use the IP address, subnet mask, and gateway specified below.
    - `DHCP`: Use DHCP to obtain an IP address.
  - **IP Address Setting**: The IP address of the ATOM controller.
  - **IP Address Monitor**: The current IP address of the ATOM controller.
  - **Subnet Mask**: The subnet mask of the ATOM controller.
  - **Gateway**: The gateway address for the ATOM controller.
- Section `2`: Network Timeout
  - The EtherNet/IP heartbeat timeout (Encapsulation Inactivity Timeout) in seconds.
  - You can configure a network timeout action to perform when the device loses communication with the PLC:
    - `None`: Do nothing

- `STOP, fault shutdown`: STOP the controller, disabling output
- `Use network timeout setpoint`: Configure an alternative setpoint to use when the controller loses communication with the PLC.

- Section `3`: Ethernet status
  - Indicates the status of both RJ45 ports, IP address configuration, conflict detection, and any other errors with the EtherNet/IP connection.

> **INFO**
>
> ### Control Panel and PLC software
>
> These settings are synchronized with your PLC environment. You do not have to use Control Panel to change these settings - you can stay in your PLC software. Control Panel merely provides them as an altnerative way to configure ATOM's EtherNet/IP settings.
>
> You can use Control Panel simultaneously with your PLC software without issues.

> **WARNING**
>
> ### IP Address Conflict Detection
>
> ATOM uses **IP Address Conflict Detection** to detect IP address conflicts on the network. If ATOM detects another device using the same IP address, it will disable all network communication until the conflict is resolved.
>
> Please ensure all devices on the network are assigned unique a IP address.

Hardware considerations

WARNING

Daisy chaining

As ATOM has two RJ45 ports, it can be easily daisy-chained. When daisy-chaining ATOM, take care to avoid a loop in the network. In some loop configurations, ATOM is susceptible to network broadcast storms, which can cause the controller to become unresponsive. If you are daisy-chaining ATOM, ensure that the network is loop-free.

ATOM works with both unmanaged and managed switches. We recommend a managed switch for larger networks to give you more control over the network topology.

Parameters

Overview

ATOM makes 30 parameters accessible to EtherNet/IP. These parameters are made available through the CIP Assembly Object (code `0x04`) and a custom ParameterLink object (code `0x64`). The assembly object is most commonly used to read and write parameters from a PLC. The ParameterLink object is a custom object defined by Control Concepts that can be used to individually control parameters and is less commonly used.

Output Assembly (Class `0x04`, Instance `0x01`)

#	Name	Type	Description	Read/Write
1	Digital setpoint	DINT	A value between 0 and 10,000 indicating the desired output current. The value is scaled to the output range of ATOM. For example,	Read/Write

#	Name	Type	Description	Read/Write
			if the output range is 0-100A, a value of 5000 would set the output to 50A.	
2	Digital run enable	BOOL	Enables or disables the output current. When disabled, the output current is set to 0A.	Read/Write

Input Assembly (Class `0x04`, Instance `0x02`)

#	Name	Type	Description	Read/Write
3	Inhibit Alarm Status	BYTE	A bitfield indicating alarms that are preventing controller operation. See Inhibit Alarm Status.	Read
4	Warning Alarm Status	BYTE	A bitfield indicating warning alarms. See Warning Alarm Status.	Read
5	Feedback Read Status	BOOL	A bitfield indicating if controller has acquired feedback. See Feedback Read Status.	Read
6	AC Line Frequency	REAL	The AC line frequency in Hz.	Read
7	AC Line Voltage	REAL	The AC line voltage in volts.	Read

#	Name	Type	Description	Read/Write
8	Load Voltage	REAL	The load voltage in volts.	Read
9	Load Current	REAL	The load current in amps.	Read
10	Load Resistance	REAL	The load resistance in ohms.	Read
11	Heatsink Temperature	REAL	Heatsink temperature, in degrees celsius.	Read
12	Output Duty Cycle %	REAL	Indicates the amount, in percent, that the output of the controller is ON	Read
13	Setpoint reference	REAL	Reference input to control compensation loop in units determined by "feedback type"	Read
14	Feedback	REAL	The control output supplied to the load in units determined by "feedback type"	Read
15	Partial Load Fault Target Resistance	REAL	Expected nominal resistance, in Ohms, of the load. Used for partial load fault detection.	Read
16	Partial Load Fault Resistance	REAL	The actual load resistance in Ohms. Compared with #15 to determine if a partial load fault has occurred.	Read

#	Name	Type	Description	Read/Write
17	Partial Load Fault Resistance Deviation	REAL	The tolerable percentage that parameter #15 and #16 may differ by until a partial load fault will be triggered.	Read
18	Firmware ID	DINT	Indicates the version of firmware that is loaded, dictating which features are available.	Read
19	Firmware major revision	DINT	Indicates which revision of the firmware is loaded. Major revisions fix critical bugs or add significant new features.	Read
20	Firmware minor revision	DINT	Indicates which minor revision of the firmware is loaded. Minor revisions fix minor issues and/or add minor improvements.	Read
21	Full Scale Voltage	DINT	The expected output voltage when the controller output is fully on.	Read
22	Full Scale Current	REAL	The expected current when the controller output is fully on.	Read
23	AC Line Status	BYTE	A bitfield indicating the status of the connected AC Line. See AC Line Status.	Read

#	Name	Type	Description	Read/Write
24	Load Status	BYTE	A bitfield indicating the load status. See Load status.	Read
25	Controller Status	BYTE	A value indicating the operational status of the controller. See Controller status.	Read
26	Controller State	BYTE	A value indicating the controller state. See Controller state.	Read
27	EEPROM Status	WORD	A bitfield indicating the EEPROM status. Seee EEPROM Status.	Read
28	EEPROM Status 2	WORD	Identical to parameter #27	Read
29	Error Latch	BYTE	A bitfield used for diagnostic troubleshooting. See Error Latch.	Read
30	Miscellaneous Status	BYTE	A bitfield indicating miscellaneous status information. See Miscellaneous Status.	Read

Additional parameter descriptions

Inhibit Alarm Status

Inhibit alarm status is a 8-bit bitfield:

7	6	5	4	3	2	1
Reserved	Reserved	Reserved	Reserved	Feedback Loss	Over Temperature	Over Current Trip

If any bit is set to 1, the controller will *not* be allowed to run.

Warning Alarm Status

Warning alarm status is a 8-bit bitfield:

7	6	5	4	3	2	1	0
Reserved	Reserved	High temperature	Shorted SCR	Open Load	Partial Load Fault	Current Limit	Voltage Limit

Warning alarms are not considered critical and will not prevent the controller from running.

Feedback Read Status

Feedback status is a 8-bit bitfield:

7	6	5	4	3	2	1	
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Ti

Indicates whether the controller has acquired feedback on the line. If any bit is set to 1, then the controller has lost feedback.

AC Line Status

AC Line status is a 8-bit bitfield:

7	6	5	4	3	2	1	0
Reserved	Reserved	Sync-Locked (to AC Line)	Pre-Lock 2	Pre-Lock 1	Reserved	AC Line B OK	AC Line A OK

Bits 5 must be set to 1 before the controller can provide power to the load.

Load Status

Load status is a 8-bit bitfield:

7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	Open Load	Reserved	Reserved	Reserved	Short SCR

Controller Status

Controller status is one of:

Value	Description
0	Disabled
1	Initialization
2	Normal, operating

Value	Description
3	Calibration
4	Diagnostic

Controller State

Controller state is one of:

Value	State	Description
0	STOP	The state the controller is in when AC Line voltage is not present.
1	RUN	The state the controller is in when AC Line voltage is present and the controller is synchronized to the AC line.
2	FAULT	A latching state of output shutdown caused by over current or over temperature alarms. A power cycle or processor reset is required to clear this state.
3	FAULT RESET	Used as a temporary state to transition from FAULT to RUN once again.

EEPROM Status

EEPROM status is an 16-bit bitfield. EEPROM is used to store controller configuration and calibration data. Any errors in EEPROM may indicate that the firmware is corrupted.

Bit	Description
0	EEPROM Initialization
1	SP Table Error
2	MFG CP Table Error
3	Calibration Table Error
4	Reserved
5	Reserved
6	Backup Calibration Table Error
7	Bottom Board Calibration Table Error
8	SP Definition Table needs updating
9	Bottom Board Calibration Backup Error
10	Reserved
11	Reserved
12	EEPROM is write protected
13	Reserved
14	Reserved

Bit	Description
15	Feedback Calibration Table has changed, store to EEPROM

Error Latch

Error latch is a 8-bit bitfield:

7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	Feedback loss	SCR timing loss	Line Frequency failure	Phase loss or missing cycle	Line Lock Loss

Error latch is provided as a diagnostic troubleshooting aid.

Miscellaneous Status

Miscellaneous status is an 8-bit bitfield:

7	6	5	4	3	2	1
Reserved	Initialization in progress	Reserved	Reserved	Waiting for ENTER key during initialization	Reserved	USB Powered

Data types

The data types listed above in the parameter table are defined in the CIP standard as:

Type	Size	Description
BOOL	1 byte	Boolean value
BYTE	1 byte	8-bit bitmap
WORD	2 bytes	16-bit bitmap
DWORD	4 bytes	32-bit bitmap
LWORD	8 bytes	64-bit bitmap
USINT	1 byte	Unsigned 8-bit integer
UINT	2 bytes	Unsigned 16-bit integer
UDINT	4 bytes	Unsigned 32-bit integer
ULINT	8 bytes	Unsigned 64-bit integer
SINT	1 byte	Signed 8-bit integer
INT	2 bytes	Signed 16-bit integer
DINT	4 bytes	Signed 32-bit integer
LINT	8 bytes	Signed 64-bit integer
REAL	4 bytes	32-bit floating point number
LREAL	8 bytes	64-bit floating point number

> **INFO**
>
> Rockwell's RSLogix Studio 5000 does not support unsigned integers. Any EDS file that contains unsigned integers will cause issues when it is imported into Studio 5000. To avoid this issue, ATOM uses signed integers for all integer types, regardless of whether the value may be negative or not. For example, parameter #1, *Digital setpoint* is represented as a signed 32-bit integer, but it may never be negative.

Other resources

> **INFO**
>
> Detailed information about ATOM's EtherNet/IP profile is also available as a **downloadable word document**.

Advanced

ATOM has many more parameters beyond the 30 made available through EtherNet/IP. The default profile listed above should be sufficient for the majority of use cases.

If this is not the case, you can use Control Panel to adjust or monitor all parameters.

In the rare case that you need more parameters available through ATOM's EtherNet/IP profile, Control Concepts does have the ability to make additional parameters available or to change the data type of included parameters. Please contact us if you would like a custom EtherNet/IP profile. There may be a service fee for custom EtherNt/IP profiles as they require new EDS files, device-reconfiguration and testing.

ATOM / Fieldbus / EtherNet/IP / RSLogix Studio 5000

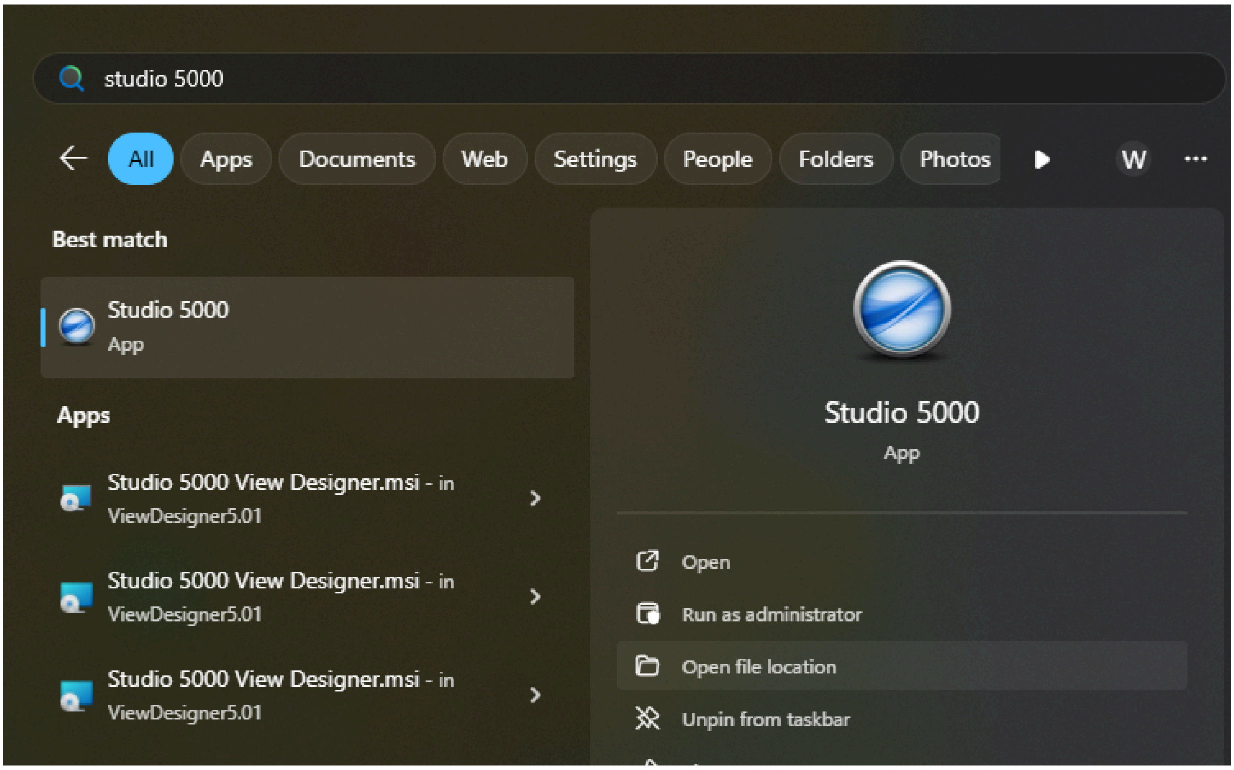
Overview

In this tutorial, you'll learn how to control ATOM over EtherNet/IP with RSLogix Studio 5000.

> **NOTE**
>
> If you'd like to skip this tutorial, download the completed example project: AtomExampleStudio5000.zip.

Prerequisites

1. A PC with RSLogix Studio 5000 installed (See Installation Troubleshooting for help with installation issues):

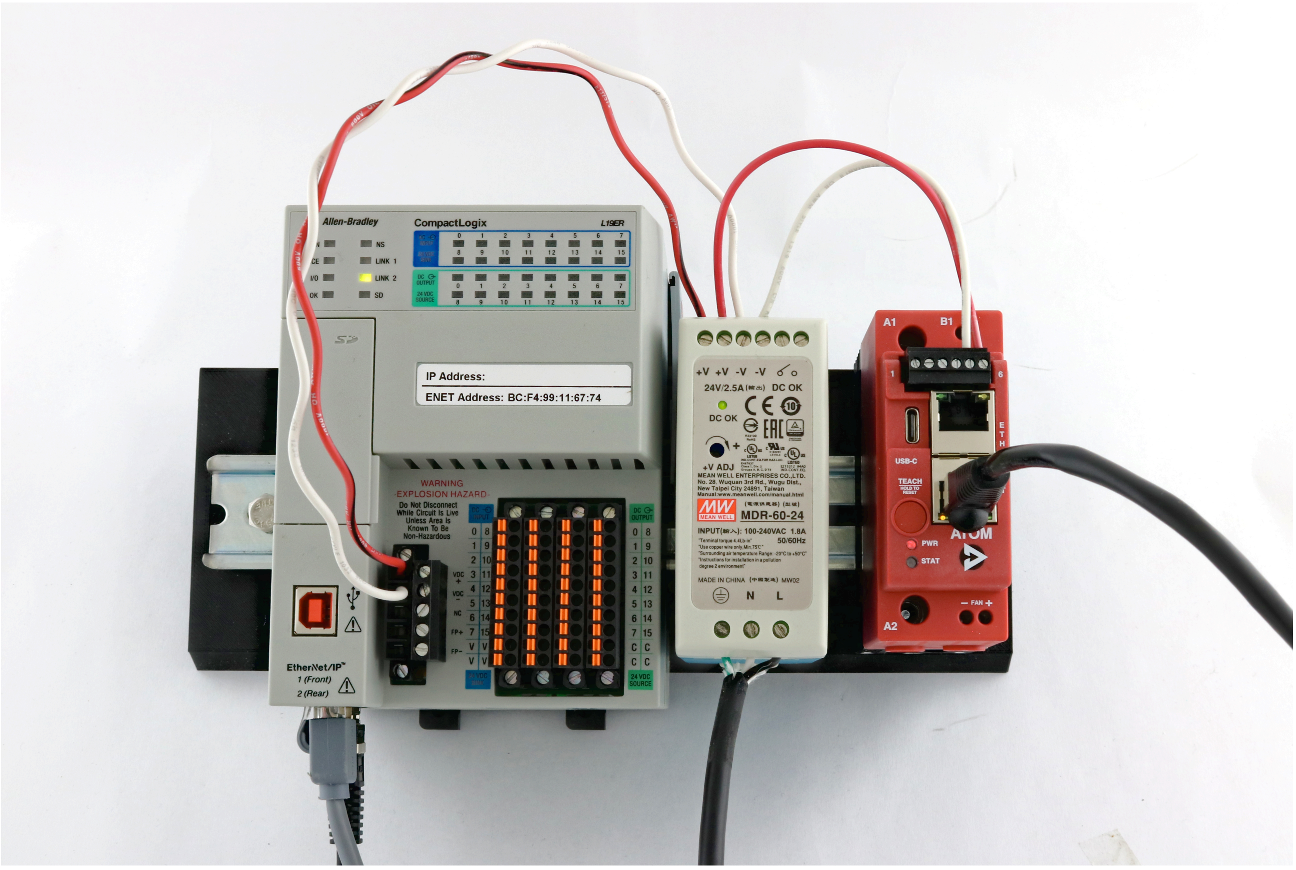


2. A Logix PLC - a CompactLogix `1769-L19ER-BB1B` is used in this example, but you can follow along with any Logix PLC that supports EtherNet/IP.
3. Download ATOM's EDS file: Atom.eds

Hardware Setup

Connections:

- Connect port `1 (Front)` on your PLC to your *PC*
- Connect port `2 (Rear)` on your PLC to *ATOM* (either port)
- Connect a 24VDC power supply to ATOM and your PLC

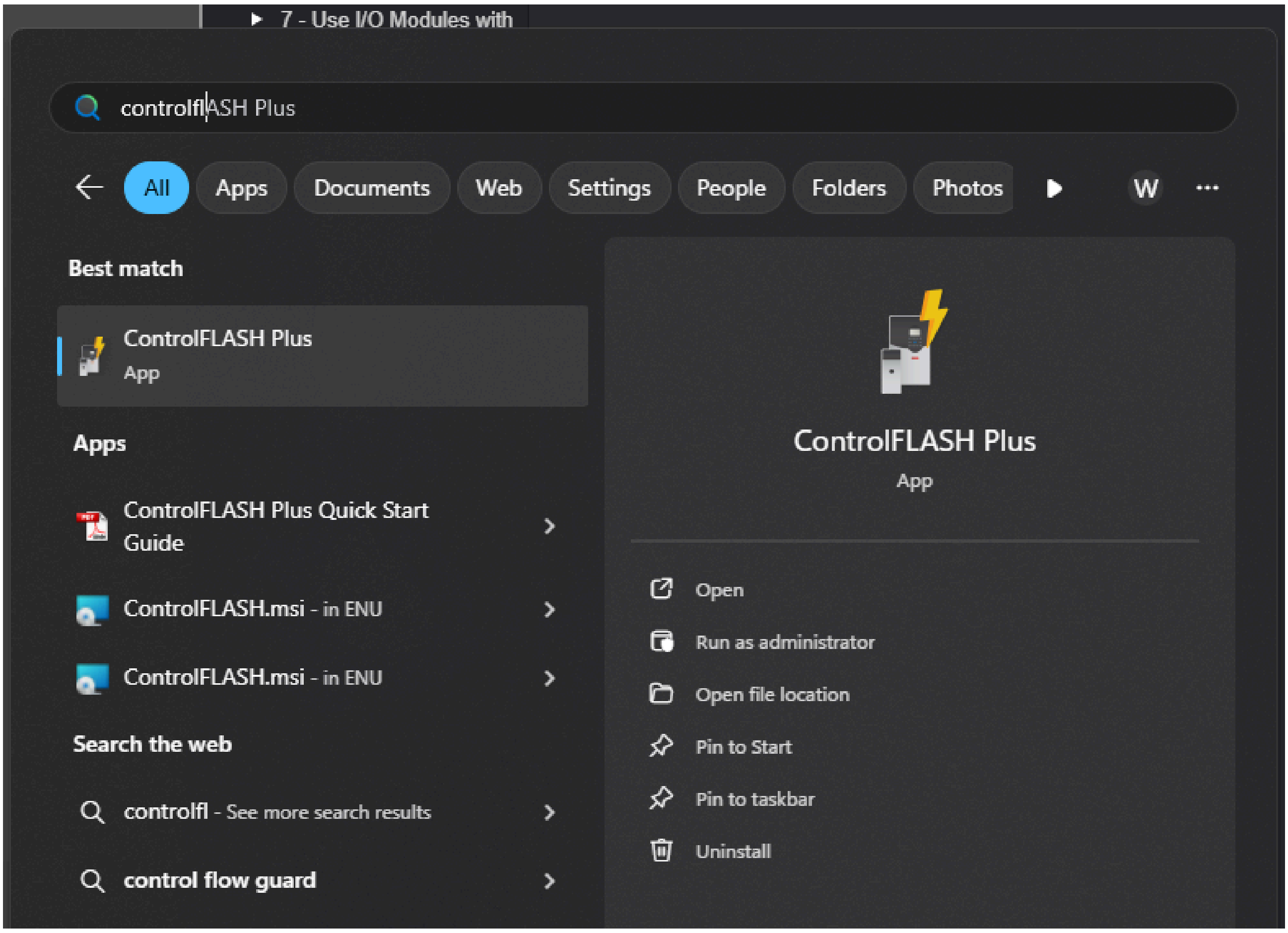


PLC Configuration

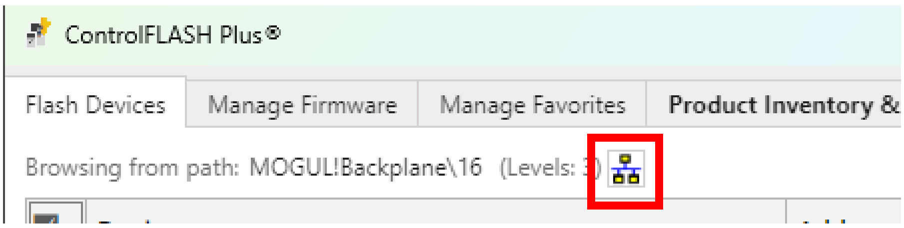
Upgrading firmware

Make sure to upgrade your PLC firmware to the lastest version to ensure compatibility with Studio 5000.

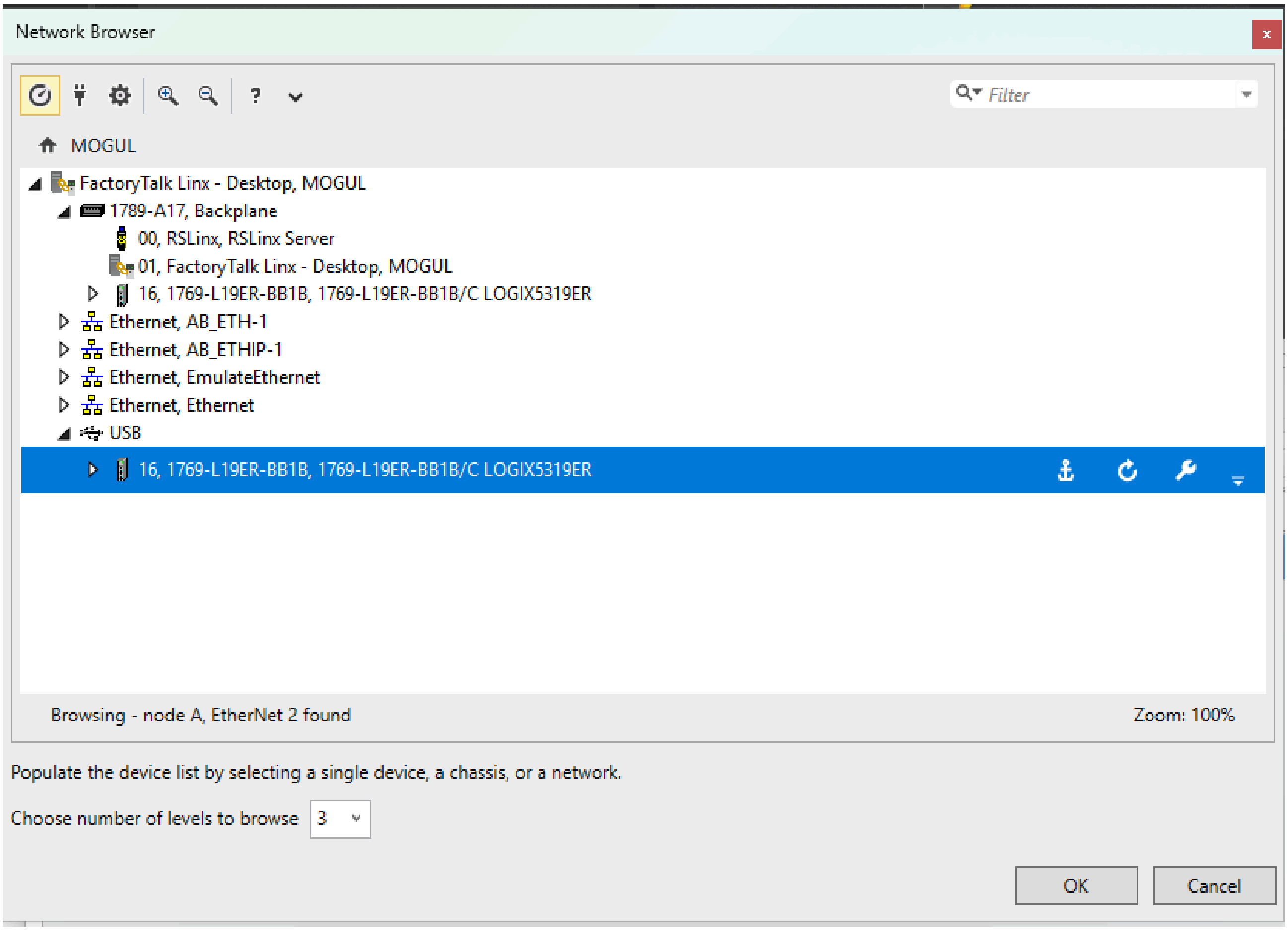
1. Connect your Logix PLC to your PC with a USB cable.
2. Launch `ControlFLASH Plus`:



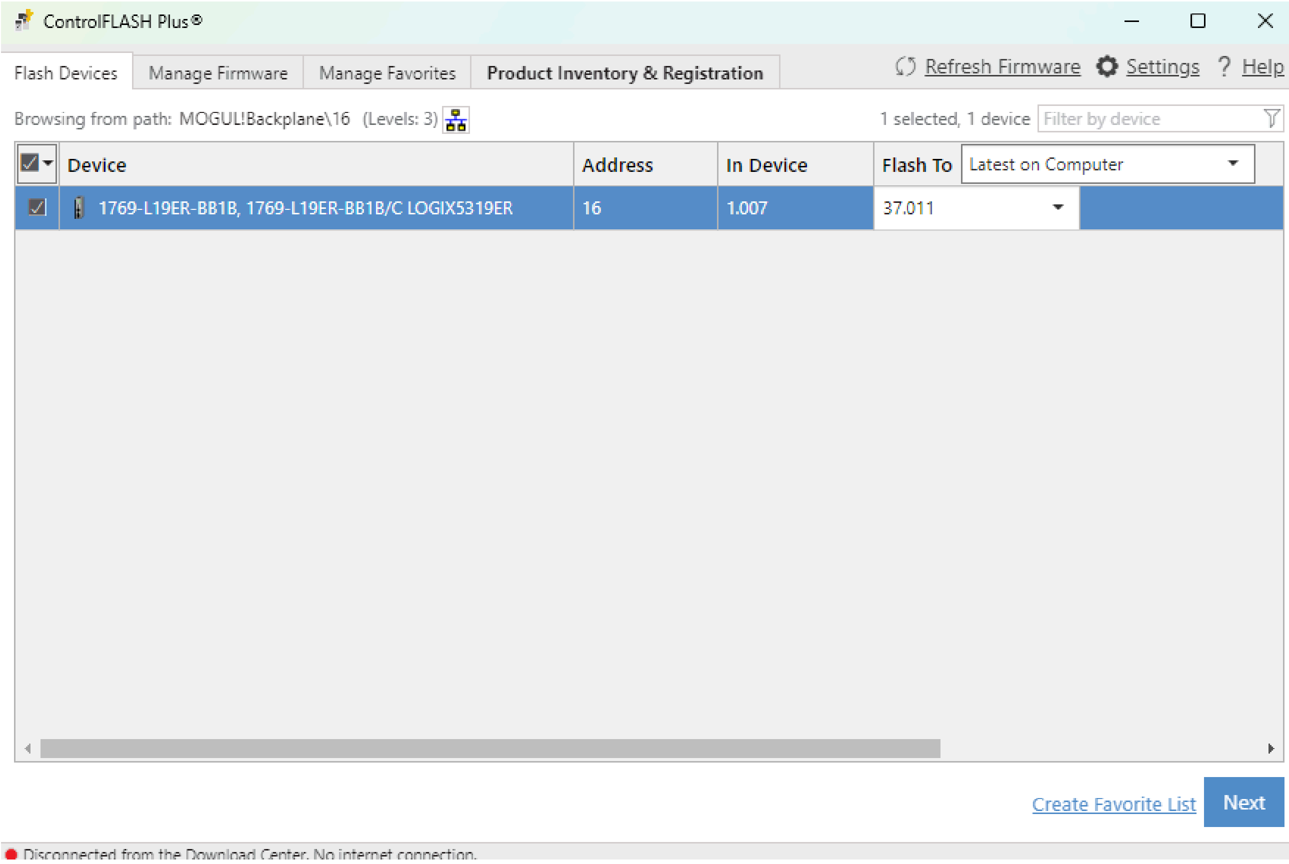
3. Open the network browser:



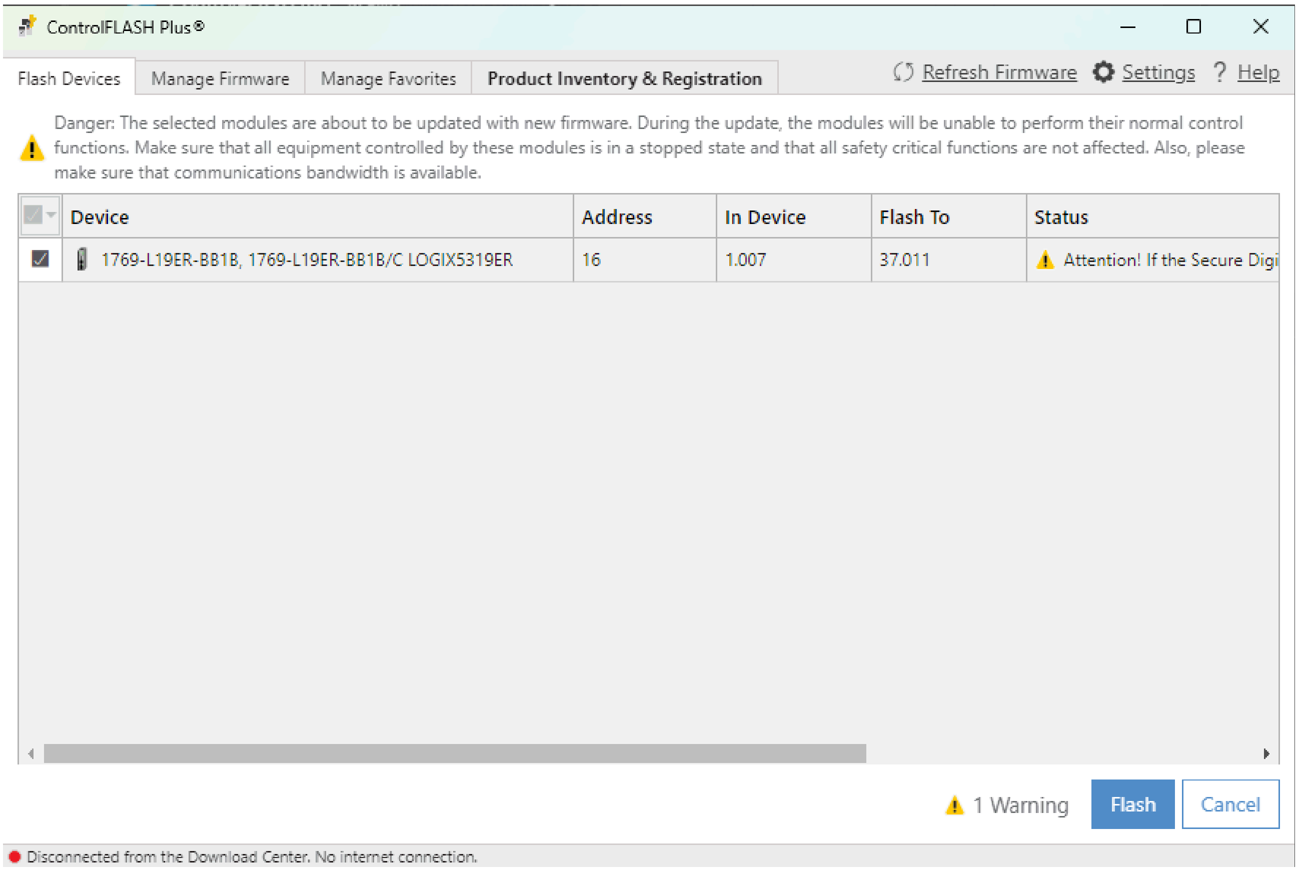
4. Select your PLC under the `USB` category:



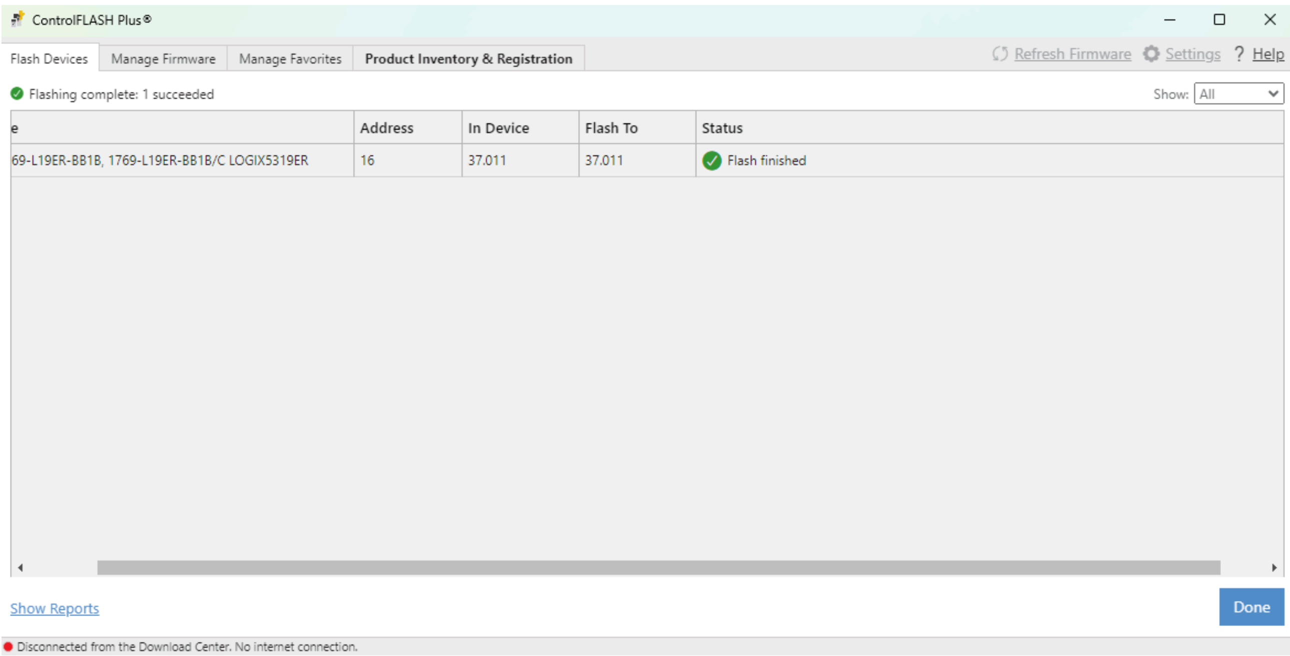
5. Select the device and latest firmware version, then click **Next**:



6. Click **Flash**:

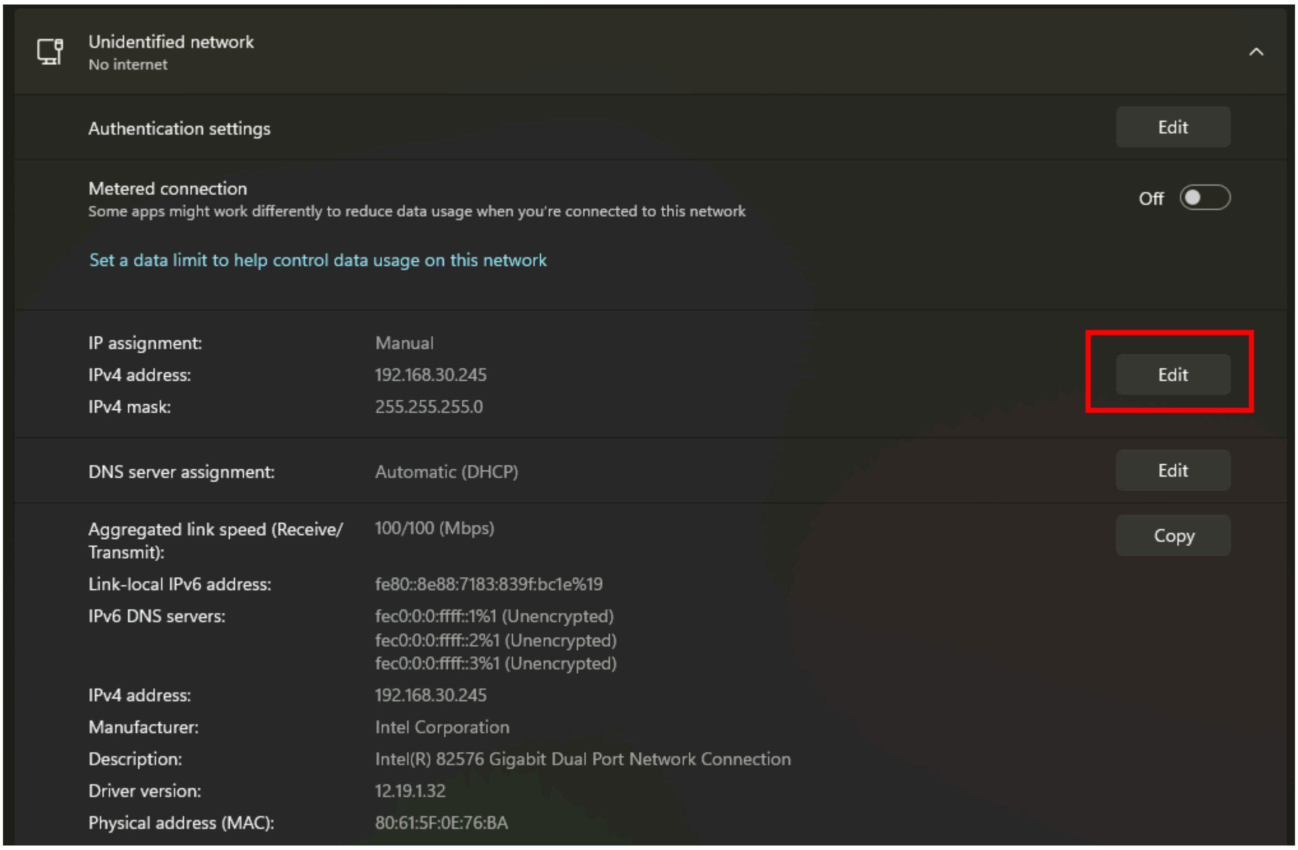


7. After flashing succeeds, reboot your PLC.



Configuring your PC's network settings

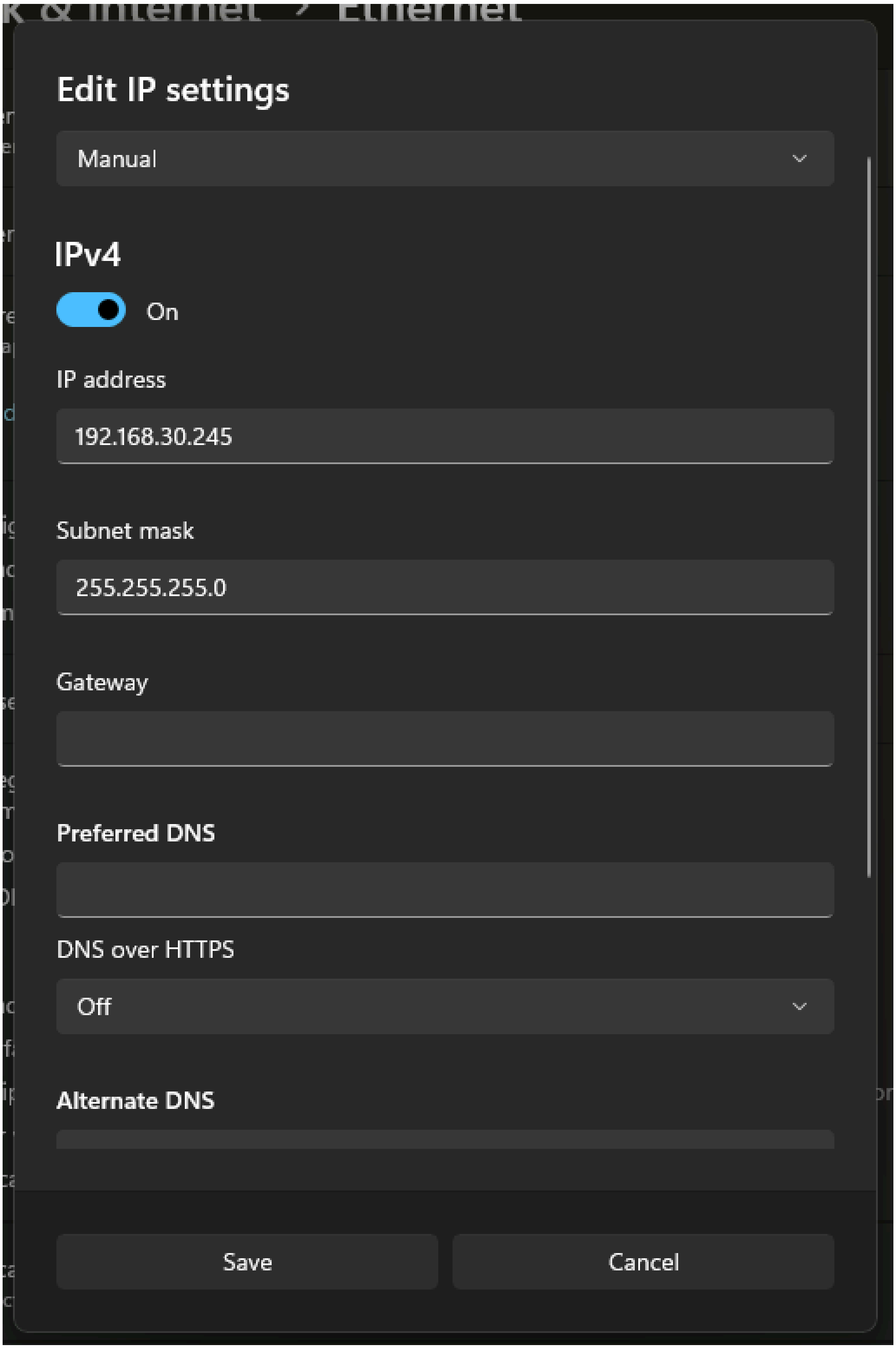
1. Open your PC's network settings and select edit on the Ethernet adapter connected to your PLC:



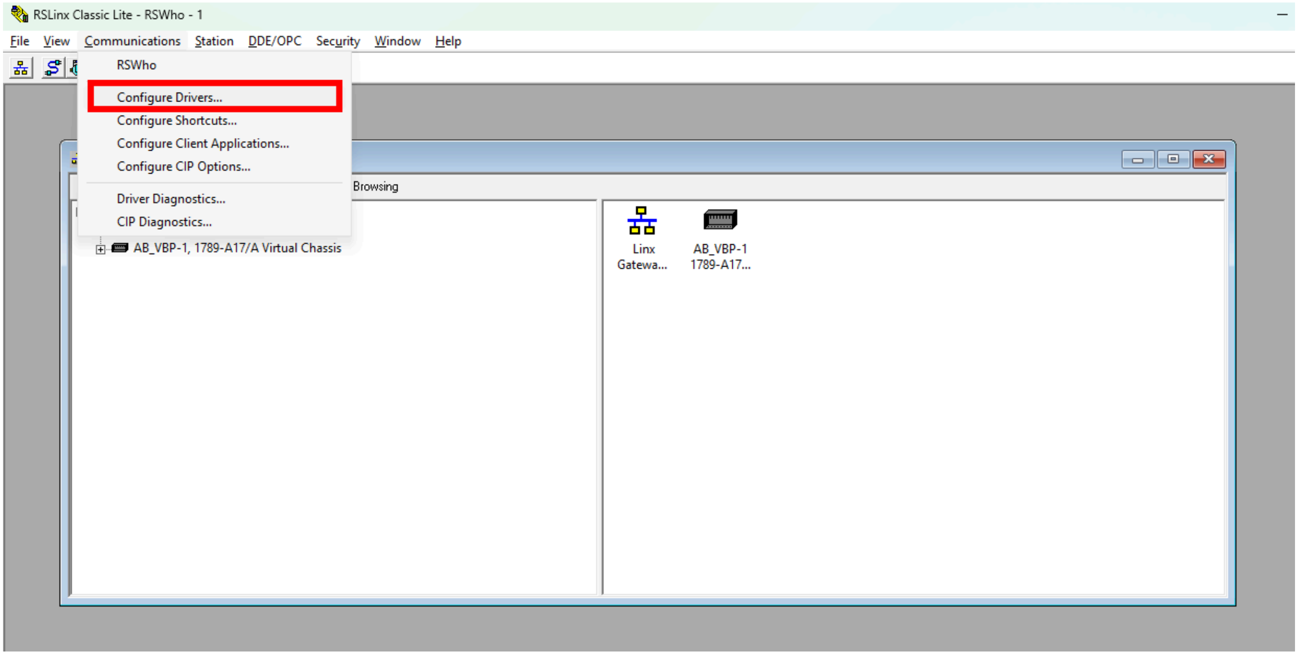
2. In the **Edit IP Settings** dialog, set:

   - **IP Address**: `192.168.30.245`
   - **Subnet Mask**: `255.255.255.0`

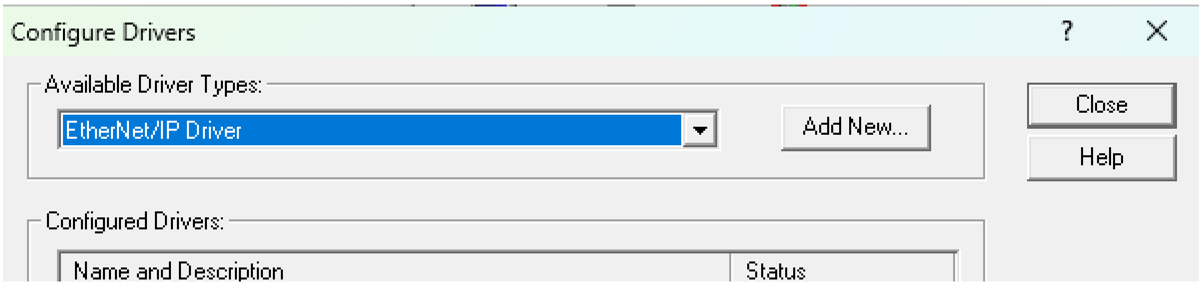
   Click **Save** to apply the settings.



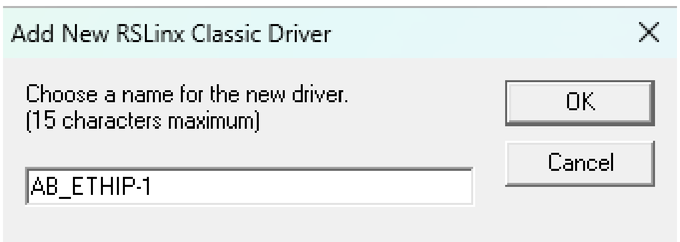
3. Open RSLogix Classic and select **Communications** > **Configure Drivers**:



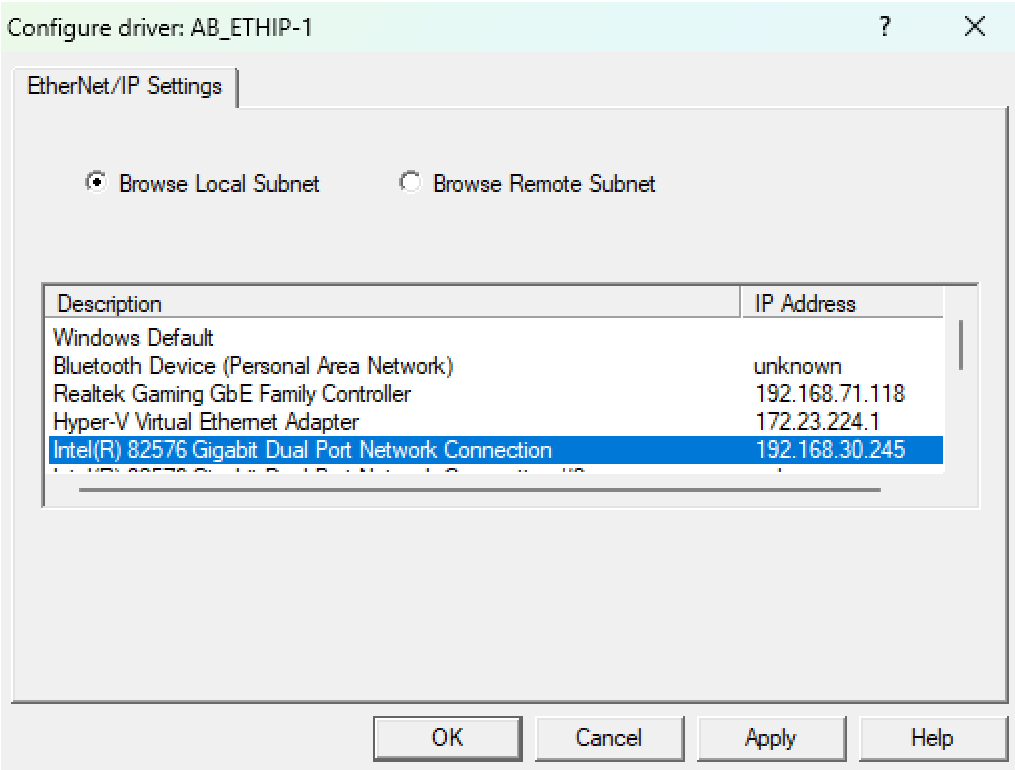
4. Select **EtherNet/IP Driver** and click **Add New**:



5. Click **OK** to add the driver:

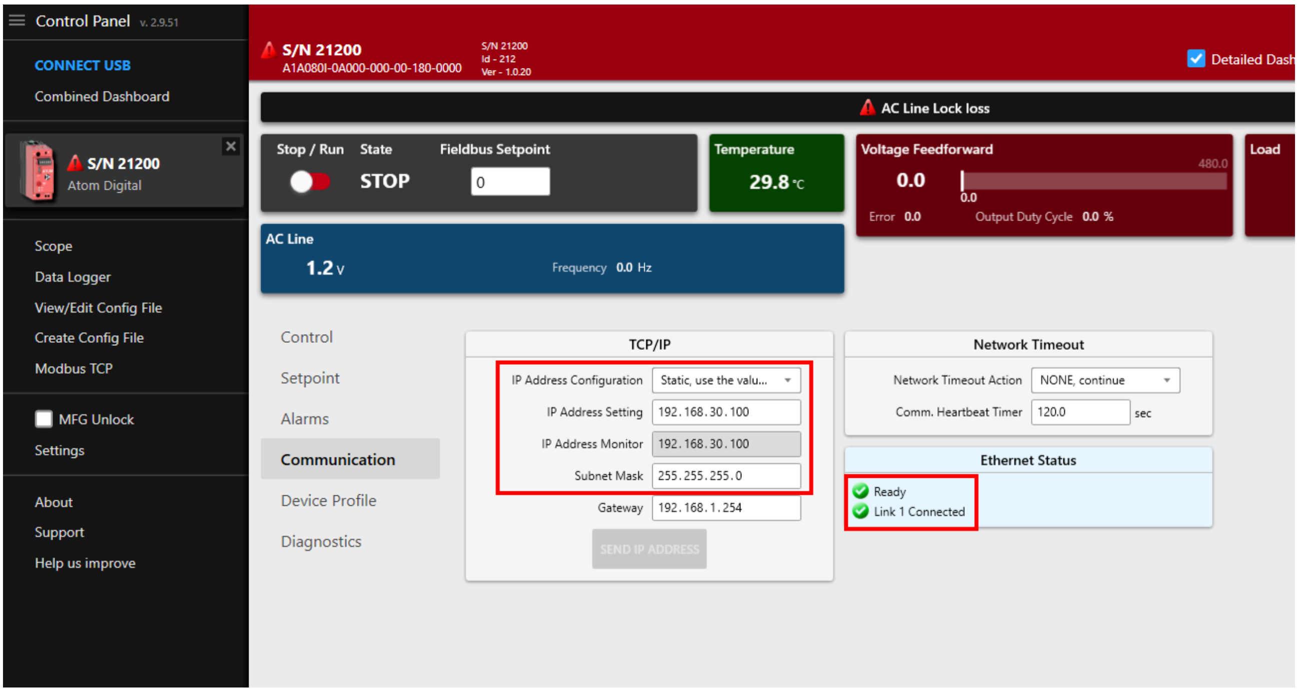


6. Select the adapter with IP address `192.168.30.245`, then hit **Apply** and **OK**:



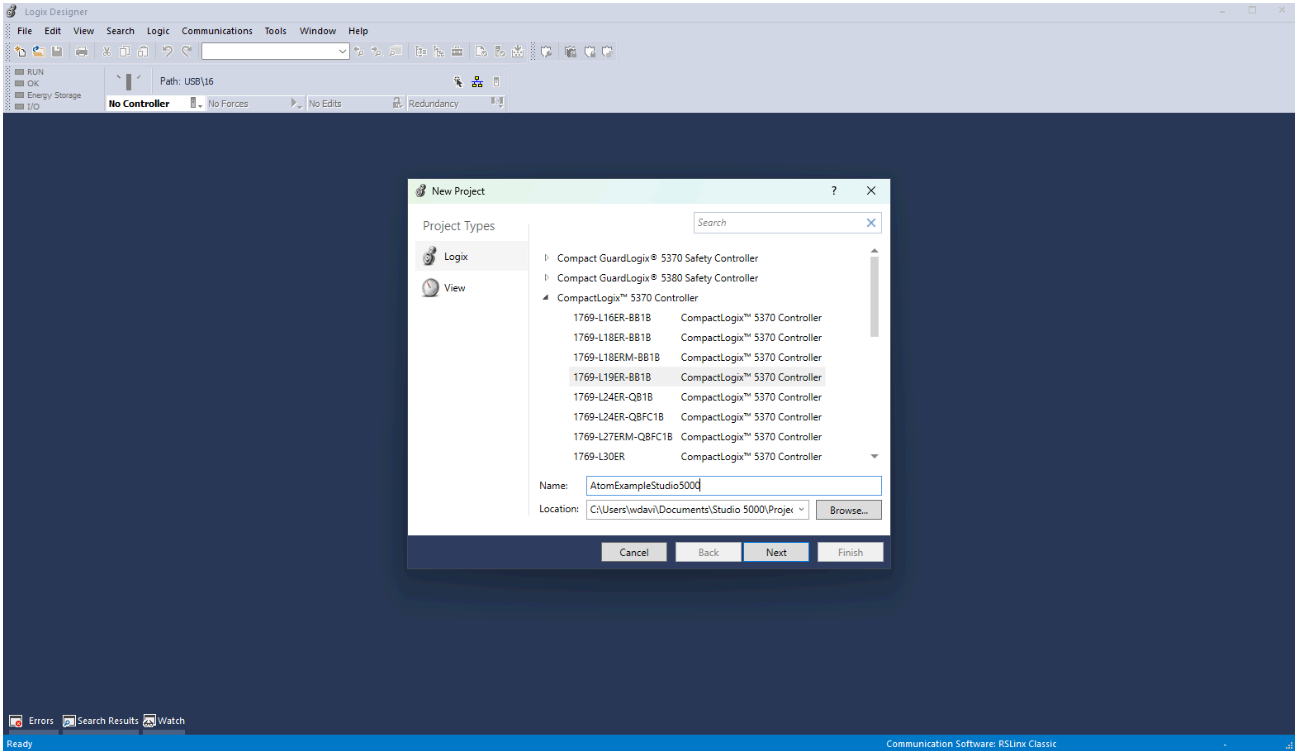
ATOM Configuration

1. Connect ATOM to your PC using a USB-C cable. Launch Control Panel and connect to your ATOM. In the **Network** tab, set the following:
   - **IP Address Configuration**: `Static`
   - **IP Address**: `192.168.30.100`
   - **Subnet Mask**: `255.255.255.0`

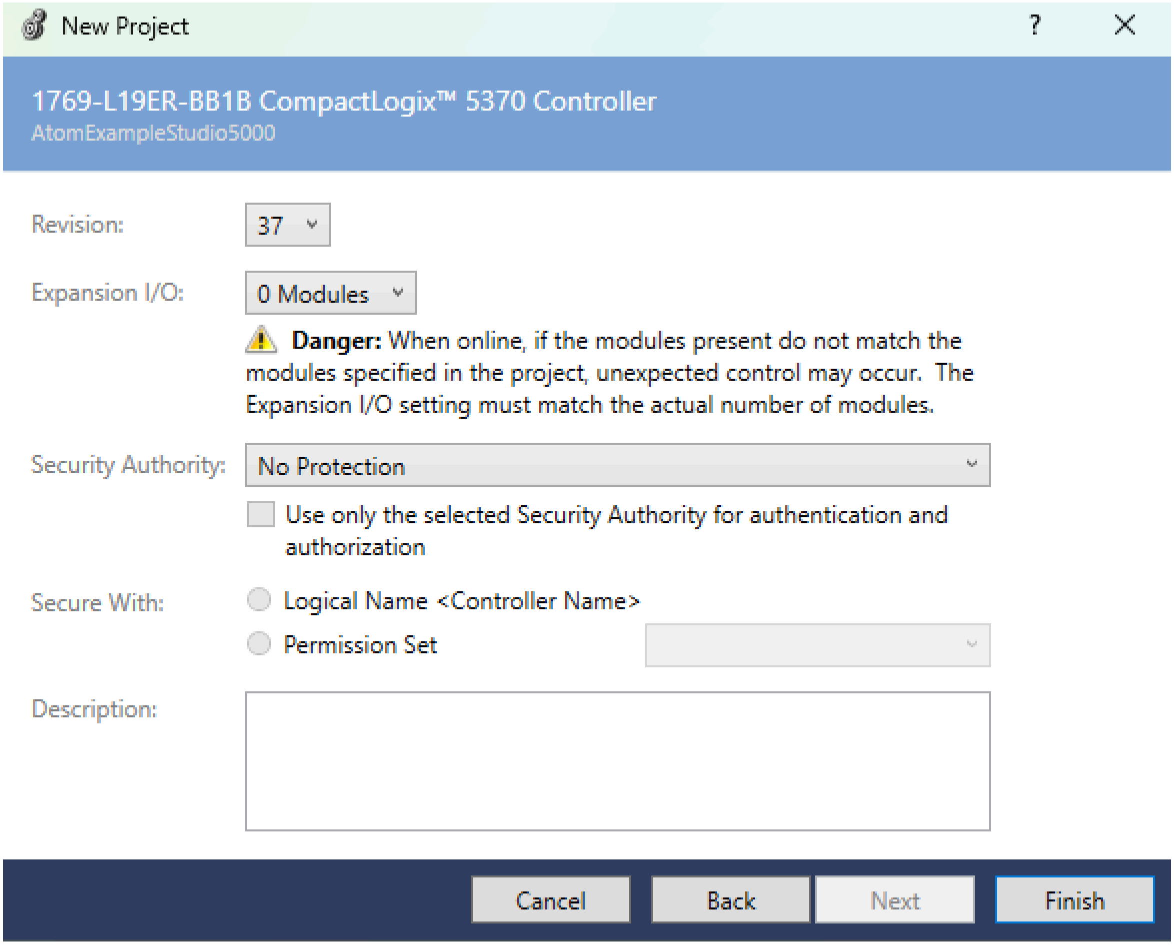


Create a Studio 5000 project and connect to your PLC

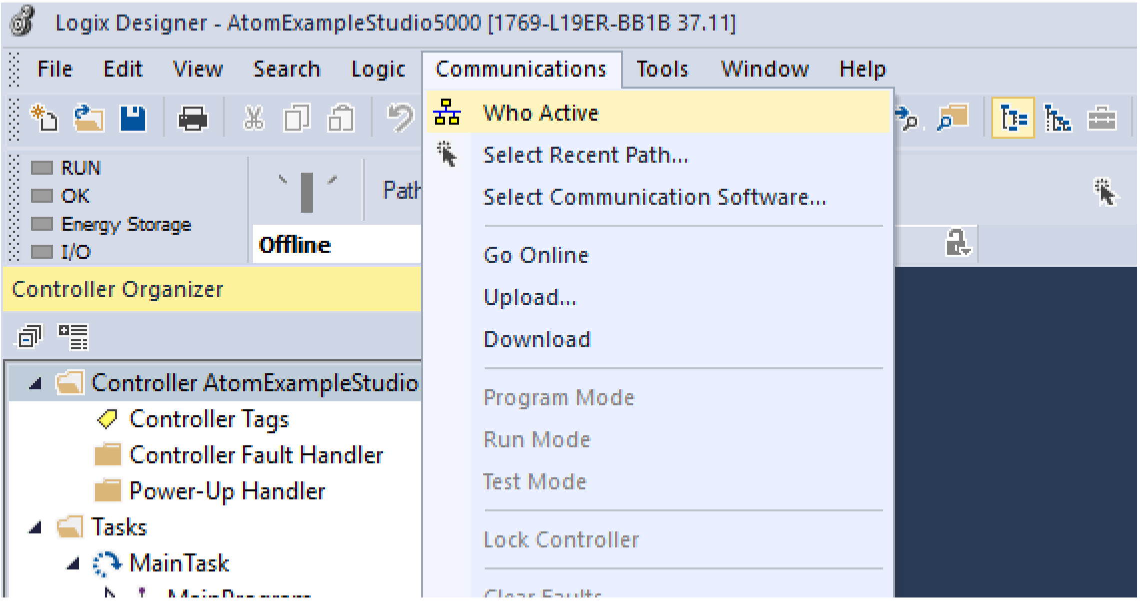
1. Launch Studio 5000, select **File** > **New Project**. Name the project `AtomExampleStudio5000` and select `1769-L19ER-BB1B` (CompactLogix 5370). Click **OK**:



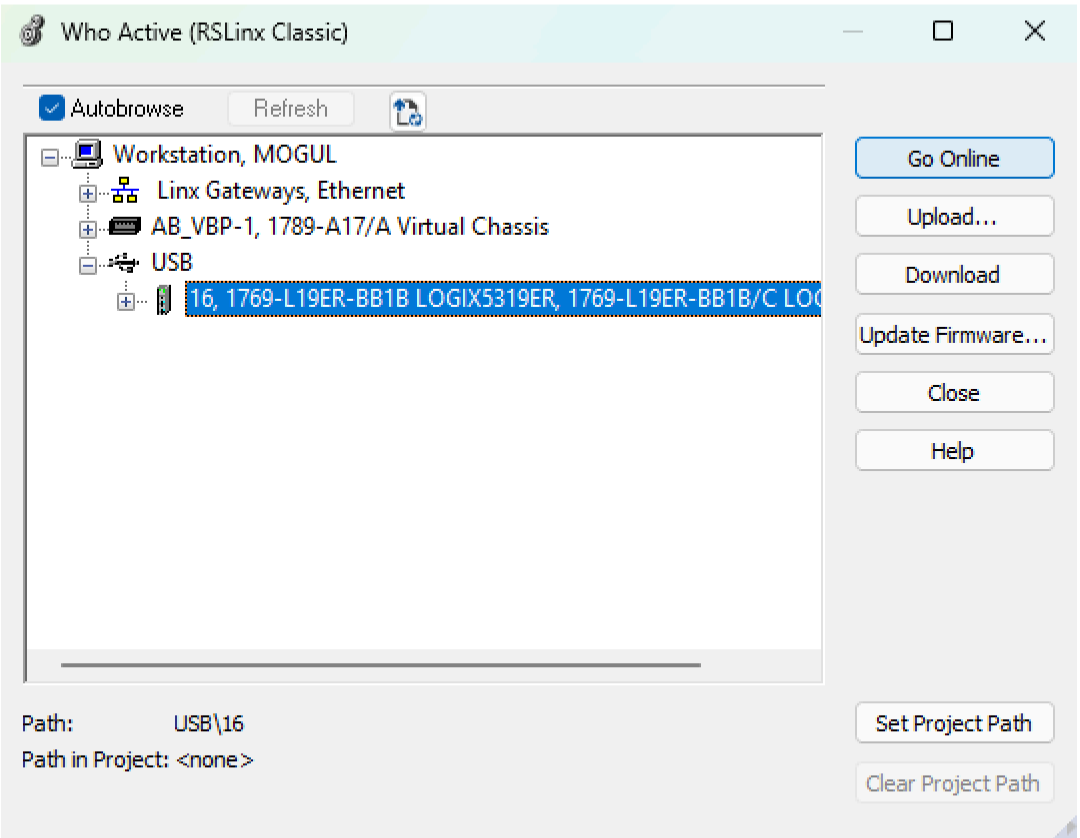
2. Select `0 Modules` under Expansion I/O, then click **Finish**:



3. Connect your PLC to your PC with a USB-B cable. In Studio 5000, select **Communications** > **Who Active**:

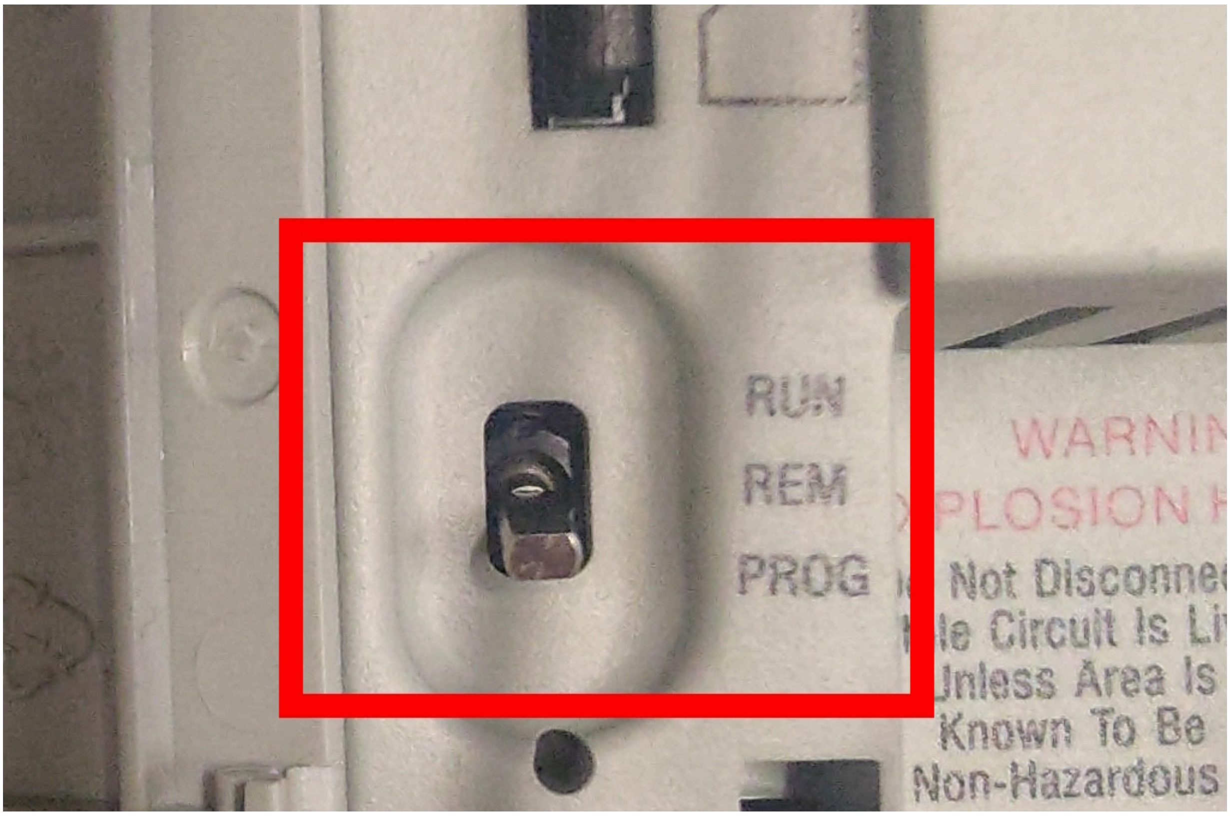


4. Select your PLC under the USB category and click **Go Online**:

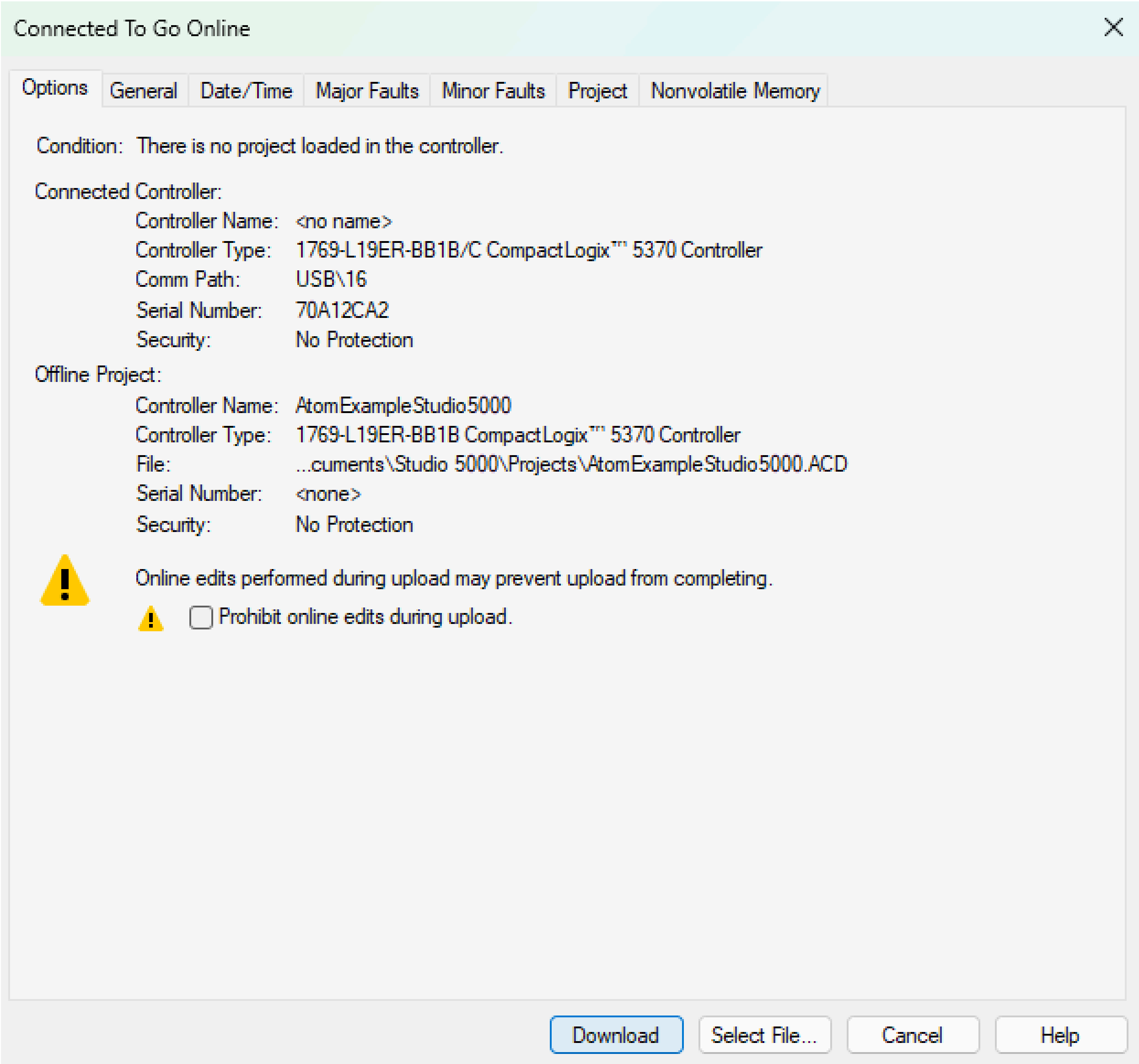


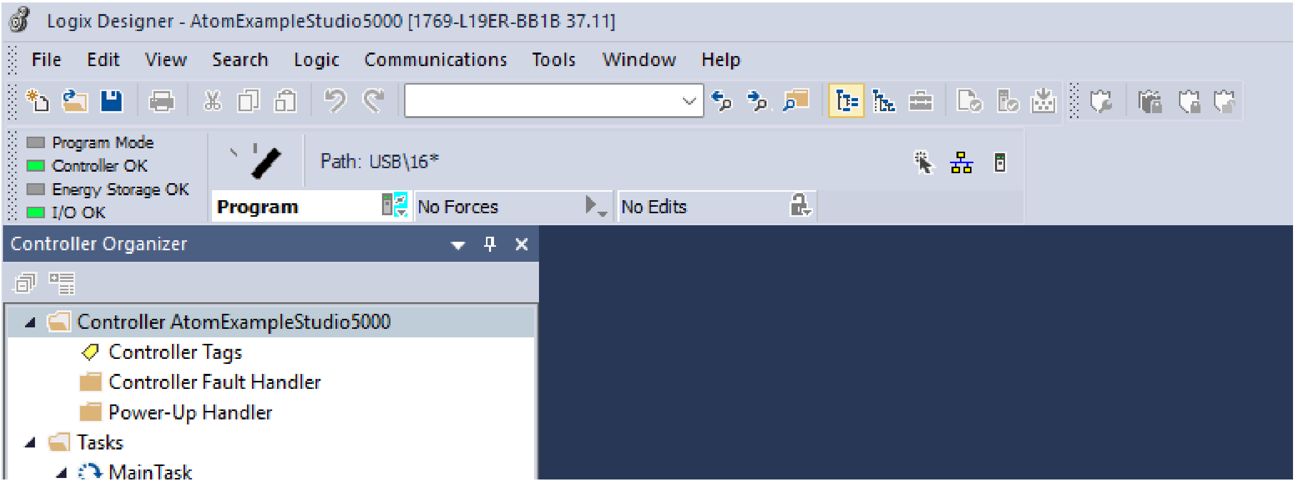
> Ensure the switch on your PLC is set to `PROG` mode before downloading.


Allen-Bradley
RUN
FORCE
I/O
OK
NS
LINK 1
LINK 2
SD

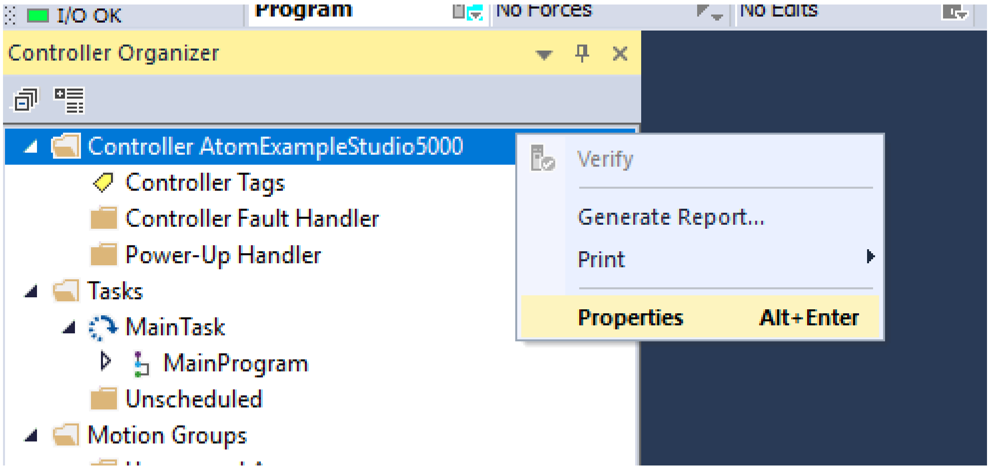


5. Select **Download** and double check that the `Controller OK` indicator light turns green:

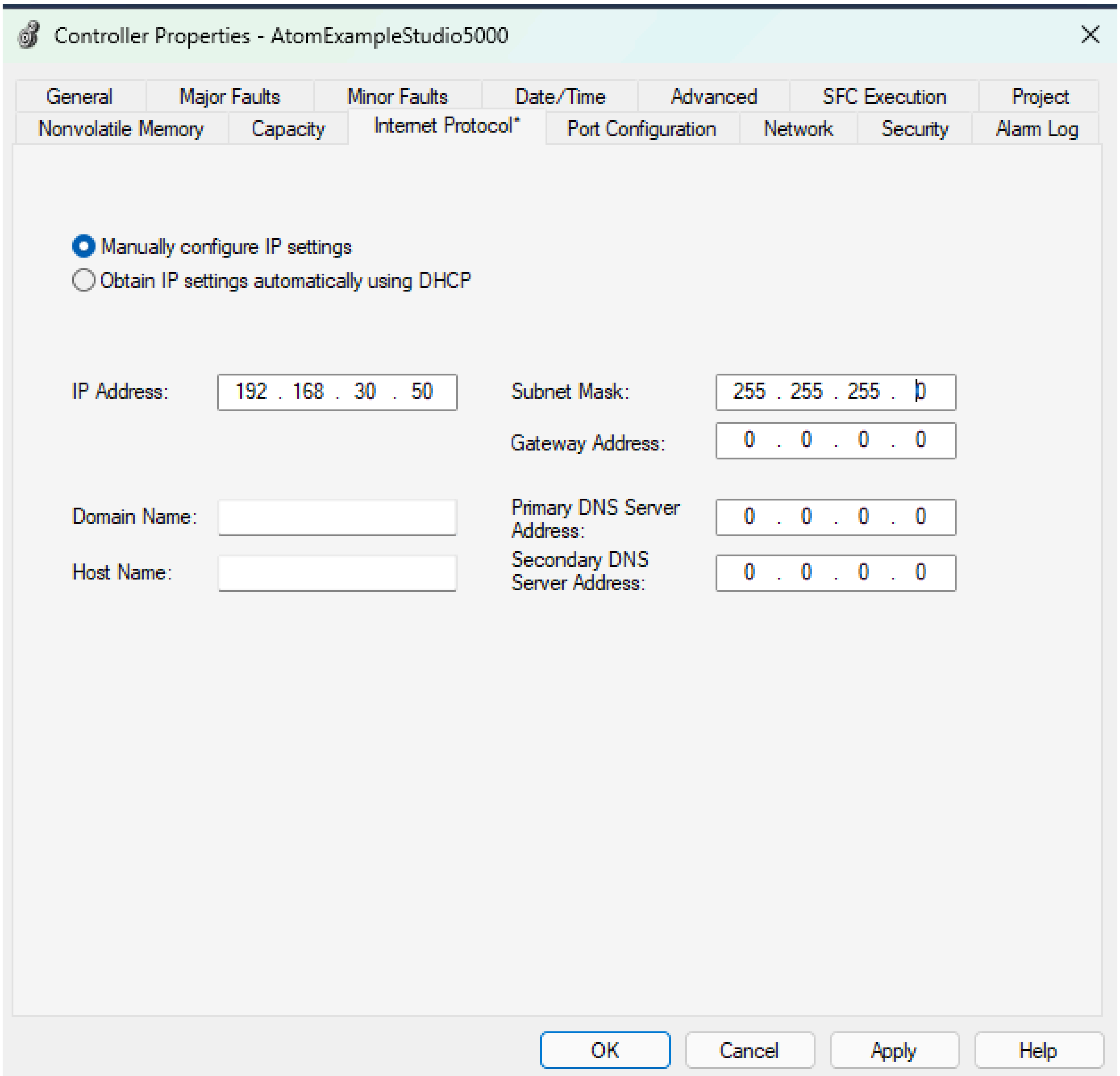

Connected To Go Online
Options
General
Date/Time
Major Faults
Minor Faults
Project
Nonvolatile Memory
Condition: There is no project loaded in the controller.
Connected Controller:
Controller Name: <no name>
Controller Type: 1769-L19ER-BB1B/C CompactLogix™ 5370 Controller
Comm Path: USB\16
Serial Number: 70A12CA2
Security: No Protection
Offline Project:
Controller Name: AtomExampleStudio5000
Controller Type: 1769-L19ER-BB1B CompactLogix™ 5370 Controller
File: ...cuments\Studio 5000\Projects\AtomExampleStudio5000.ACD
Serial Number: <none>
Security: No Protection
Online edits performed during upload may prevent upload from completing.
Prohibit online edits during upload.
Download
Select File...
Cancel
Help



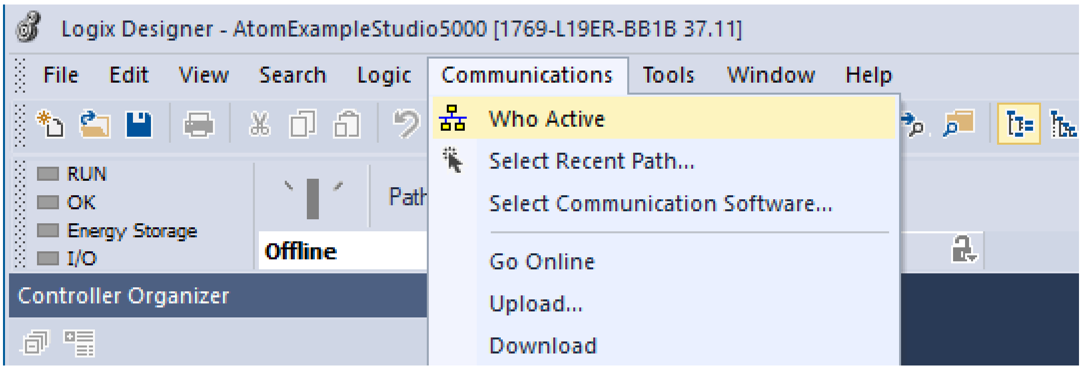
6. Right click `Controller AtomExampleStudio5000` and select **Properties**:

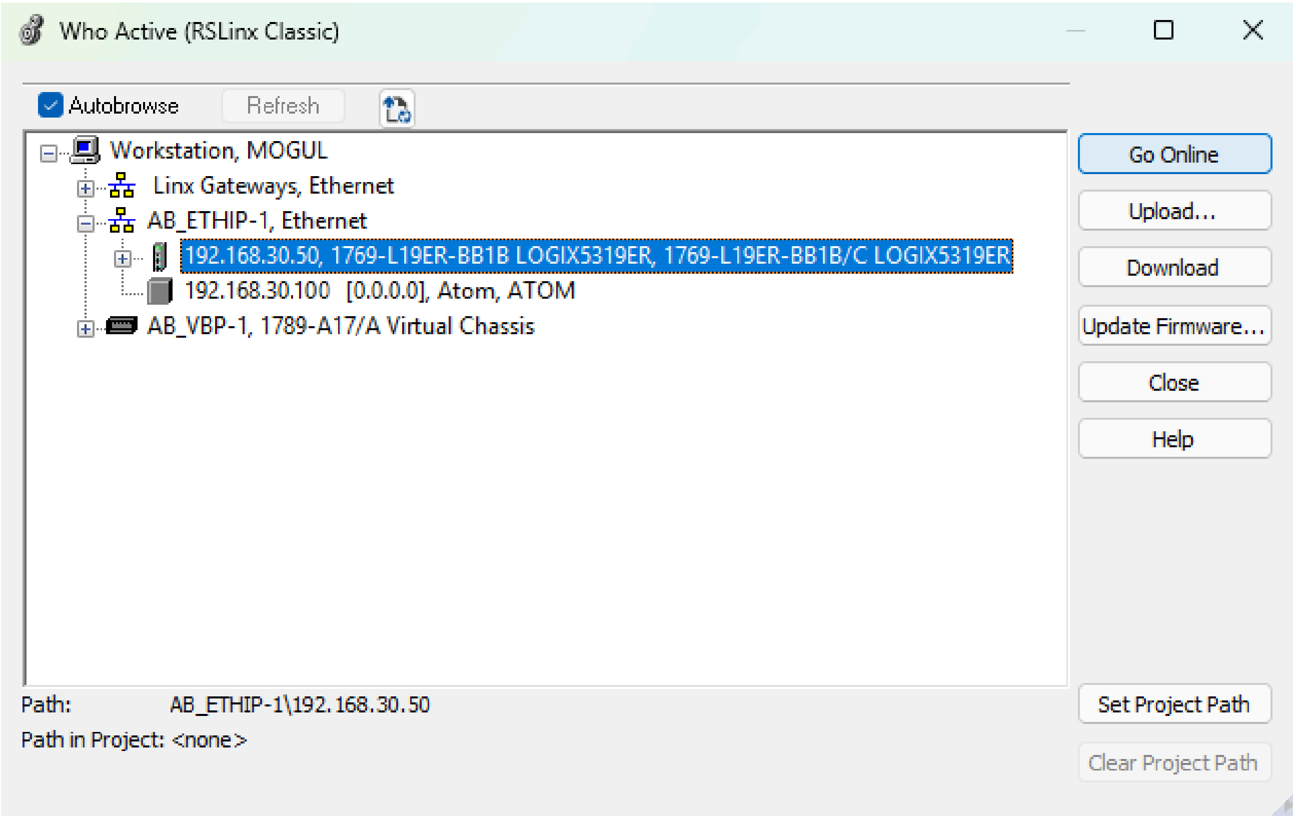


7. In the **Internet Protocol** tab, set the following and hit **Apply** and **OK**:
    - Manually configure IP settings checked
    - **IP Address**: `192.168.30.50`
    - **Subnet Mask**: `255.255.255.0`



8. Disconnect the USB cable from your PLC. In Studio 5000, select **Communications** > **Who Active** again, then select your PLC under the Ethernet category and click **Go Online**:

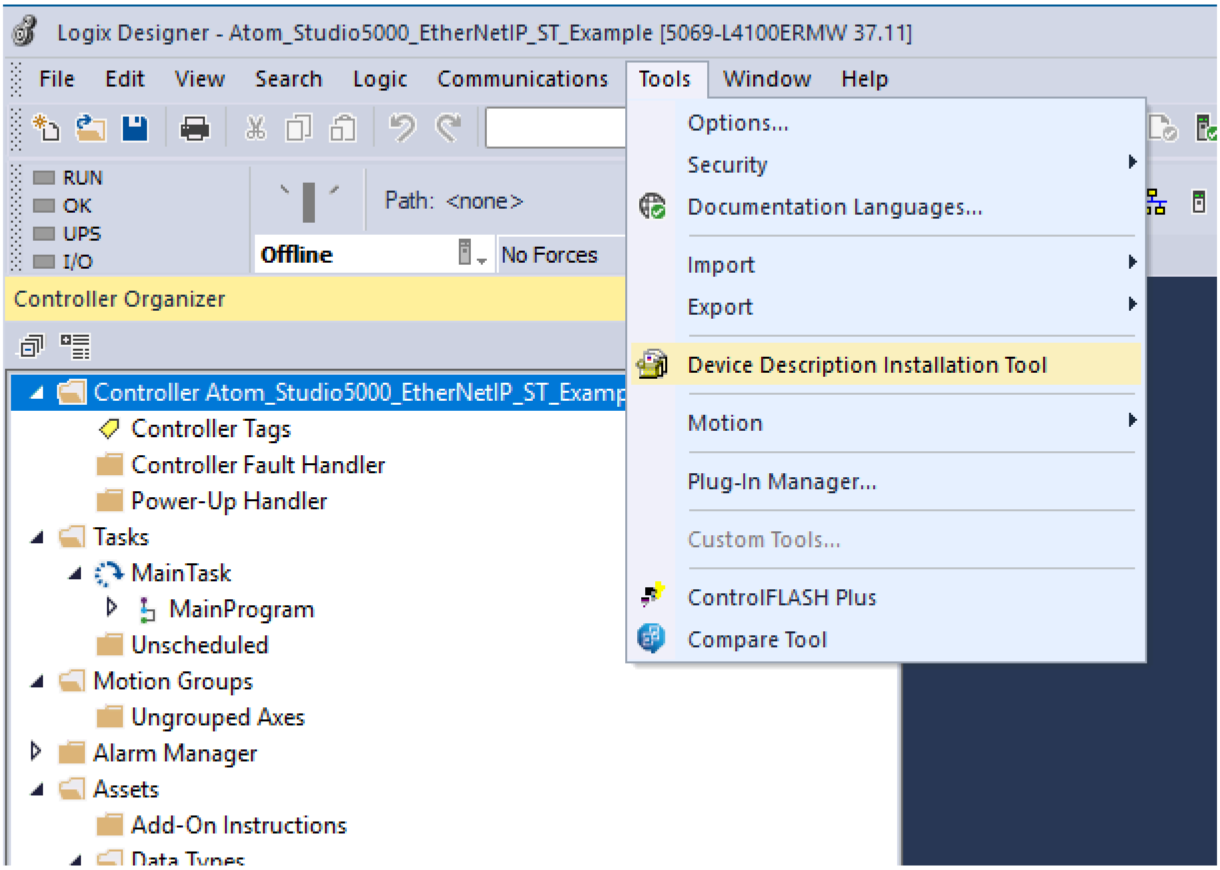


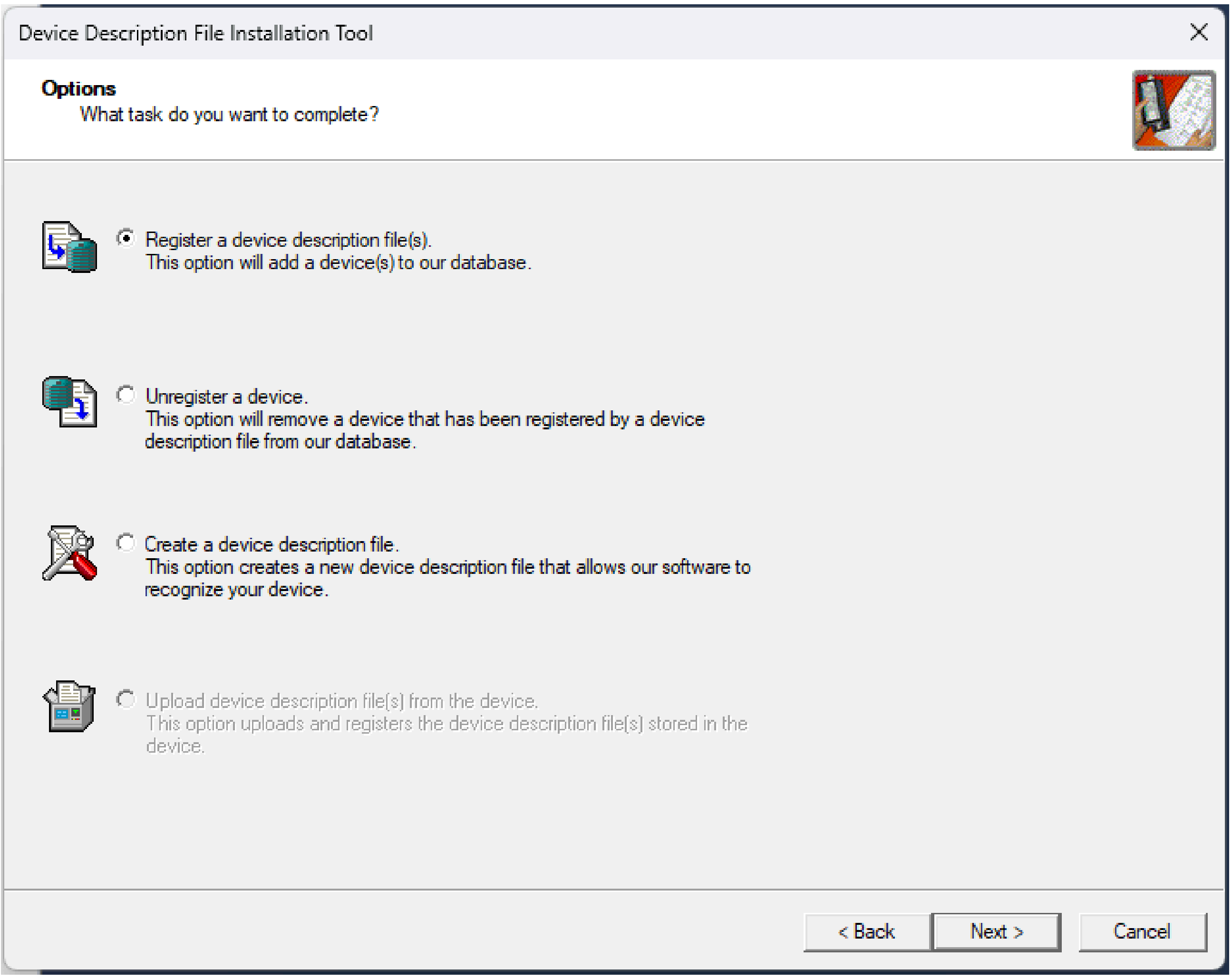


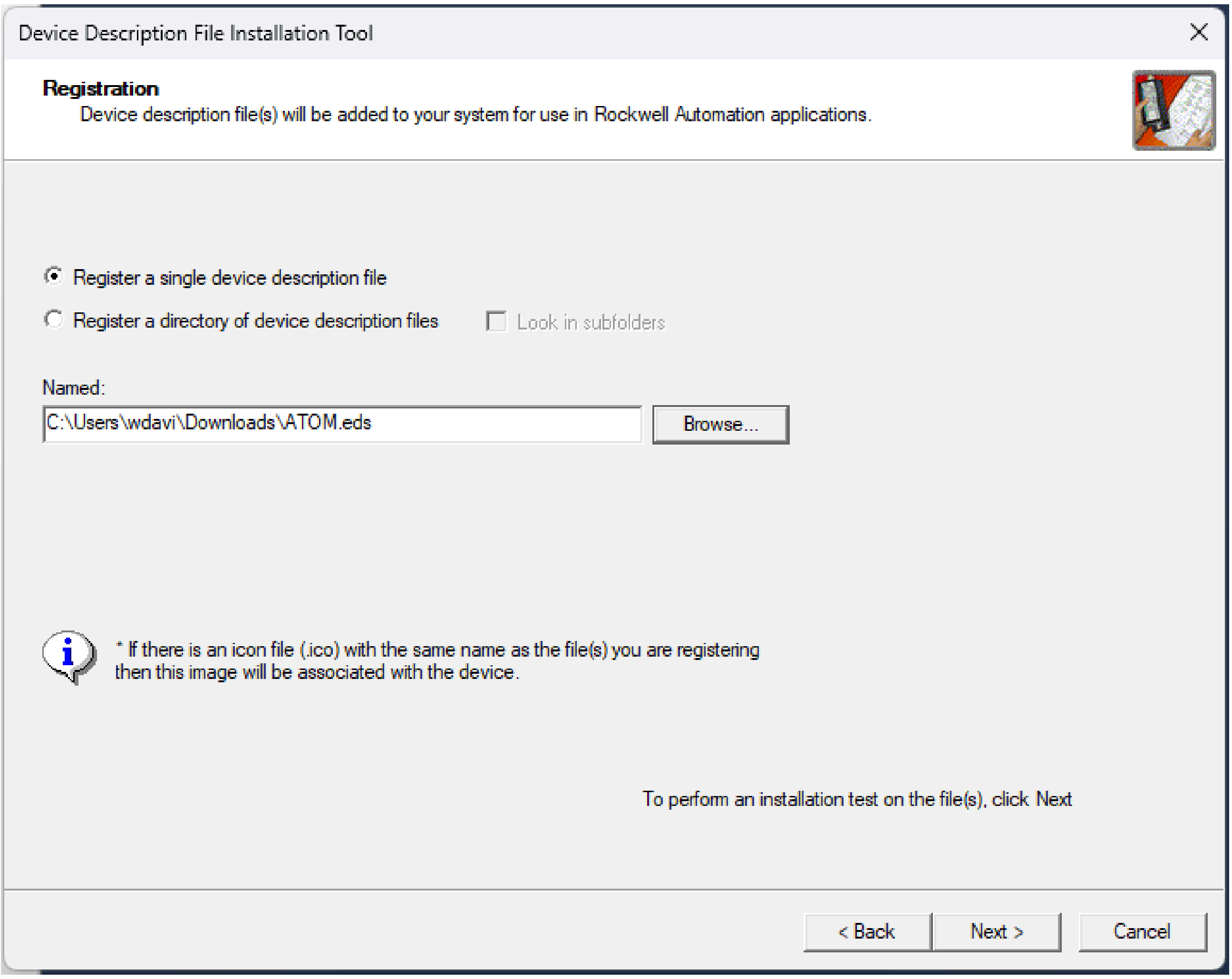
> You should also see ATOM (with IP address `192.168.30.100`) under the `AB_ETHIP-1` category.

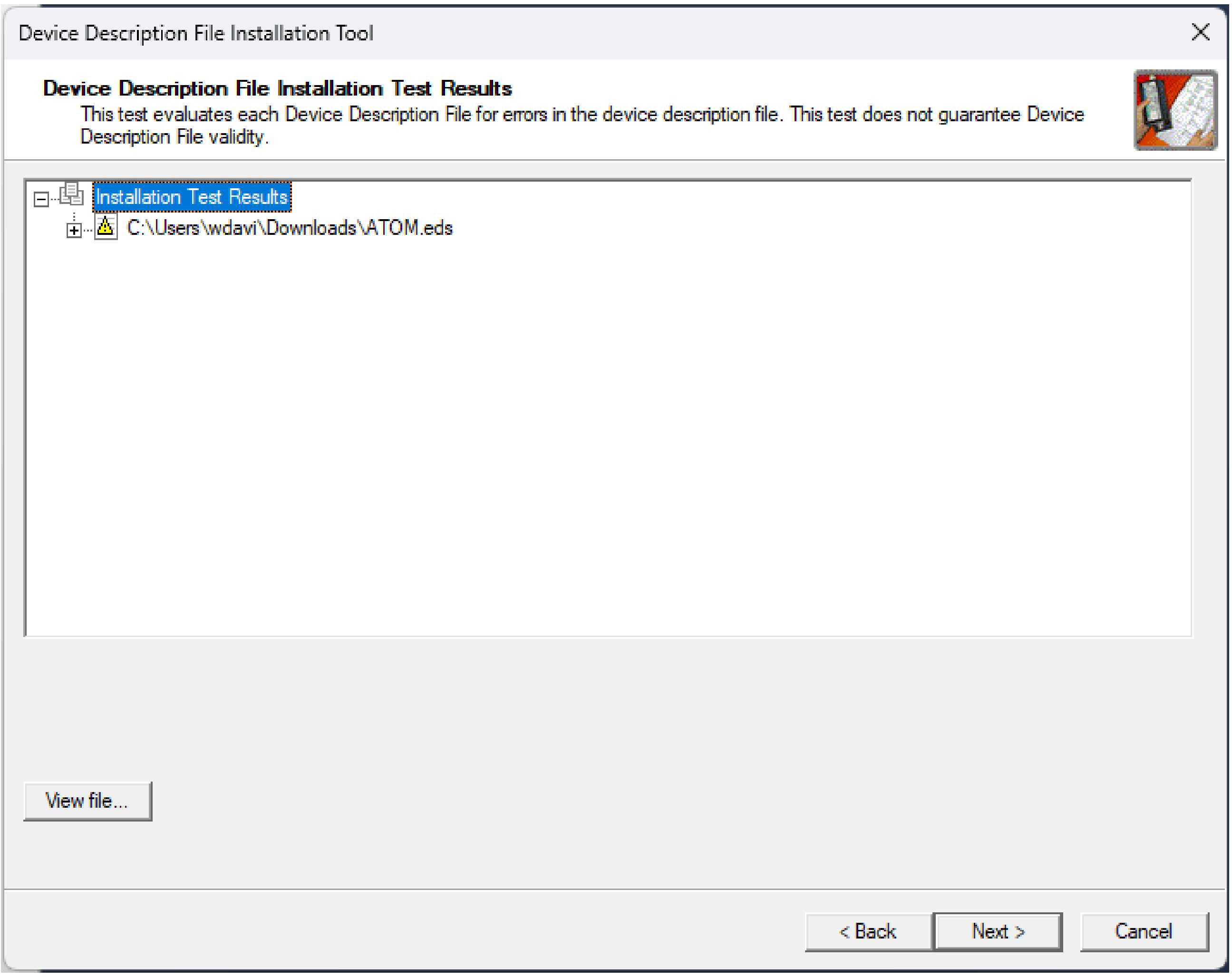
Import EDS file

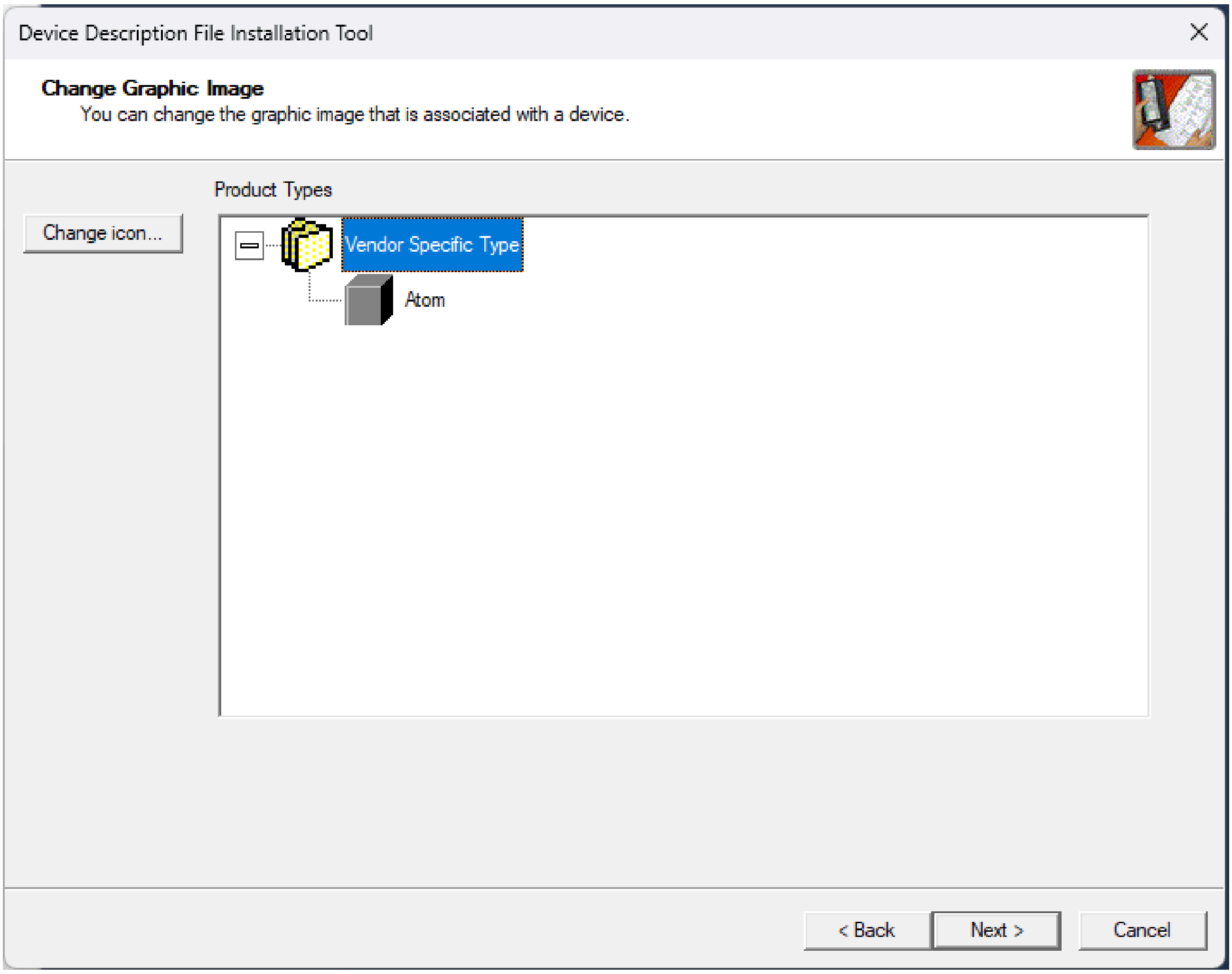
Select **Tools** > **Device Description Installation Tool** (some versions call it **EDS Hardware Installation Tool**)

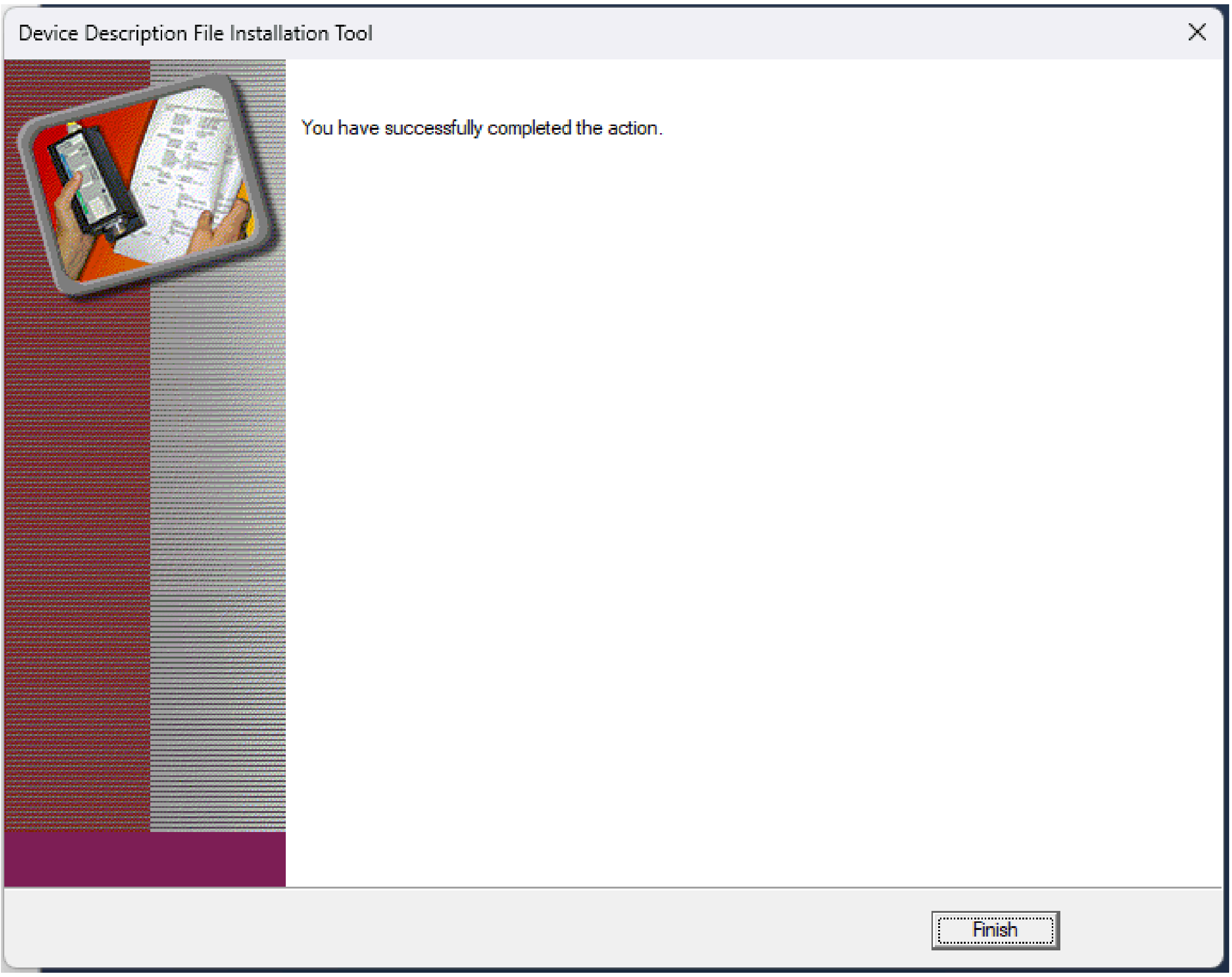

Logix Designer - Atom_Studio5000_EtherNetIP_ST_Example [5069-L4100ERMW 37.11]
File Edit View Search Logic Communications Tools Window Help
RUN
OK
UPS
I/O
Path: <none>
Offline
No Forces
Controller Organizer
Options...
Security
Documentation Languages...
Import
Export
Device Description Installation Tool
Motion
Plug-In Manager...
Custom Tools...
ControlFLASH Plus
Compare Tool
Controller Atom_Studio5000_EtherNetIP_ST_Examp
Controller Tags
Controller Fault Handler
Power-Up Handler
Tasks
MainTask
MainProgram
Unscheduled
Motion Groups
Ungrouped Axes
Alarm Manager
Assets
Add-On Instructions
Data Types


Device Description File Installation Tool
Options
What task do you want to complete?
Register a device description file(s).
This option will add a device(s) to our database.
Unregister a device.
This option will remove a device that has been registered by a device description file from our database.
Create a device description file.
This option creates a new device description file that allows our software to recognize your device.
Upload device description file(s) from the device.
This option uploads and registers the device description file(s) stored in the device.
< Back
Next >
Cancel


Device Description File Installation Tool
Registration
Device description file(s) will be added to your system for use in Rockwell Automation applications.
Register a single device description file
Register a directory of device description files
Look in subfolders
Named:
C:\Users\wdavi\Downloads\ATOM.eds
Browse...
* If there is an icon file (.ico) with the same name as the file(s) you are registering then this image will be associated with the device.
To perform an installation test on the file(s), click Next
< Back
Next >
Cancel

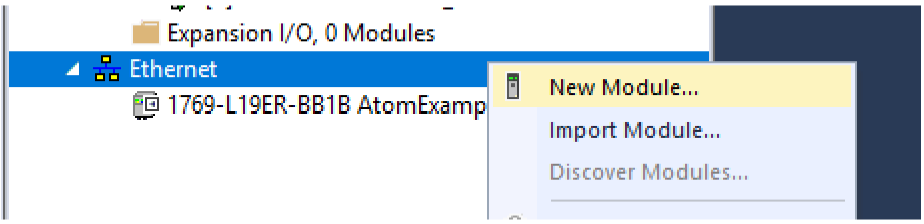

Device Description File Installation Tool
Device Description File Installation Test Results
This test evaluates each Device Description File for errors in the device description file. This test does not guarantee Device Description File validity.
Installation Test Results
C:\Users\wdavi\Downloads\ATOM.eds
View file...
< Back
Next >
Cancel


Device Description File Installation Tool
Change Graphic Image
You can change the graphic image that is associated with a device.
Product Types
Change icon...
Vendor Specific Type
Atom
< Back
Next >
Cancel

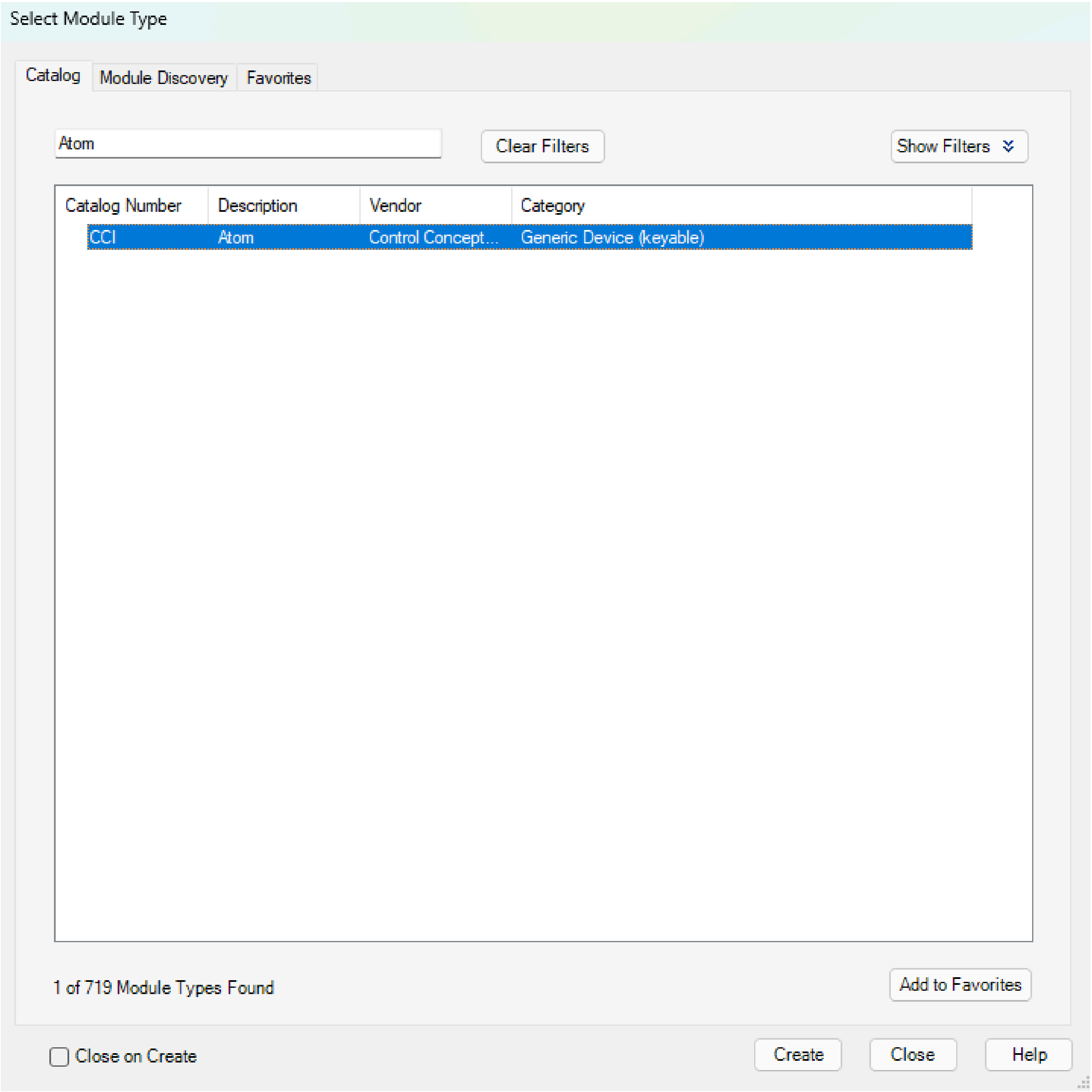


Add Atom to the project

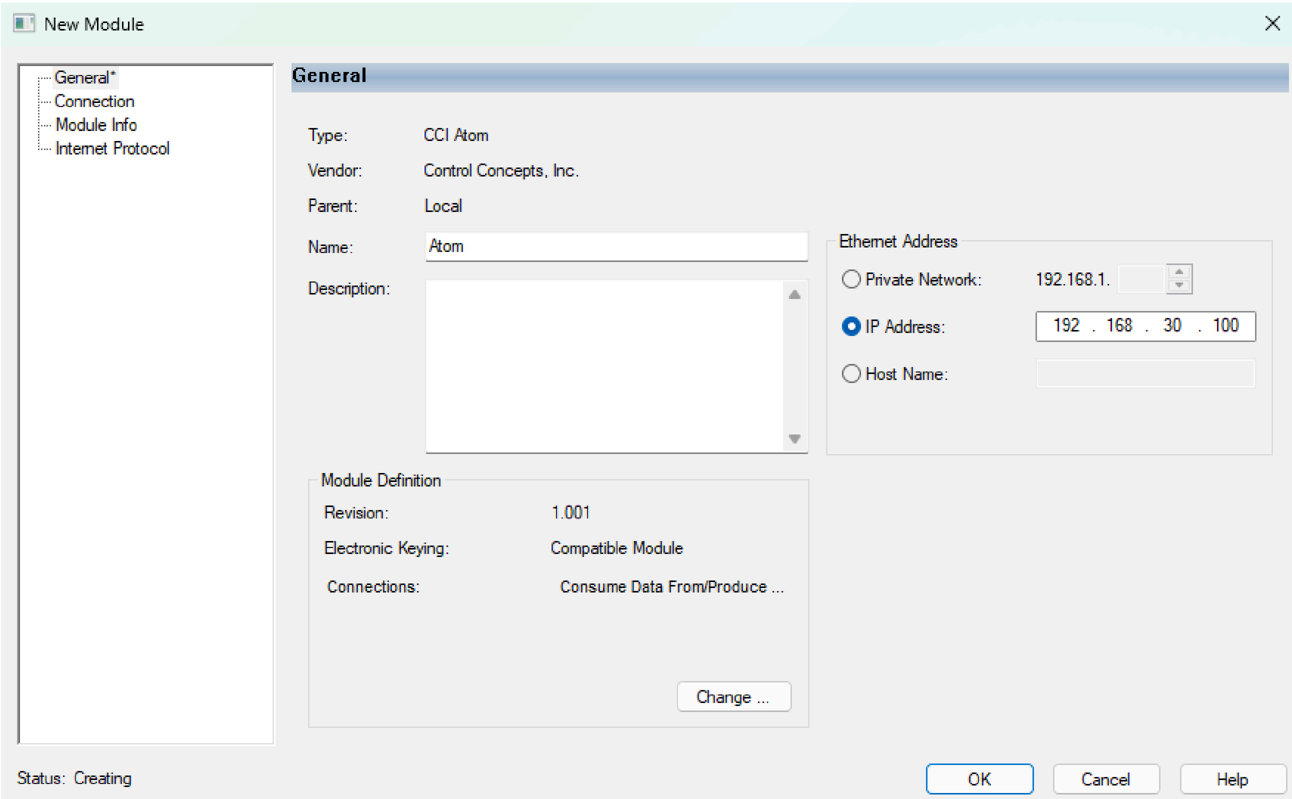
1. Right-click **Ethernet** and select **New Module**:



2. In the **Catalog** tab, search for `Atom`, select it, and click **Create**:

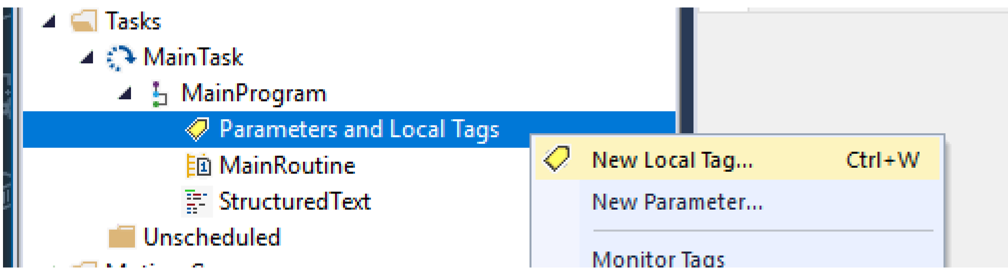


3. In the **General** tab, set the **IP Address** to `192.168.30.100` and click **OK**:



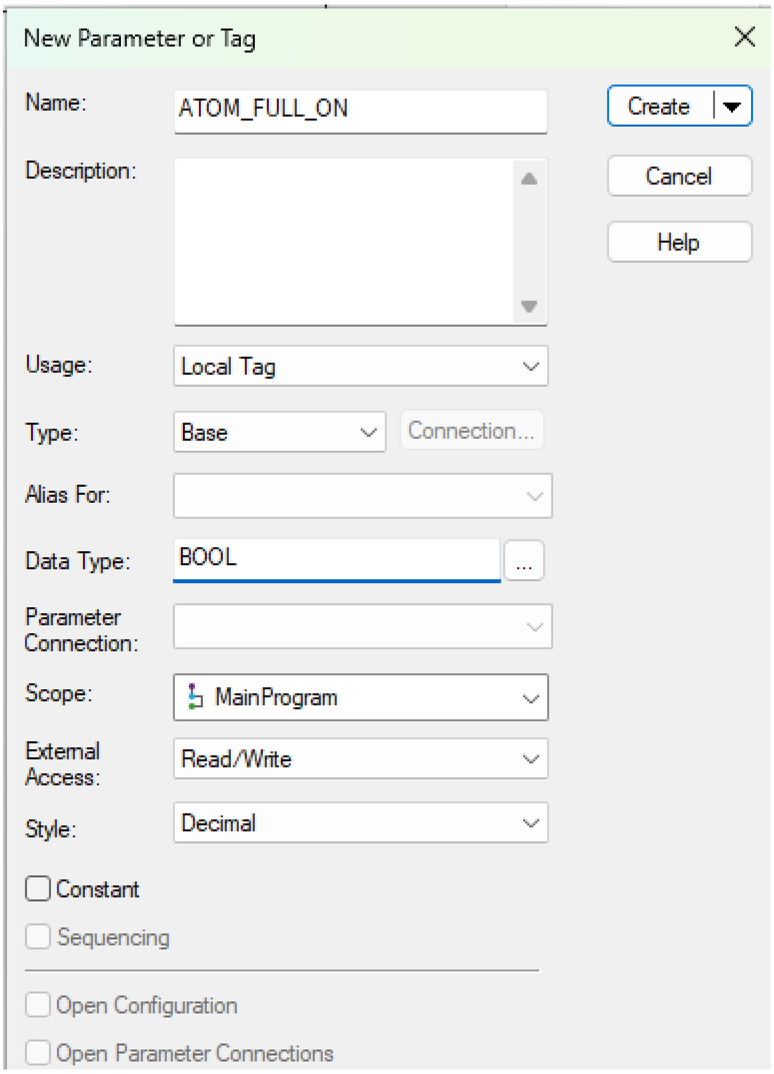
A basic example program

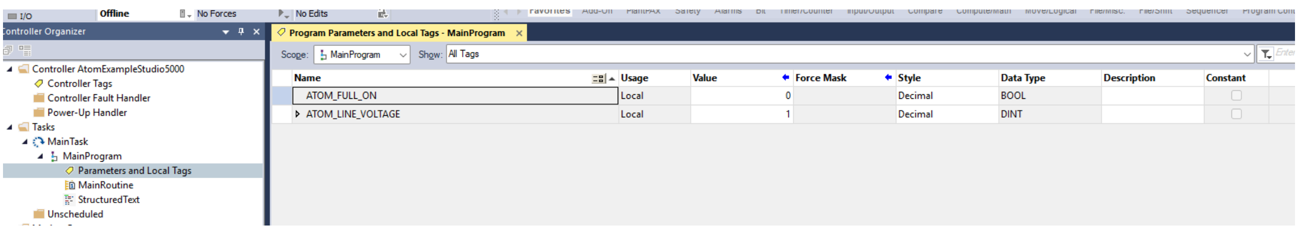
1. Right-click **Parameters and Local Tags** under `MainProgram` and select **New Tag** to create a tag:



2. Create *two_* new tags:
    - Tag 1
        - **Name**: `ATOM_FULL_ON`
        - **Data Type**: `BOOL`

- Tag 2
  - **Name**: `ATOM_LINE_VOLTAGE`
  - **Data Type**: `DINT`

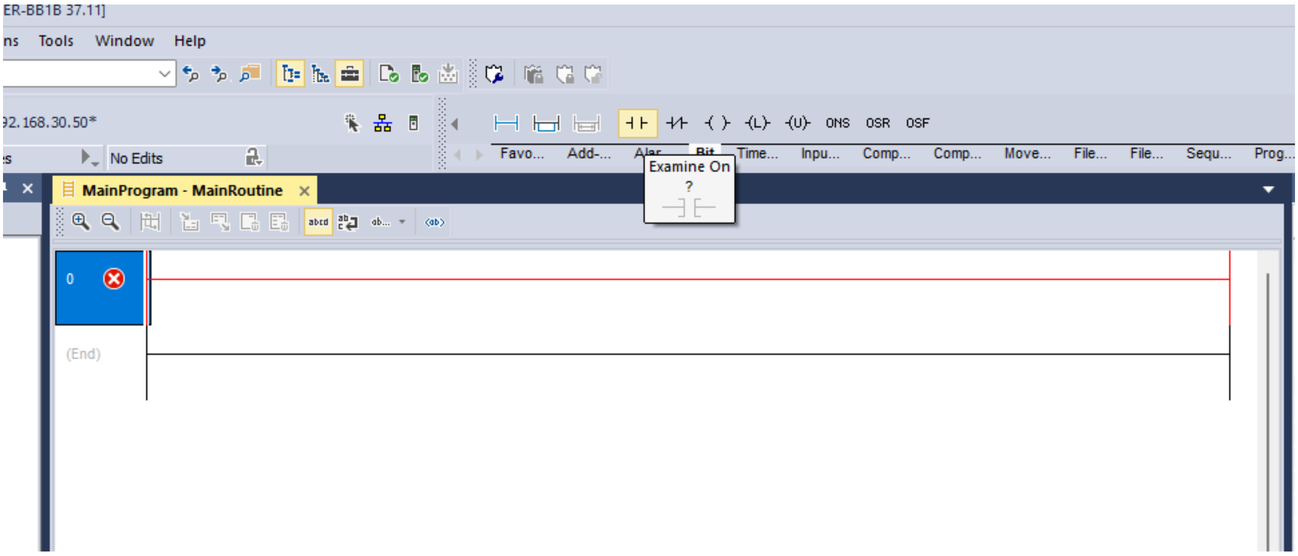




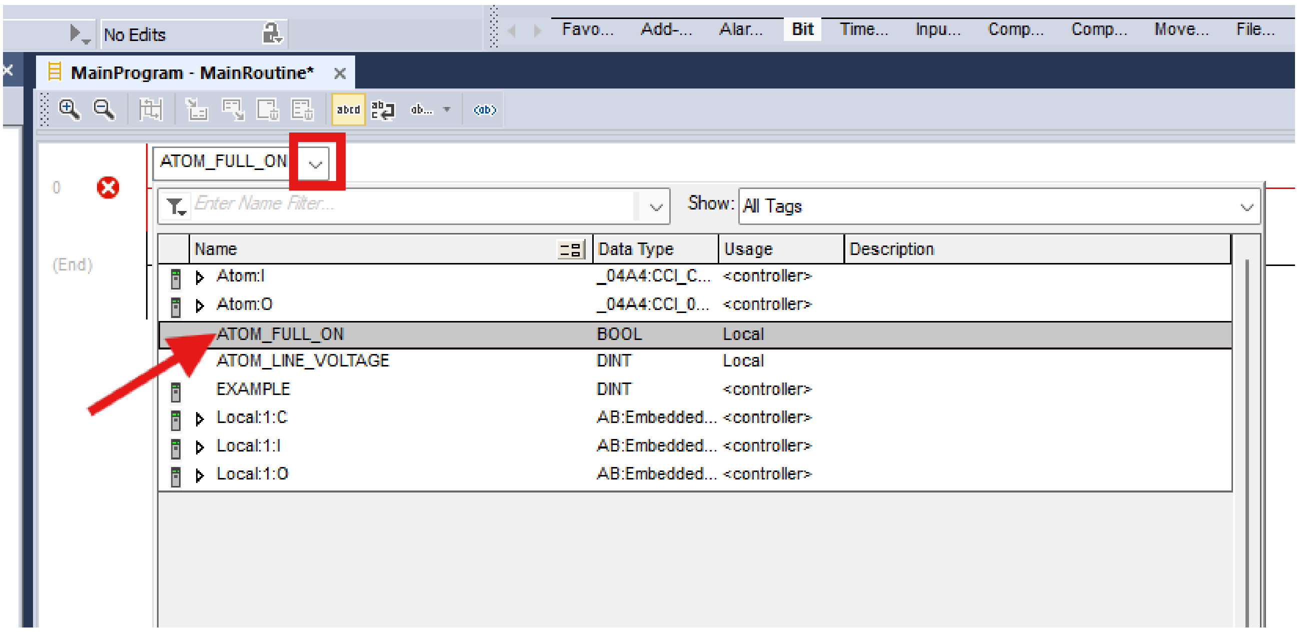
You can follow along with either the Structured Text or Ladder Logic examples below.

Ladder Logic

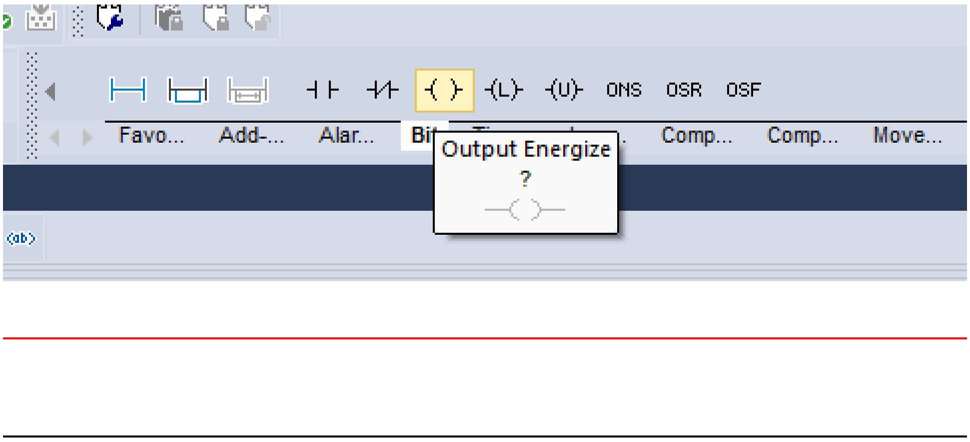
1. In the `MainRoutine` file, select `Ring 0` and add an `Examine On` instruction:

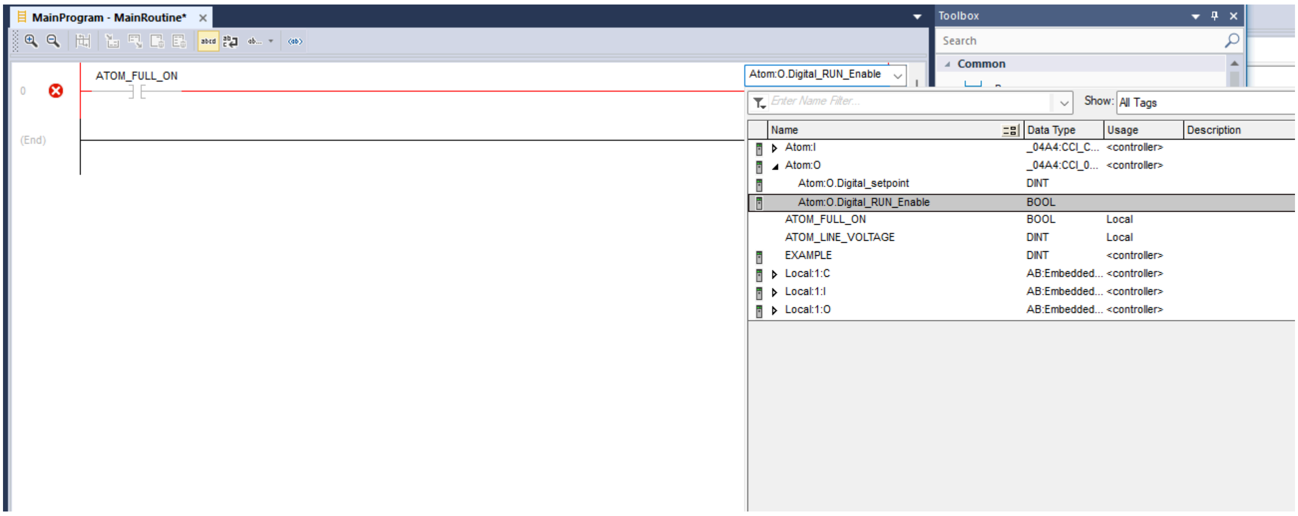


2. Configure this instruction to examine the `ATOM_FULL_ON` tag:

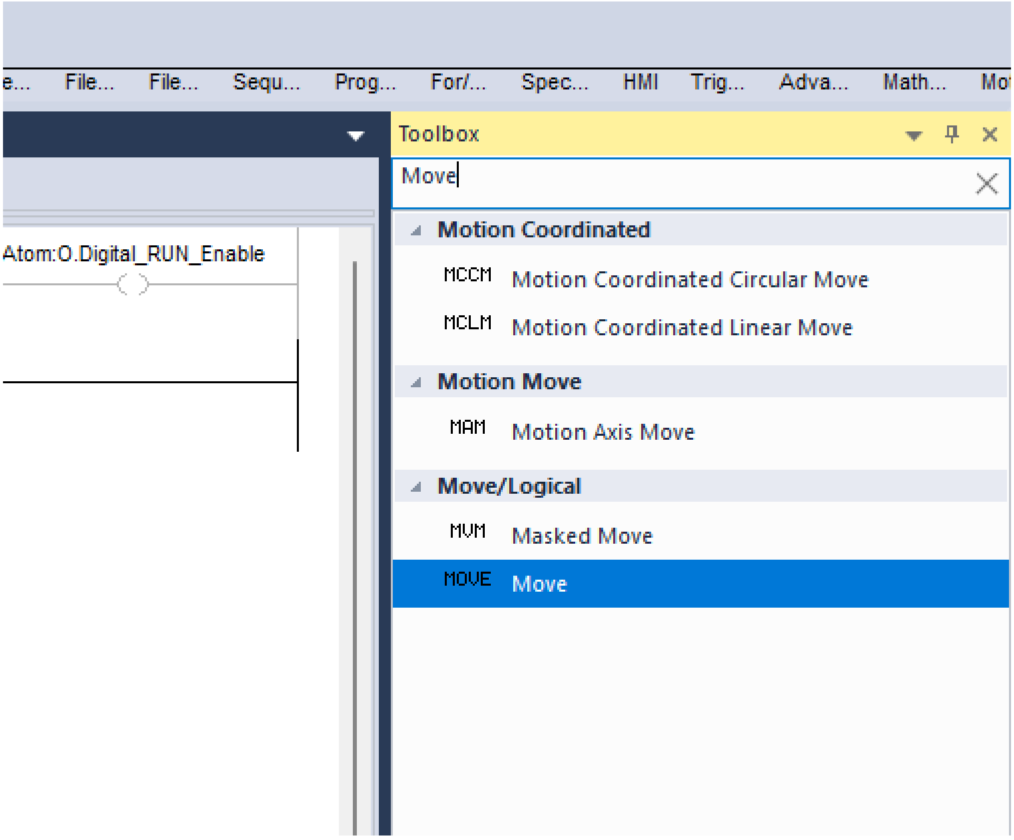


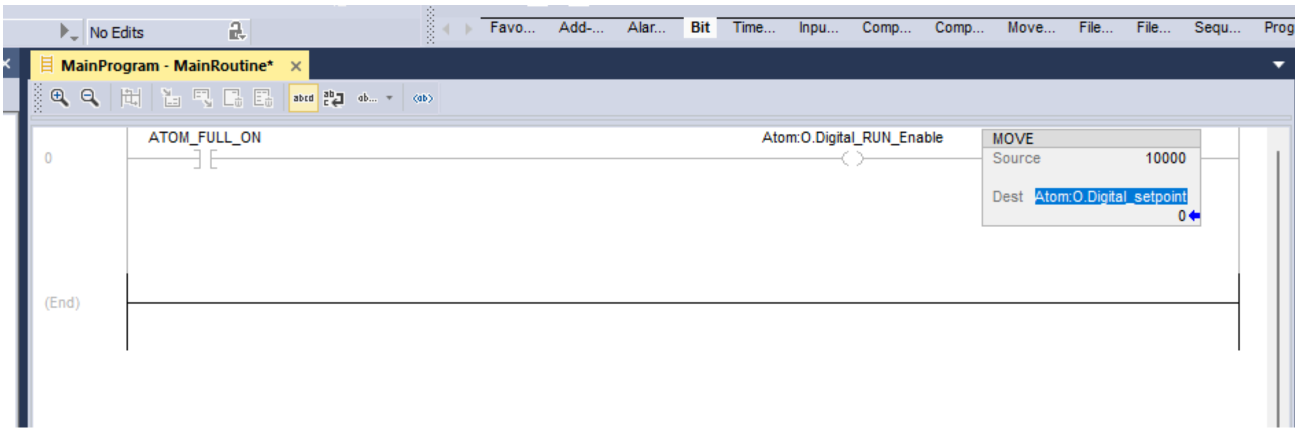
3. Add an `Output Energize` instruction and select `Atom:O.Digital_RUN_Enable`:



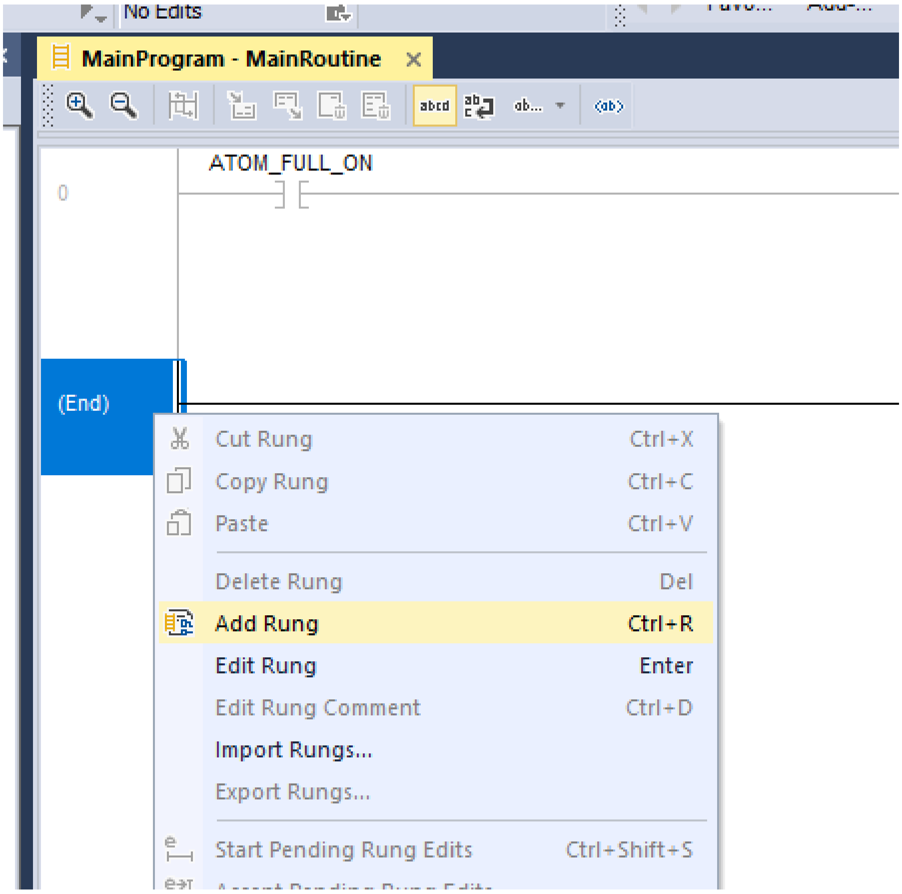


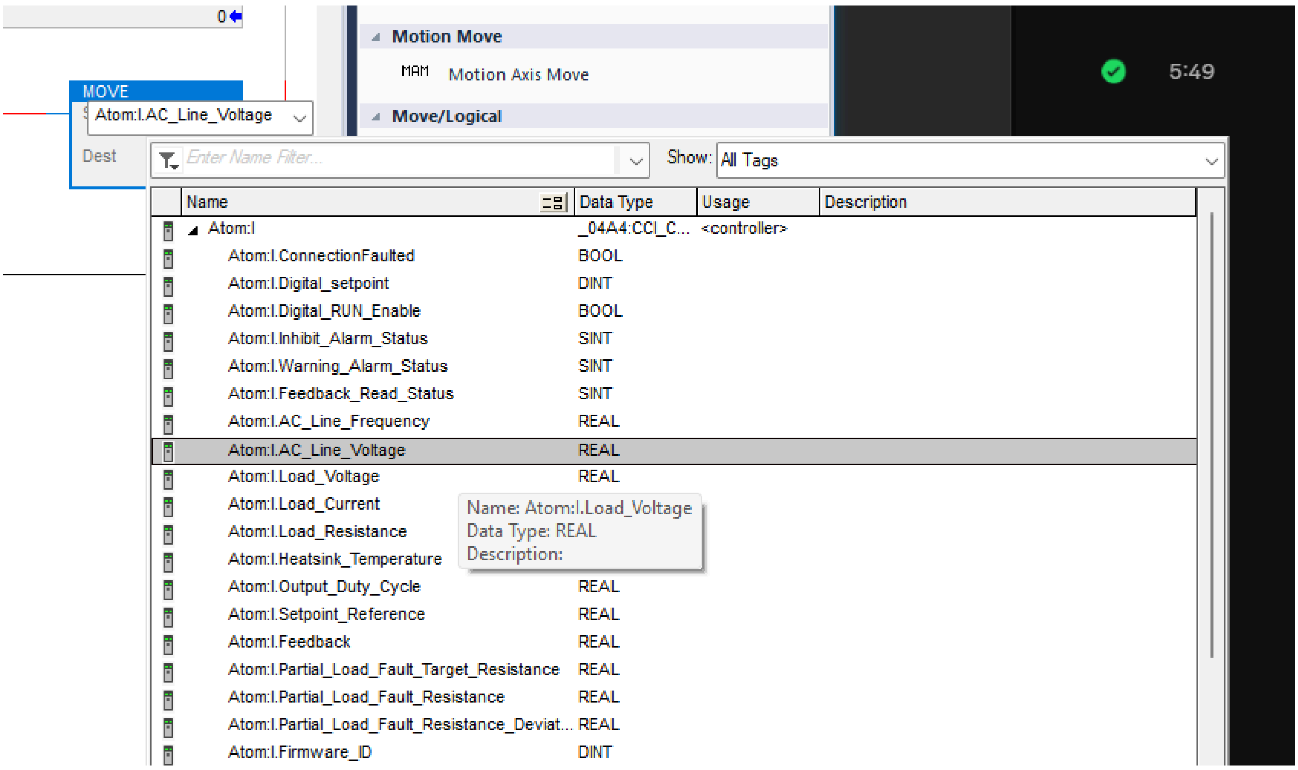
4. Add a `Move` instruction and set *source* to `10000` and *dest* to `Atom:O.Digital_setpoint`.





5. Right-click and select **Add rung**. In this new rung, add a `MOVE` instruction and set *source* to `Atom:I.AC_Line_Voltage` and *dest* to `ATOM_LINE_VOLTAGE`:

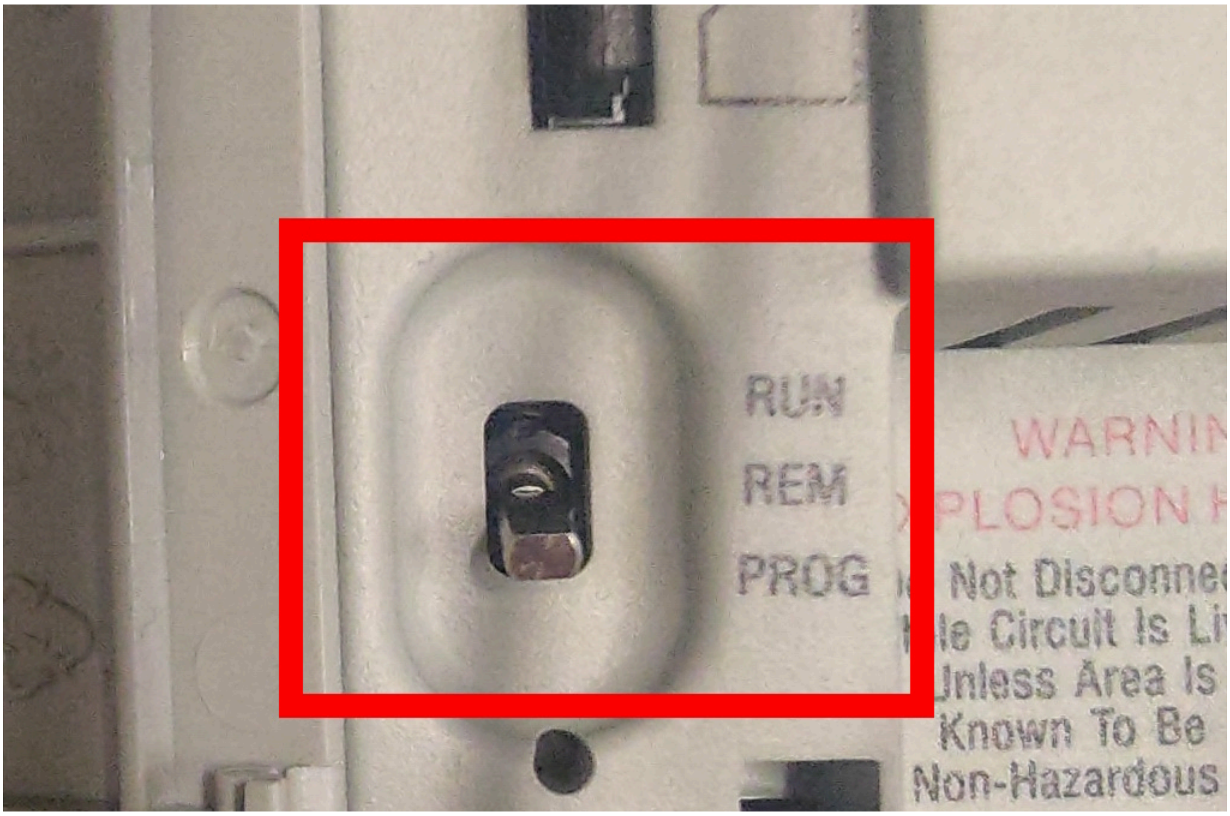

No Edits
MainProgram - MainRoutine
ATOM_FULL_ON
0
(End)
Cut Rung Ctrl+X
Copy Rung Ctrl+C
Paste Ctrl+V
Delete Rung Del
Add Rung Ctrl+R
Edit Rung Enter
Edit Rung Comment Ctrl+D
Import Rungs...
Export Rungs...
Start Pending Rung Edits Ctrl+Shift+S

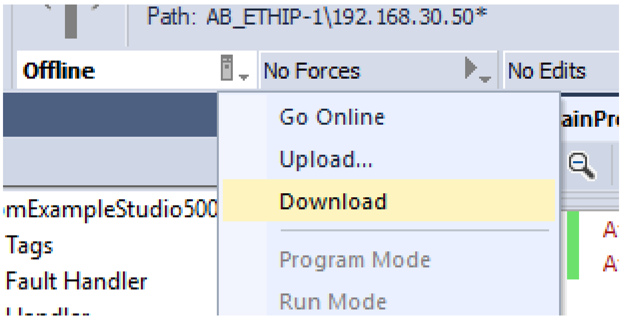

MOVE
Atom:I.AC_Line_Voltage
Dest
Motion Move
Motion Axis Move
Move/Logical
5:49
Enter Name Filter...
Show: All Tags
Name
Data Type
Usage
Description
Atom:I _04A4:CCI_C... <controller>
Atom:I.ConnectionFaulted BOOL
Atom:I.Digital_setpoint DINT
Atom:I.Digital_RUN_Enable BOOL
Atom:I.Inhibit_Alarm_Status SINT
Atom:I.Warning_Alarm_Status SINT
Atom:I.Feedback_Read_Status SINT
Atom:I.AC_Line_Frequency REAL
Atom:I.AC_Line_Voltage REAL
Atom:I.Load_Voltage REAL
Atom:I.Load_Current
Atom:I.Load_Resistance
Atom:I.Heatsink_Temperature
Name: Atom:I.Load_Voltage
Data Type: REAL
Description:
Atom:I.Output_Duty_Cycle REAL
Atom:I.Setpoint_Reference REAL
Atom:I.Feedback REAL
Atom:I.Partial_Load_Fault_Target_Resistance REAL
Atom:I.Partial_Load_Fault_Resistance REAL
Atom:I.Partial_Load_Fault_Resistance_Deviat... REAL
Atom:I.Firmware_ID DINT

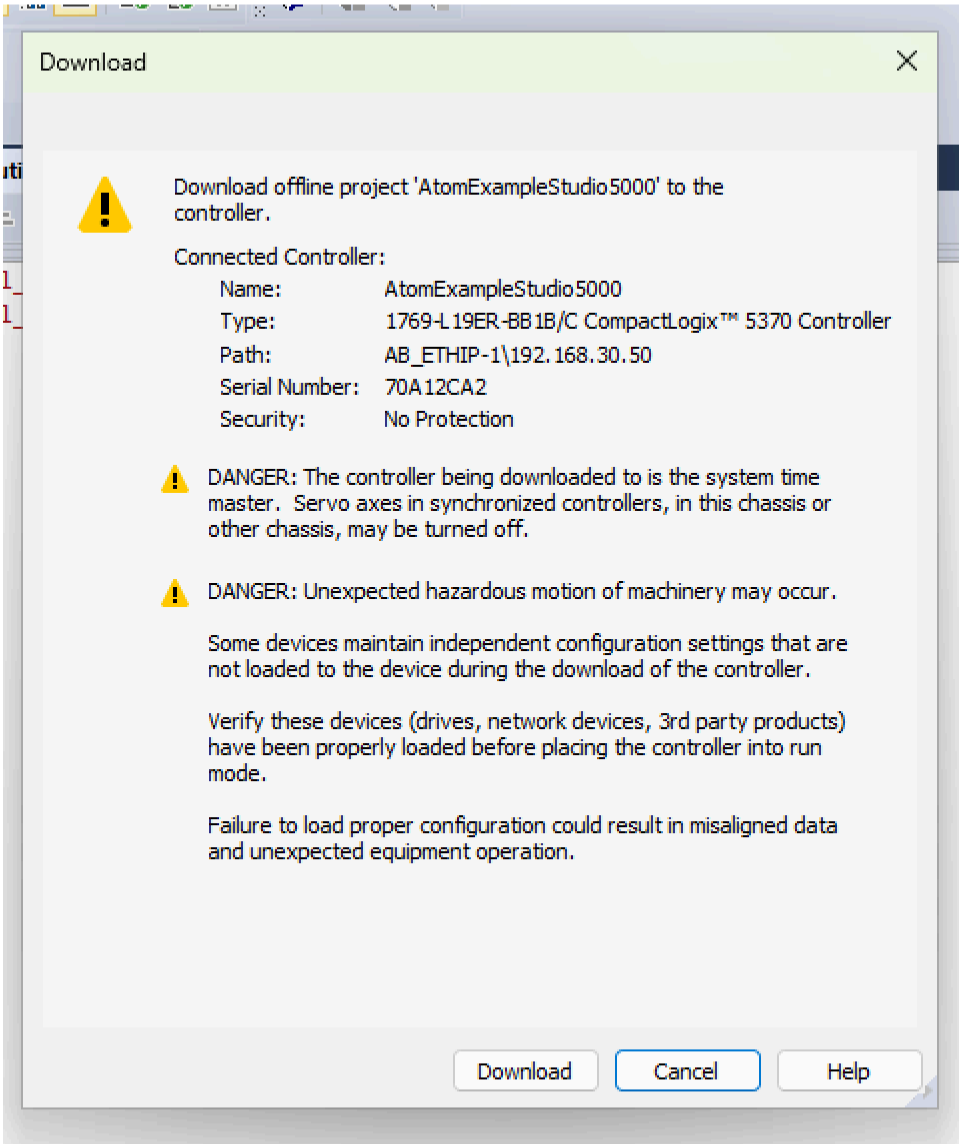
6. Select the PLC dropdown and click **Download** to download the program to your PLC:

> Ensure the switch on your PLC is set to `PROG` mode before downloading.


Allen-Bradley
RUN
FORCE
I/O
OK
NS
LINK 1
LINK 2
SD


RUN
REM
PROG
WARNIN
PLOSION H
Not Disconnec
le Circuit Is Li
Jnless Area Is
Known To Be
Non-Hazardous

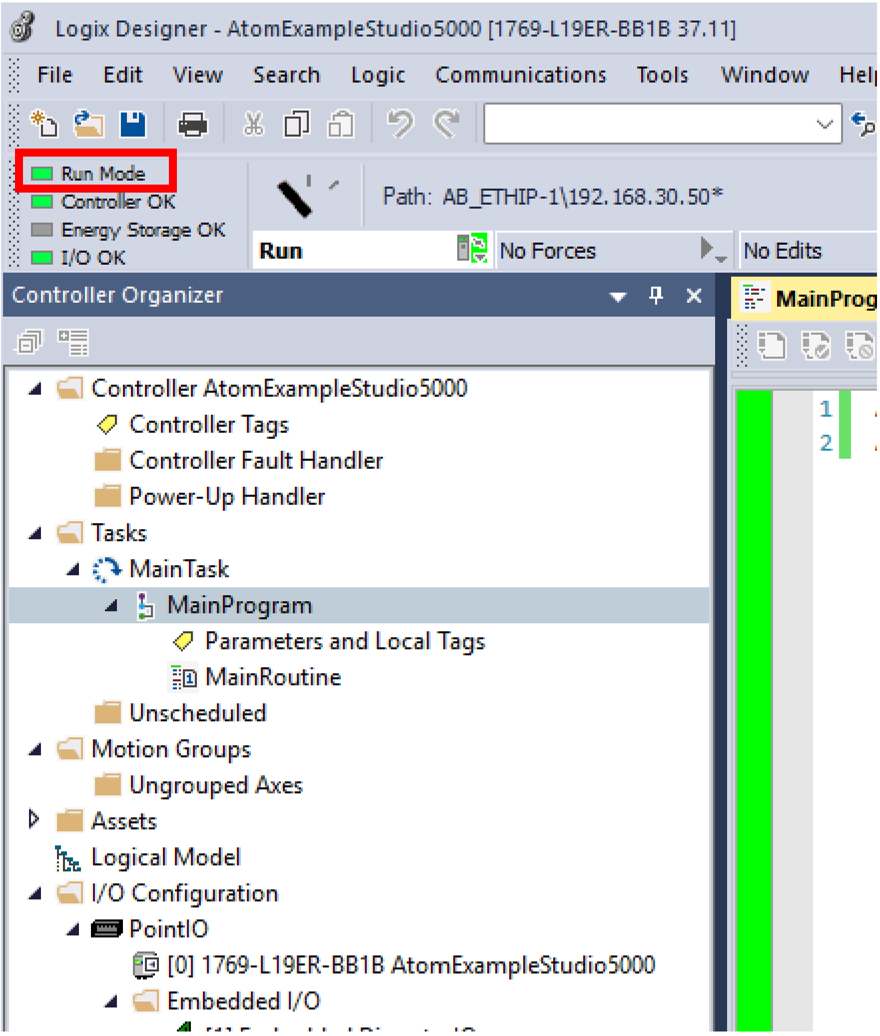

Path: AB_ETHIP-1\192.168.30.50*
Offline
No Forces
No Edits
Go Online
Upload...
Download
Program Mode
Run Mode
mExampleStudio500
Tags
Fault Handler



7. Flip the switch on your PLC to `RUN` mode.


Allen-Bradley
RUN
NS
FORCE
LINK 1
I/O
LINK 2
OK
SD

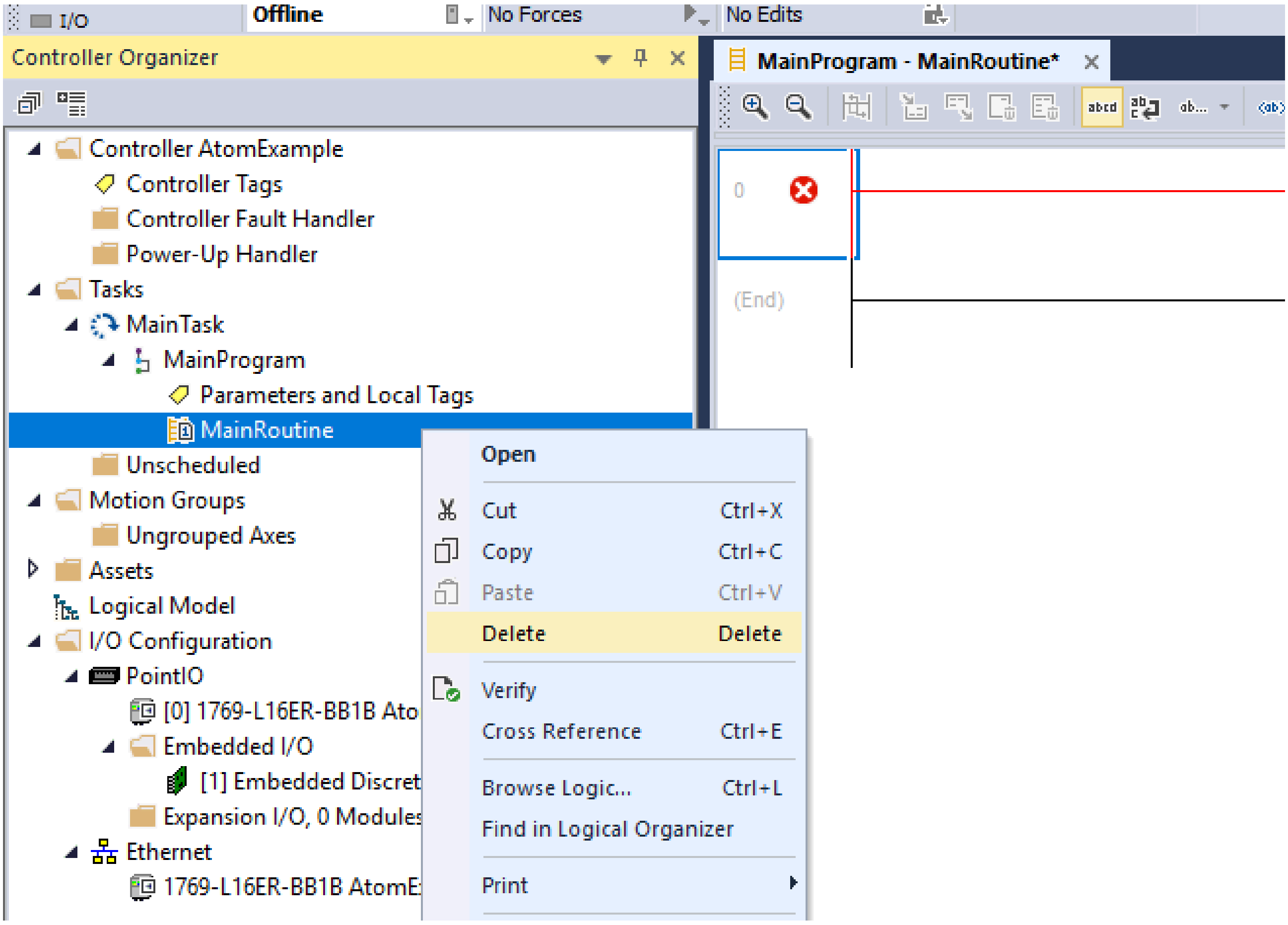
8. If everything worked properly, the controller **Run Mode** indicator light should turn green:



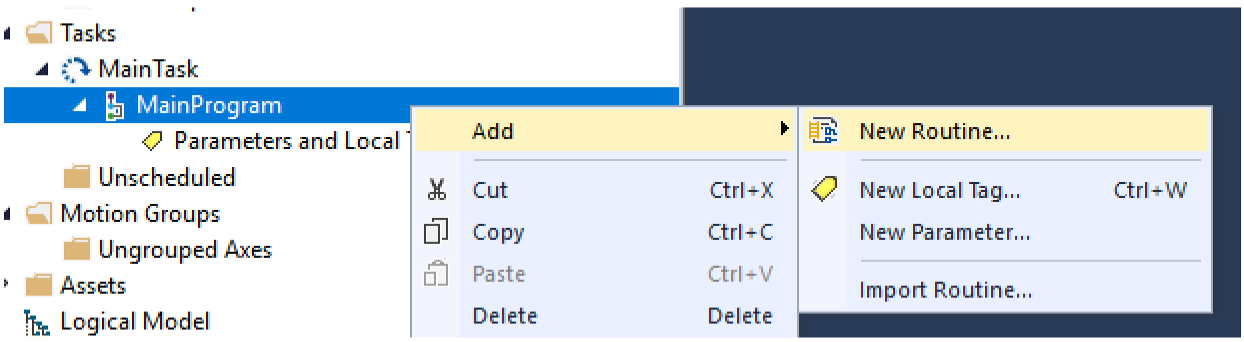
Next, jump to the Creating a user interface section.

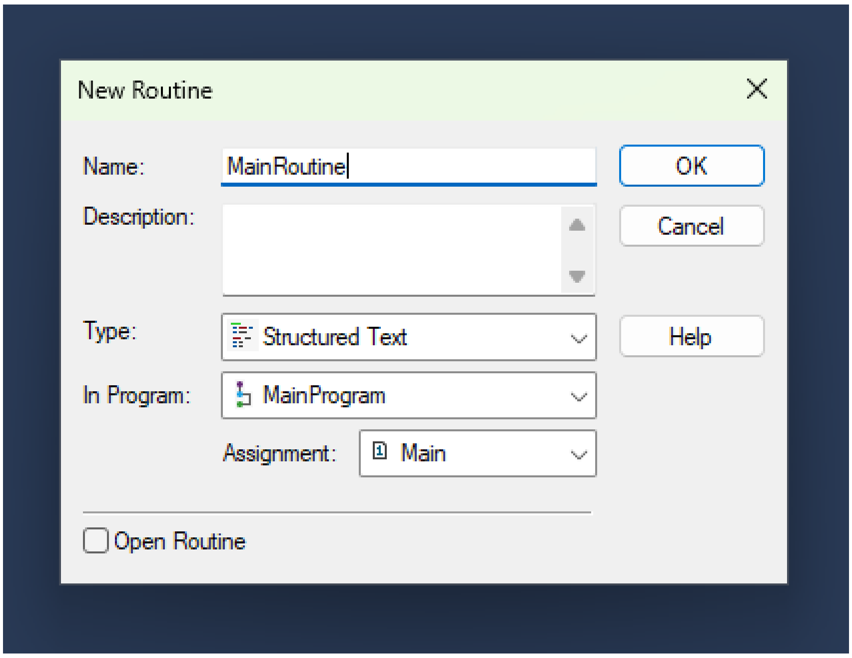
Structured Text

1. Delete the default `MainRoutine`:

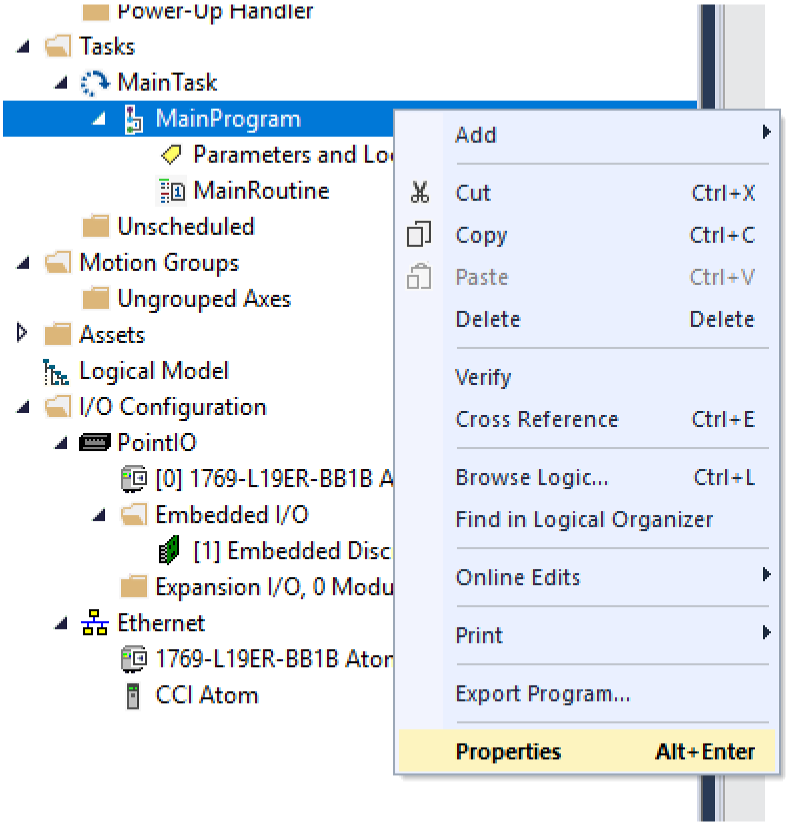


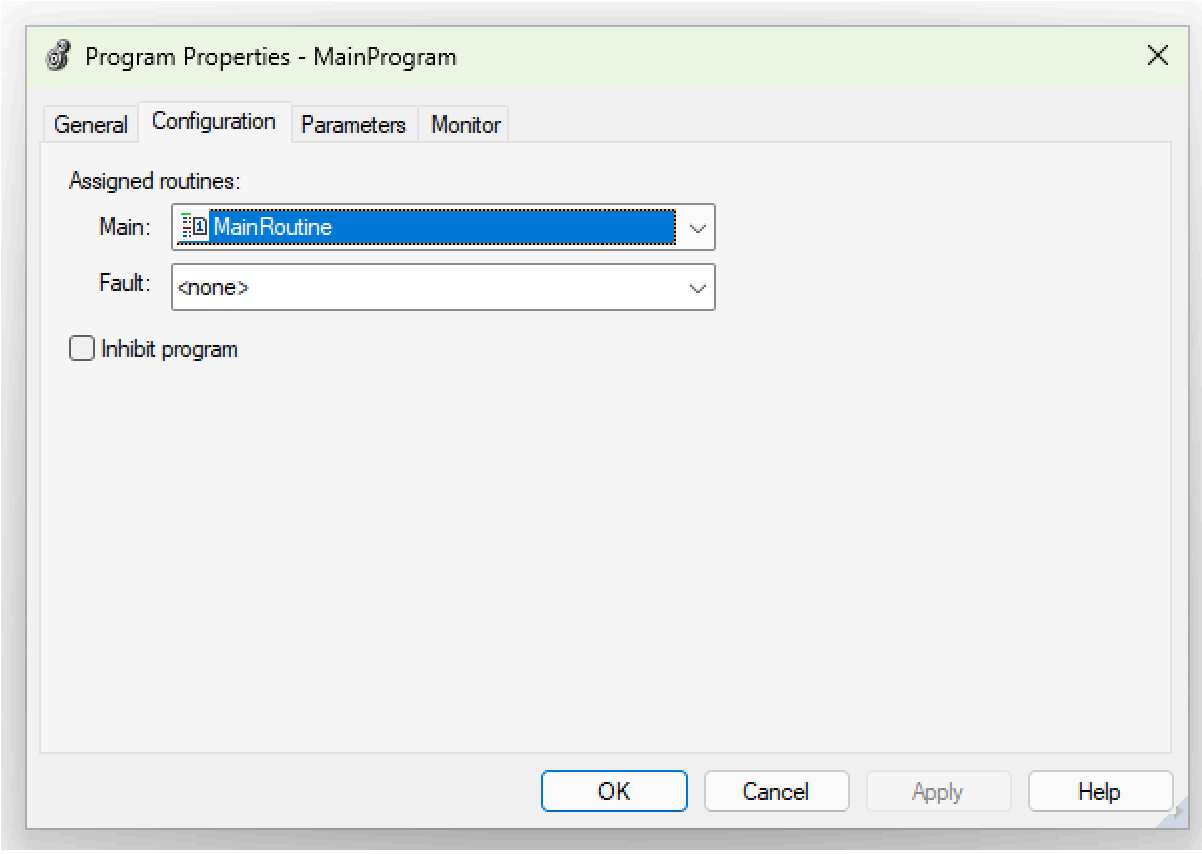
2. Right-click `MainProgram` and select **Add Routine**. Name it `MainRoutine`, set the *Type* to `Structured Text`, and click **OK**:



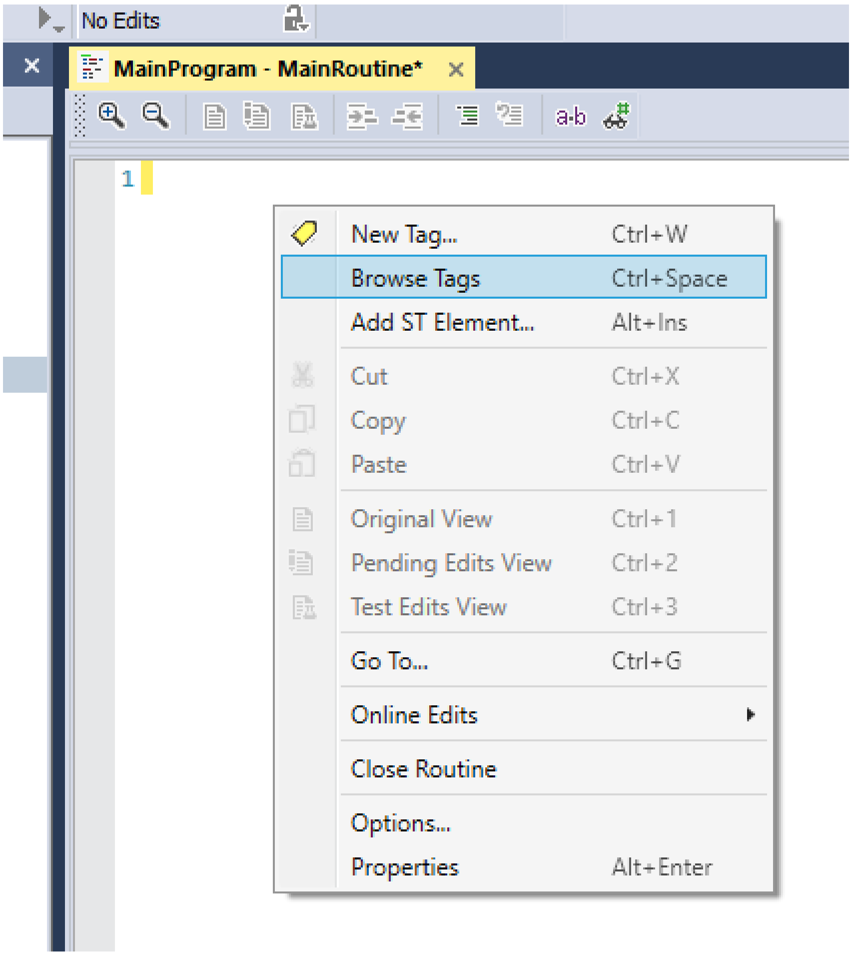


3. Right-click `MainRoutine`, select **Properties**, and ensure `MainRoutine` is set as the *Main Routine* in the **Configuration** tab:

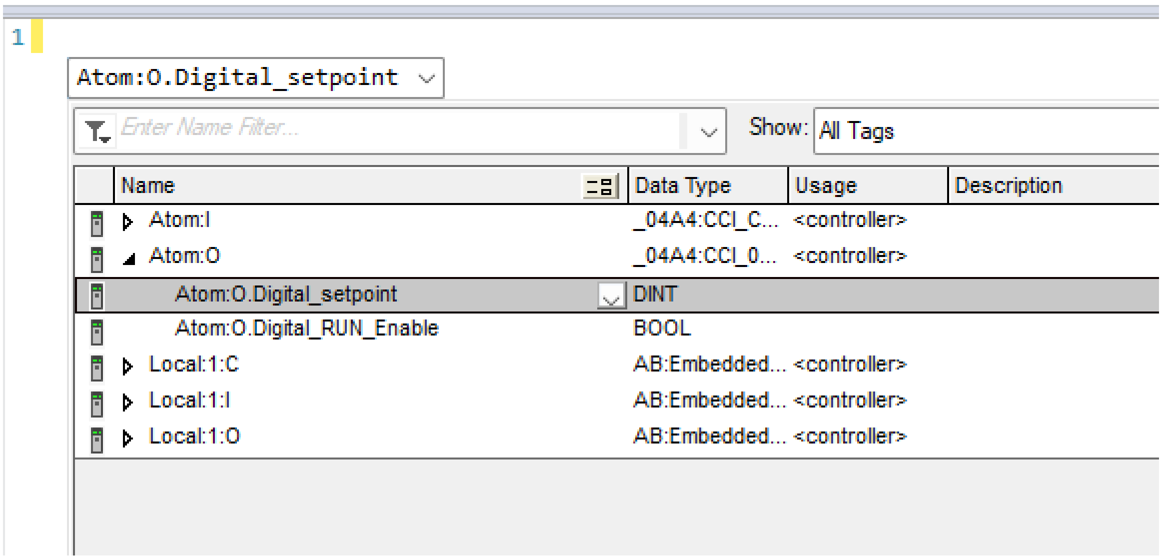




4. Insert tags by right-clicking in `MainRoutine` and selecting **Browse Tags**.

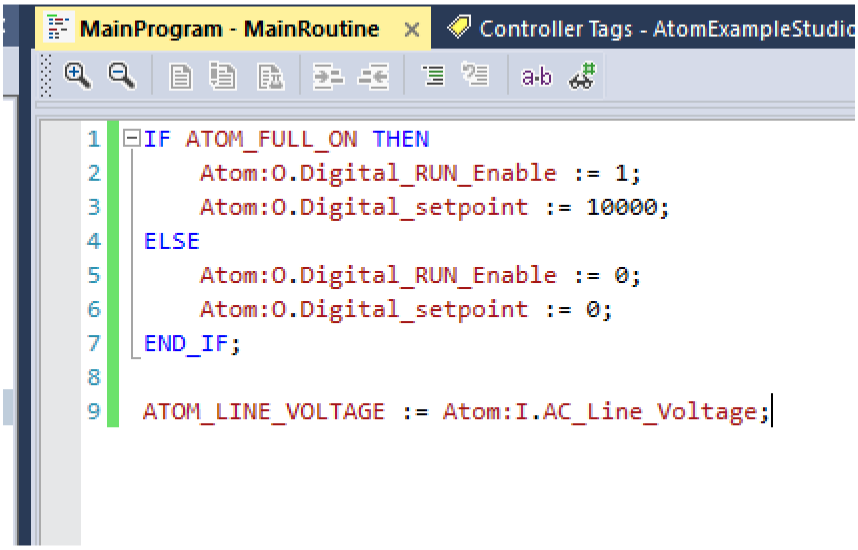


5. You can insert `Atom:I` (input) and `Atom:O` (output) tags to control ATOM:



6. Add the following code to `MainRoutine`:

```
IF ATOM_FULL_ON THEN
        Atom:O.Digital_RUN_Enable := 1;
        Atom:O.Digital_setpoint := 10000;
ELSE
        Atom:O.Digital_RUN_Enable := 0;
        Atom:O.Digital_setpoint := 0;
END_IF;

ATOM_LINE_VOLTAGE := Atom:I.AC_Line_Voltage;
```

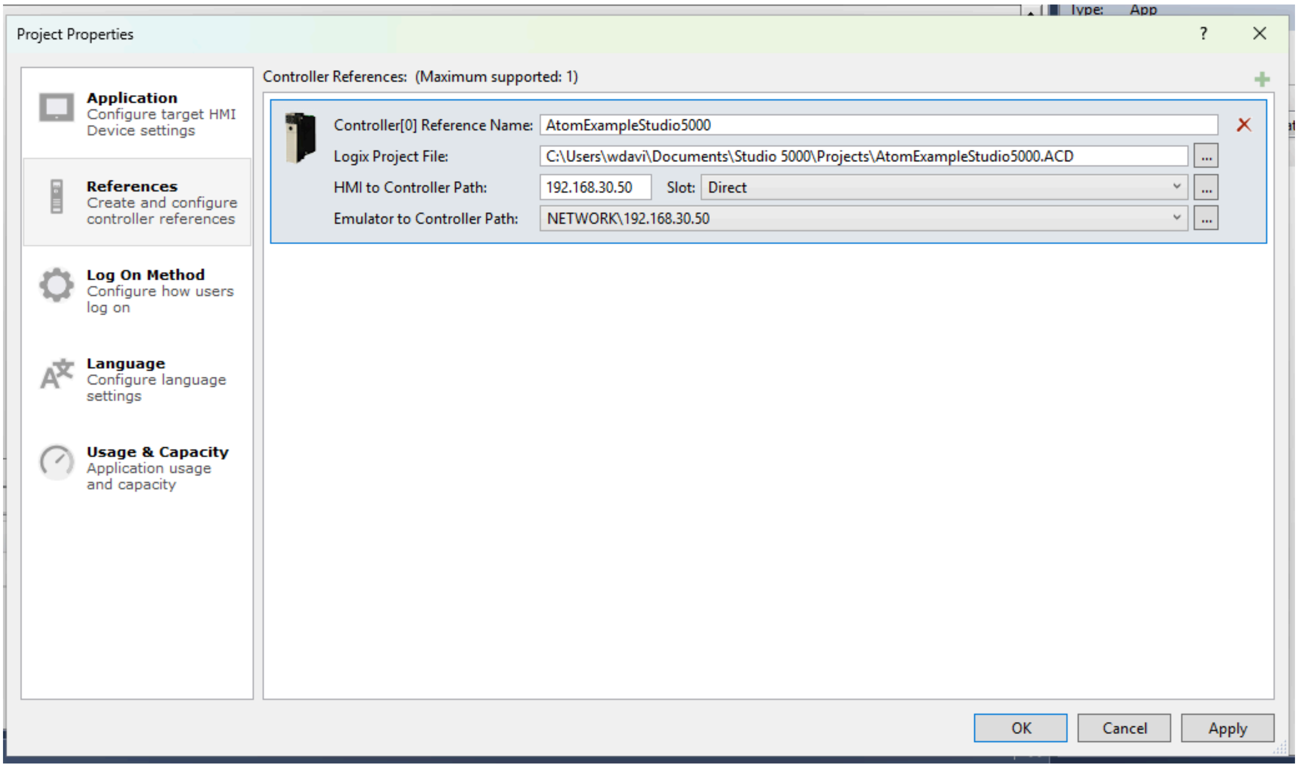


Next, jump to the Creating a user interface section.

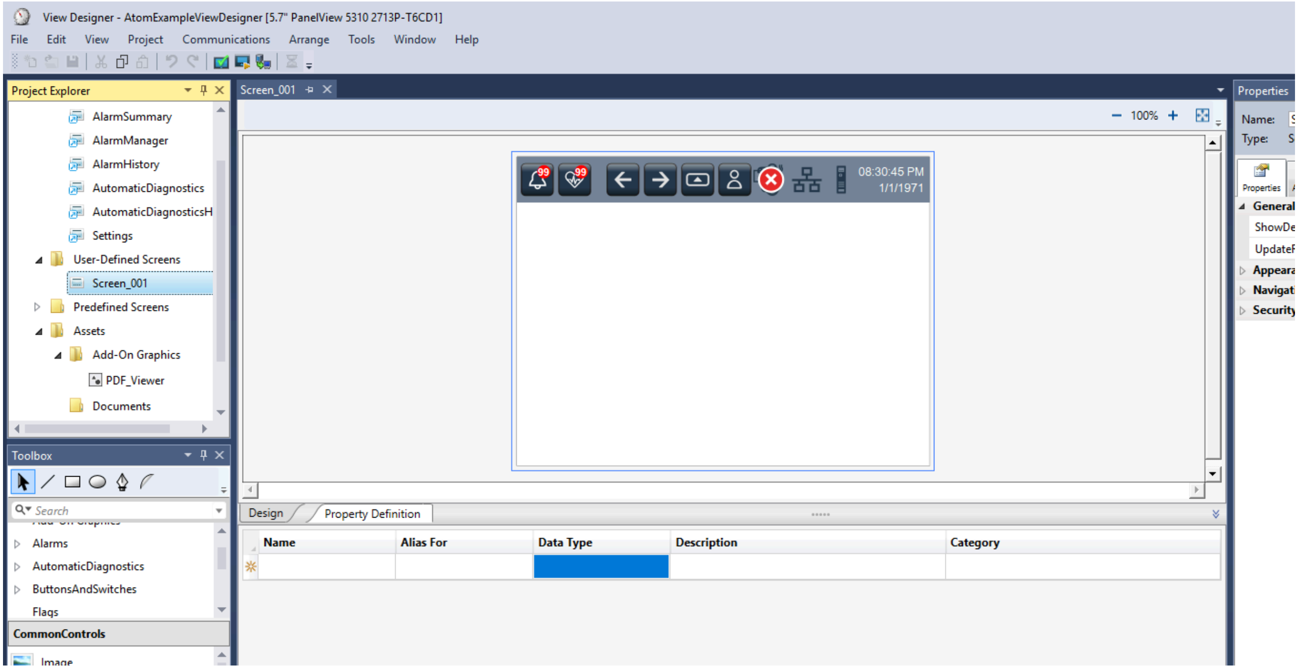
Creating a user interface

> **INFO**
>
> Studio 5000 comes with a separate program called **View Designer** for creating user interfaces. It's usually installed at `C:\Program Files (x86)\Rockwell Software\Studio 5000\View Designer\ENU\V10\ViewDesigner.exe`

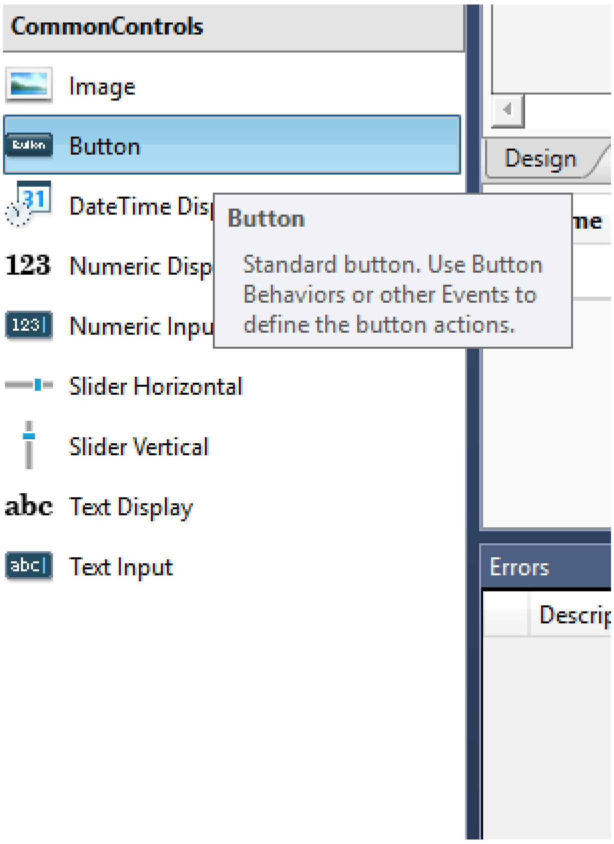
1. Launch View Designer and create a new project with the following settings:
    - **Controller[0] Reference Name**: `AtomExampleStudio5000`
    - **Logix Project File**: `path-to-your-project\AtomExampleStudio5000.ACD`
    - **HMI to Controller Path**: `192.168.30.50`
    - **Emulerator to Controller Path**: `NETWORK\192.168.30.50`

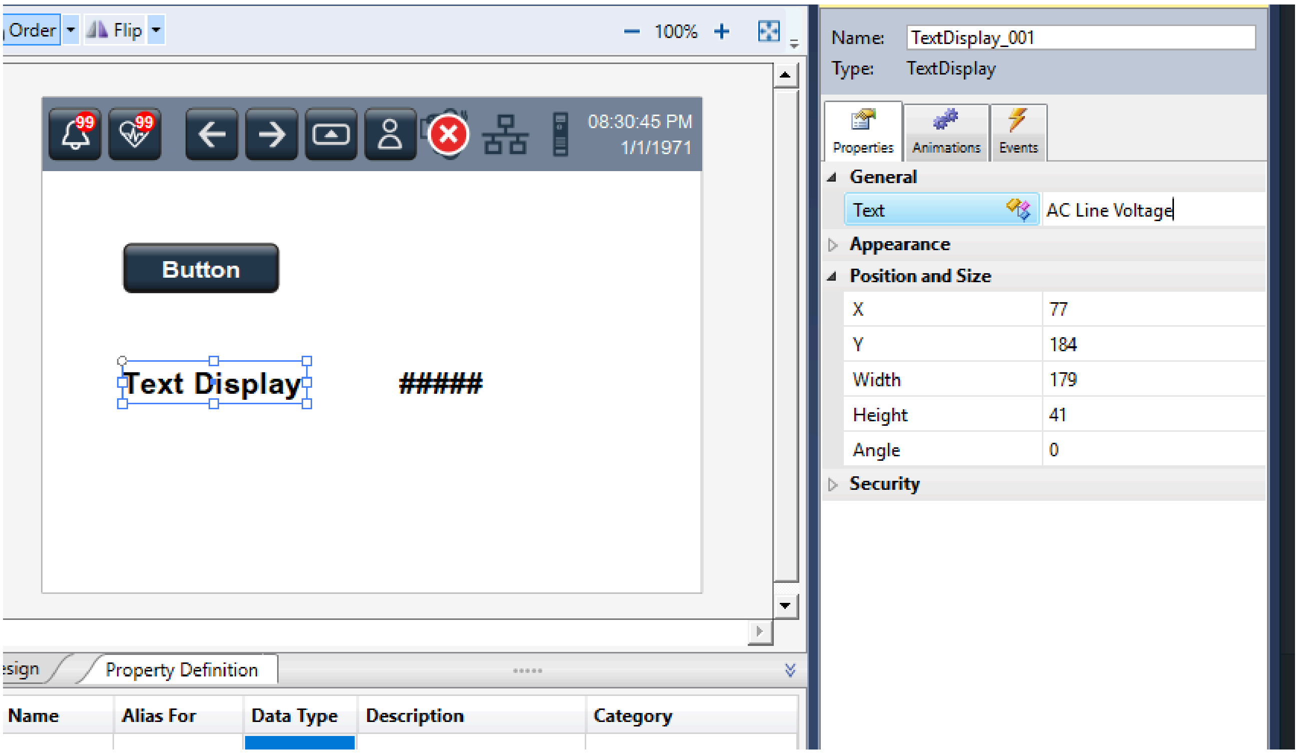


2. Open `Screen_001`:

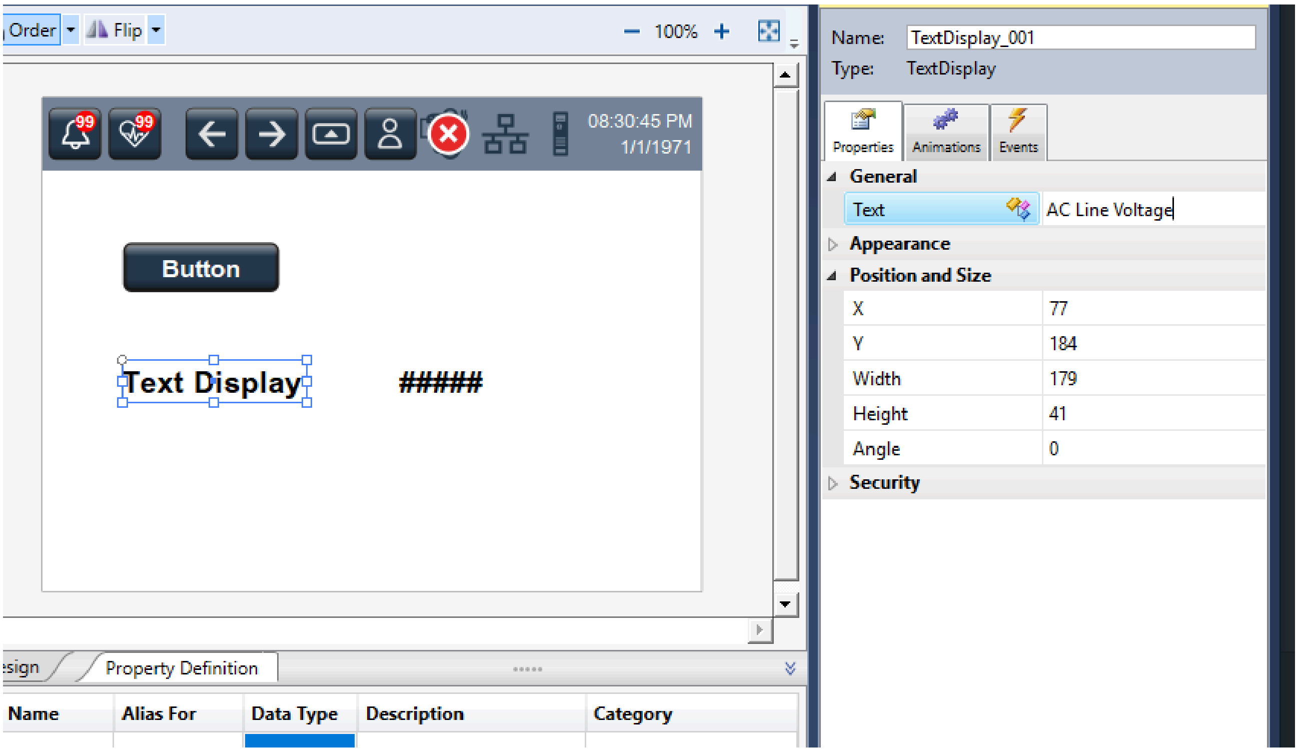


3. In the **CommonControls** toolbox, drag three components onto the screen:
    - **Button**
    - **Numeric Display**
    - **Text Display**

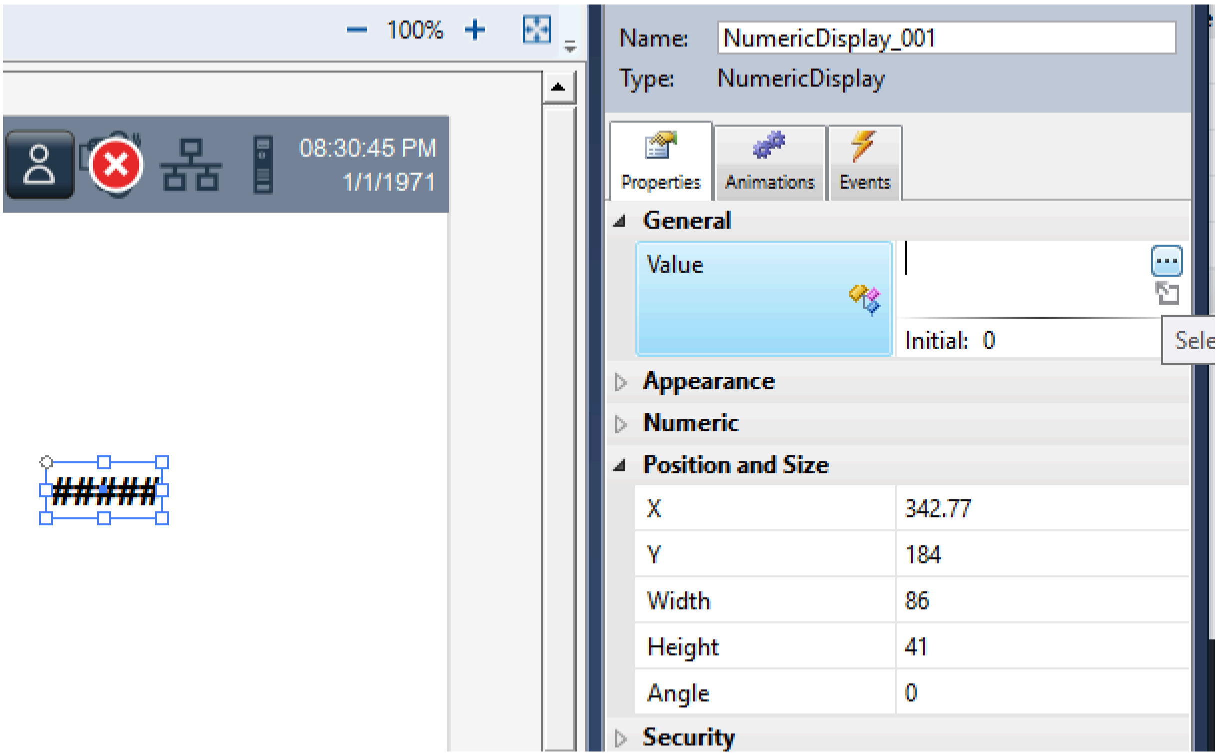

CommonControls
Image
Button
DateTime Dis
Numeric Disp
Numeric Inpu
Slider Horizontal
Slider Vertical
Text Display
Text Input
Button
Standard button. Use Button Behaviors or other Events to define the button actions.
Design
Errors
Descrip

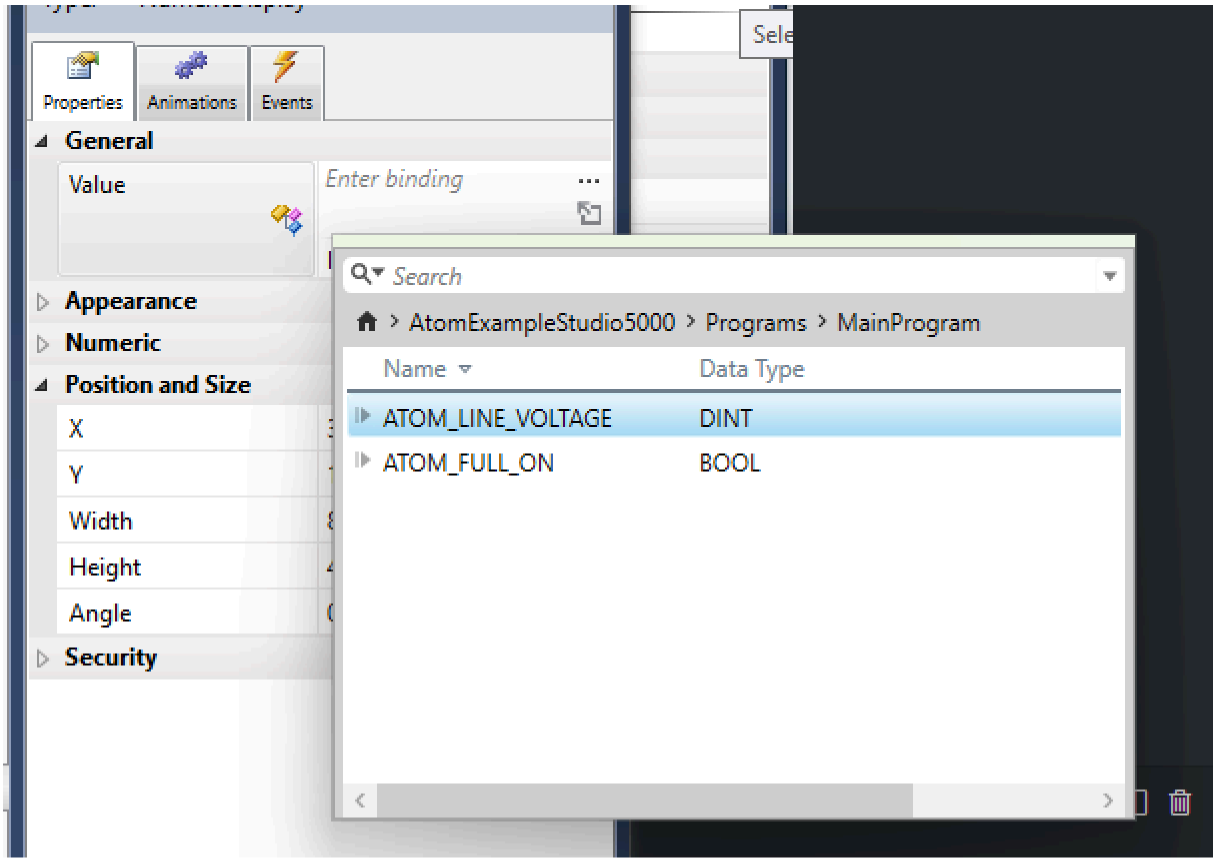

Order
Flip
100%
08:30:45 PM
1/1/1971
Button
Text Display
#####
Property Definition
Name
Alias For
Data Type
Description
Category
Name: TextDisplay_001
Type: TextDisplay
Properties
Animations
Events
General
Text
AC Line Voltage
Appearance
Position and Size
X
77
Y
184
Width
179
Height
41
Angle
0
Security

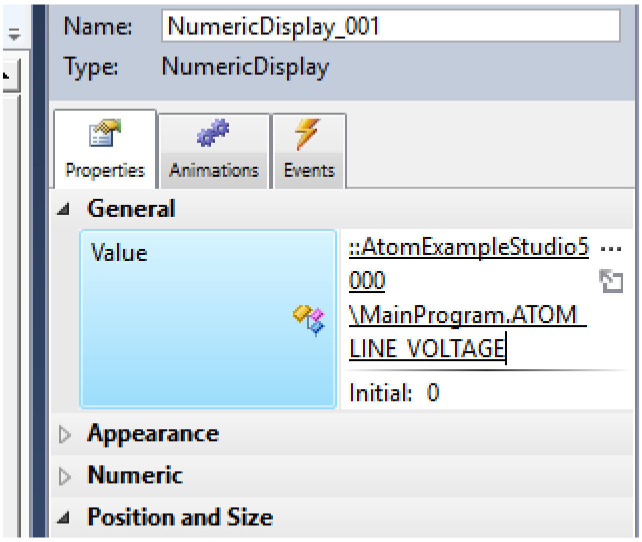
4. Select the **Text Display** component and set the text to `AC Line Voltage`:



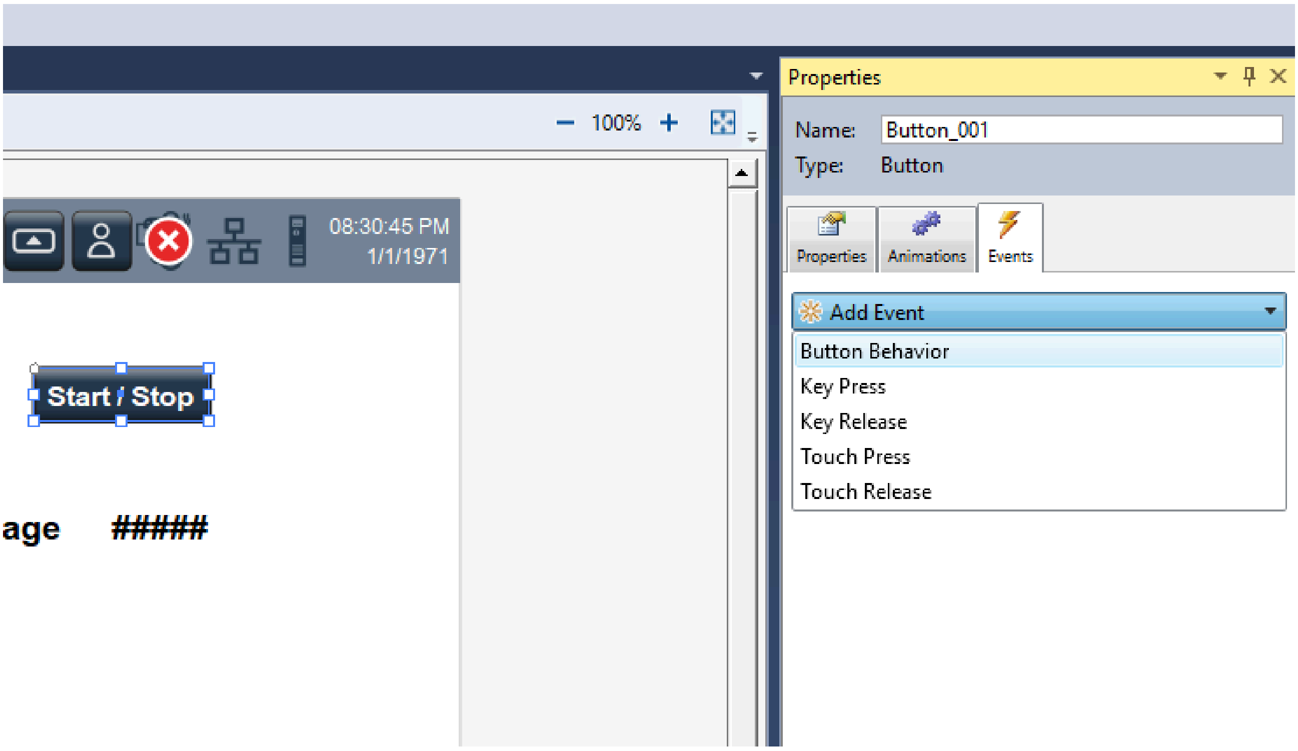
5. Select the **Numeric Display** component and set the **Value** (in the **Properties** panel) to `ATOM_LINE_VOLTAGE`:


100%
08:30:45 PM
1/1/1971
Name: NumericDisplay_001
Type: NumericDisplay
Properties
Animations
Events
General
Value
Initial: 0
Appearance
Numeric
Position and Size
X 342.77
Y 184
Width 86
Height 41
Angle 0
Security

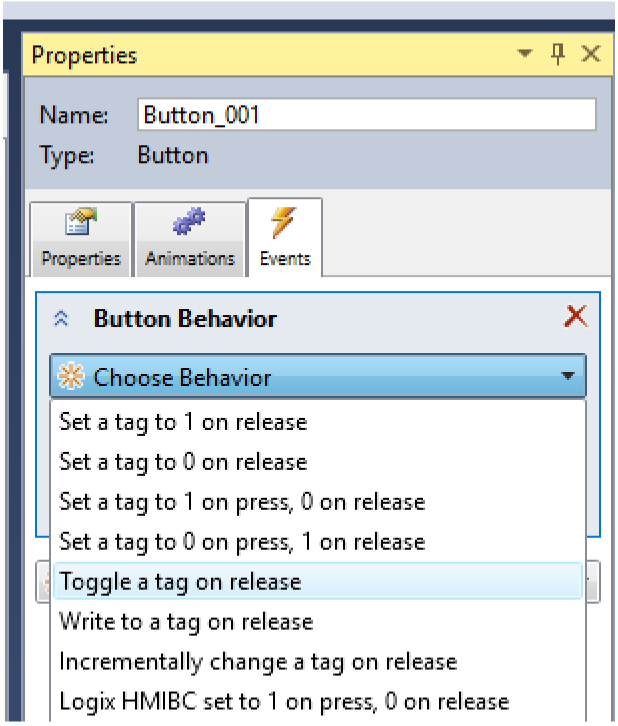

Properties
Animations
Events
General
Value
Enter binding
Appearance
Numeric
Position and Size
X
Y
Width
Height
Angle
Security
Search
AtomExampleStudio5000 > Programs > MainProgram
Name
Data Type
ATOM_LINE_VOLTAGE DINT
ATOM_FULL_ON BOOL

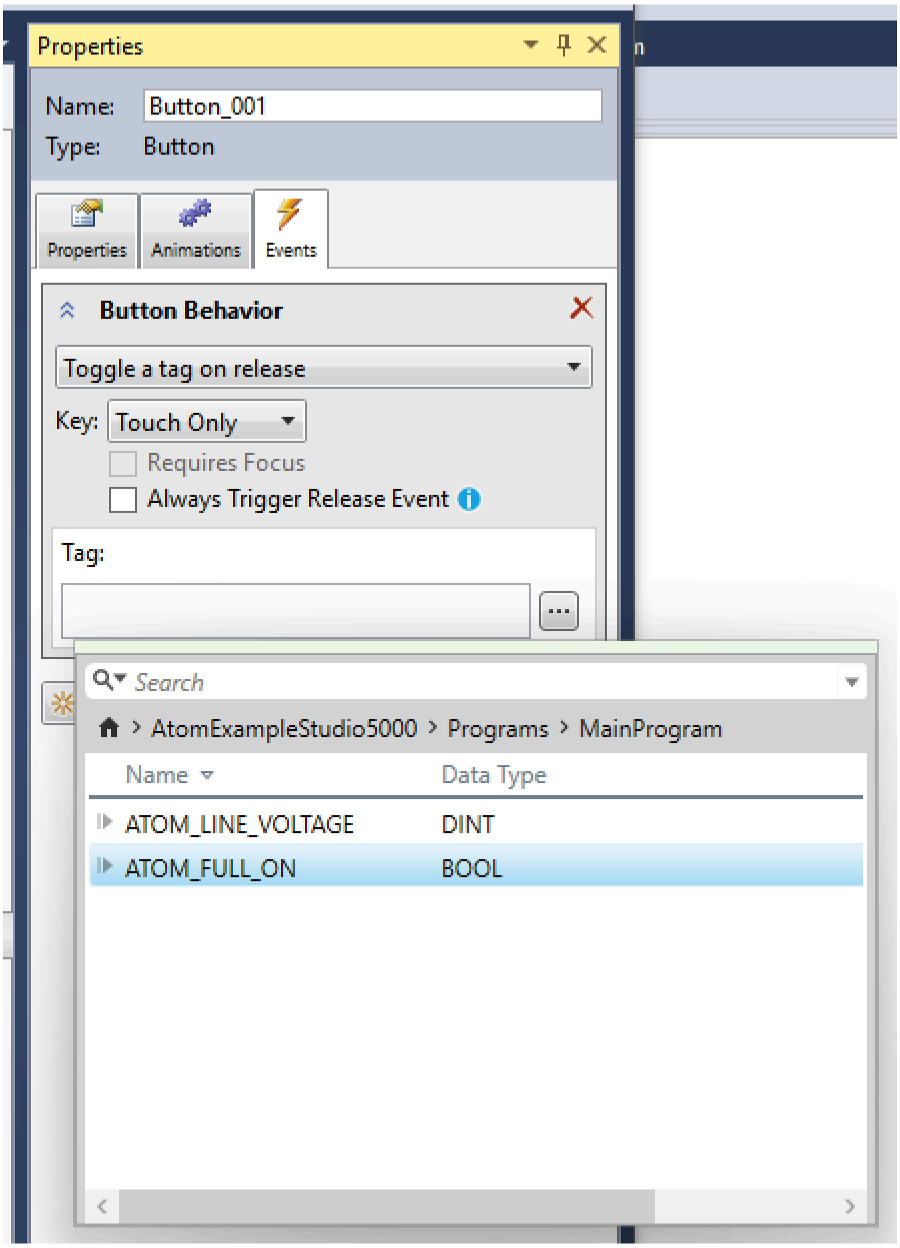


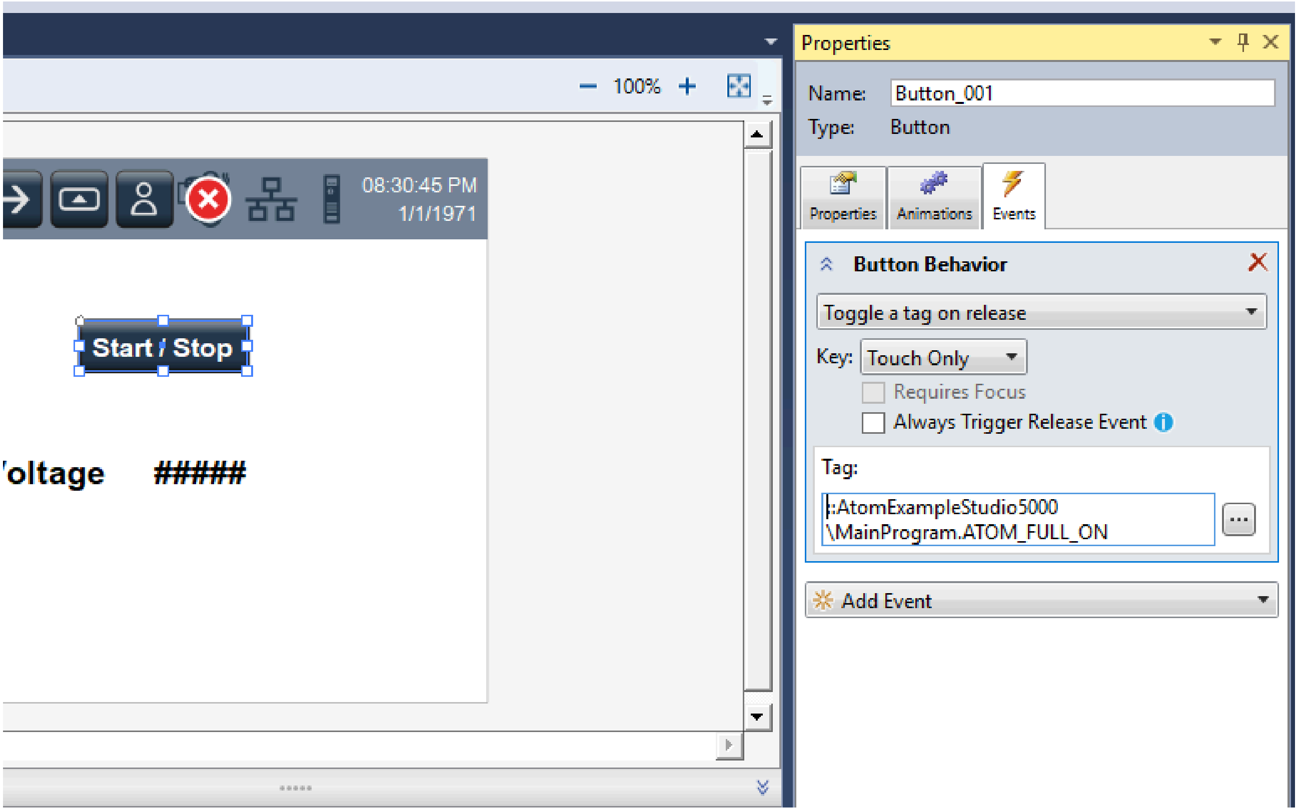
6. Select the *Button* component and set the text to `Start / Stop`. In the **Events** panel, click **Add Event**, **Button Behavior**:



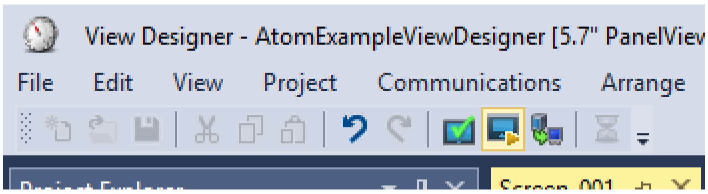
7. Select **Toggle a tag on release** and set the tag to `ATOM_FULL_ON`:


Properties
Name: Button_001
Type: Button
Properties
Animations
Events
Button Behavior
Choose Behavior
Set a tag to 1 on release
Set a tag to 0 on release
Set a tag to 1 on press, 0 on release
Set a tag to 0 on press, 1 on release
Toggle a tag on release
Write to a tag on release
Incrementally change a tag on release
Logix HMIBC set to 1 on press, 0 on release


Properties
Name: Button_001
Type: Button
Properties
Animations
Events
Button Behavior
Toggle a tag on release
Key: Touch Only
Requires Focus
Always Trigger Release Event
Tag:
Search
AtomExampleStudio5000 > Programs > MainProgram
Name
Data Type
ATOM_LINE_VOLTAGE
DINT
ATOM_FULL_ON
BOOL

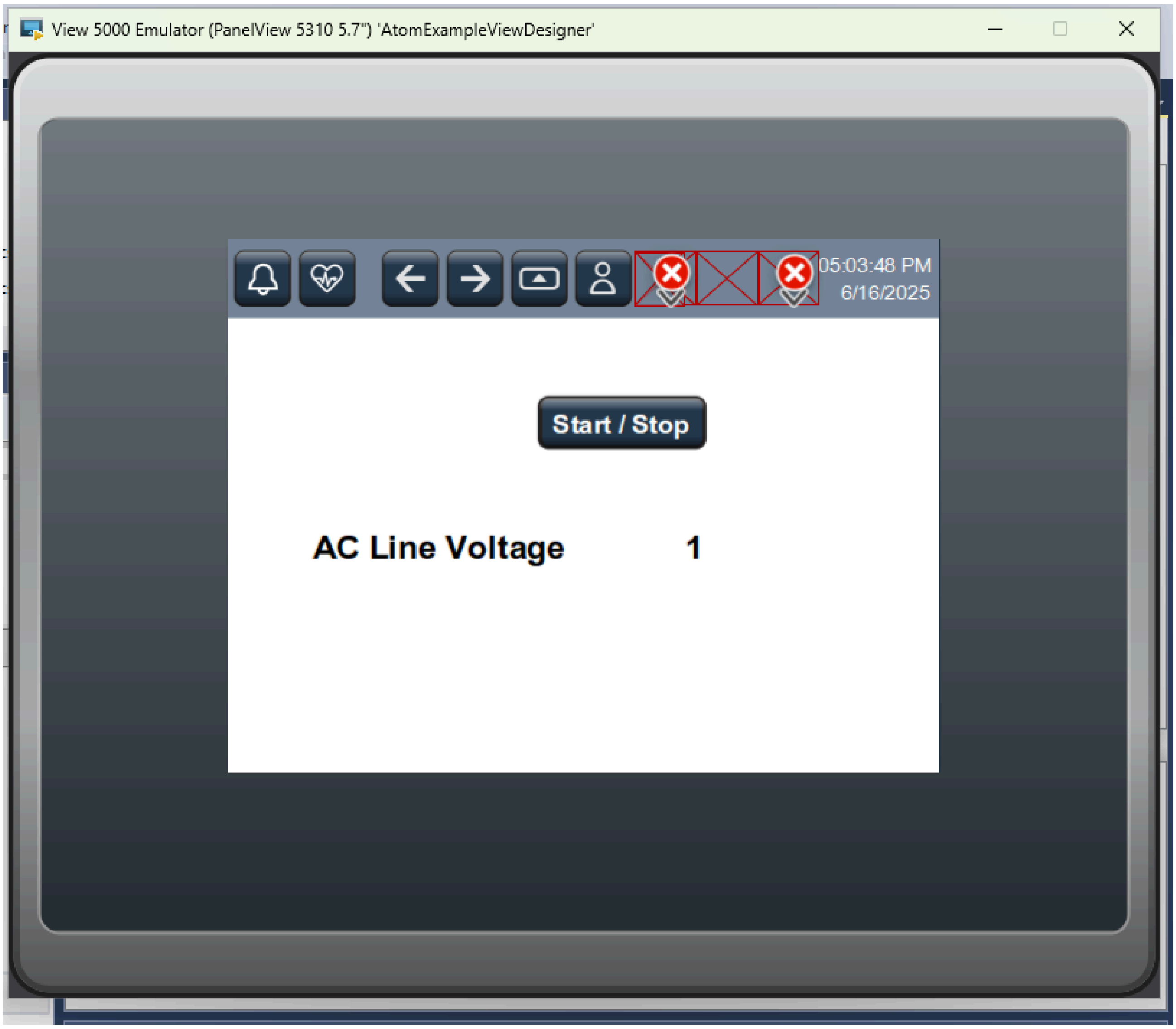


8. Select the **Emulate** button to launch the HMI emulator:

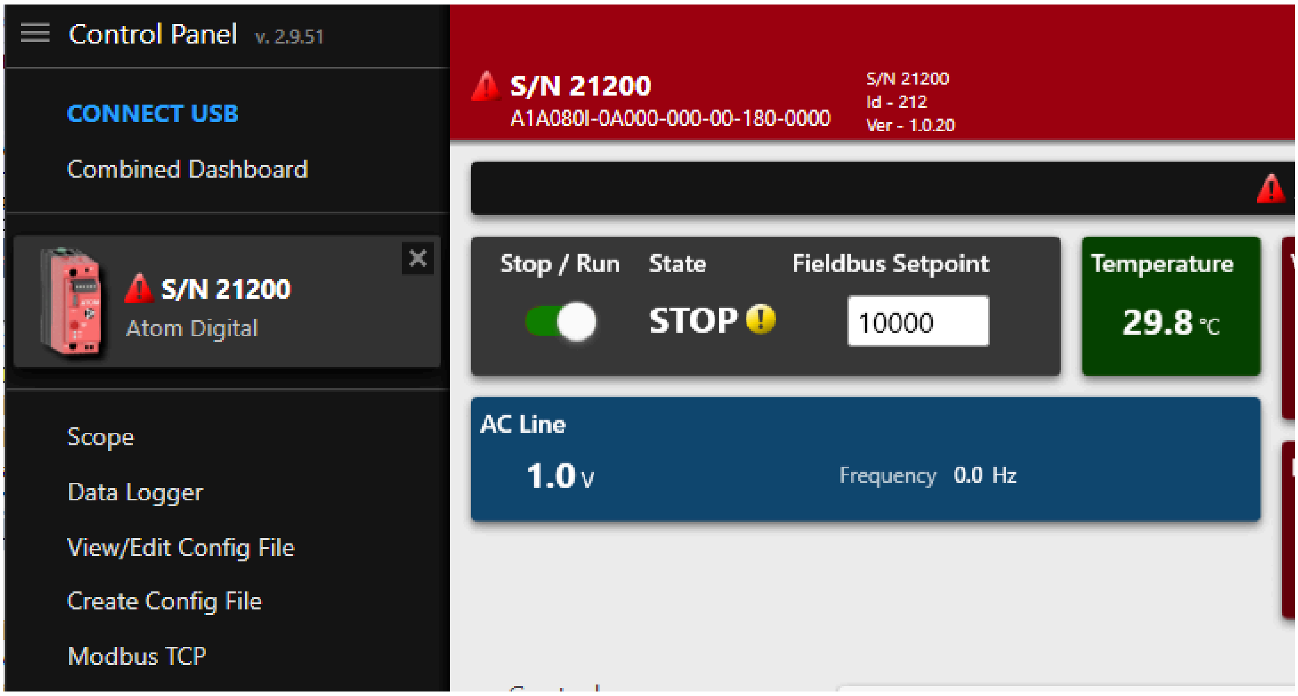


9. Ensure your PLC is in `RUN` if it is not already.

10. In the emulator, you can click `Start / Stop` to toggle ATOM's operation. The **AC Line Voltage** display should show the current line voltage (in tenths of volts (e.g., `2300` for `230.0V`)):



If you are connected to ATOM with Control Panel, you can watch the **Stop / Run** and **Fieldbus setpoint** controls change as you toggle the button in the Rockwell emulator.



Troubleshooting

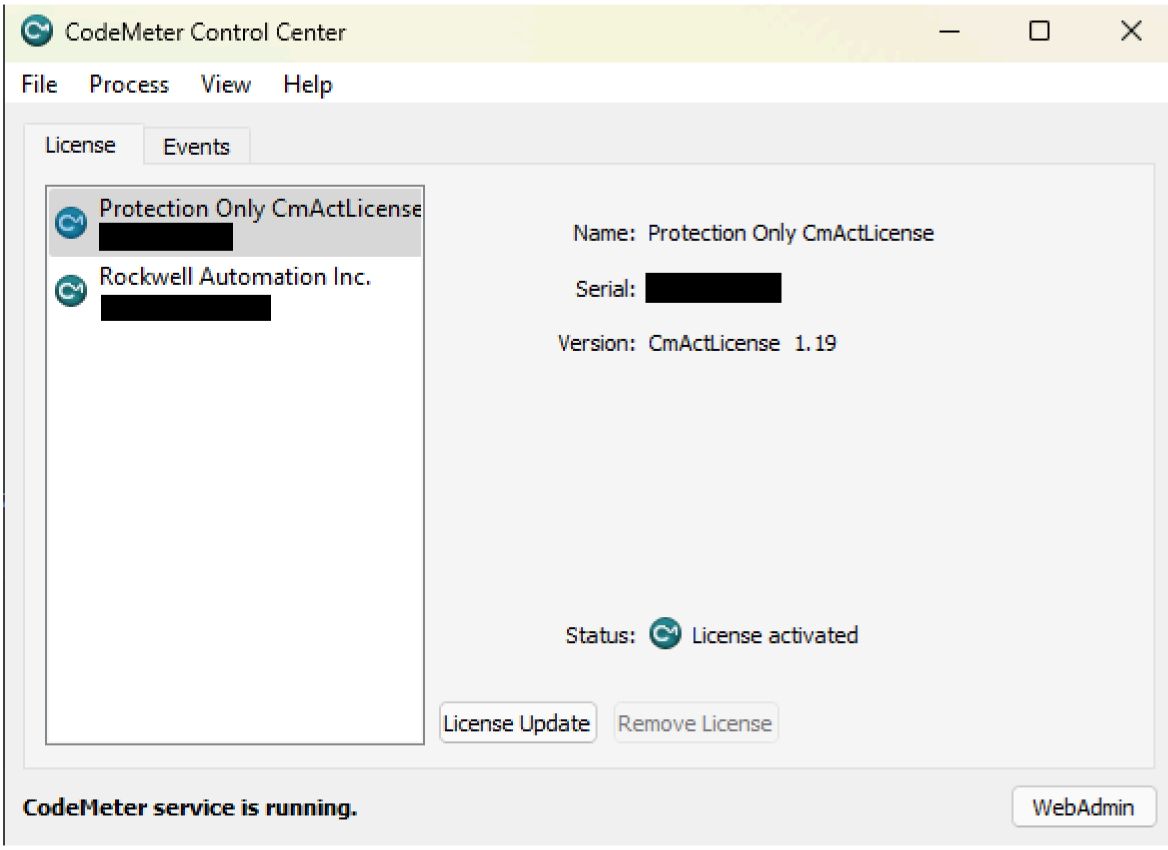
Installation troubleshooting

Activation issues

If you use your PC for multiple PLC environments (like Siemens TIA, Codesys, etc.) you may run into activation issues caused by CodeMeter licenses.

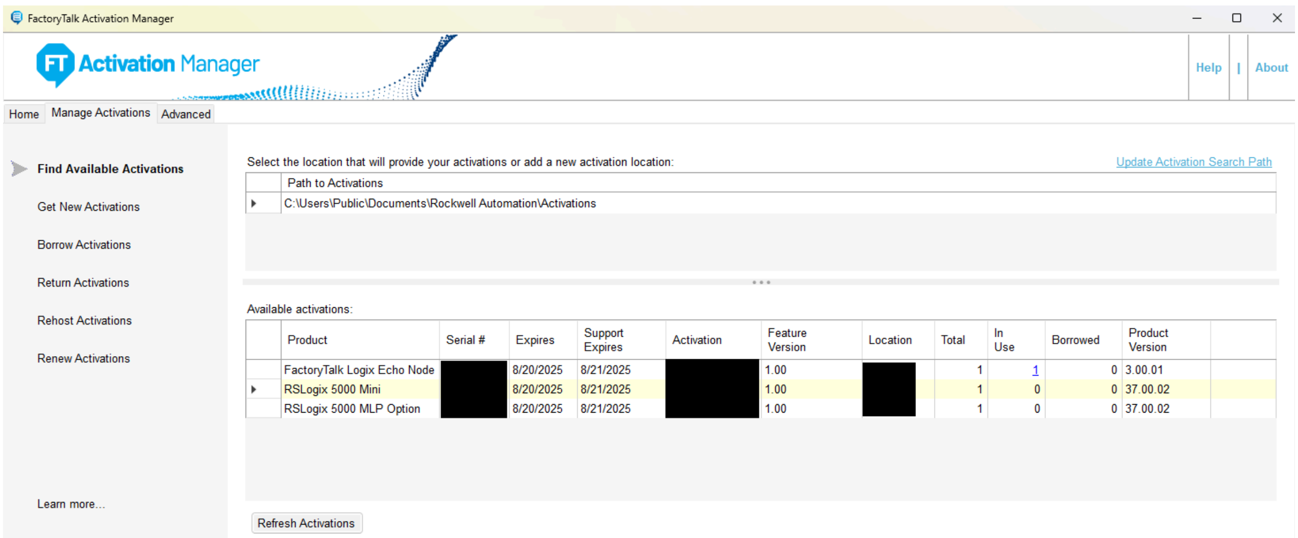
Follow this guide to delete other CodeMeter licenses as Studio 5000 requires exclusive access to the CodeMeter license manager.

Your CodeMeter should look like this:



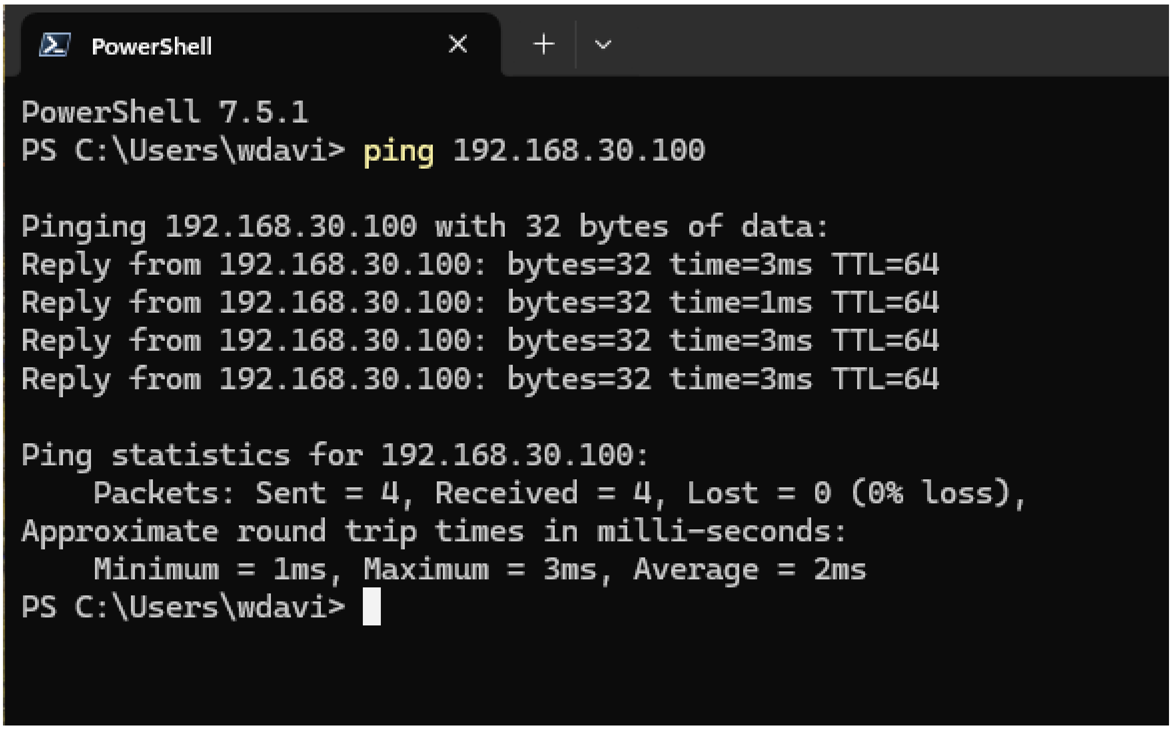
Factory Talk activation

Use **Factory Talk Activation Manager** to ensure your have a valid Studio 5000 license.



Can't connect to PLC or ATOM

Use the `ping` utility on Windows to check if your PC can reach the PLC/ATOM:



If:

- Ping is successful - you have a configuration problem with your PC
- Ping is unsuccessful - you have a hardware configuration, PLC configuration, or ATOM configuration problem.

ATOM / Fieldbus / EtherNet/IP / Codesys

In this tutorial, you'll learn how to use Codesys with the SoftPLC emulator to connect to ATOM using EtherNet/IP and perform some basic operations and monitor data. You can follow along using the SoftPLC emulator or your own PLC.

We provide examples for both ladder logic and structured text.

If you haven't yet, please review ATOM's EtherNet/IP Profile.

If you'd like to skip the tutorial, you can download a completed example project:

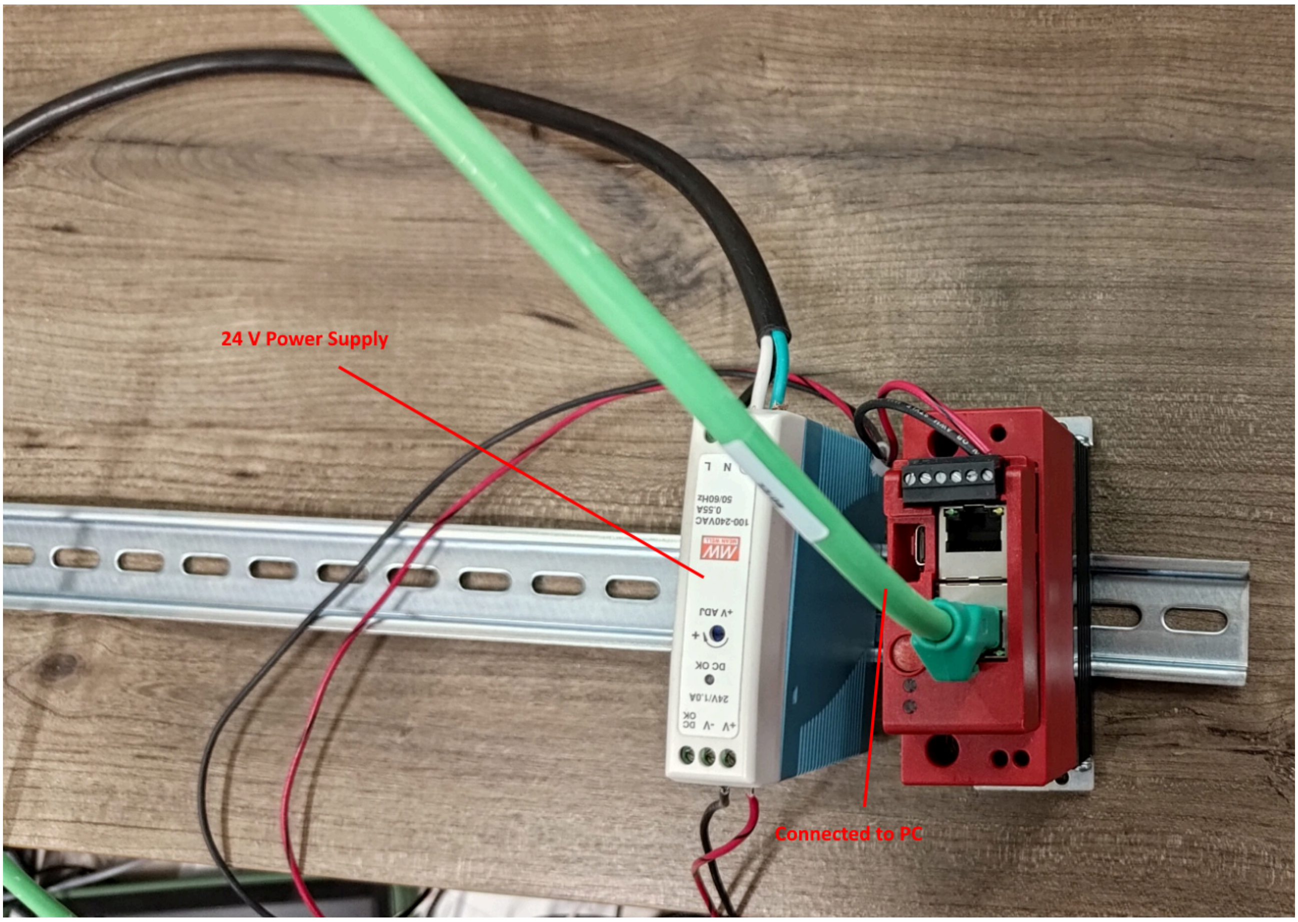
- Download ATOM_Codesys_LadderLogic_Example.zip
- Download ATOM_Codesys_StructuredText_Example.zip

Prerequisites

1. Install Codesys
2. Download ATOM's EDS file

Hardware setup

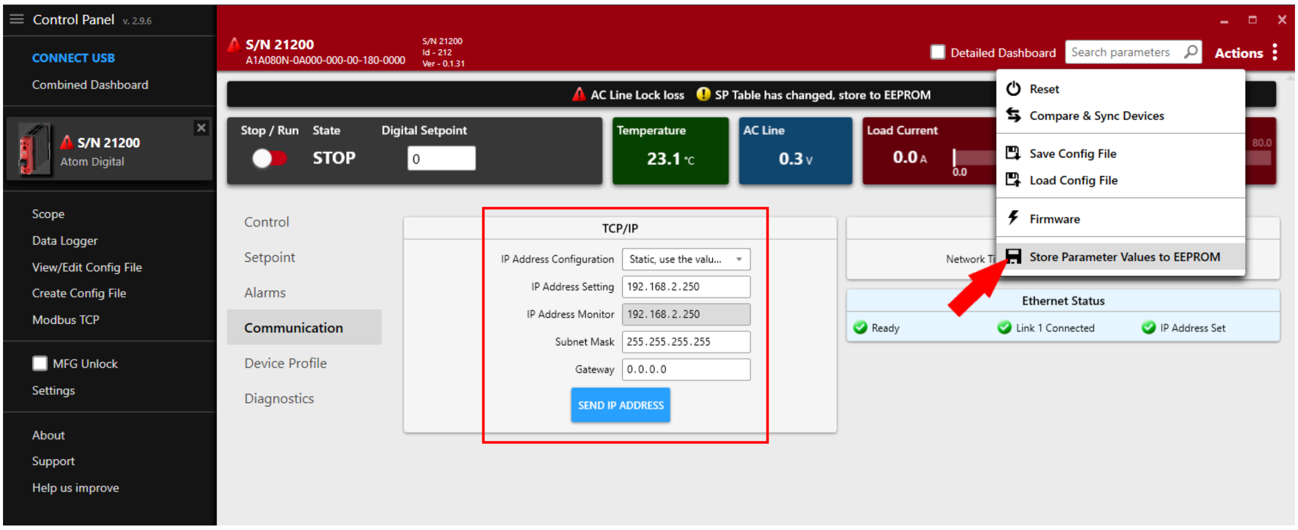
Connect 24V to your PLC and Atom unit with the provided power cable. Connect Atom to your PC with an Ethernet cable.



> **INFO**
>
> To simplify this diagram, we have not connected a load to Atom. You may connect a load or leave it disconnected, either way is fine for the purposes of this tutorial.
>
> If you do not connect a load, you can still verify your PLC is working by connecting a USB cable to Atom and using Control Panel to watch the parameters change/verify the PLC is receiving the correct monitor data.

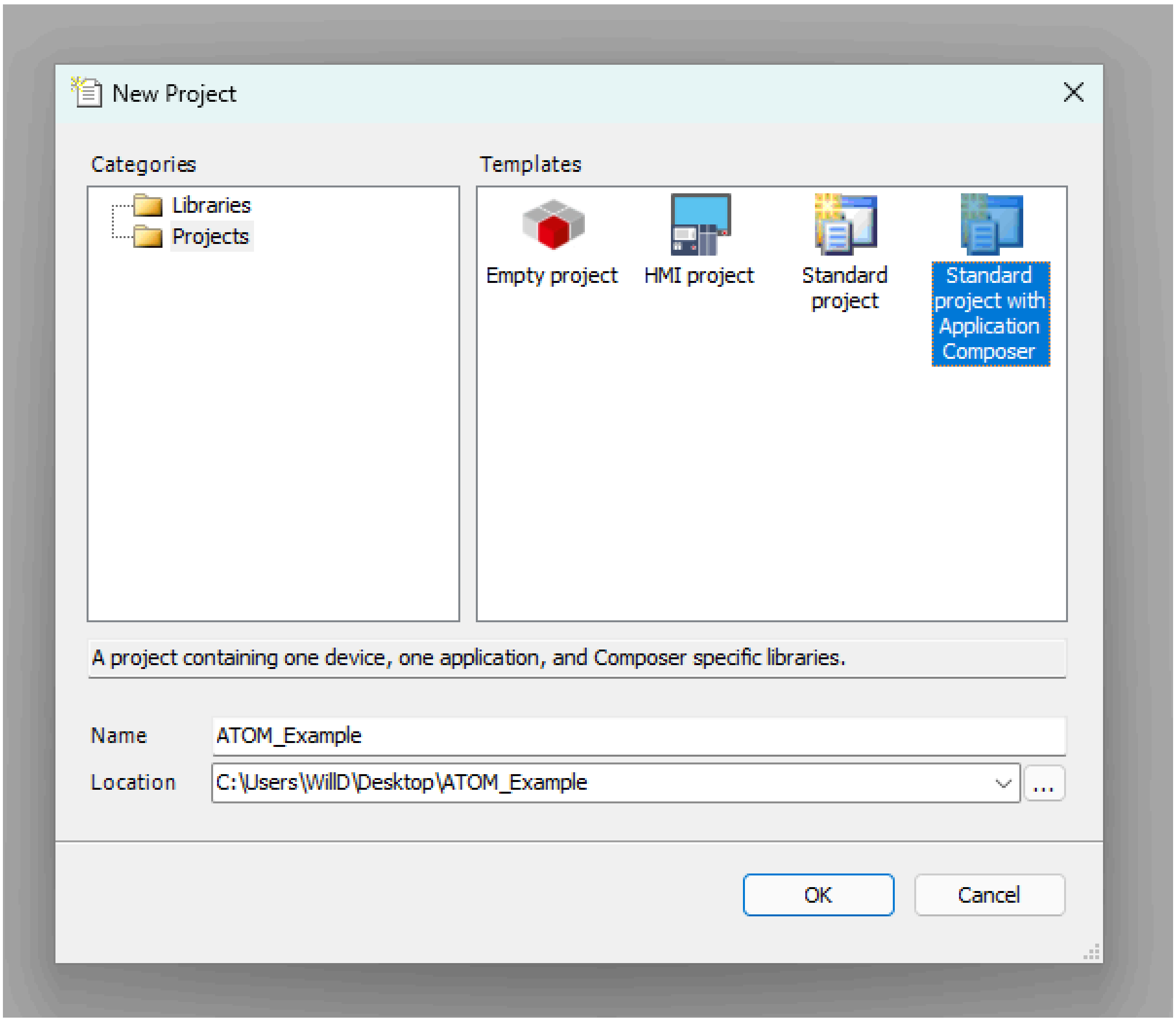
Configuring Atom network settings

Connect your Atom unit to your PC using a USB cable. Open Control Panel and update your Atom's communication parameters. When you're finished, click **Send IP Address**, then go to **Actions** in the upper right and select **Store Parameter Values to EEPROM**:

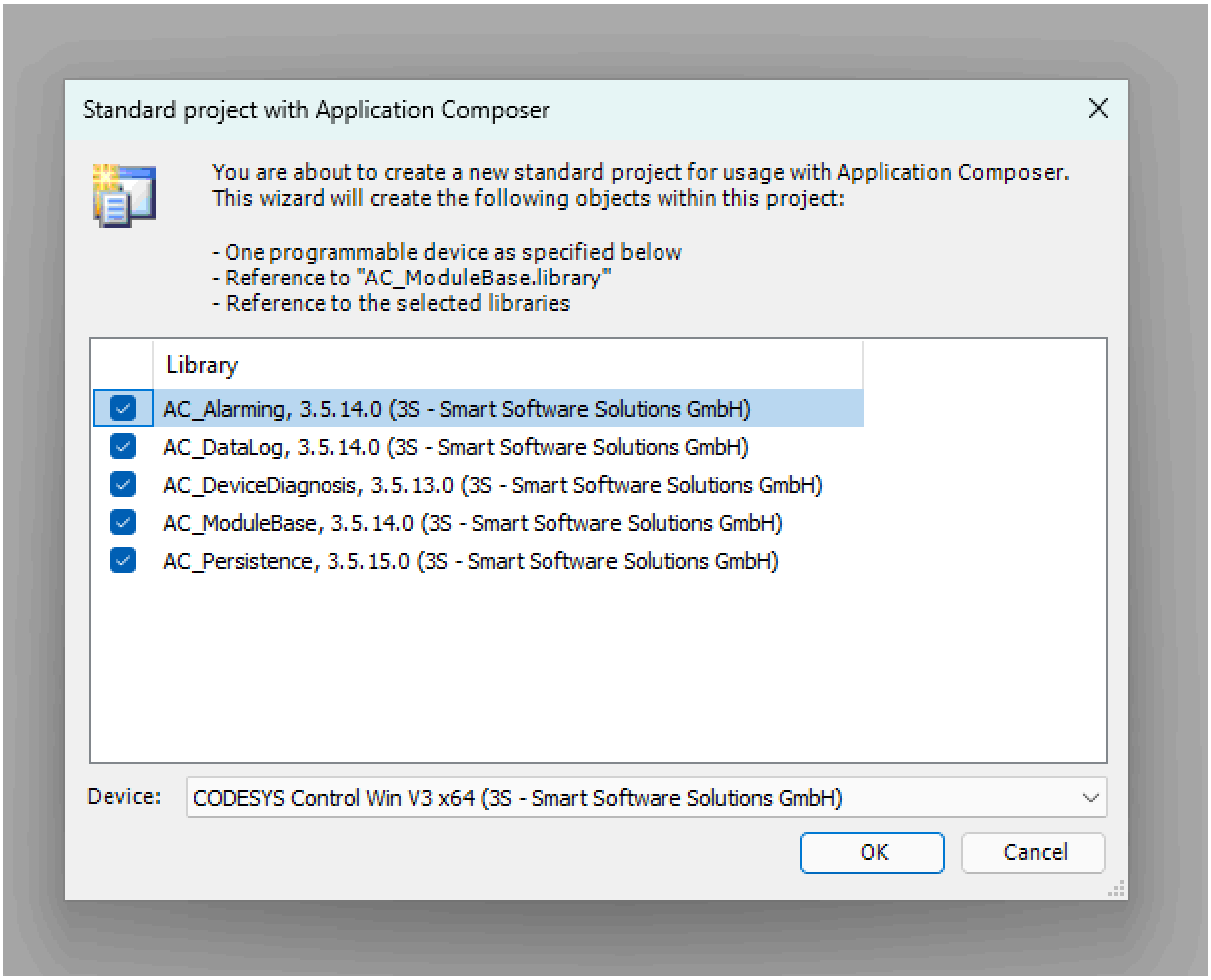


Create a Codesys project

Create a new Codesys project using the **Standard project with Application Composer** template:



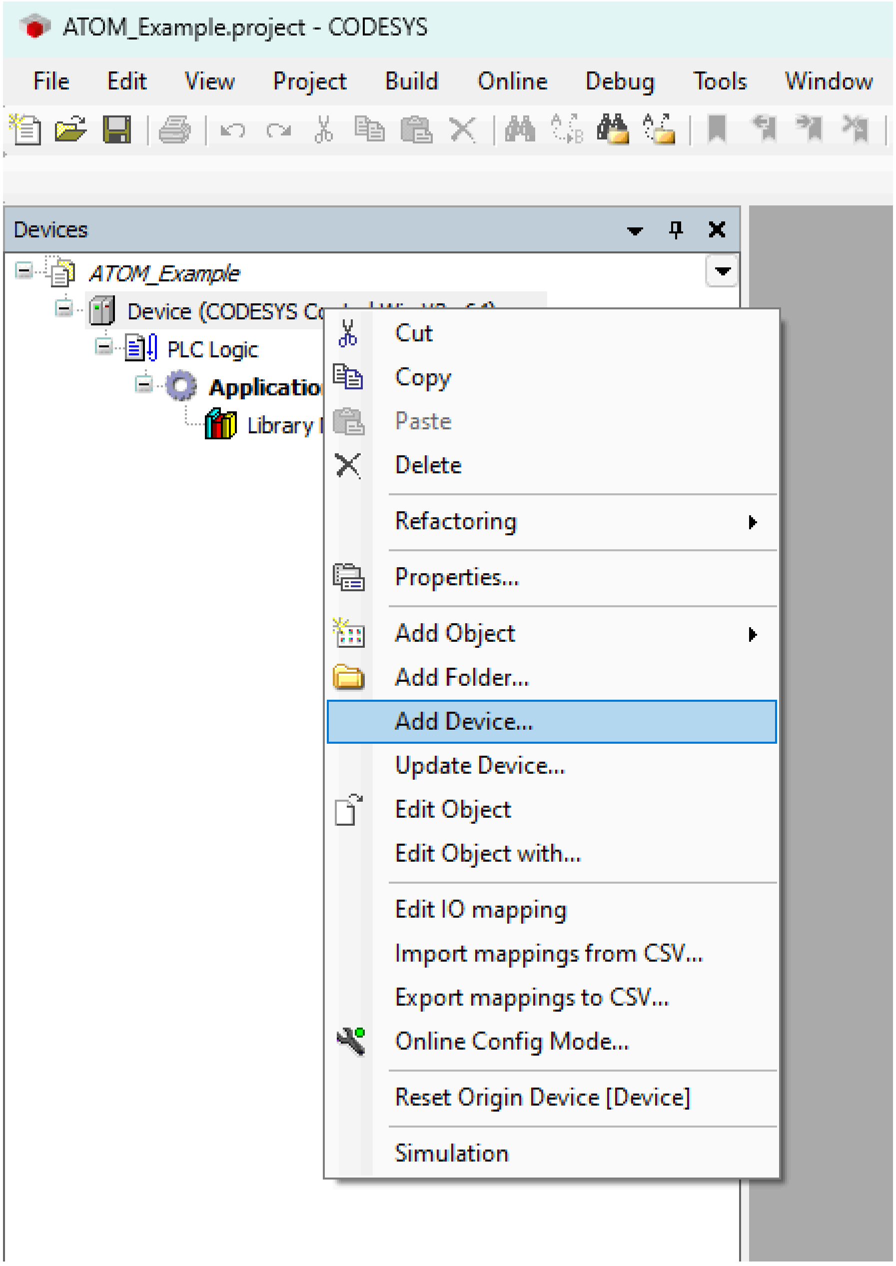
Check each library to include it in the project and select **CODESYS Control WIN V3 x64** as the device:



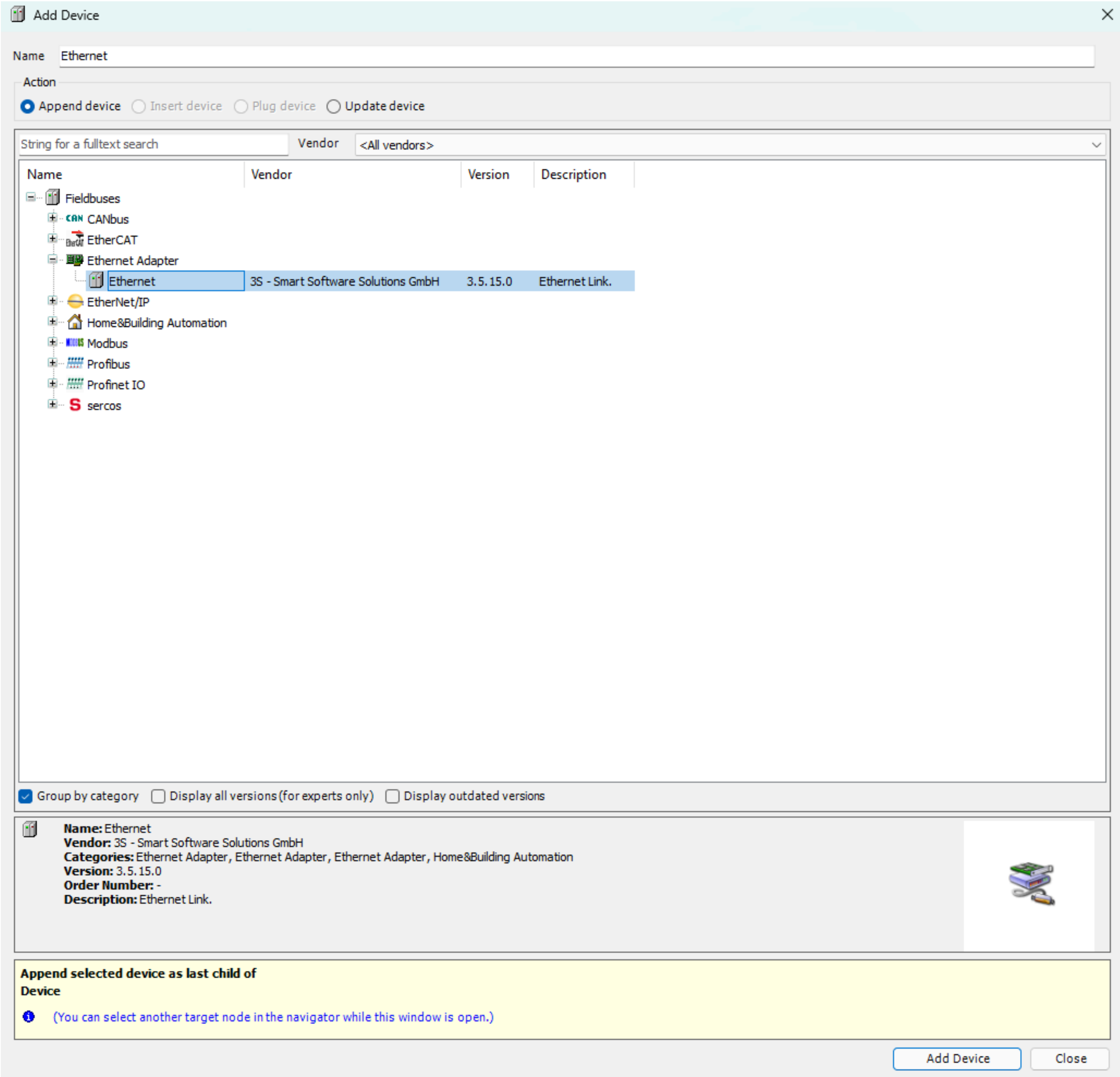
Adding an EtherNet/IP Scanner

Next we'll add an EtherNet/IP Scanner module. This allows the PLC to discover EtherNet/IP devices on the network (in our case, ATOM) and establish a connection with them.

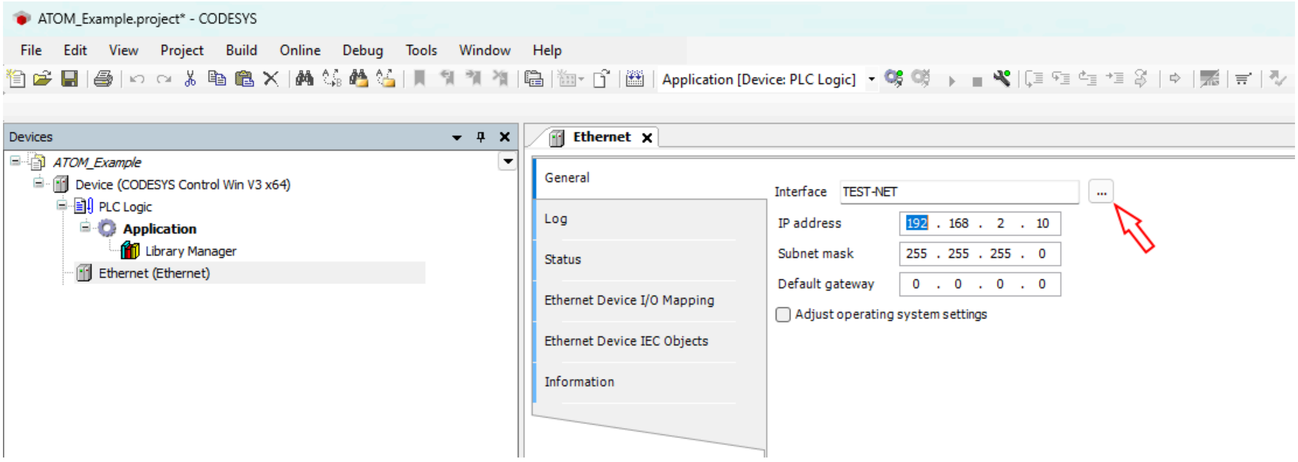
First, right click **Device** and select **Add Device**:



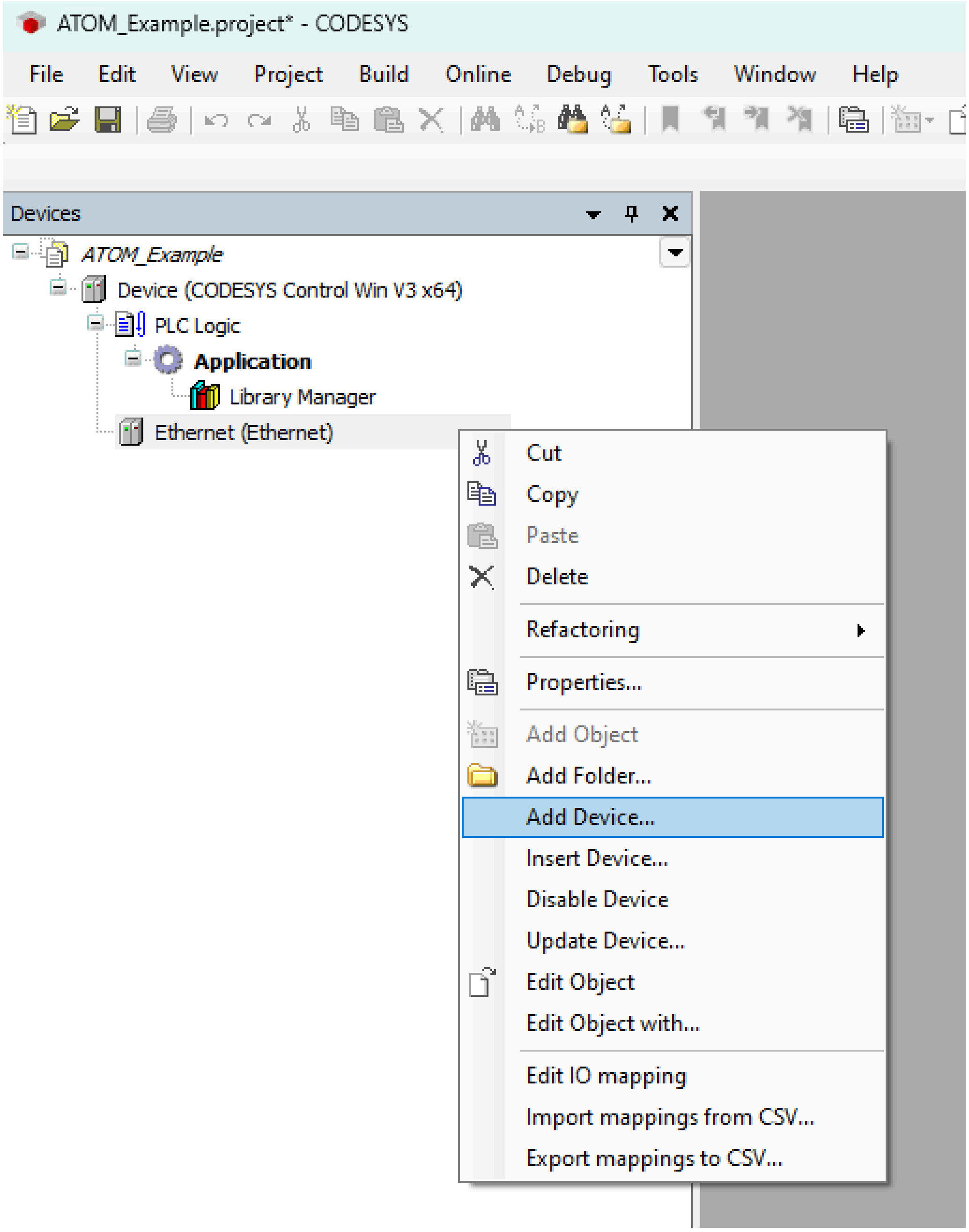
Next, expand **Ethernet Adapter** and select **Ethernet**, then click **Add Device**:



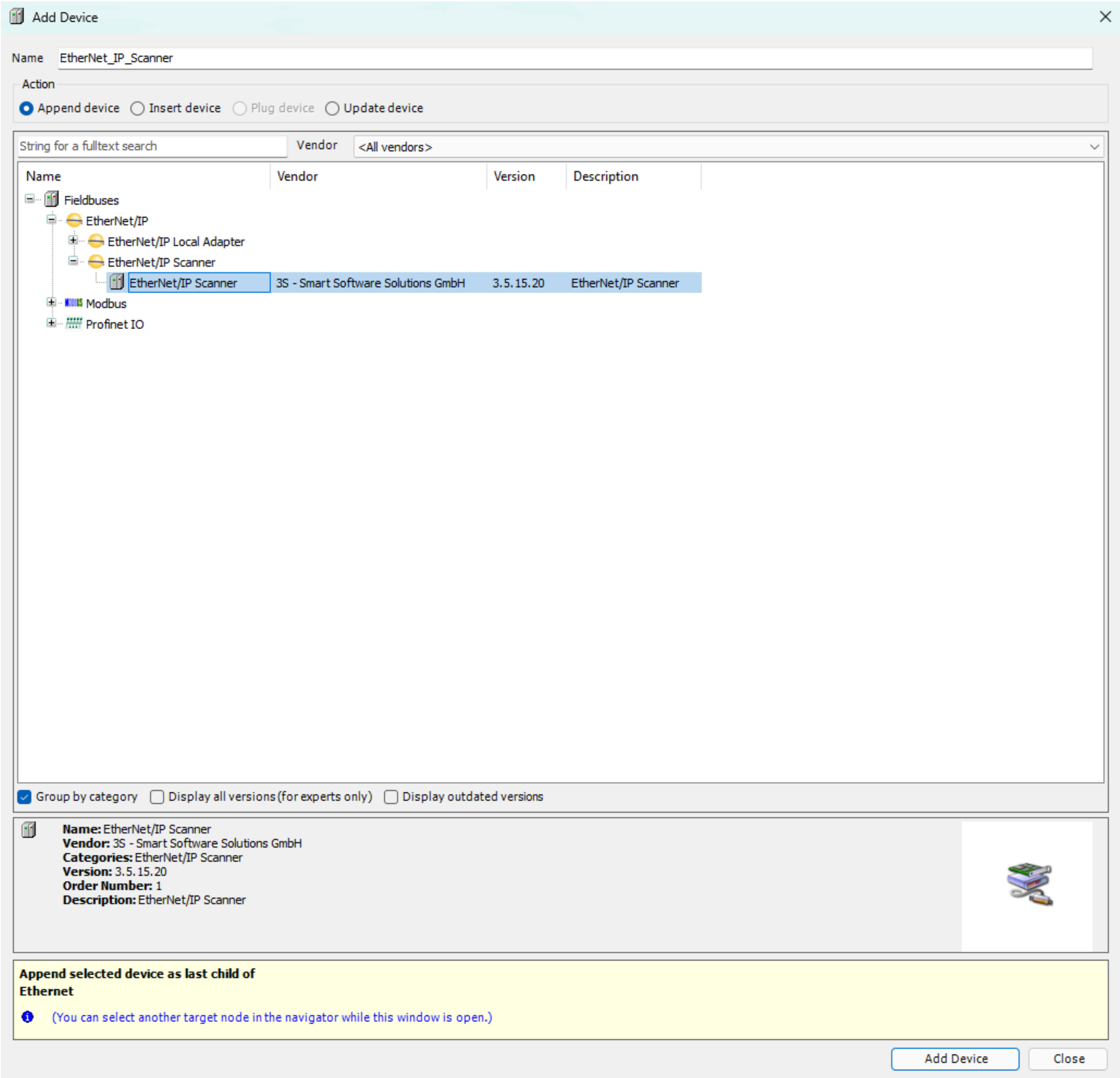
The newly added **Ethernet** device will now appear in the device tree. Double click **Ethernet (Ethernet)** to open its configuration tab. Within the **General** configuration tab, use the button indicated by the red arrow to select the network interface of the host machine that will be used to communicate with ATOM. In our case, we have a `TEST-NET` interface but this will be different for you.



Next, right click **Ethernet (Ethernet)** and select **Add Device**:



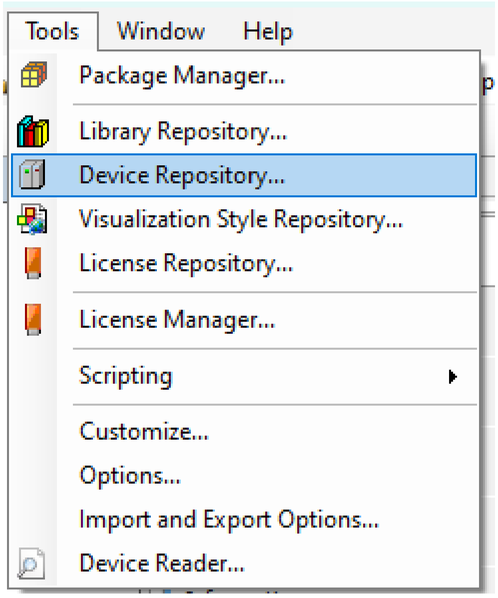
Expand **EtherNet/IP Scanner**, select **EtherNet/IP Scanner**, then click **Add Device**:



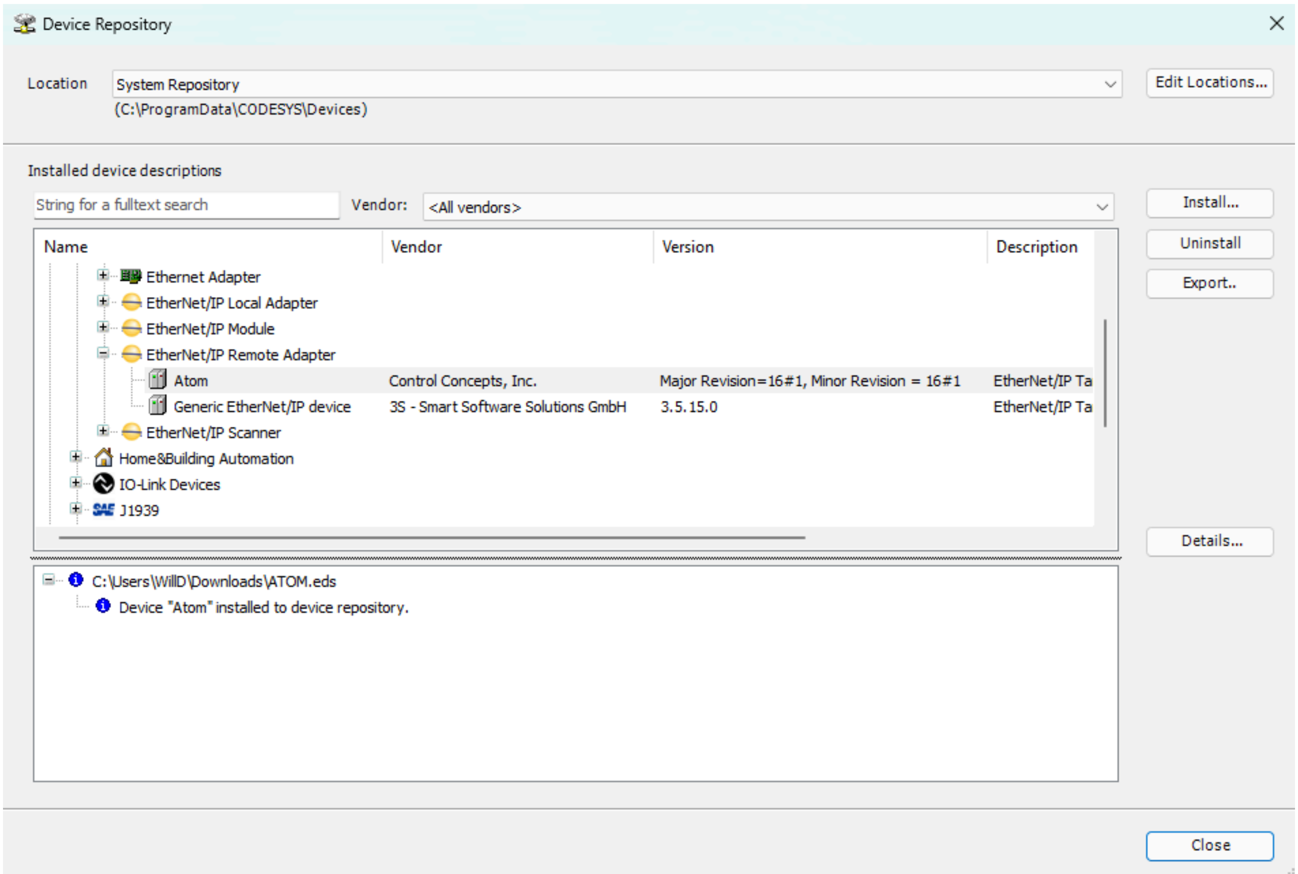
Your device tree should update to include the **EtherNet/IP Scanner** device.

Adding ATOM to the scanner

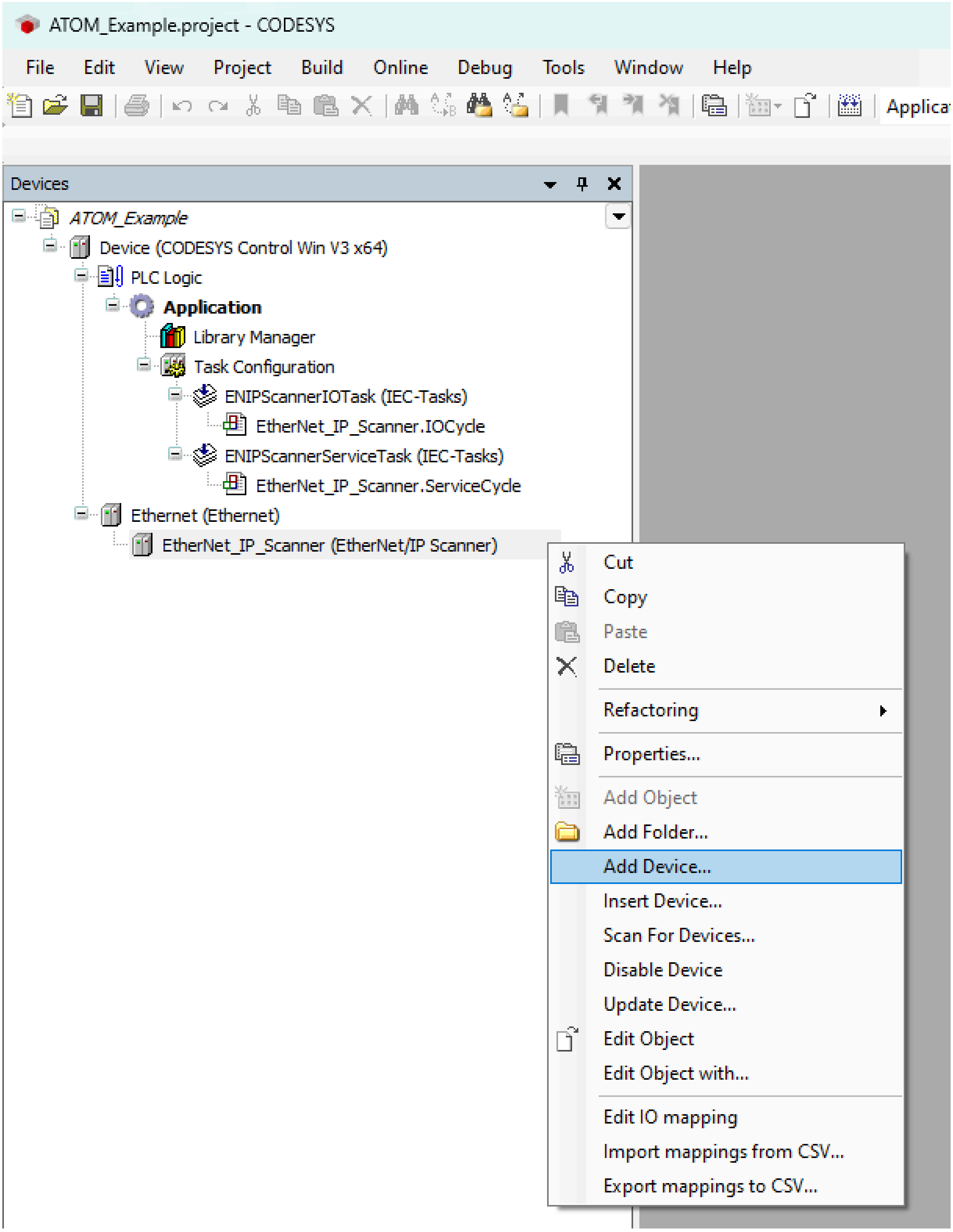
First, we'll import ATOM's EDS file you downloaded earlier into our Codesys device library. Open the tools menu and select **Device repository**:



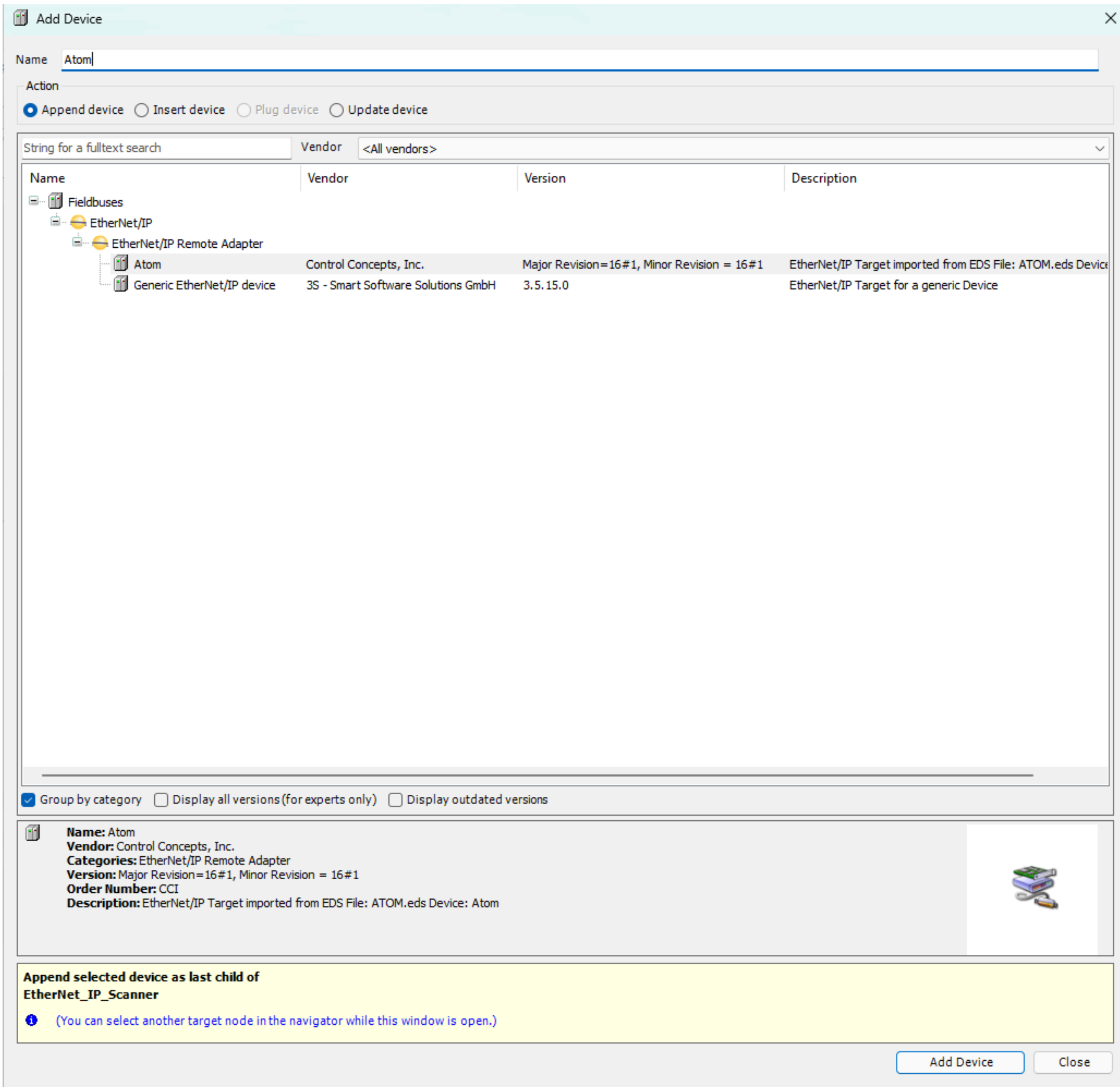
Next, click **Install** and select the `ATOM.eds` file. After you click install, **Atom** will appear under the **EtherNet/IP Remote Adapter** category. Click **Close** to dismiss the dialog:



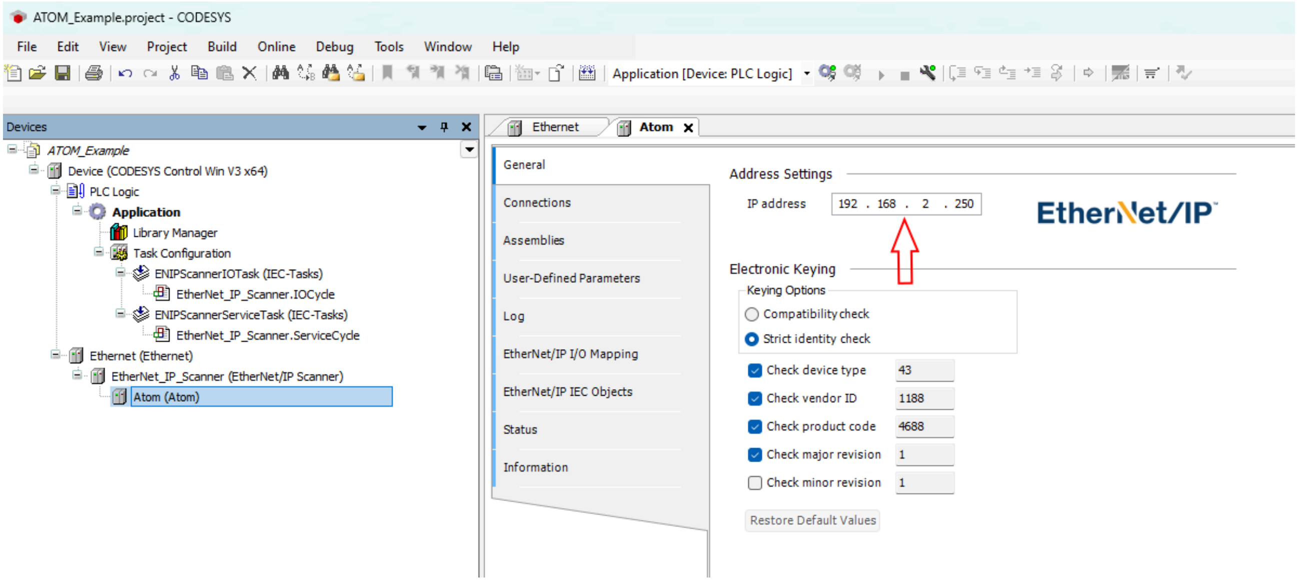
Now, we'll add ATOM to the scanner. Right click **EtherNet/IP Scanner (EtherNet/IP Scanner)** and select **Add Device**:



Expand **EtherNet/IP Remote Adapter** and select **Atom**, then click **Add Device**:



Finally, double click **Atom (Atom)** to open its configuration tab. In the **General** tab, set the **IP Address** to the IP address of your ATOM device:



Create a program

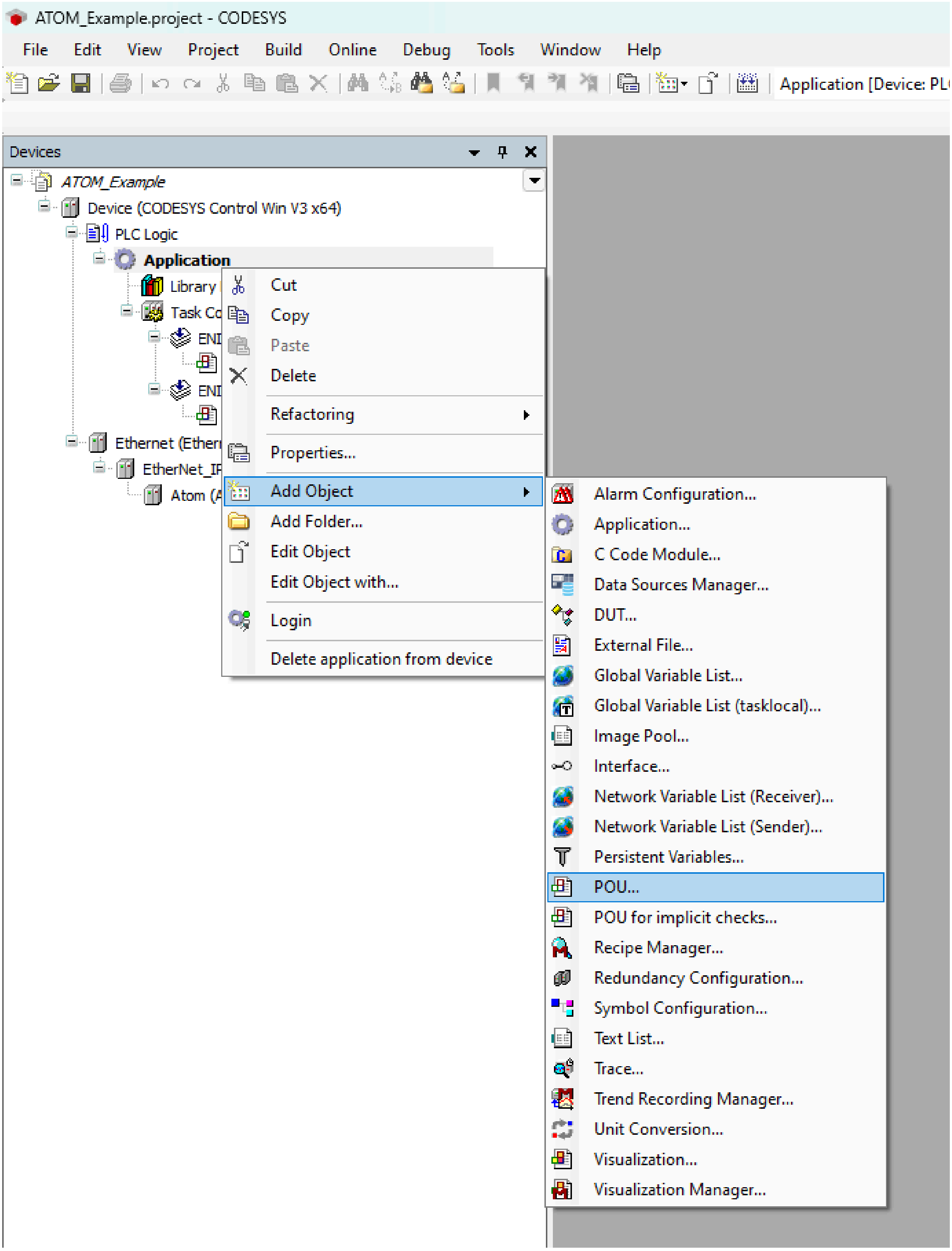
Next, we'll create a PLC program. We provide examples for both ladder logic and structured text:

- Program with ladder logic
- Program with structured text

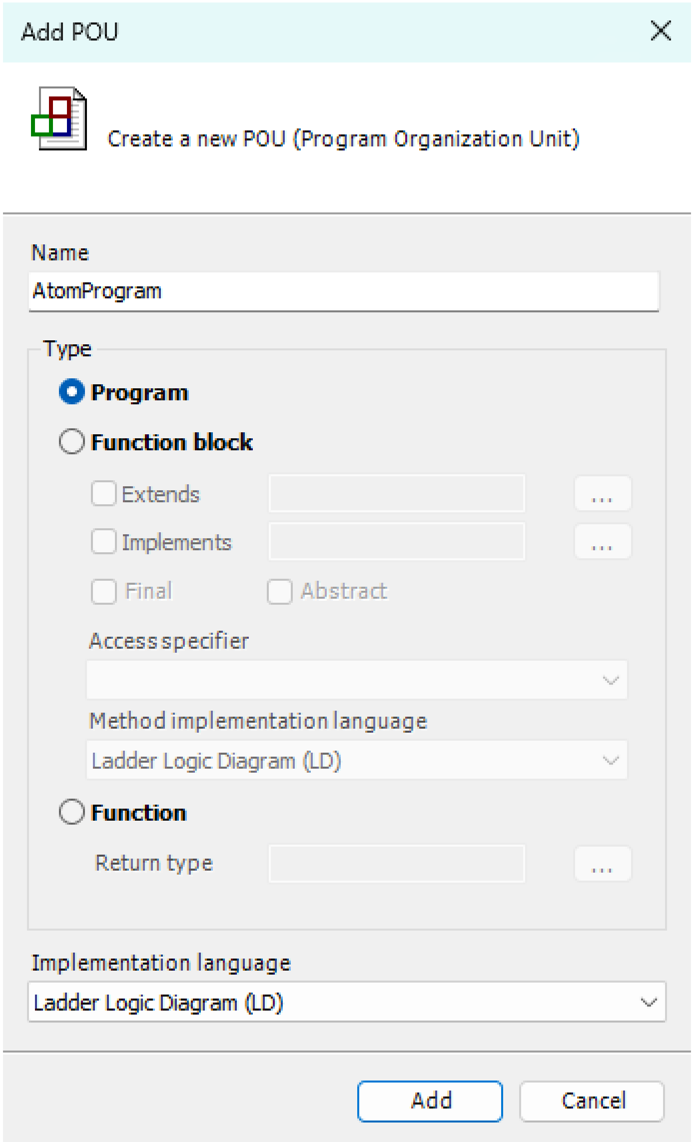
Example: Ladder logic

Creating the program

Right click **Application** and select **Add Object > POU**:


ATOM_Example.project - CODESYS
File
Edit
View
Project
Build
Online
Debug
Tools
Window
Help
Devices
ATOM_Example
Device (CODESYS Control Win V3 x64)
PLC Logic
Application
Library
Task Co
Ethernet (Ether
EtherNet_IP
Atom (A
Cut
Copy
Paste
Delete
Refactoring
Properties...
Add Object
Add Folder...
Edit Object
Edit Object with...
Login
Delete application from device
Alarm Configuration...
Application...
C Code Module...
Data Sources Manager...
DUT...
External File...
Global Variable List...
Global Variable List (tasklocal)...
Image Pool...
Interface...
Network Variable List (Receiver)...
Network Variable List (Sender)...
Persistent Variables...
POU...
POU for implicit checks...
Recipe Manager...
Redundancy Configuration...
Symbol Configuration...
Text List...
Trace...
Trend Recording Manager...
Unit Conversion...
Visualization...
Visualization Manager...

Set the name to `AtomProgram` and select **Ladder Diagram (LD)** as the Implementation language:



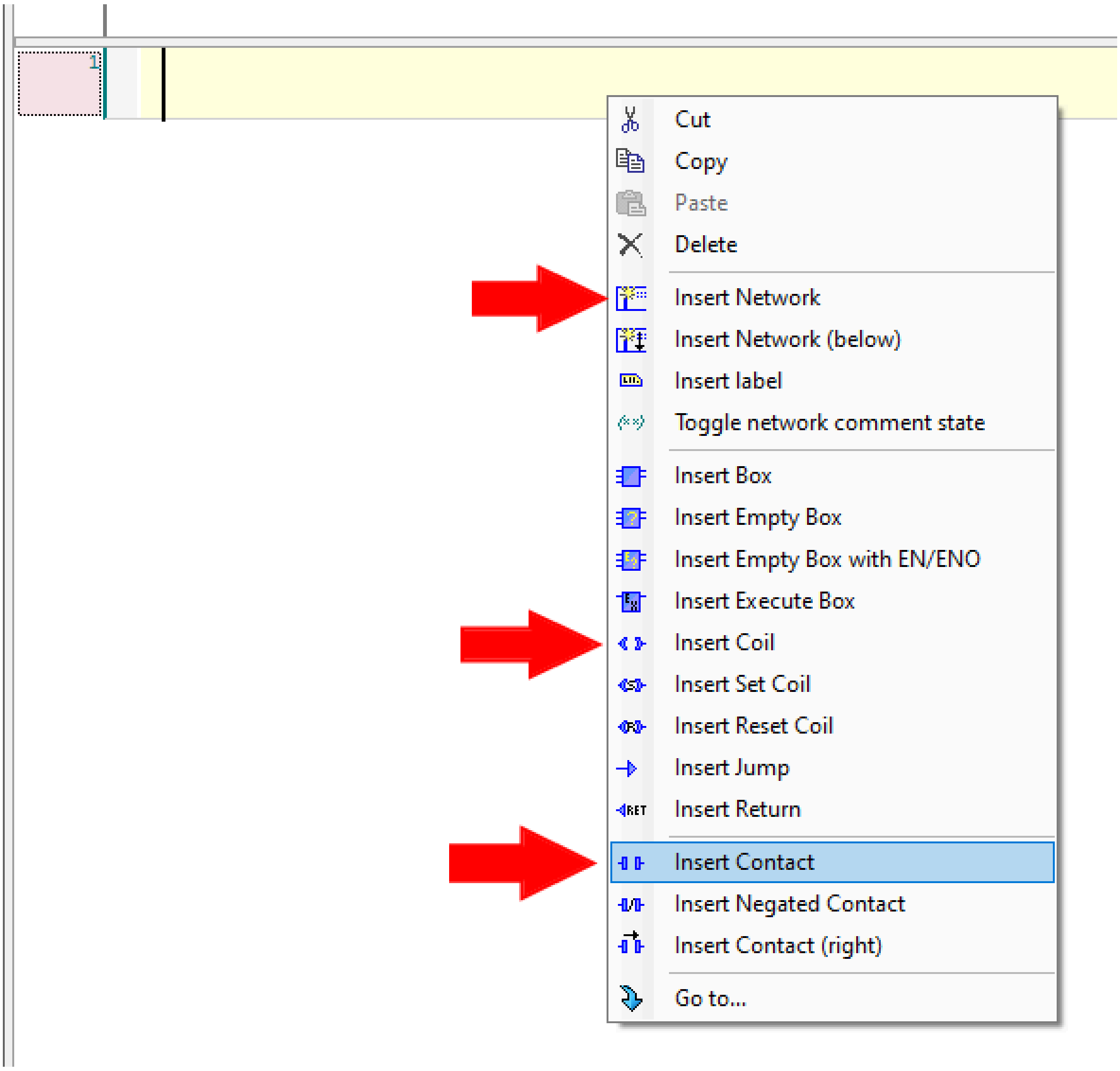
Copy the following code into the top panel of the **AtomProgram** editor:

```
PROGRAM AtomProgram
VAR

RUN_SWITCH: BOOL;
SETPOINT: DINT;
TEMP: REAL;

ATOM_OUTPUT_SETPOINT: DINT;
ATOM_OUTPUT_RUN_ENABLE: BOOL;
ATOM_INPUT_TEMP: REAL;

END_VAR
```

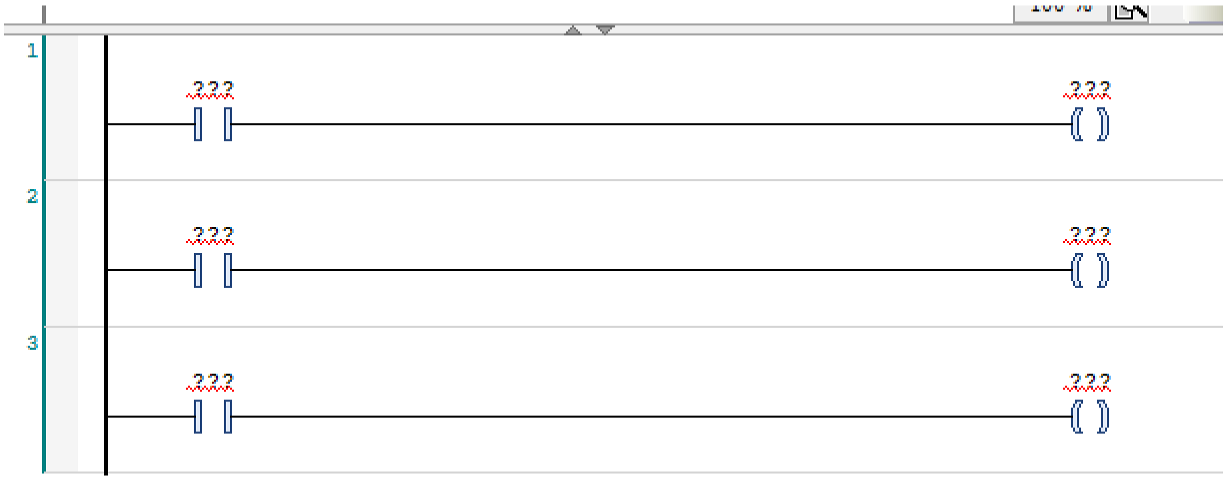
After you've copied the code over, the editor for **AtomProgram** should look like this:

In the bottom panel of the editor, we'll create a simple ladder logic program using the variables we just added above.

1. Create **3** networks total by right-clicking and selecting **Insert Network**
2. For each network, right click and insert **one** contact and **one** coil



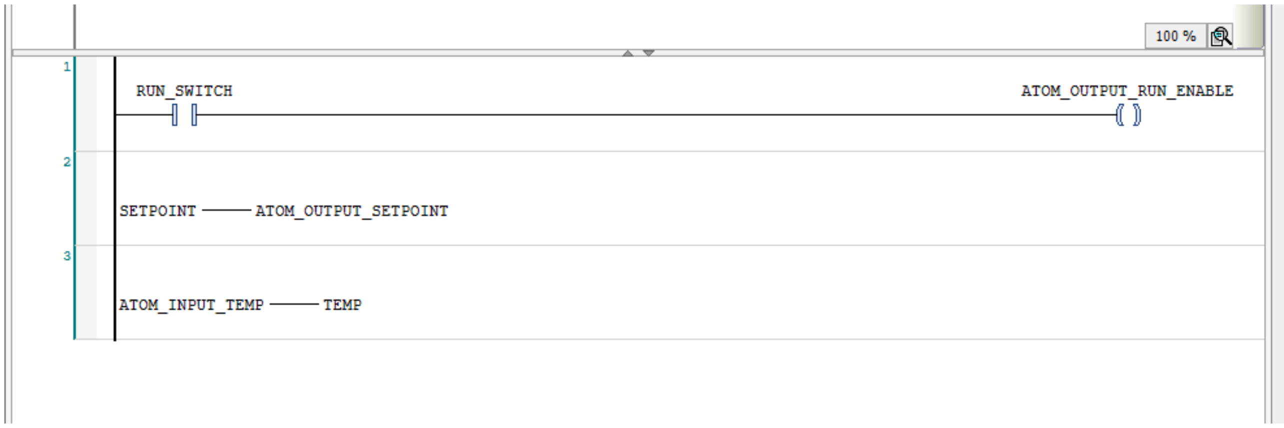
After you're finished, your ladder logic program should look like:



For each rung, replace the `???` with the corresponding variables:

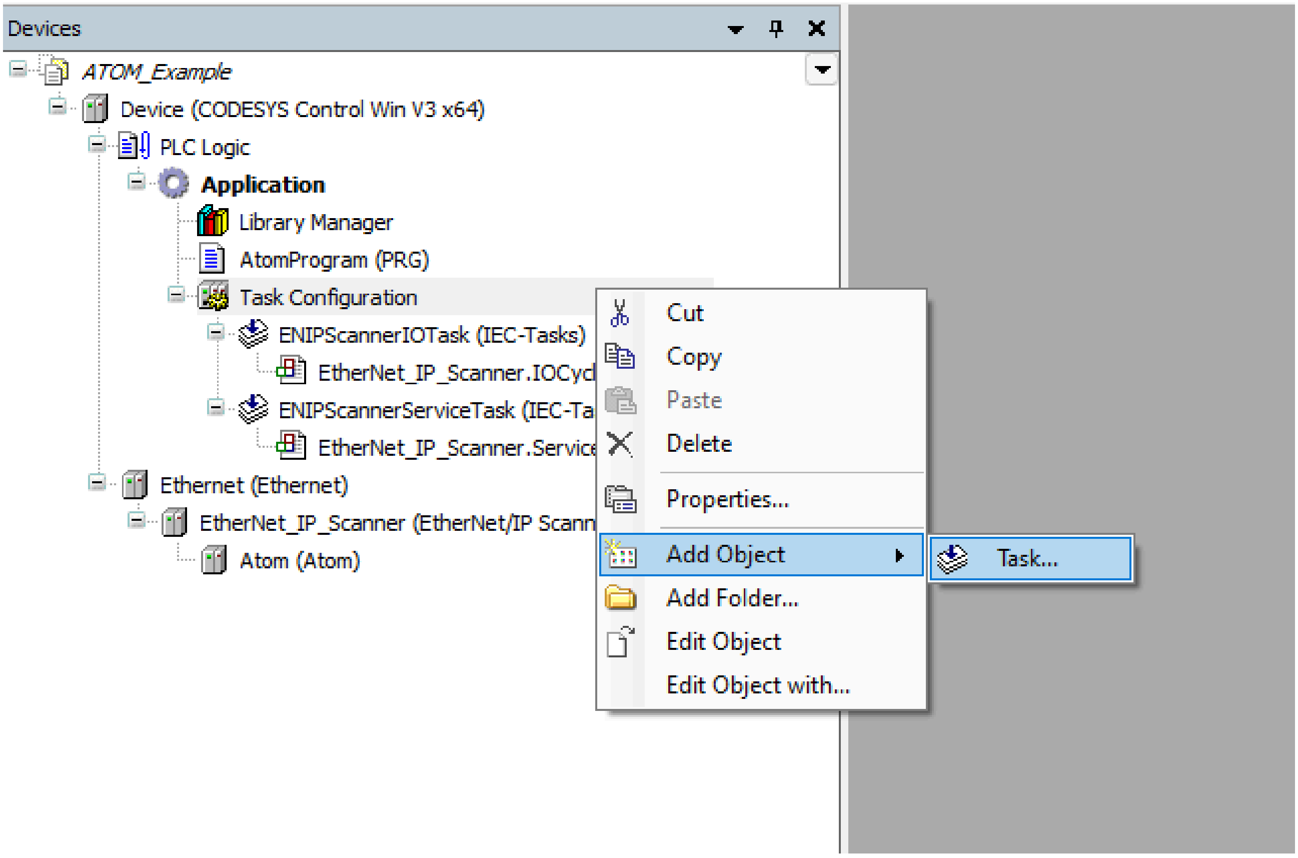
1. **Rung #1** - `RUN_SWITCH` and `ATOM_OUTPUT_RUN_ENABLE`
2. **Rung #2** - `SETPOINT` and `ATOM_OUTPUT_SETPOINT`
3. **Rung #3** - `ATOM_INPUT_TEMP` and `TEMP`

After you're finished, your ladder logic program should look like:

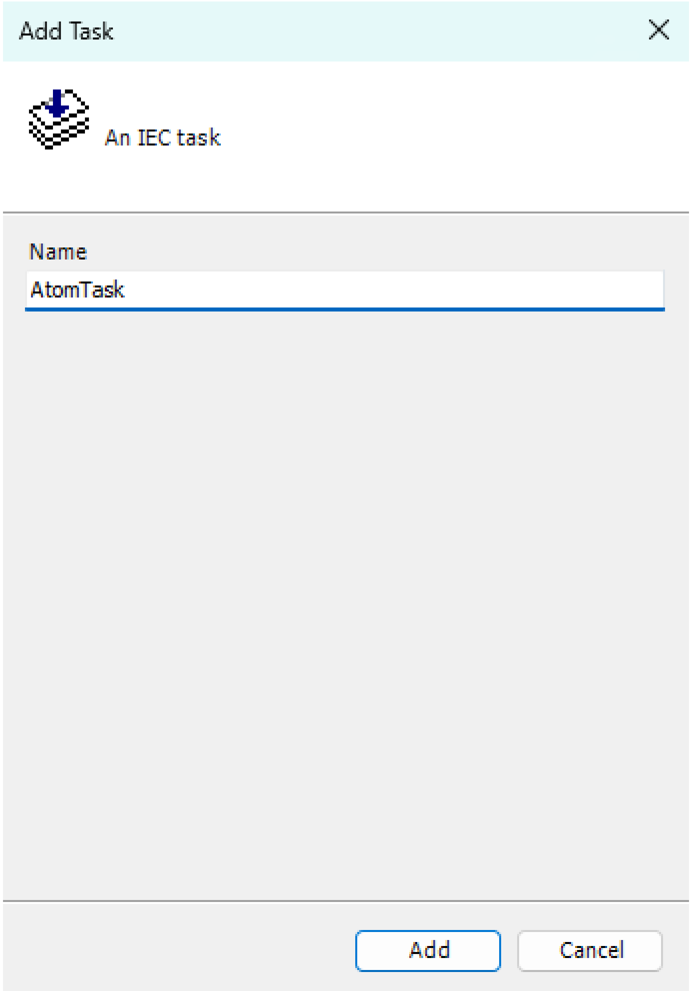


Finally, we'll add a task to call **AtomProgram** from the PLC's control loop:

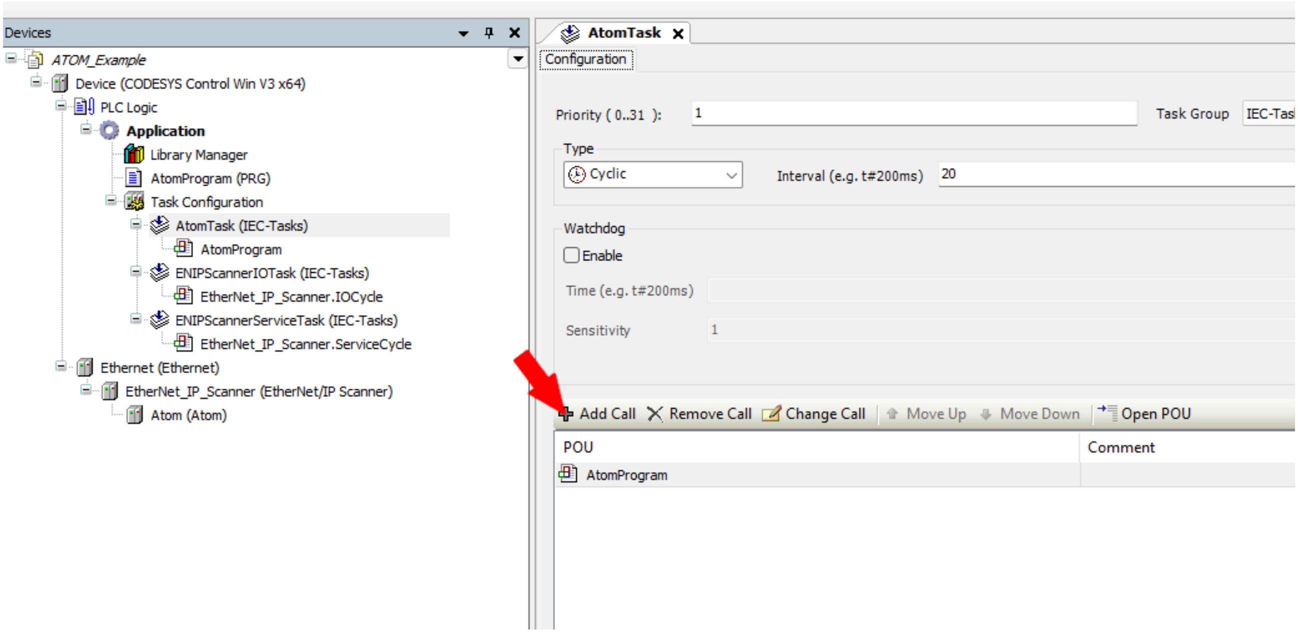
Right click **Task Configuration** and select **Add Object > Task**:



Name your task `AtomTask` and click **OK**:



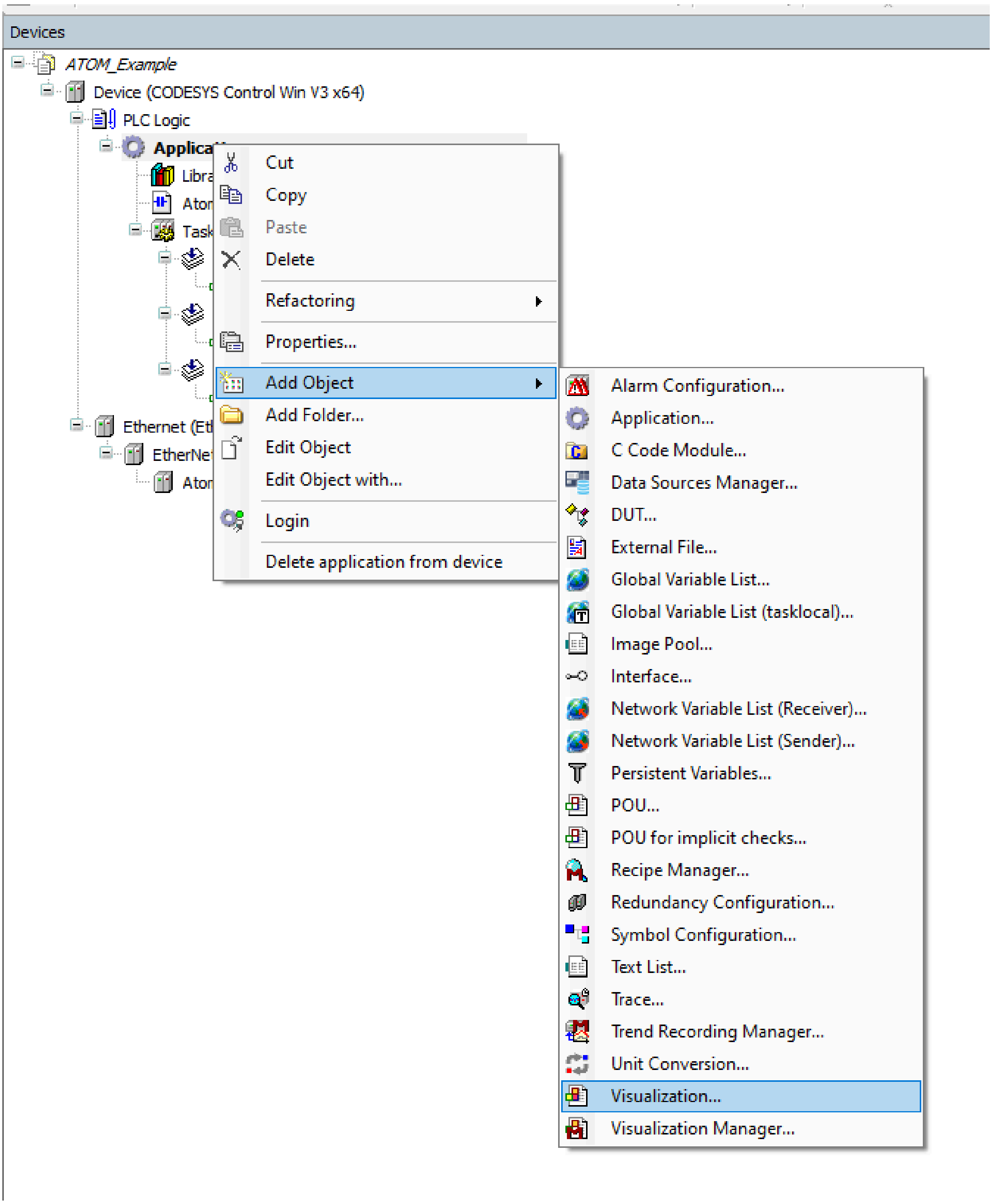
Next, double click **AtomTask (IEC-Tasks)** to open its configuration tab. Click **Add Call** and select **Application > AtomProgram**. After doing so, AtomTask's configuration should look like:



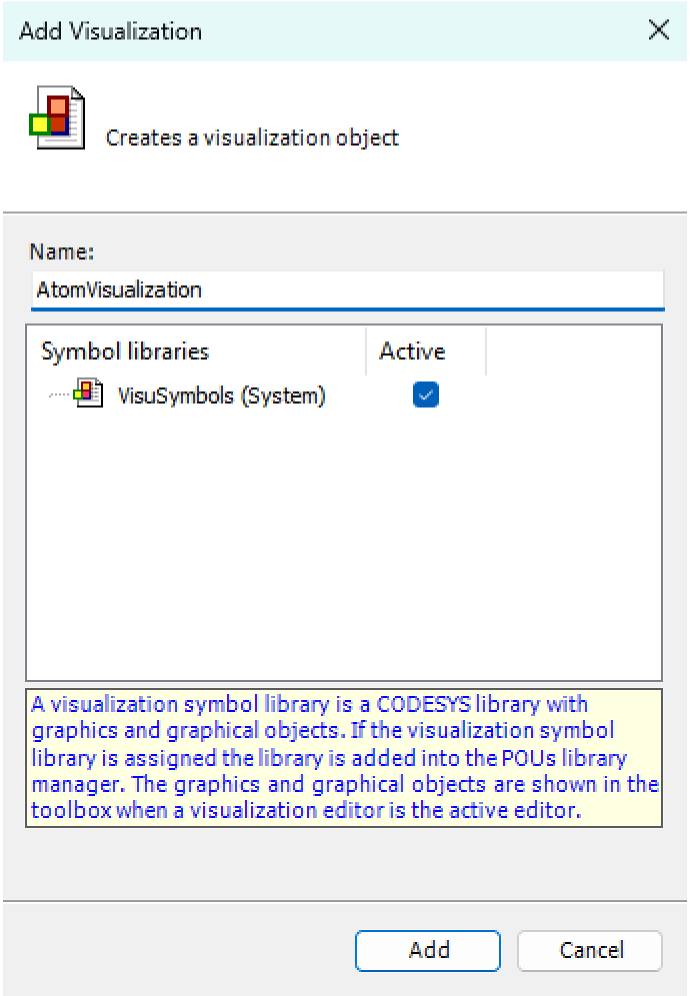
Setting up visualization

Next, we'll set up a simple visualization display to control and monitor ATOM.

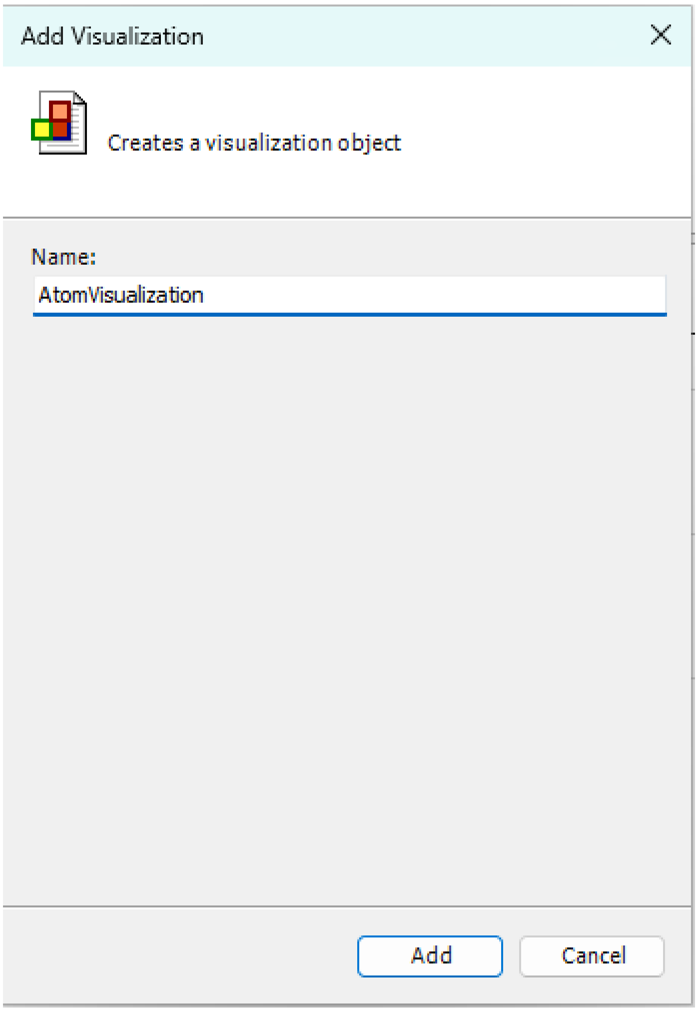
Right click **Application** and select **Add Object > Visualization**:



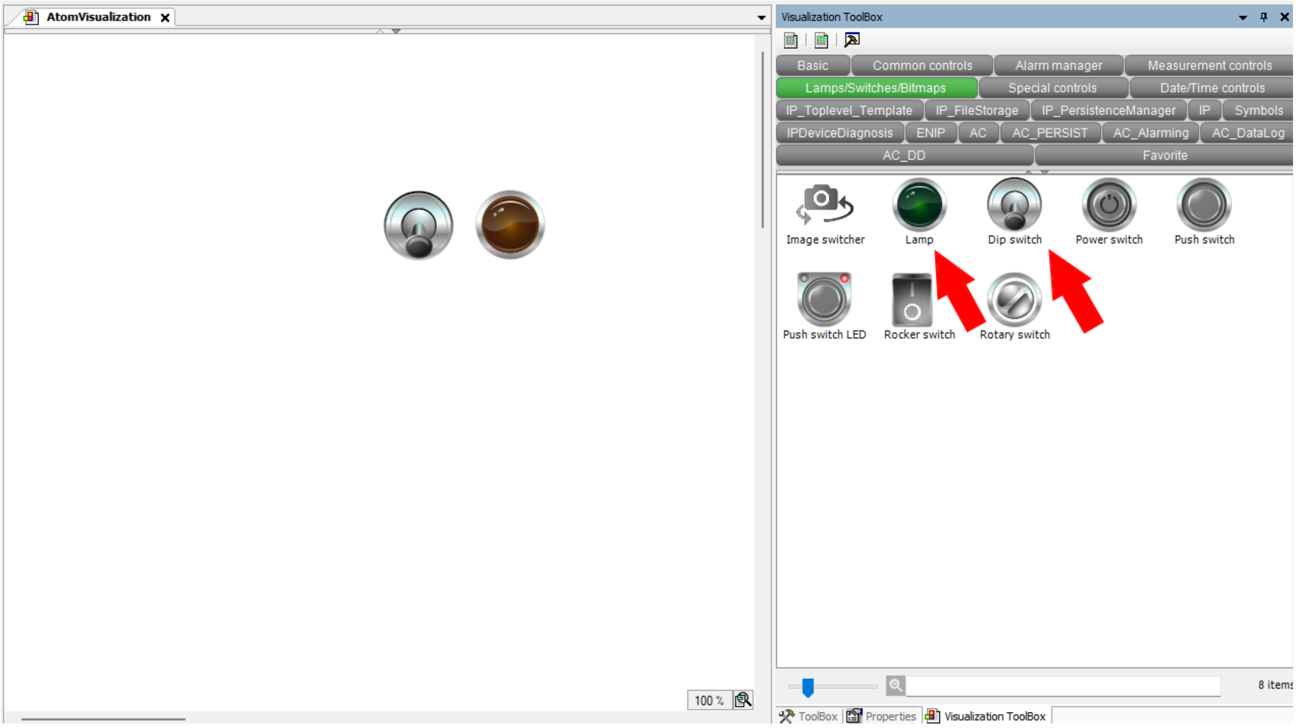
Make sure to check **Active** for **VisuSymbols (System)**, then click **Add**:



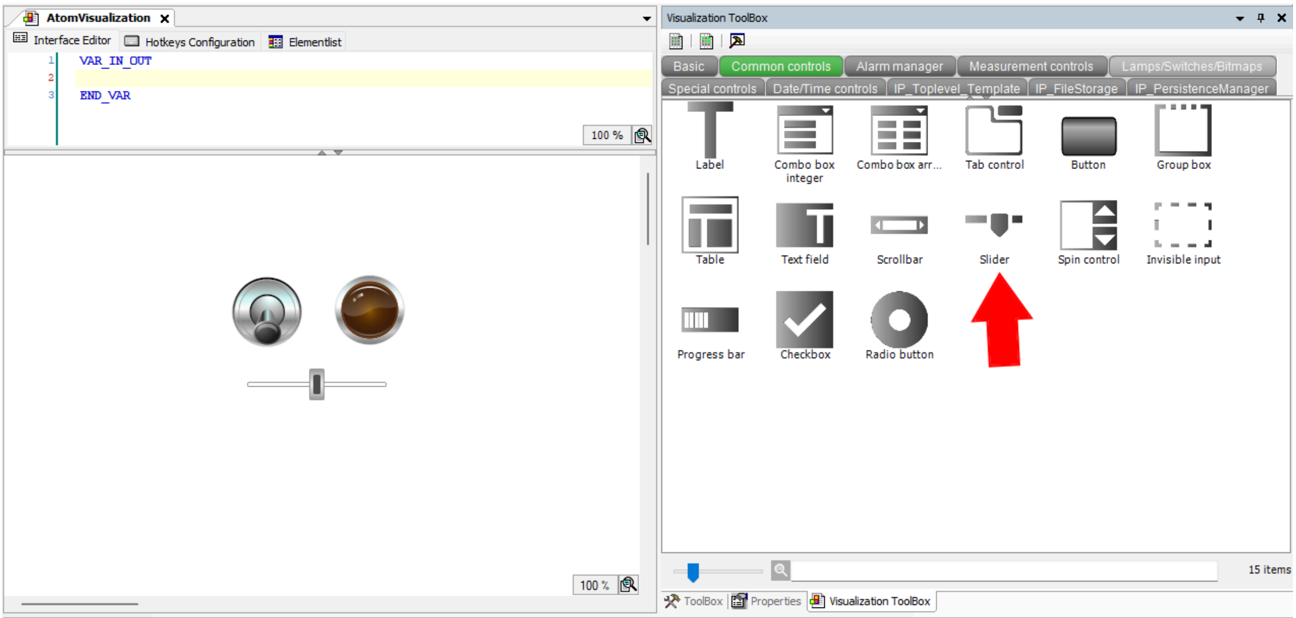
Name your visualization `AtomVisualization` and click **Add**:



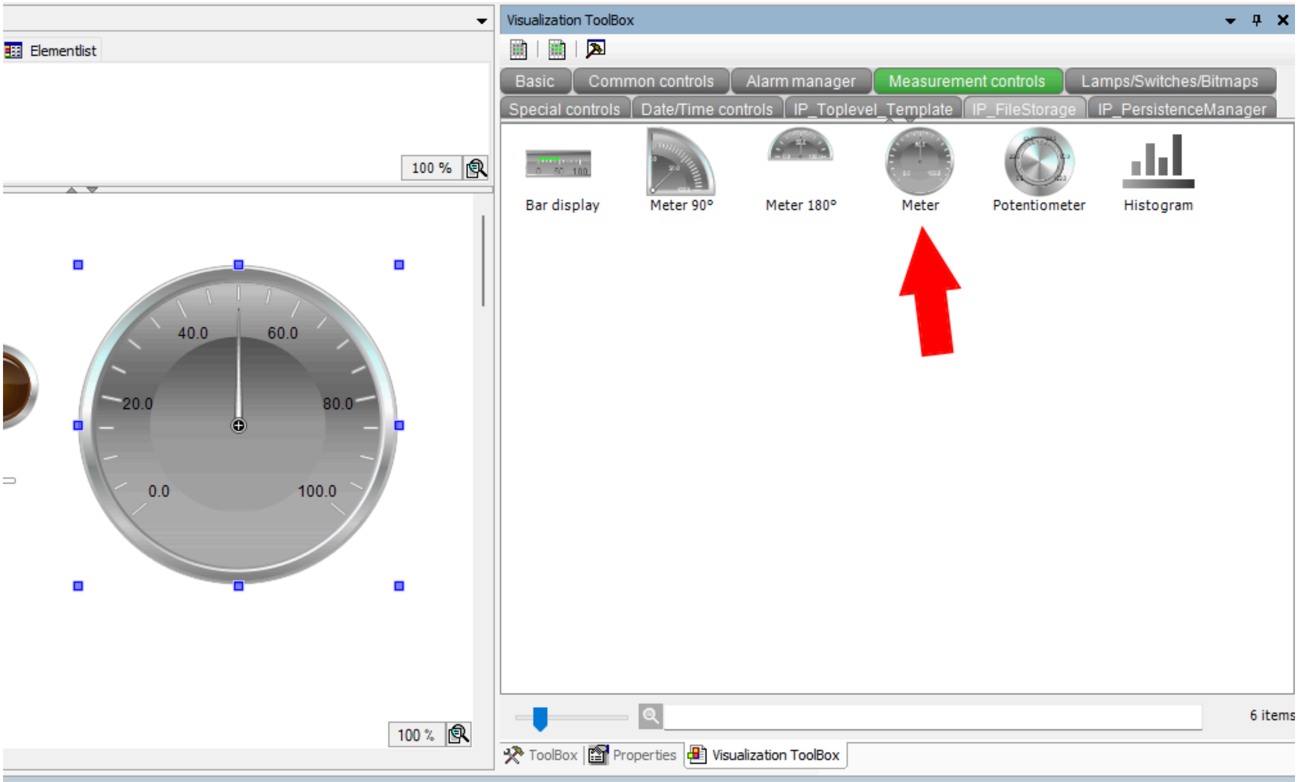
Double click **AtomVisualization** to open its configuration editor. From the **Visualization ToolBox** panel on the right, select the **Lamps/Switches/Bitmaps** category and add a lamp and a dip switch:



Next, in the **Common controls** category, add a slider:

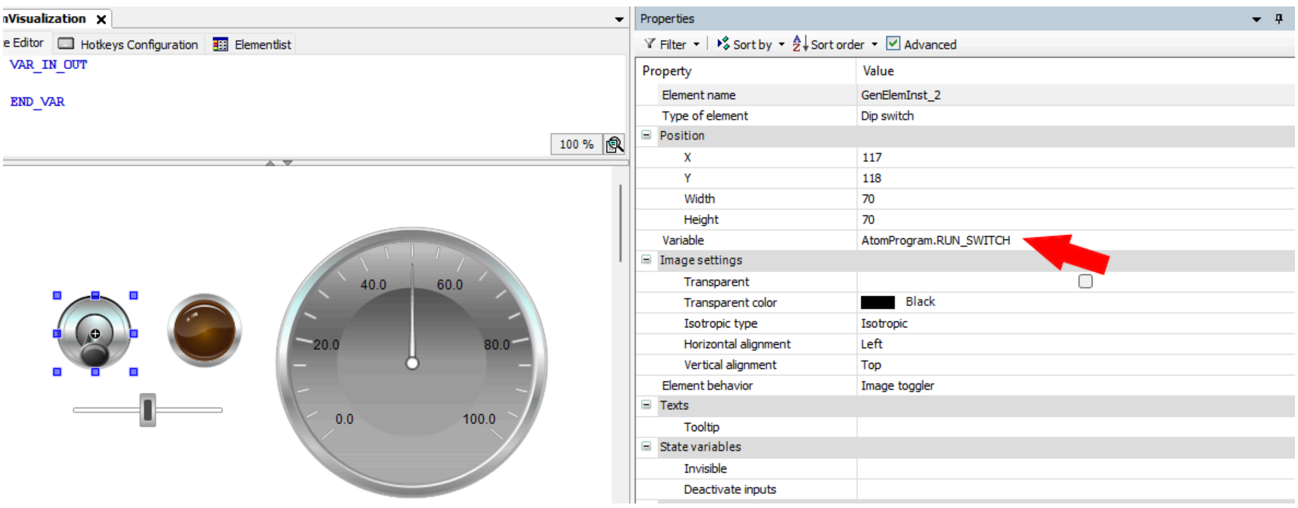


Finally, in the **Measurement controls** category, add a meter:

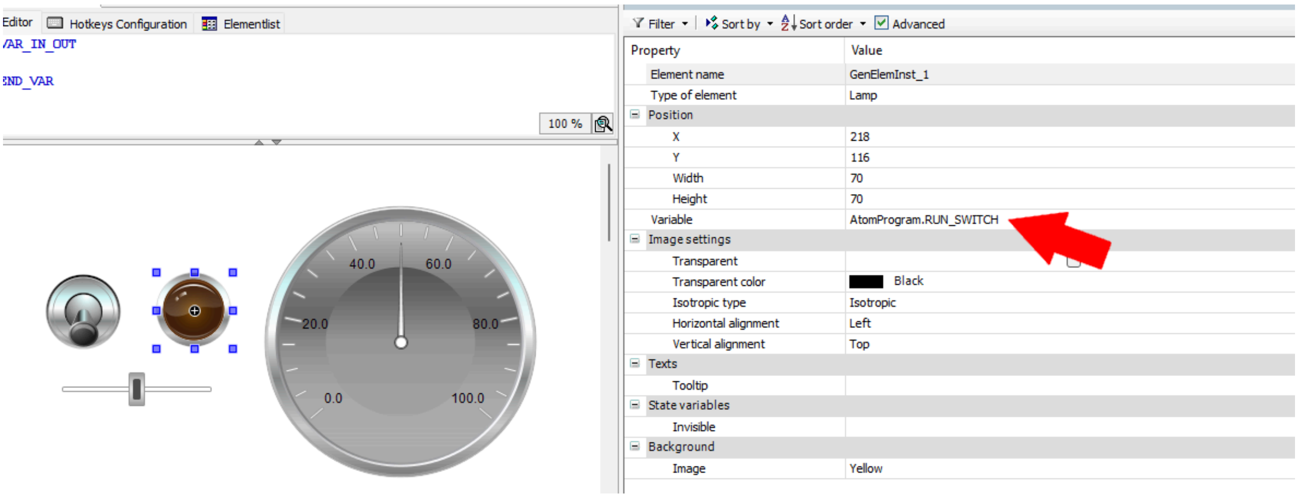


Wiring up the controls

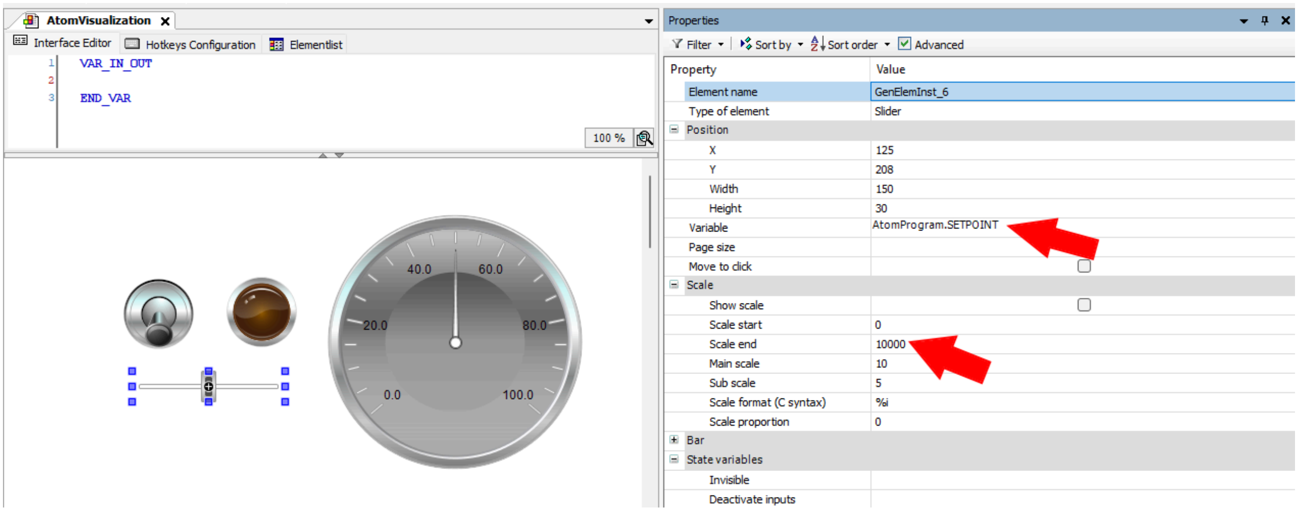
Next, we'll connect the controls to our PLC program. Select the dip switch and set the **Variable** field to `AtomProgram.RUN_SWITCH` as indicated by the red arrow:



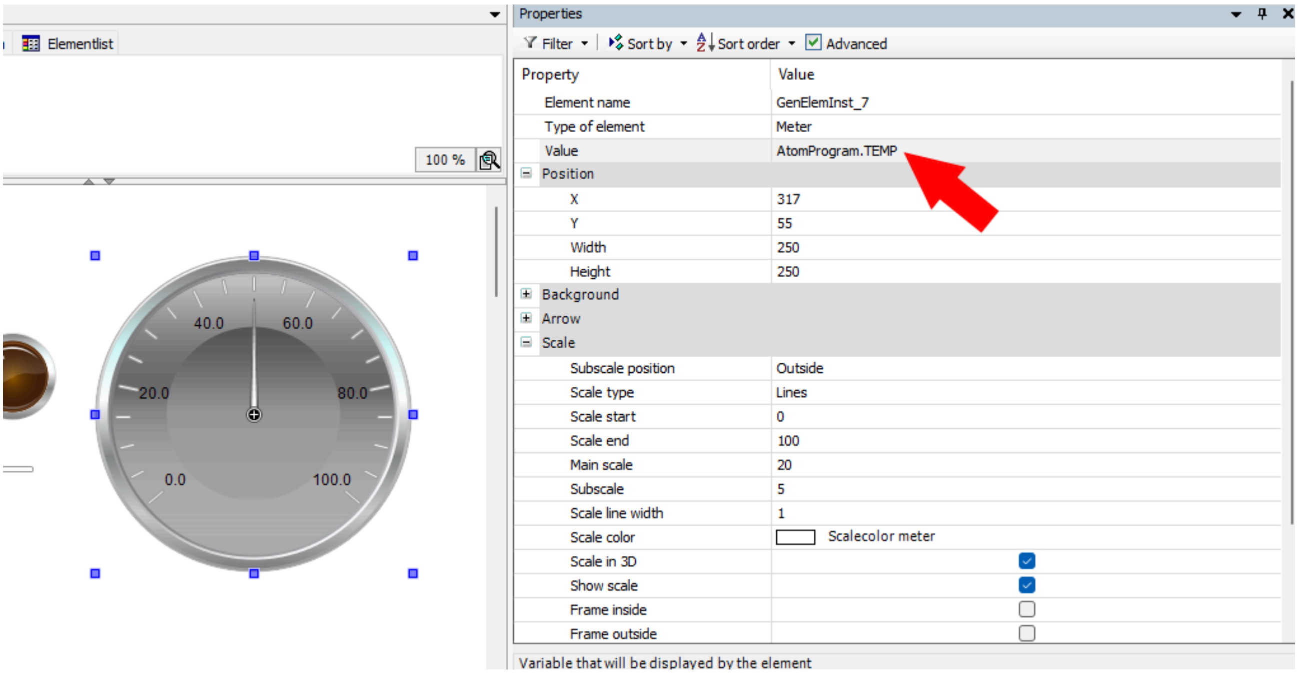
Select the lamp and set the **Variable** field to `AtomProgram.RUN_SWITCH` as indicated by the red arrow:



Select the slider and set the **Variable** field to `AtomProgram.SETPOINT` and set **Scale end** to `10000`:

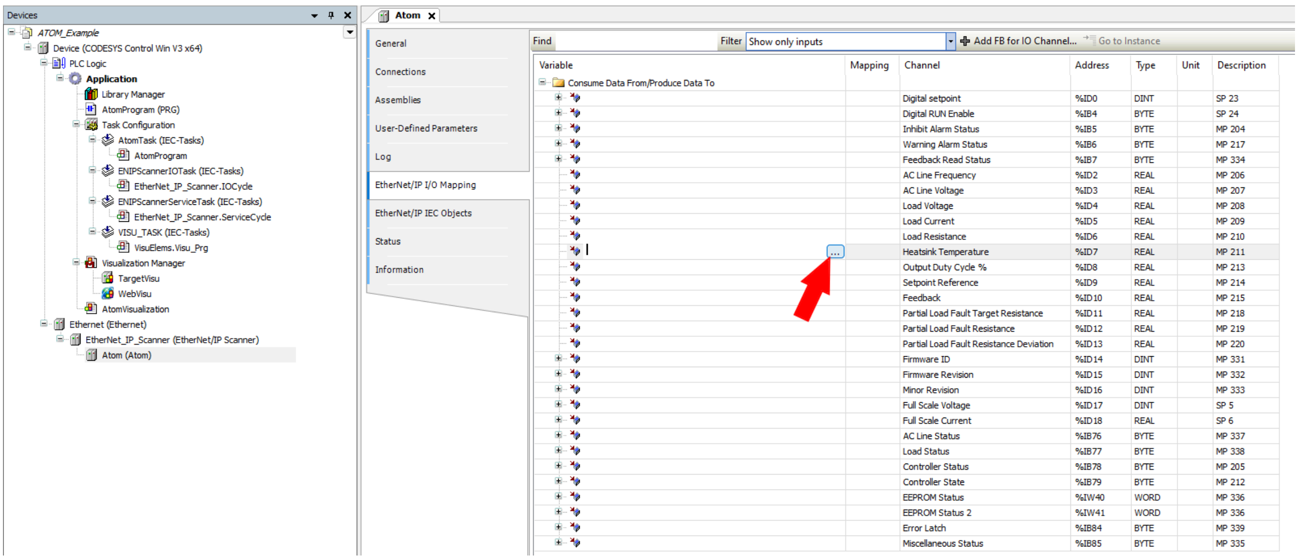


Select the meter and set the **Variable** field to `AtomProgram.TEMP`:

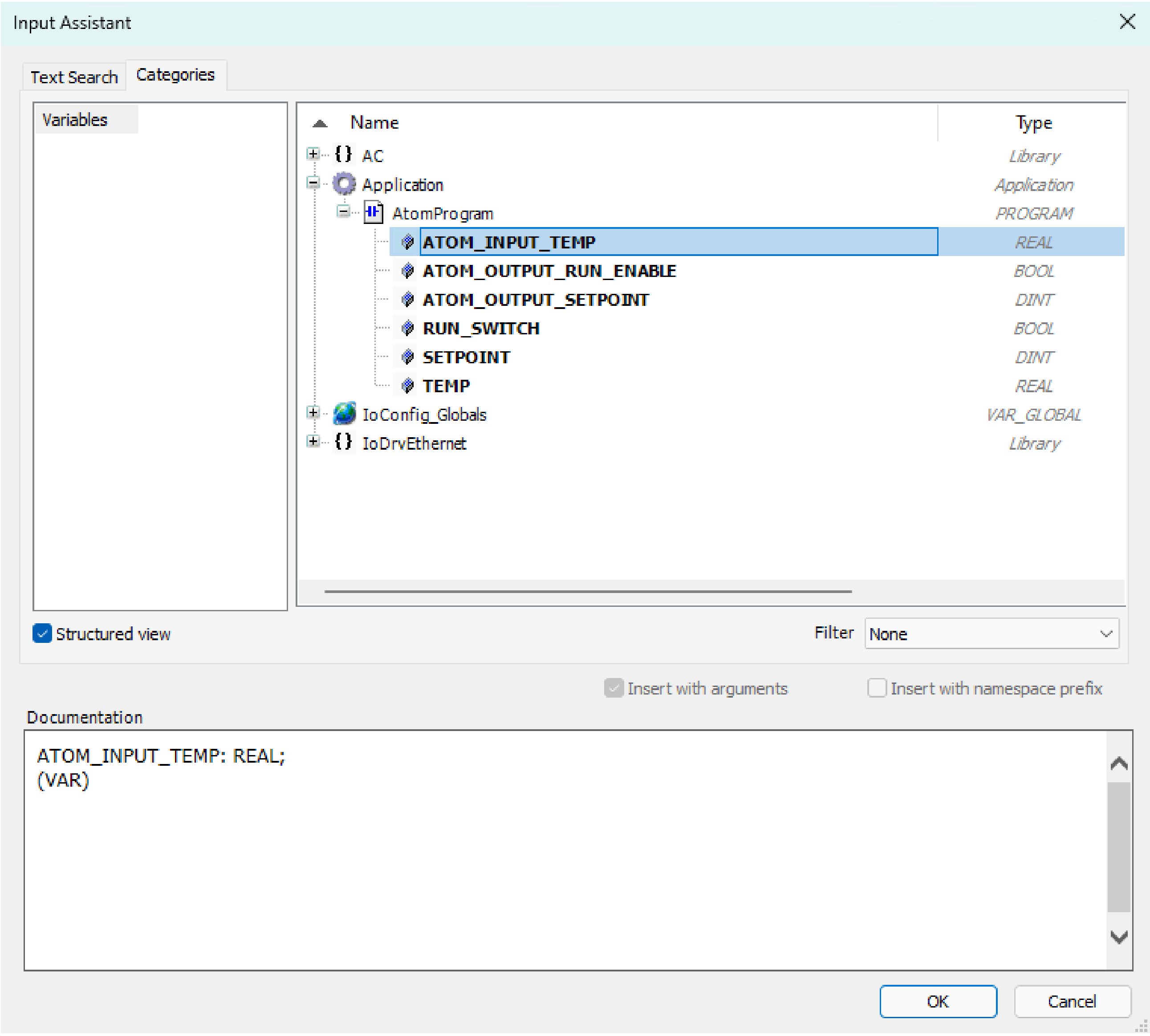


Mapping variables

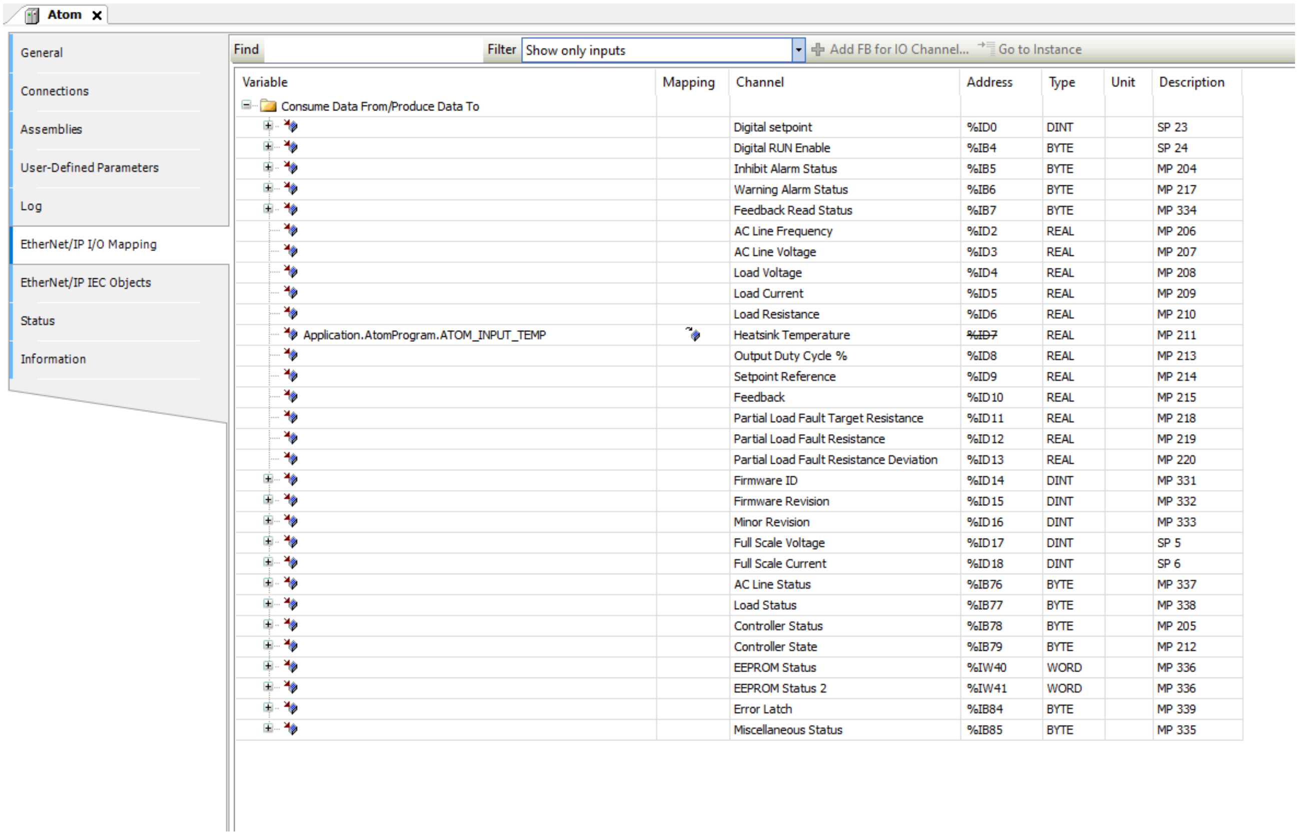
Finally, we'll map our PLC variables to ATOM. Double click **Atom** in the device tree to open its configuration window. Select the **EtherNet/IP I/O Mapping** tab and set **Filter** to **Show only inputs**:



Above, select the button indicated by the red arrow. This will open the **Input Assistant** dialog. Select **Application > AtomProgram > ATOM_INPUT_TEMP** and click **Add**:

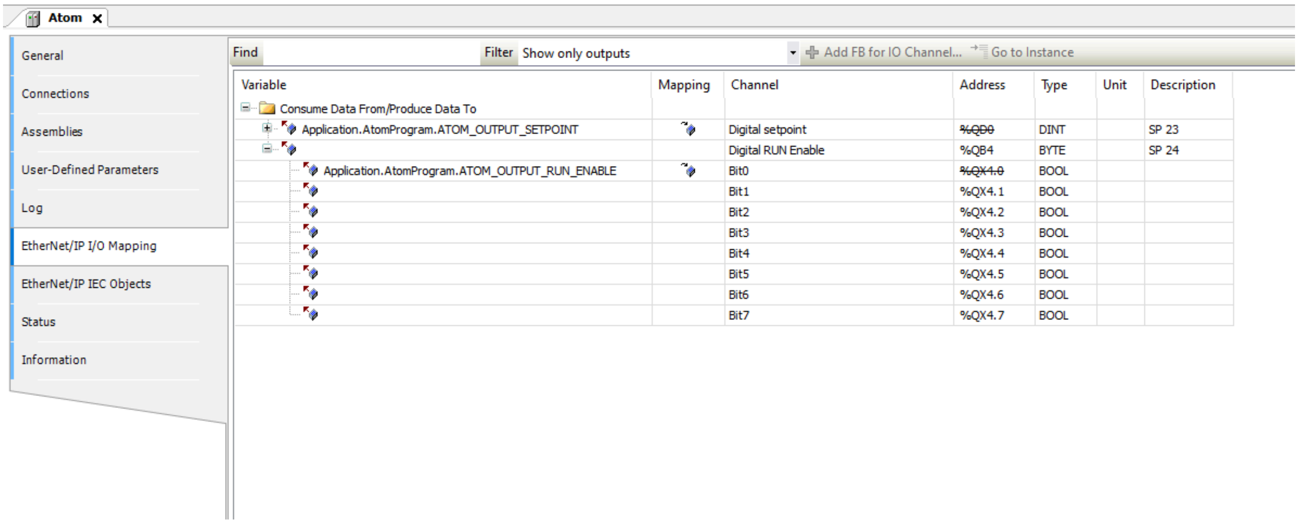


After doing so, your input I/O mappings should look like:



Change the **Filter** to **Show only outputs** and repeat the process for the outputs. Map **Digital setpoint** to `Application.AtomProgram.ATOM_OUTPUT_SETPOINT` and **Digital RUN Enable** to `Application.AtomProgram.ATOM_OUTPUT_RUN_ENABLE`.

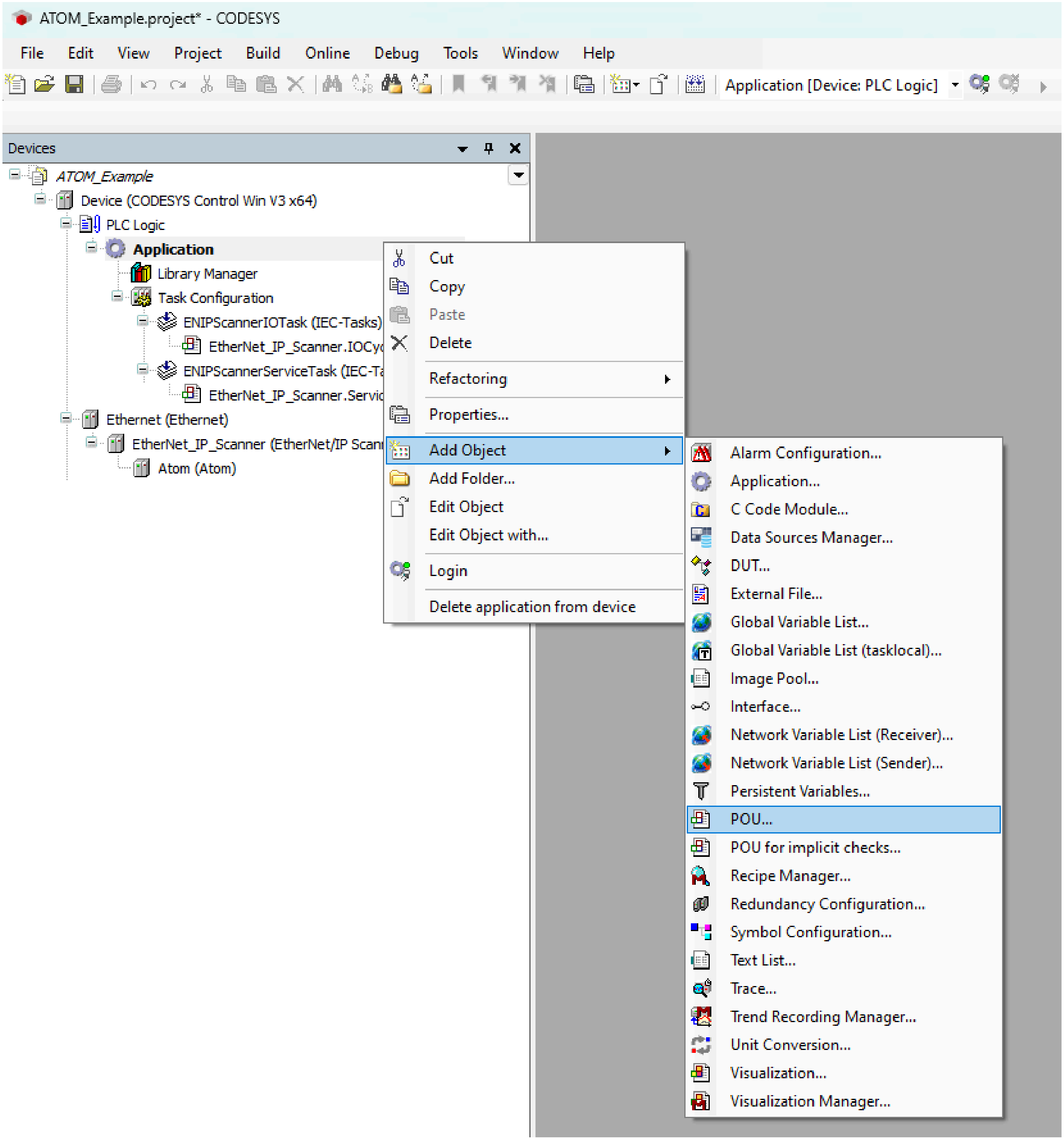
> **TAKE CARE**
>
> Make sure you map **Bit0** of **Digital RUN Enable** to **ATOM_OUTPUT_RUN_ENABLE**, NOT **Digital RUN Enable** itself.



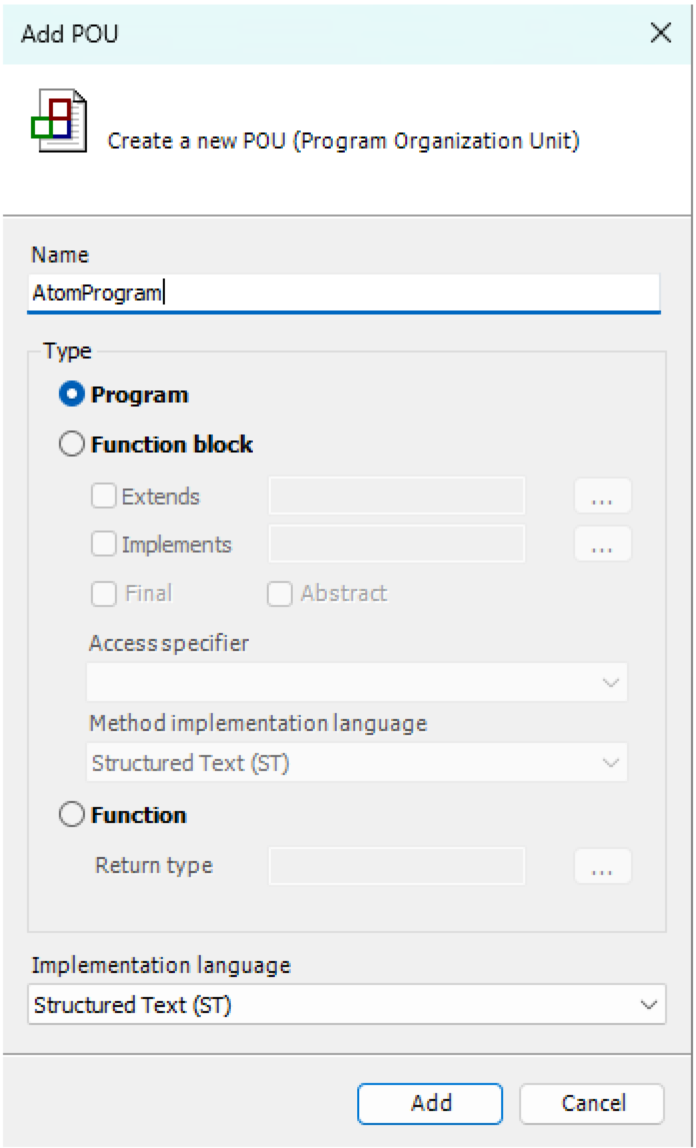
Example: Structured text

Creating the program

Right click **Application** and select **Add Object > POU**:



Name your **POU** `AtomProgram` and select **Structured Text (ST)** as the language:



Next, let's create a basic program. We'll check to make sure no alarms are active and then write a setpoint value of `8000` and set run enable to `true`.

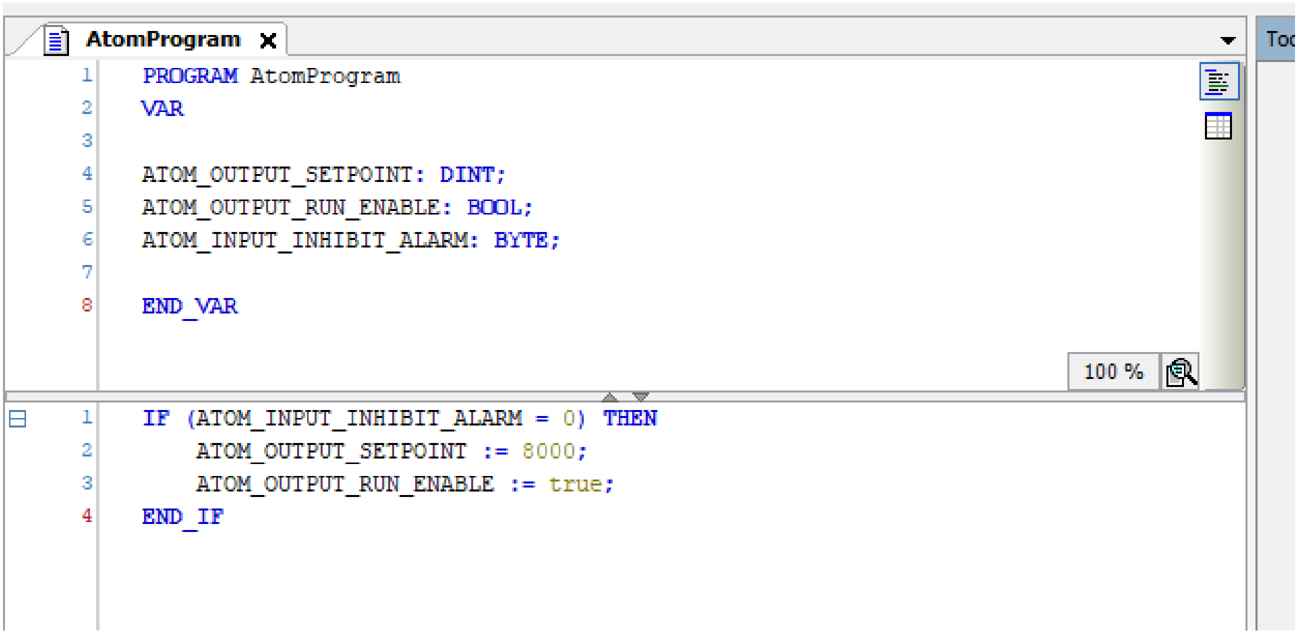
Copy the following code into the top panel of the **AtomProgram** editor:

```
PROGRAM AtomProgram
VAR

ATOM_OUTPUT_SETPOINT: BOOL;
ATOM_OUTPUT_RUN_ENABLE: BYTE;
ATOM_INPUT_INHIBIT_ALARM: BYTE;

END_VAR
```

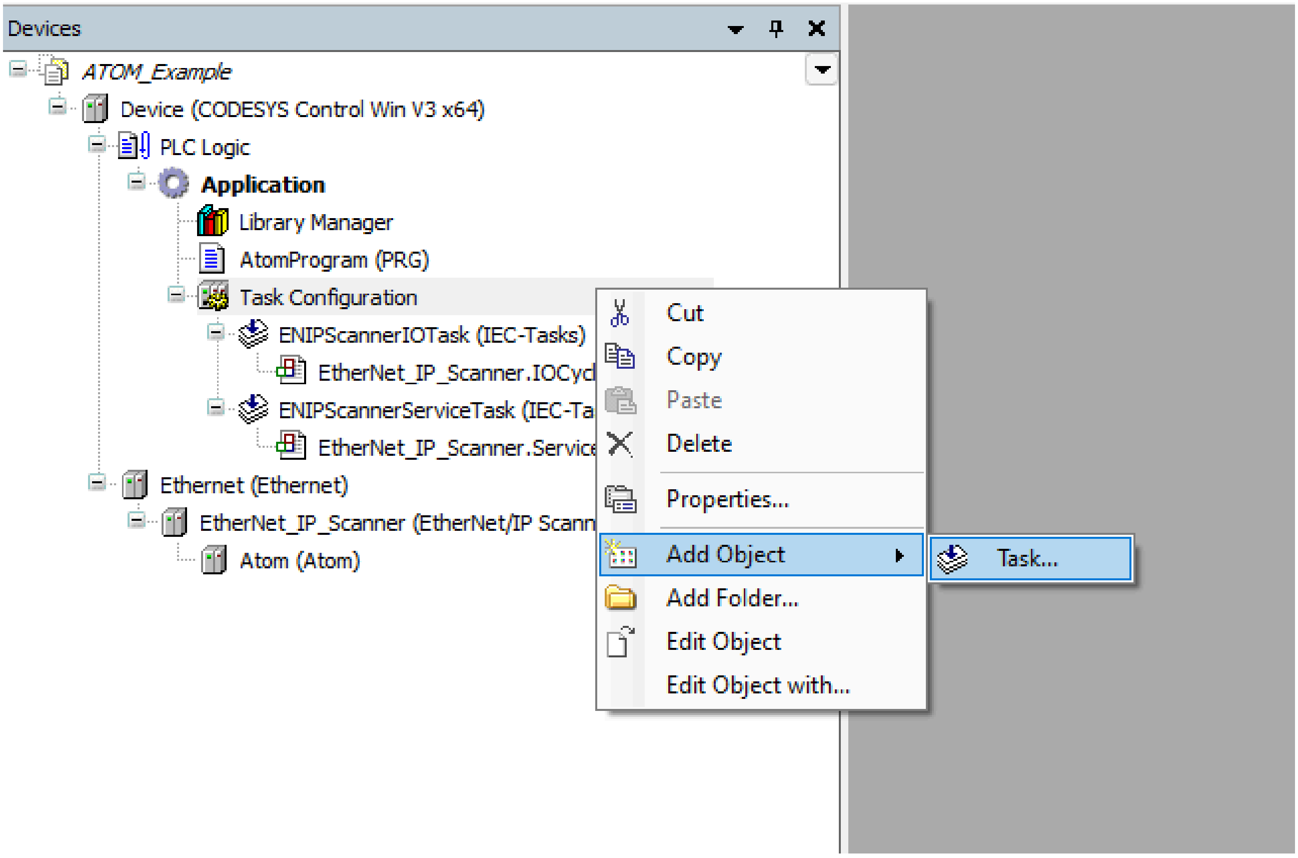
Copy the following code into the main program section:

```
IF (ATOM_INPUT_INHIBIT_ALARM = 0) THEN
        ATOM_OUTPUT_SETPOINT := 8000;
        ATOM_OUTPUT_RUN_ENABLE := true;
END_IF
```

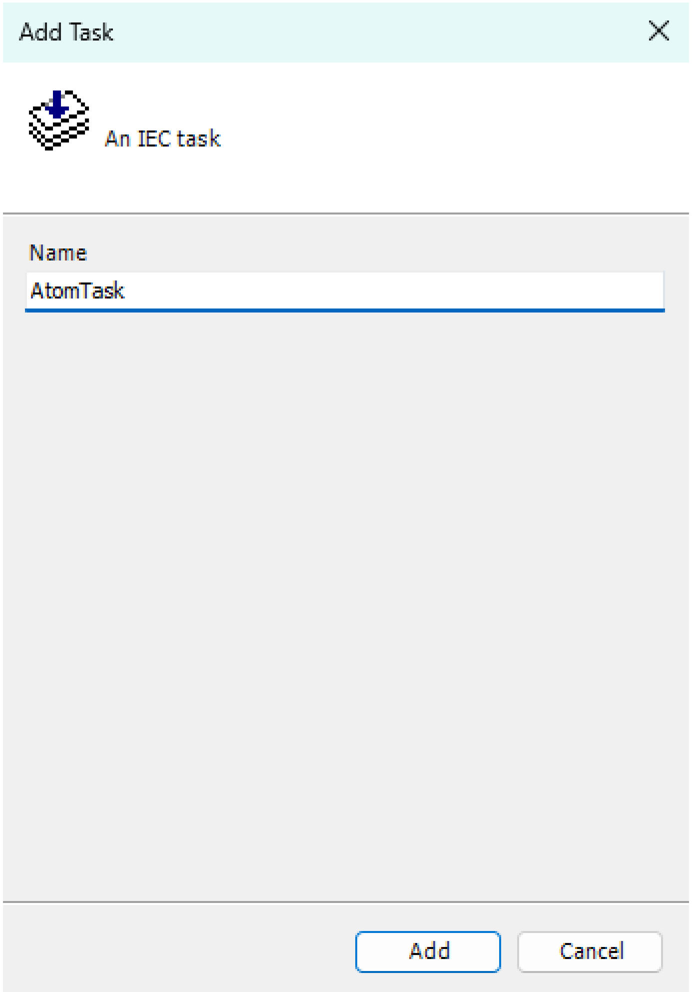
Your editor should look like:



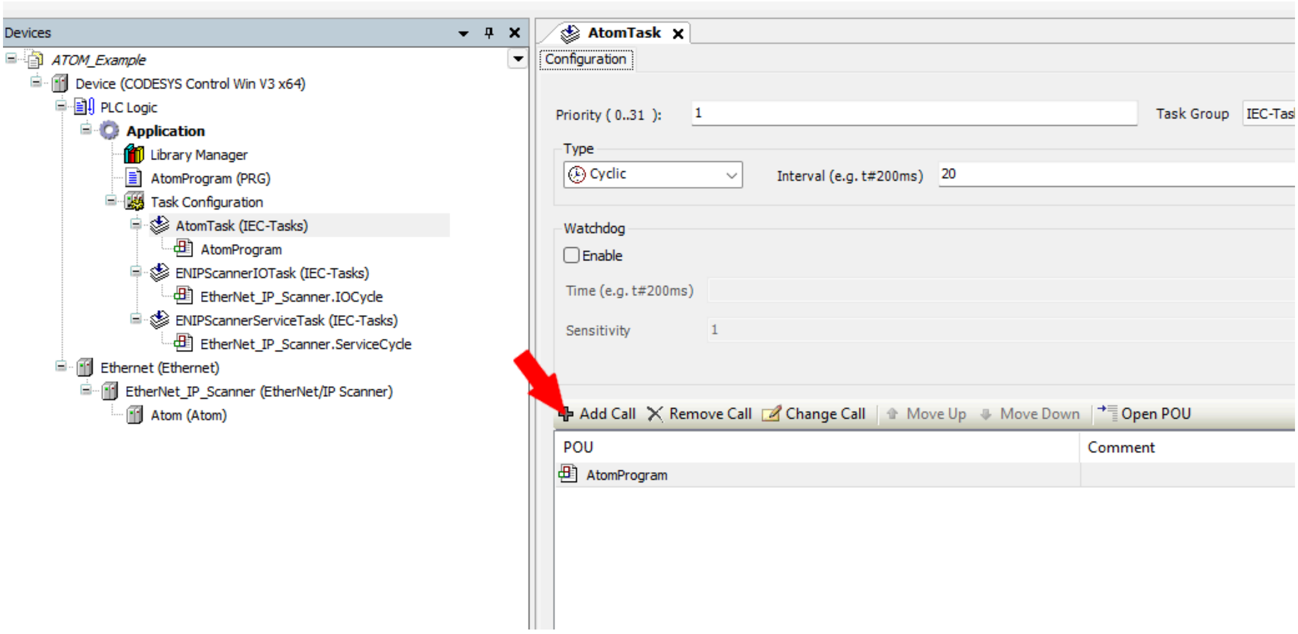
Next, we'll add a new task to call our program. Right click **Task Configuration** and Select **Add Object > Task**:



Name your task `AtomTask` and click **Add**:

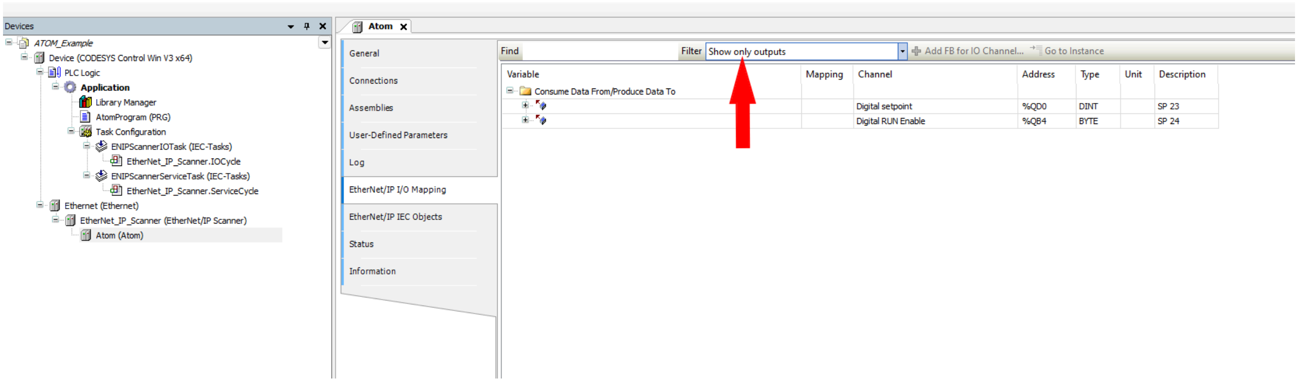


Next, double click **AtomTask (IEC-Tasks)** to open its configuration tab. Click **Add Call** and select **Application > AtomProgram**. After doing so, **AtomTask**'s configuration should look like:

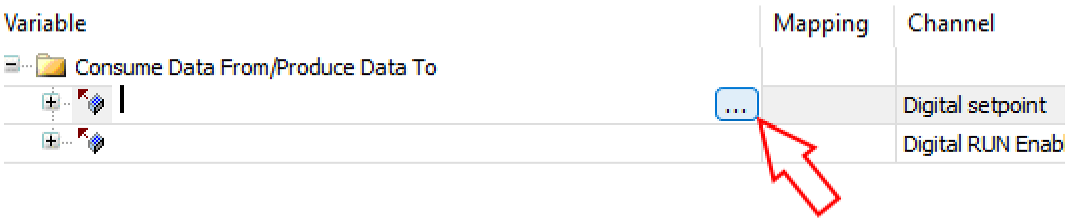


Mapping variables

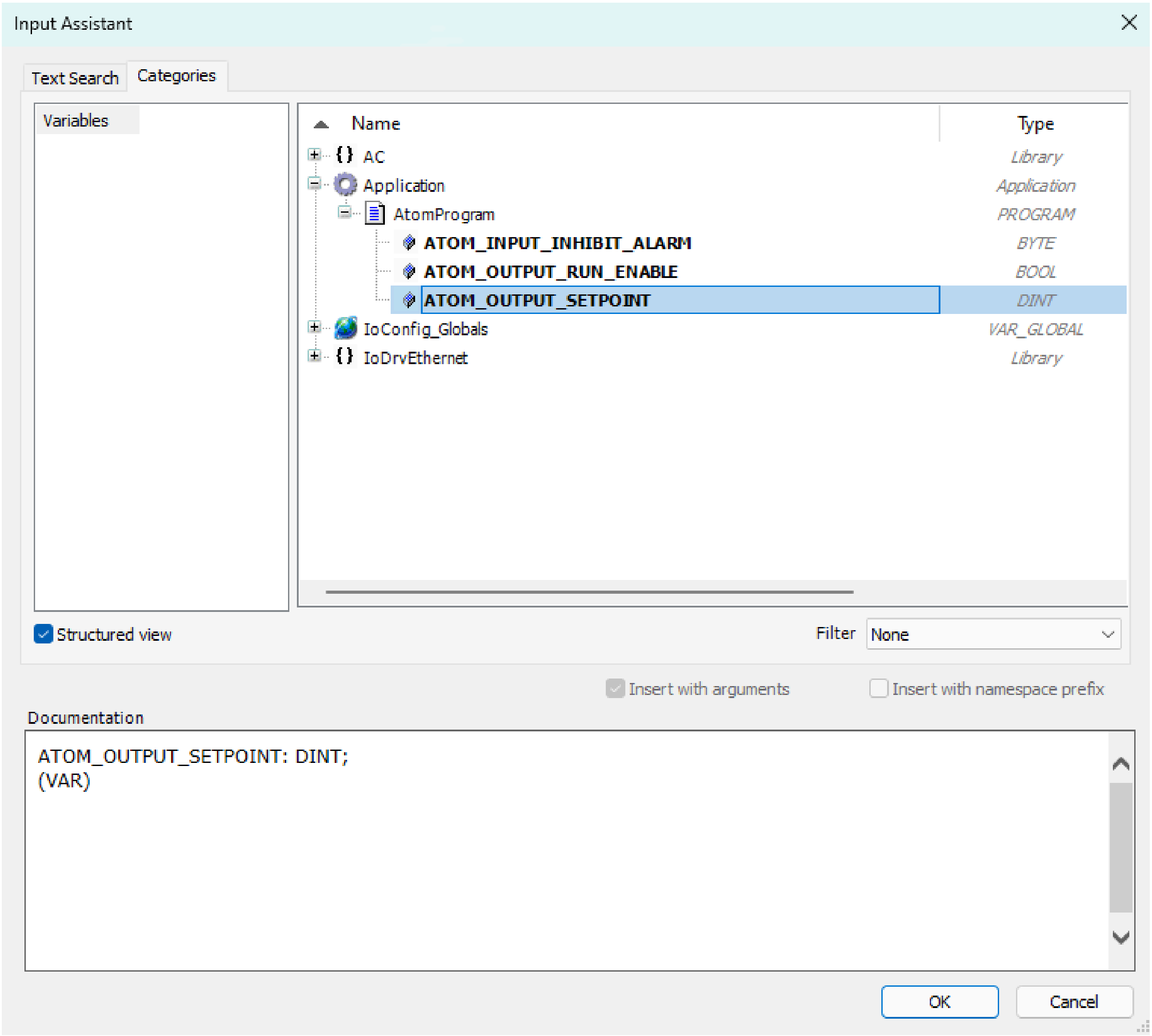
Next, we'll map our ATOM's I/O configuration to our program variables. Double click **Atom (Atom)** to open its configuration window, then select the **EtherNet/IP I/O Configuration** tab. On the **Filter** dropdown indicated by the red arrow, select **Show only outputs**:



Click the button indicated by the red arrow to map the **Digital setpoint** value:

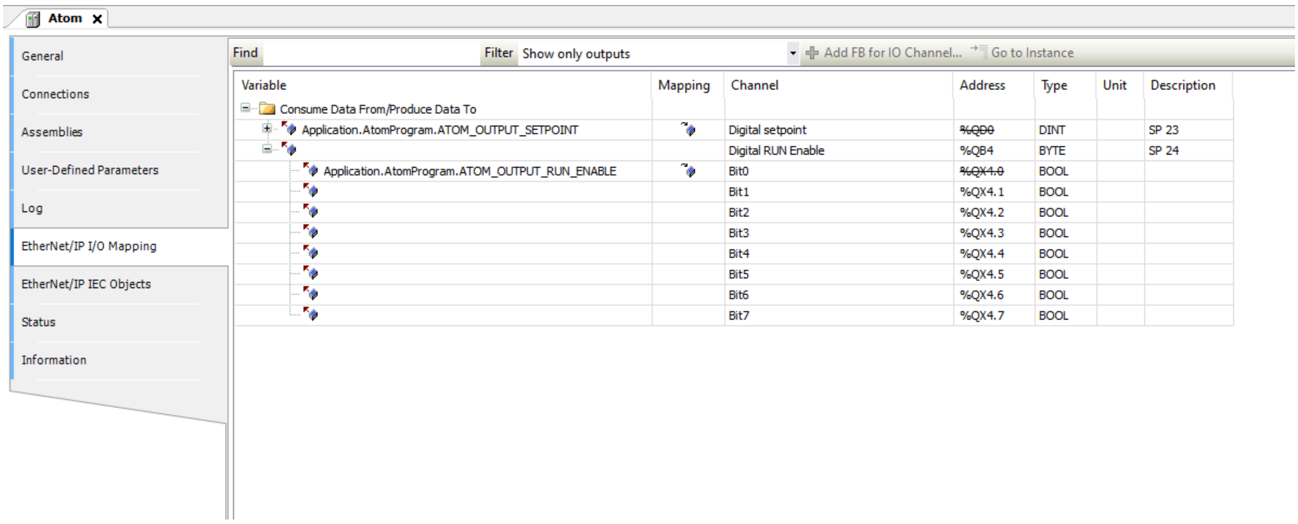


This button will open the **Input Assistant** dialog. Select the corresponding variable from your program and click **Ok**:

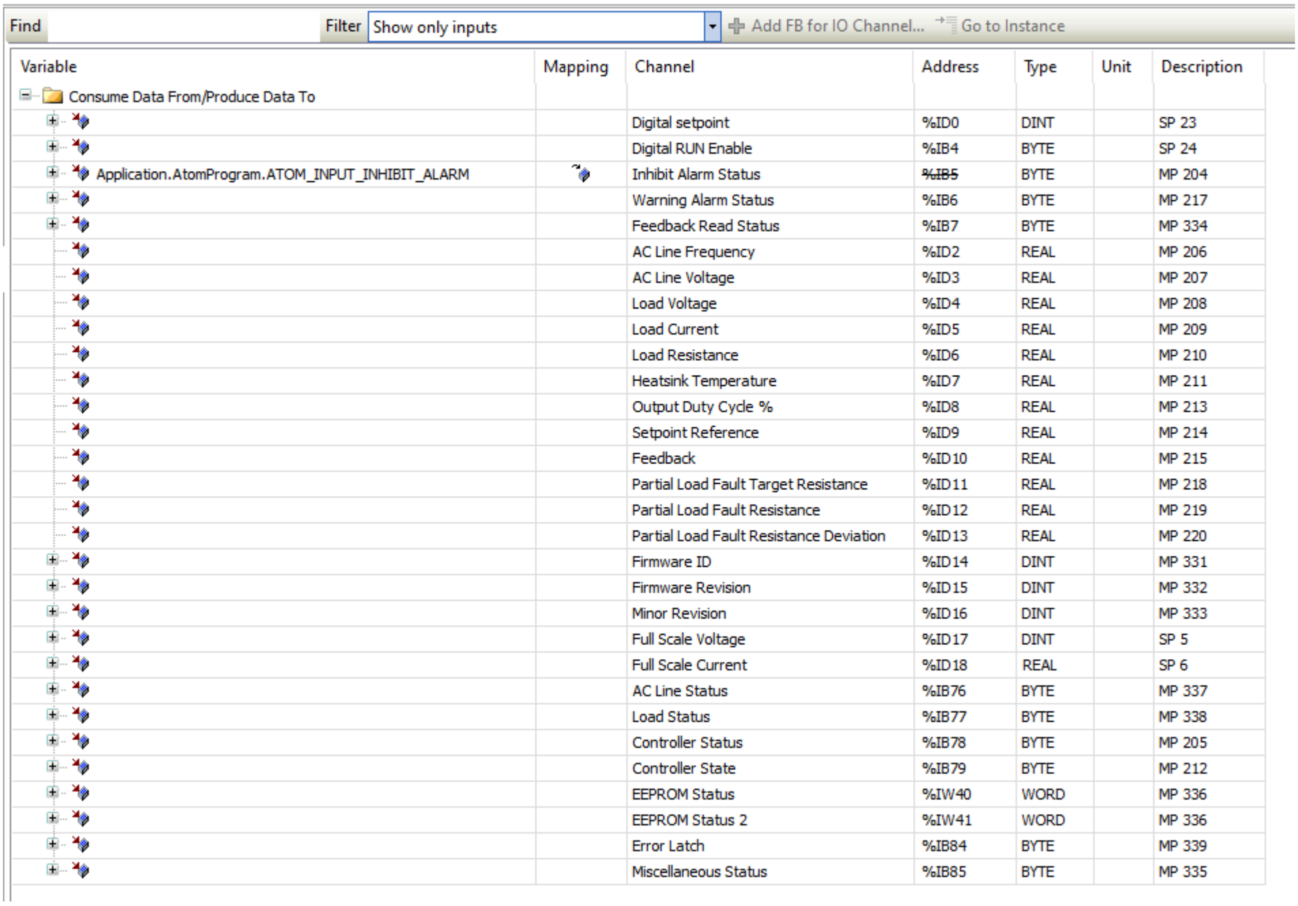


Repeat this process so that your output variables are mapped like so:

> **TAKE CARE**
>
> Make sure you map **Bit0** of **Digital RUN Enable** to **ATOM_OUTPUT_RUN_ENABLE**, NOT **Digital RUN Enable** itself.



Switch the filter to **Show only inputs** and then map the **Inhibit alarm status** variable:



Variable	Mapping	Channel	Address	Type	Unit	Description
Consume Data From/Produce Data To						
		Digital setpoint	%ID0	DINT		SP 23
		Digital RUN Enable	%IB4	BYTE		SP 24
Application.AtomProgram.ATOM_INPUT_INHIBIT_ALARM		Inhibit Alarm Status	~~%IB5~~	BYTE		MP 204
		Warning Alarm Status	%IB6	BYTE		MP 217
		Feedback Read Status	%IB7	BYTE		MP 334
		AC Line Frequency	%ID2	REAL		MP 206
		AC Line Voltage	%ID3	REAL		MP 207
		Load Voltage	%ID4	REAL		MP 208
		Load Current	%ID5	REAL		MP 209
		Load Resistance	%ID6	REAL		MP 210
		Heatsink Temperature	%ID7	REAL		MP 211
		Output Duty Cycle %	%ID8	REAL		MP 213
		Setpoint Reference	%ID9	REAL		MP 214
		Feedback	%ID10	REAL		MP 215
		Partial Load Fault Target Resistance	%ID11	REAL		MP 218
		Partial Load Fault Resistance	%ID12	REAL		MP 219
		Partial Load Fault Resistance Deviation	%ID13	REAL		MP 220
		Firmware ID	%ID14	DINT		MP 331
		Firmware Revision	%ID15	DINT		MP 332
		Minor Revision	%ID16	DINT		MP 333
		Full Scale Voltage	%ID17	DINT		SP 5
		Full Scale Current	%ID18	REAL		SP 6
		AC Line Status	%IB76	BYTE		MP 337
		Load Status	%IB77	BYTE		MP 338
		Controller Status	%IB78	BYTE		MP 205
		Controller State	%IB79	BYTE		MP 212
		EEPROM Status	%IW40	WORD		MP 336
		EEPROM Status 2	%IW41	WORD		MP 336
		Error Latch	%IB84	BYTE		MP 339
		Miscellaneous Status	%IB85	BYTE		MP 335

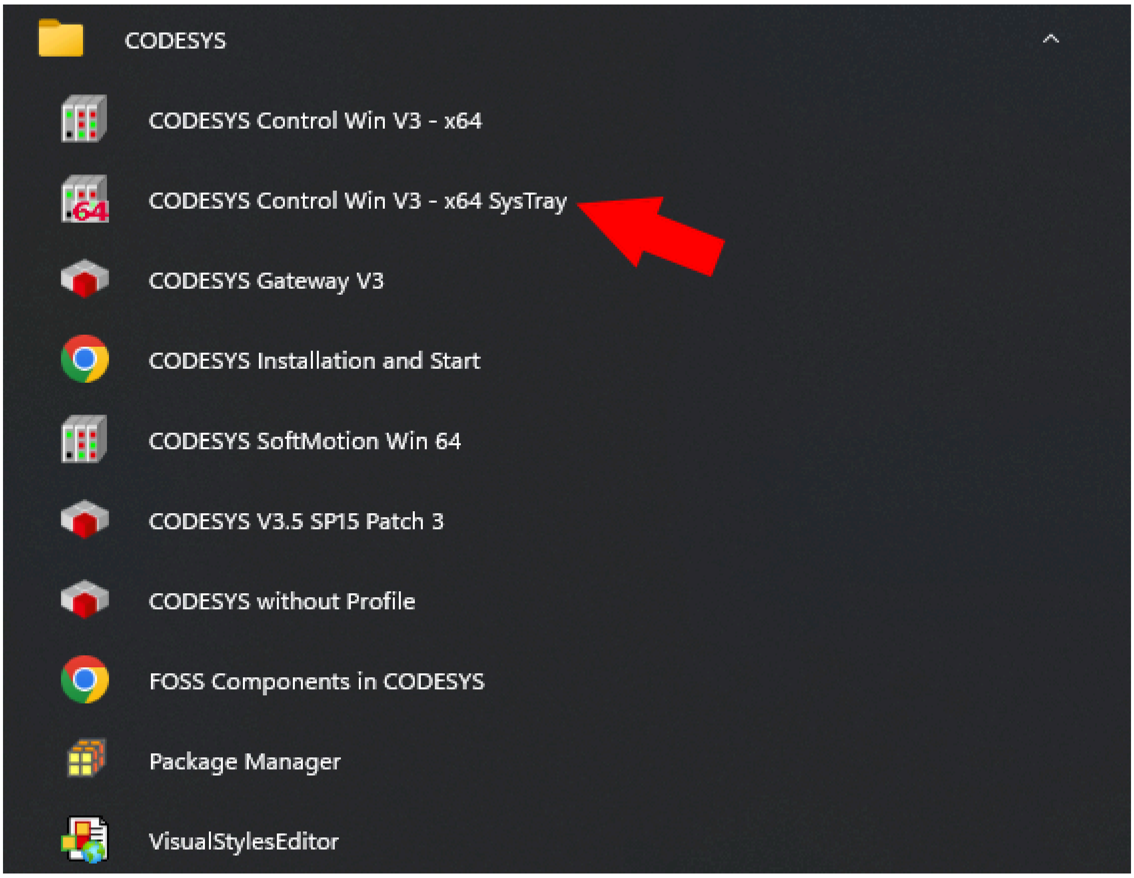
Running the program with SoftPLC

INFO

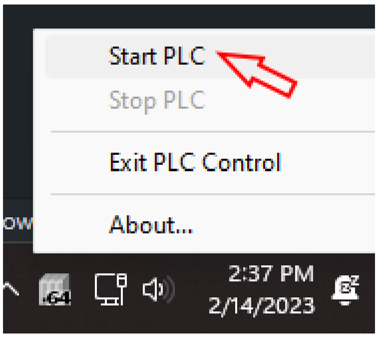
The instructions to run your program are the same regardless of whether you are using ladder logic or structured text.

The only difference is that in the ladder logic example, a visualization window will open that allows you to control ATOM.

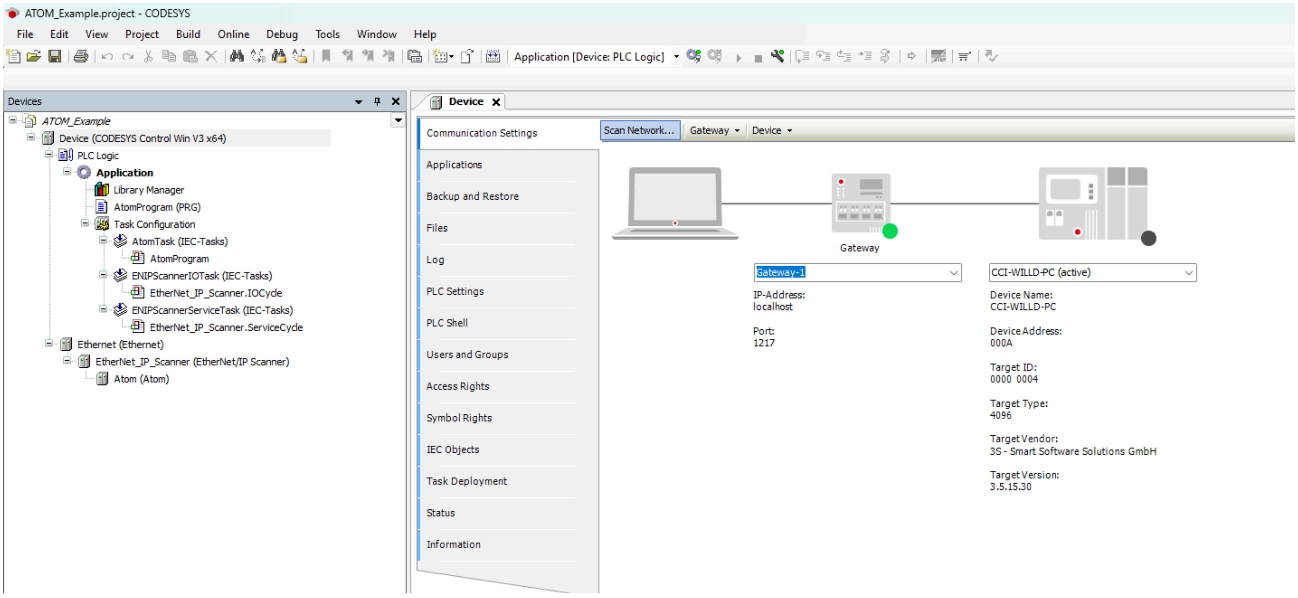
To debug the program, first make sure you start **Codesys WIN Control V3 - x64 SysTray**



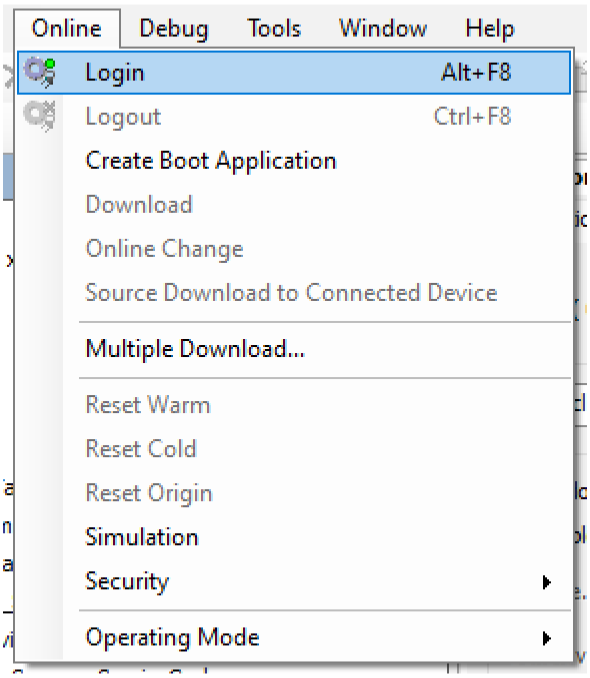
This will launch the Codesys SoftPLC. You should see an icon appear in your systray and you can right click it and select **Start PLC** to start the SoftPLC:



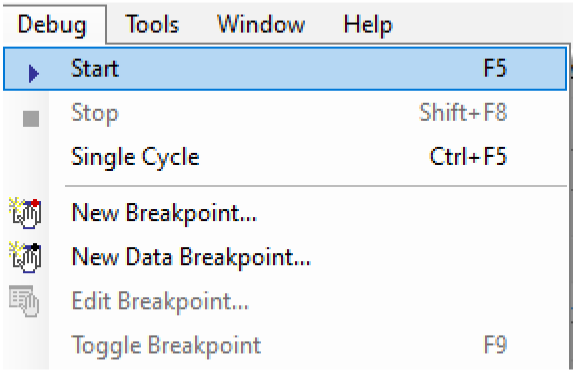
Next, in Codesys double click **Application** to open its configuration window. Here you can select **Scan Network** to discover your SoftPLC:



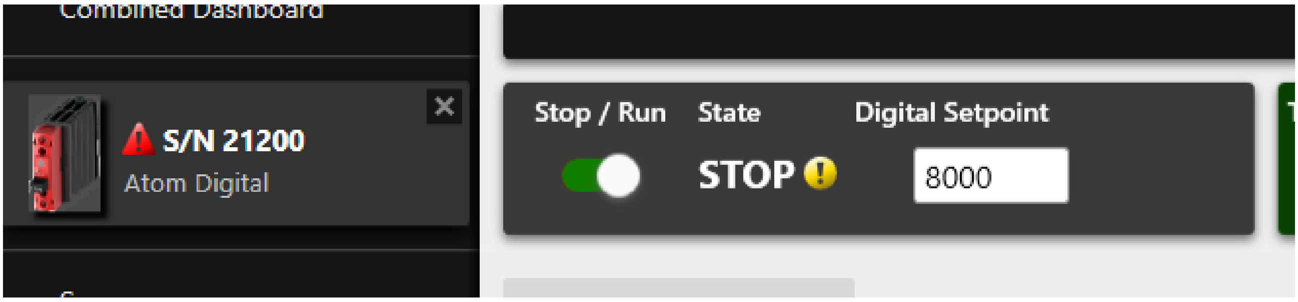
Finally, **Login** to your SoftPLC:



Then you can start debugging the program:



If you use Control Panel to monitor ATOM, you should see the **Stop / Run** state and the **Digital Setpoint** values change to reflect the PLC program's instructions. If you followed the structured text example, the values will change once and remain fixed. If you followed the ladder logic example, a visualization control panel will appear. Flipping the dip switch or adjusting the slider will immediately update ATOM and the changes should reflect in real-time:



ATOM / Fieldbus / EtherNet/IP / Labview

> **NOTE**
>
> You do NOT need to purchase the NI-Industrial Communications for EtherNet/IP add-on.
>
> This is only used if you want your Labview application to operate as an EtherNet/IP adapter device. In our case, Labview will operate as a scanner that connects to ATOM to control it. This functionality is included in the default version of Labview.

> **NOTE**
>
> We use **explicit messaging** to connect to ATOM from Labview because Labview does not support I/O connections.

Using our pre-built VI

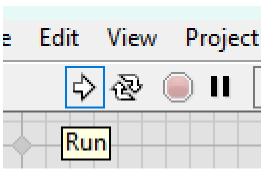
Download our ATOM control panel VI to quickly control and monitor ATOM from Labview.

Most of the parameters listed in our EtherNet/IP Profile are available through the VI controls, although you can easily extend or alter the VI to suite your needs.

Download ATOM.vi

To setup the VI, enter the IP address of your ATOM unit in the **ATOM IP Address** box. Below, we've entered `192.168.2.250` as the IP address. You can check or change your ATOM's IP address by using Control Panel.

After you've set the IP address, you can run the VI by clicking the run icon:



After the VI starts, you can adjust the Digital setpoint and RUN Enable controls. All the other indicators will update every 500 ms to reflect the current state of the ATOM unit.

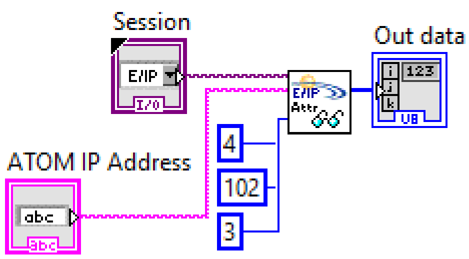
Advanced

Reading and parsing data from ATOM

If you need to modify or extend the VI that we provide, this section is provided to help make it easier.

The main way to pull data from ATOM is by fetching the *producing* assembly object (class 4, instance 102, attribute 3) using the Get Attribute Single service (0x0E).

Below is a simple example of fetching ATOM's assembly object. After this block diagram executes, `Out data` will be a byte array of length `86` containing the 30 parameters listed in ATOM's EtherNet/IP Profile:



To pull a parameter out of this byte array, you can calculate its index within `Out data` by finding the sum of the parameter data type sizes before it.

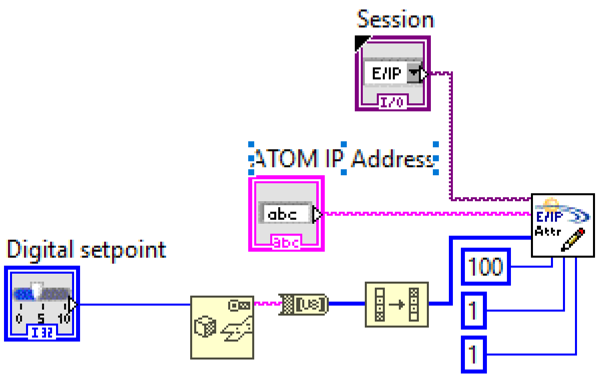
> **EXAMPLE**
>
> To pull out the AC Line Voltage parameter, we can calculate the sum of the parameter data type sizes before it. AC Line Voltage is parameter #7, so we sum the size of the first 6 parameters, which each have sizes of:
>
> ```
> Digital setpoint = DINT = 4 bytes
> Digital run enable = BOOL = 1 byte
> Inhibit alarm status = BYTE = 1 byte
> Warning alarm status = BYTE = 1 byte
> Feedback read setatus = BOOL = 1 byte
> AC Line Frequency = REAL = 4 bytes
> ```
>
> Summing these up, we get:
>
> ```
> 4 (DINT) + 1 (BOOL) + 1 (BYTE) + 1 (BYTE) + 1 (BYTE) + 4 (REAL) = 12
> ```
>
> AC Line Voltage itself is a `REAL`, so AC Line Voltage resides at index `12` within `Out data` and extends for `4` bytes.
>
> To properly display this within your VI, you will need to reverse the order of these four bytes before interpreting them as a single precision decimal in Labview. This is because Labview is by default big-endian, while ATOM is little-endian.

Writing data to ATOM

Below is a simple diagram that updates the digital setpoint parameter on ATOM.

An alternative method to this one is to compile together an assembly object and use the Set Attribute Single service (0x10) to write to the consuming assembly object (class 4, instance 101, attribute 3). The one downside to this is that it requires you to send all parameters within the consuming assembly object together.

Instead, we provide the custom ParameterLink object (class 100, instance `i`, attribute 1), where `i` is the parameter number you wish to interact with. Below, we write to ParameterLink instance `1`, attribute `1`. Consult the EtherNet/IP Profile to determine the data type of the parameter you wish to write. Here, we're writing digital setpoint, which is a `DINT`, so we'll make sure that we use an `i32` in Labview. Note that before we write the data, we have to reverse the bytes as Labview is by default big-endian, while ATOM is little-endian:

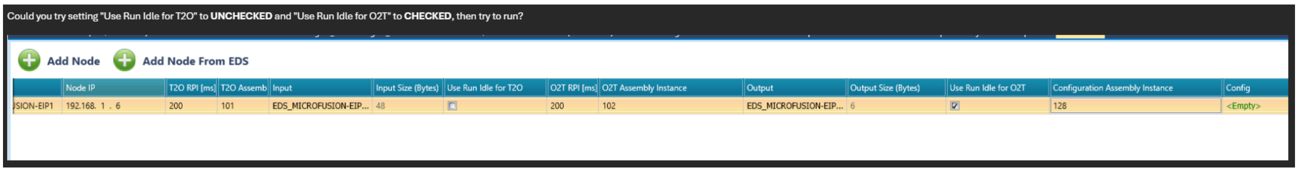


ATOM / Fieldbus / EtherNet/IP / Other

Unilogic PLCs

For Unilogic PLCs, ensure that:

- `Use Run Idle for T2O` = UNCHECKED
- `Use Run Idle for O2T` = CHECKED



ATOM / Fieldbus / PROFINET / Overview

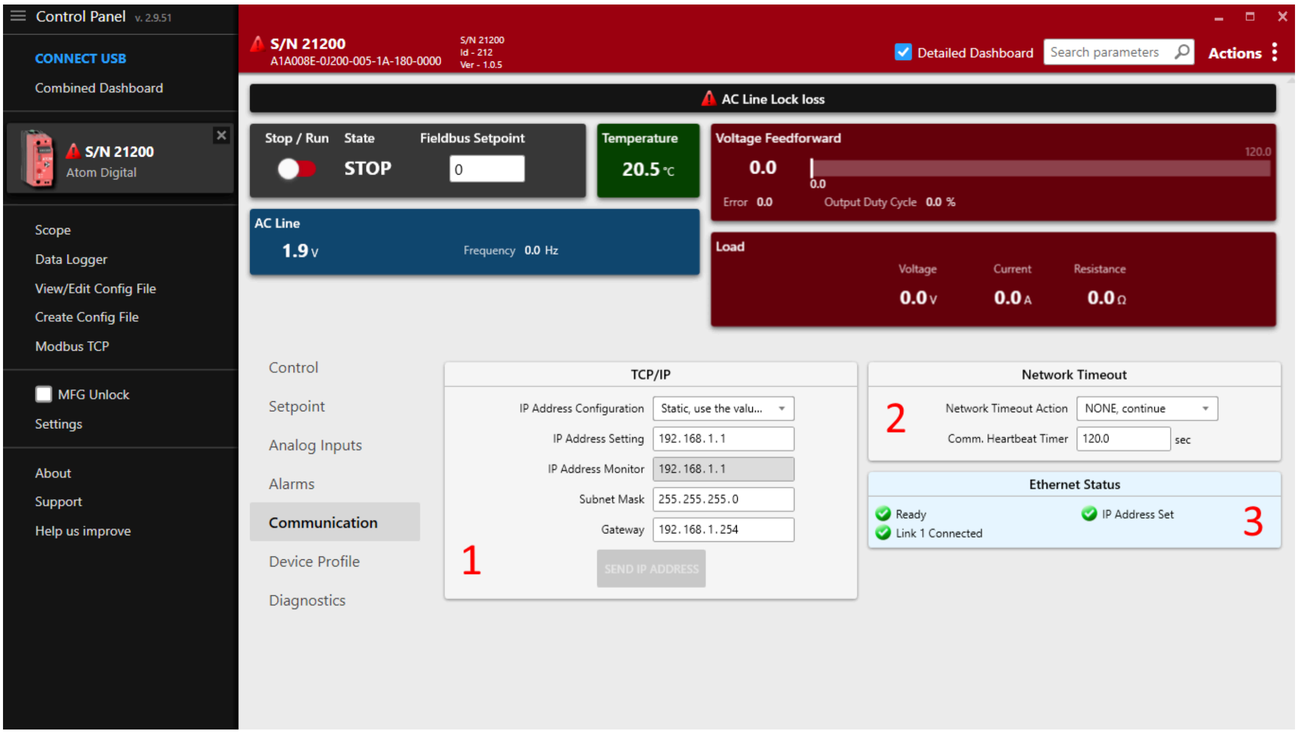
> **INFO**

GSDML

A general station description markup language file (**gsdml**) for ATOM is available. This file can be imported into most PLC software to easily interface with ATOM.

> **INFO**
>
> You can download the GSDML file for ATOM here.

Control Panel Communication Settings



Some communication settings can be configured in the **Communication** tab in Control Panel.

- Section `1`: TCP/IP settings
  - **IP Address Configuration**
    - `Static`: Use the IP address, subnet mask, and gateway specified below.
    - `DHCP`: Use DHCP to obtain an IP address.
  - **IP Address Setting**: The IP address of the ATOM controller.
  - **IP Address Monitor**: The current IP address of the ATOM controller.
  - **Subnet Mask**: The subnet mask of the ATOM controller.
  - **Gateway**: The gateway address for the ATOM controller.
- Section `2`: Network Timeout
  - The EtherNet/IP heartbeat timeout (Encapsulation Inactivity Timeout) in seconds.
  - You can configure a network timeout action to perform when the device loses communication with the PLC:
    - `None`: Do nothing

        - `STOP, fault shutdown`: STOP the controller, disabling output
        - `Use network timeout setpoint`: Configure an alternative setpoint to use when the controller loses communication with the PLC.
- Section `3`: Ethernet status
    - Indicates the status of both RJ45 ports, IP address configuration, conflict detection, and any other errors with the EtherNet/IP connection.

> **INFO**
>
> ### Control Panel and PLC software
>
> These settings are synchronized with your PLC environment. You do not have to use Control Panel to change these settings - you can stay in your PLC software. Control Panel merely provides them as an altnerative way to configure ATOM's EtherNet/IP settings.
>
> You can use Control Panel simultaneously with your PLC software without issues.

> **WARNING**
>
> ### IP Address Conflict Detection
>
> ATOM uses **IP Address Conflict Detection** to detect IP address conflicts on the network. If ATOM detects another device using the same IP address, it will disable all network communication until the conflict is resolved.
>
> Please ensure all devices on the network are assigned unique a IP address.

Hardware considerations

> ⚠ **WARNING**
>
> ## Daisy chaining
>
> As ATOM has two RJ45 ports, it can be easily daisy-chained. When daisy-chaining ATOM, take care to avoid a loop in the network. In some loop configurations, ATOM is susceptible to network broadcast storms, which can cause the controller to become unresponsive. If you are daisy-chaining ATOM, ensure that the network is loop-free.
>
> ATOM works with both unmanaged and managed switches. We recommend a managed switch for larger networks to give you more control over the network topology.

Parameters

Overview

ATOM makes 30 parameters accessible to Profinet. These parameters are made available through the input and output modules.

Table

#	Name	Type	Description	Read/Write
1	Digital setpoint	DINT	A value between 0 and 10,000 indicating the desired output current. The value is scaled to the output range of ATOM. For example, if the output range is 0-	Read/Write

#	Name	Type	Description	Read/Write
			100A, a value of 5000 would set the output to 50A.	
2	Digital run enable	BOOL	Enables or disables the output current. When disabled, the output current is set to 0A.	Read/Write
3	Inhibit Alarm Status	BYTE	A bitfield indicating alarms that are preventing controller operation. See Inhibit Alarm Status.	Read
4	Warning Alarm Status	BYTE	A bitfield indicating warning alarms. See Warning Alarm Status.	Read
5	Feedback Read Status	BOOL	A bitfield indicating if controller has acquired feedback. See Feedback Read Status.	Read
6	AC Line Frequency	REAL	The AC line frequency in Hz.	Read
7	AC Line Voltage	REAL	The AC line voltage in volts.	Read
8	Load Voltage	REAL	The load voltage in volts.	Read
9	Load Current	REAL	The load current in amps.	Read
10	Load Resistance	REAL	The load resistance in ohms.	Read

#	Name	Type	Description	Read/Write
11	Heatsink Temperature	REAL	Heatsink temperature, in degrees celsius.	Read
12	Output Duty Cycle %	REAL	Indicates the amount, in percent, that the output of the controller is ON	Read
13	Setpoint reference	REAL	Reference input to control compensation loop in units determined by "feedback type"	Read
14	Feedback	REAL	The control output supplied to the load in units determined by "feedback type"	Read
15	Partial Load Fault Target Resistance	REAL	Expected nominal resistance, in Ohms, of the load. Used for partial load fault detection.	Read
16	Partial Load Fault Resistance	REAL	The actual load resistance in Ohms. Compared with #15 to determine if a partial load fault has occurred.	Read
17	Partial Load Fault Resistance Deviation	REAL	The tolerable percentage that parameter #15 and #16 may differ by until a partial load fault will be triggered.	Read
18	Firmware ID	DINT	Indicates the version of firmware that is loaded, dictating which	Read

#	Name	Type	Description	Read/Write
			features are available.	
19	Firmware major revision	DINT	Indicates which revision of the firmware is loaded. Major revisions fix critical bugs or add significant new features.	Read
20	Firmware minor revision	DINT	Indicates which minor revision of the firmware is loaded. Minor revisions fix minor issues and/or add minor improvements.	Read
21	Full Scale Voltage	DINT	The expected output voltage when the controller output is fully on.	Read
22	Full Scale Current	REAL	The expected current when the controller output is fully on.	Read
23	AC Line Status	BYTE	A bitfield indicating the status of the connected AC Line. See AC Line Status.	Read
24	Load Status	BYTE	A bitfield indicating the load status. See Load status.	Read
25	Controller Status	BYTE	A value indicating the operational status of the controller. See Controller status.	Read

#	Name	Type	Description	Read/Write
26	Controller State	BYTE	A value indicating the controller state. See Controller state.	Read
27	EEPROM Status	WORD	A bitfield indicating the EEPROM status. Seee EEPROM Status.	Read
28	EEPROM Status 2	WORD	Identical to parameter #27	Read
29	Error Latch	BYTE	A bitfield used for diagnostic troubleshooting. See Error Latch.	Read
30	Miscellaneous Status	BYTE	A bitfield indicating miscellaneous status information. See Miscellaneous Status.	Read

Additional parameter descriptions

Inhibit Alarm Status

Inhibit alarm status is a 8-bit bitfield:

7	6	5	4	3	2	1
Reserved	Reserved	Reserved	Reserved	Feedback Loss	Over Temperature	Over Current Trip

If any bit is set to 1, the controller will *not* be allowed to run.

Warning Alarm Status

Warning alarm status is a 8-bit bitfield:

7	6	5	4	3	2	1	0
Reserved	Reserved	High temperature	Shorted SCR	Open Load	Partial Load Fault	Current Limit	Voltage Limit

Warning alarms are not considered critical and will not prevent the controller from running.

Feedback Read Status

Feedback status is a 8-bit bitfield:

7	6	5	4	3	2	1	
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Ti

Indicates whether the controller has acquired feedback on the line. If any bit is set to 1, then the controller has lost feedback.

AC Line Status

AC Line status is a 8-bit bitfield:

7	6	5	4	3	2	1	0
Reserved	Reserved	Sync-Locked (to AC Line)	Pre-Lock 2	Pre-Lock 1	Reserved	AC Line B OK	AC Line A OK

Bits 5 must be set to 1 before the controller can provide power to the load.

Load Status

Load status is a 8-bit bitfield:

7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	Open Load	Reserved	Reserved	Reserved	Short SCR

Controller Status

Controller status is one of:

Value	Description
0	Disabled
1	Initialization
2	Normal, operating
3	Calibration

Value	Description
4	Diagnostic

Controller State

Controller state is one of:

Value	State	Description
0	STOP	The state the controller is in when AC Line voltage is not present.
1	RUN	The state the controller is in when AC Line voltage is present and the controller is synchronized to the AC line.
2	FAULT	A latching state of output shutdown caused by over current or over temperature alarms. A power cycle or processor reset is required to clear this state.
3	FAULT RESET	Used as a temporary state to transition from FAULT to RUN once again.

EEPROM Status

EEPROM status is an 16-bit bitfield. EEPROM is used to store controller configuration and calibration data. Any errors in EEPROM may indicate that the firmware is corrupted.

Bit	Description
0	EEPROM Initialization

Bit	Description
1	SP Table Error
2	MFG CP Table Error
3	Calibration Table Error
4	Reserved
5	Reserved
6	Backup Calibration Table Error
7	Bottom Board Calibration Table Error
8	SP Definition Table needs updating
9	Bottom Board Calibration Backup Error
10	Reserved
11	Reserved
12	EEPROM is write protected
13	Reserved
14	Reserved
15	Feedback Calibration Table has changed, store to EEPROM

Error Latch

Error latch is a 8-bit bitfield:

7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	Feedback loss	SCR timing loss	Line Frequency failure	Phase loss or missing cycle	Line Lock Loss

Error latch is provided as a diagnostic troubleshooting aid.

Miscellaneous Status

Miscellaneous status is an 8-bit bitfield:

7	6	5	4	3	2	1
Reserved	Initialization in progress	Reserved	Reserved	Waiting for ENTER key during initialization	Reserved	USB Powered

Data types

The data types listed above in the parameter table are defined in the CIP standard as:

Type	Size	Description
BOOL	1 byte	Boolean value

Type	Size	Description
BYTE	1 byte	8-bit bitmap
WORD	2 bytes	16-bit bitmap
DWORD	4 bytes	32-bit bitmap
LWORD	8 bytes	64-bit bitmap
USINT	1 byte	Unsigned 8-bit integer
UINT	2 bytes	Unsigned 16-bit integer
UDINT	4 bytes	Unsigned 32-bit integer
ULINT	8 bytes	Unsigned 64-bit integer
SINT	1 byte	Signed 8-bit integer
INT	2 bytes	Signed 16-bit integer
DINT	4 bytes	Signed 32-bit integer
LINT	8 bytes	Signed 64-bit integer
REAL	4 bytes	32-bit floating point number
LREAL	8 bytes	64-bit floating point number

Advanced

ATOM has many more parameters beyond the 30 made available through Prrofinet. The default profile listed above should be sufficient for the majority of use cases.

If this is not the case, you can use Control Panel to adjust or monitor all parameters.

In the rare case that you need more parameters available through ATOM's Profinet profile, Control Concepts does have the ability to make additional parameters available or to change the data type of included parameters. Please contact us if you would like a custom Profinet profile. There may be a service fee for custom Profinet profiles as they require new GSDML files, device-reconfiguration and testing.

ATOM / Fieldbus / PROFINET / TIA Portal V18

In this tutorial, you'll learn how to connect Atom to a Siemens PLC over Profinet. You'll learn how to update the run/stop and setpoint parameters, and monitor the heatsink temperature.

If you haven't yet, please review ATOM's Profinet Profile.

If you'd like to skip the tutorial, you can download a completed example project in TIA Portal V18:

- Download AtomExample.zip

Requirements

1. TIA Portal V18
2. A Siemens PLC (optional, you can still follow along by simulating the PLC on your PC).
    i. We use a `S7-1511-1 6ES7 511-1AK02-0BA0` in this example.
    ii. You can easily follow along with a different PLC if you have a different one.
3. Download Atom's GSDML file

Hardware setup

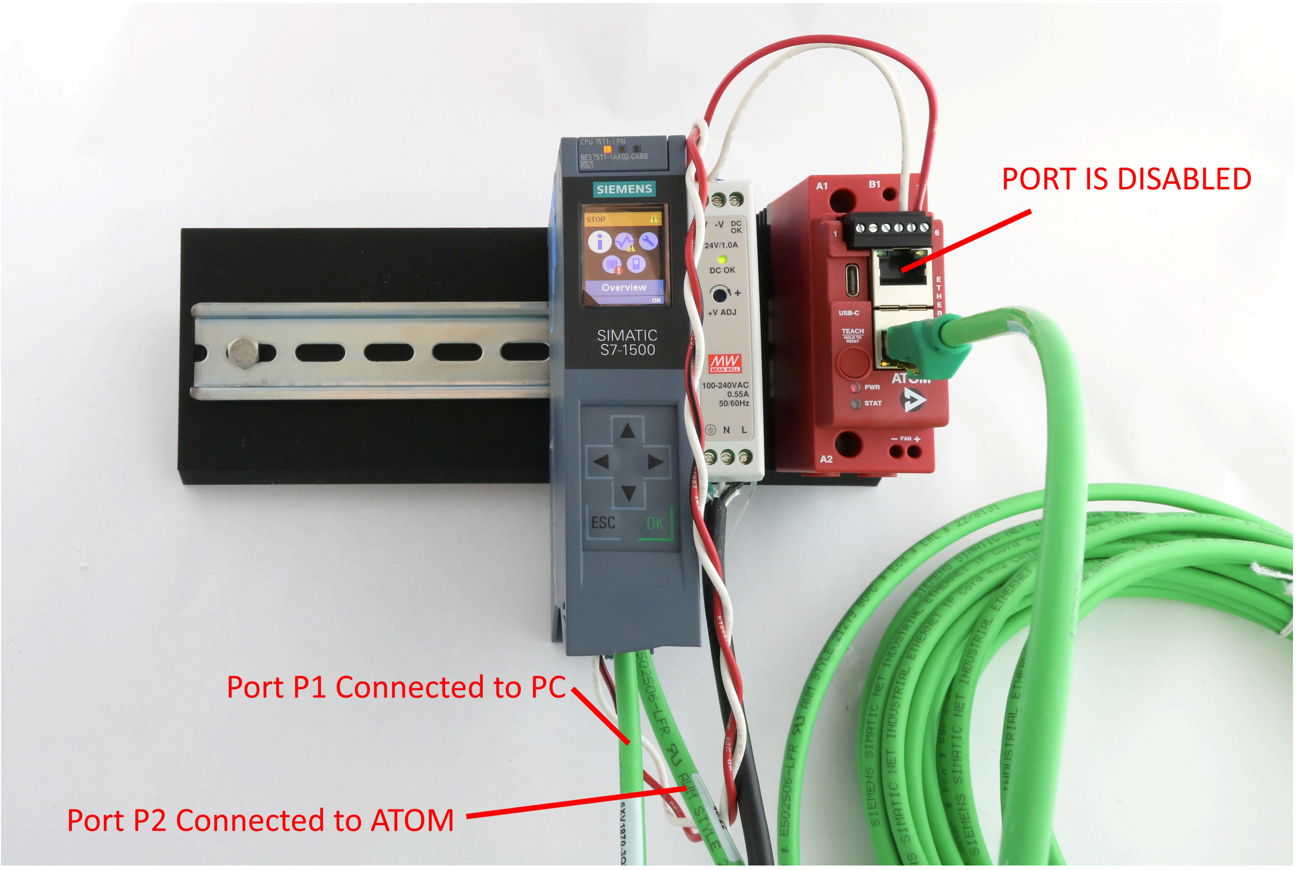
> **INFO**
>
> If you're using the PLC simulator, skip to the next section.

IMPORTANT

When Atom is configured for Profinet, the Ethernet port closest to the 24V power connector is **disabled**.

Connect 24V to your PLC and Atom unit with the provided power cable. Connect two Ethernet cables to the PLC:

- P1: Connect to your PC
- P2: Connect to your Atom unit



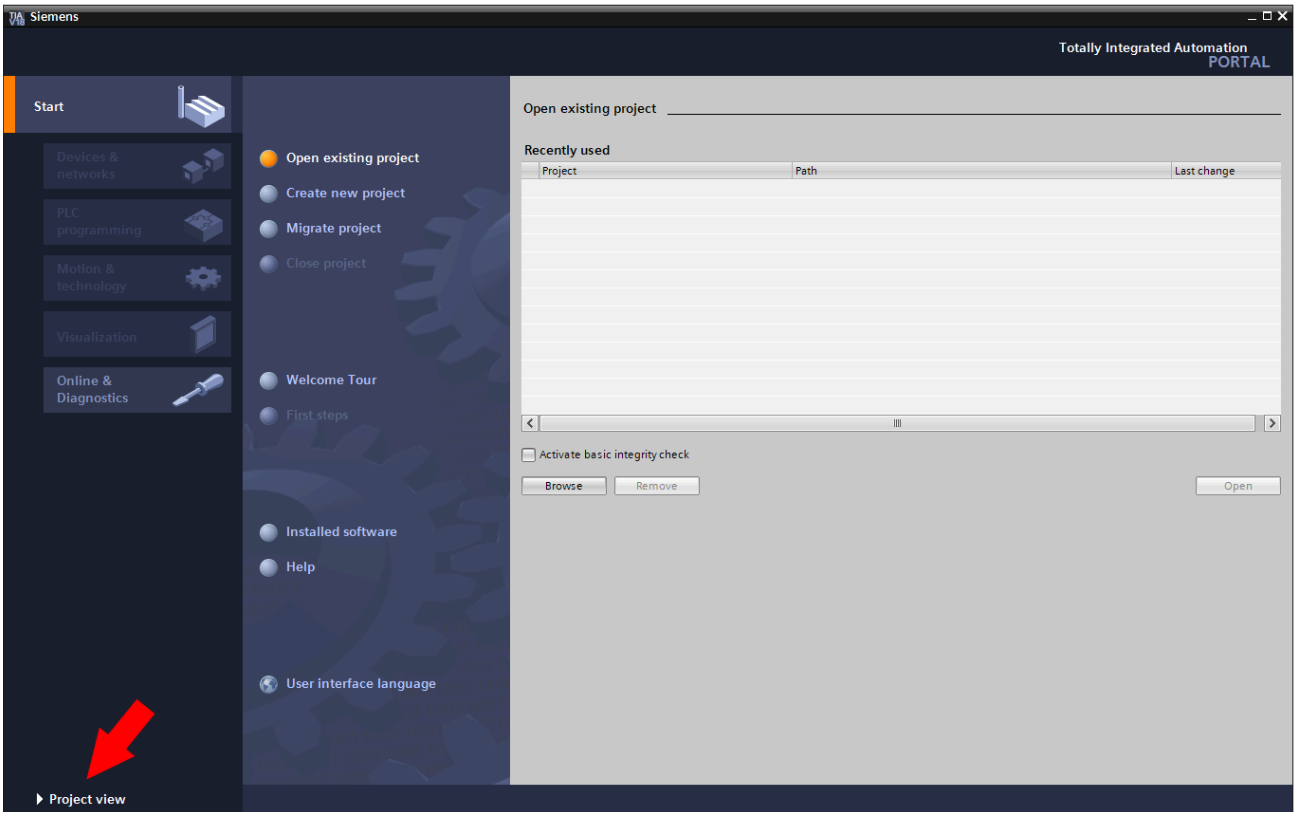
INFO

To simplify this diagram, we have not connected a load to Atom.

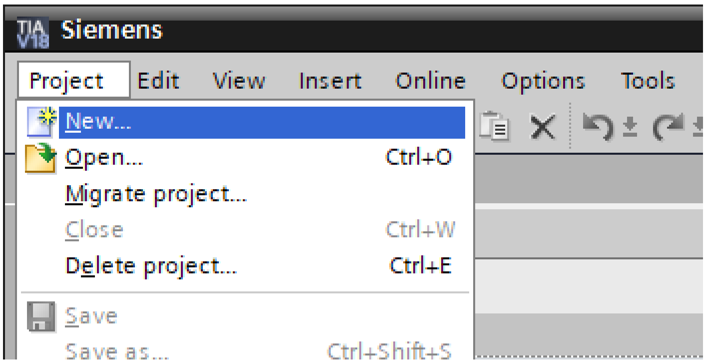
If you do not connect a load, you can still verify your PLC is working by connecting a USB cable to Atom and using Control Panel to watch the parameters change/verify the PLC is receiving the correct monitor data.

Creating a project in TIA Portal V18

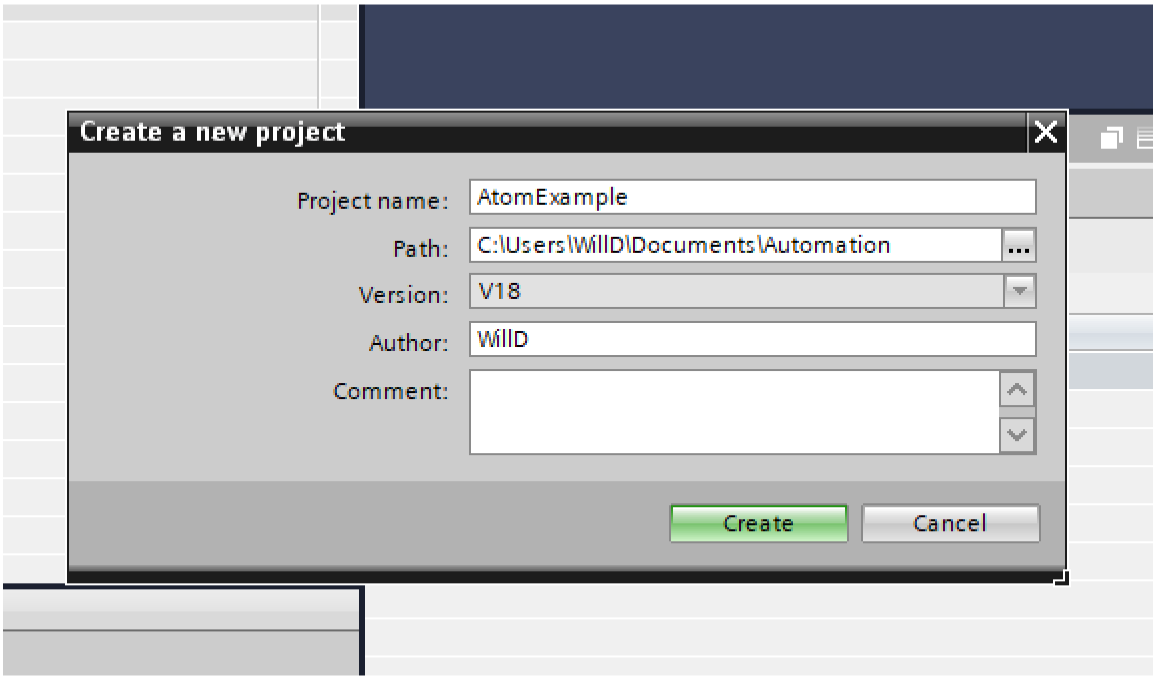
Open TIA Portal V18. If you're in the *Portal view* (shown below), switch to *Project view* by clicking in the lower left:



Next, select **Project** > **New**:



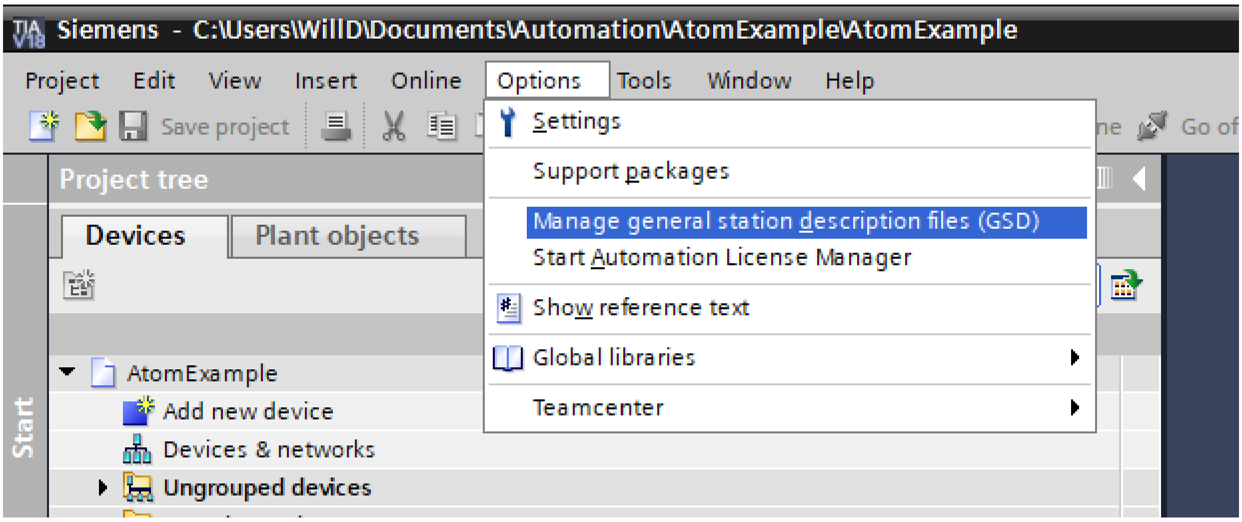
Give your project a name, like `AtomExample`, then click **Create**:



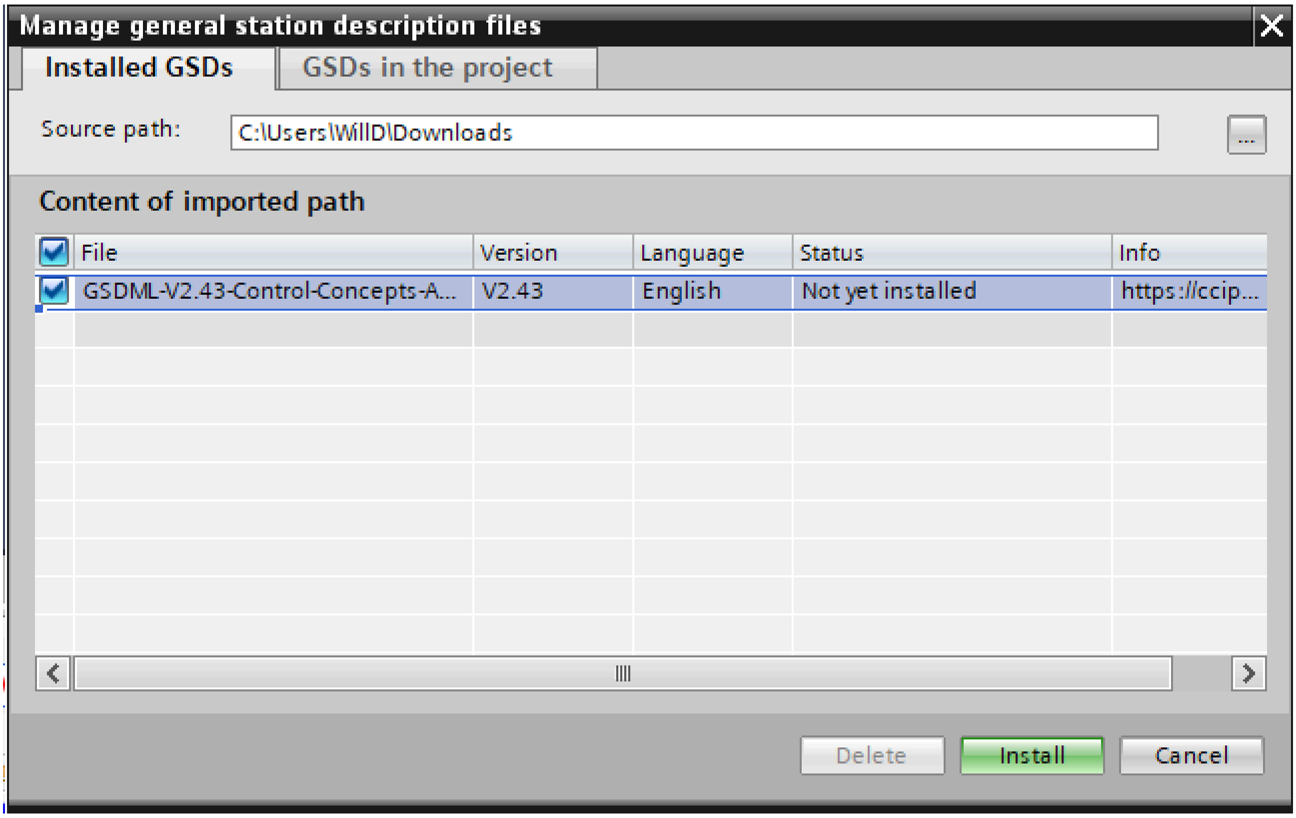
Importing Atom's GSDML file into TIA

A GSDML file describes the capabilities of a Profinet device — the parameters, communication modes, diagnostics (and more) that the device supports. We'll import Atom's GSDML file into TIA so that TIA knows how to talk to Atom.

Select **Options** > **Manage general station description files**:



Enter the path of the folder containing your GSDML file, check `GSDML-V2.43-Control-Concepts-ATOM-20231108.xml`, then click **Install**:

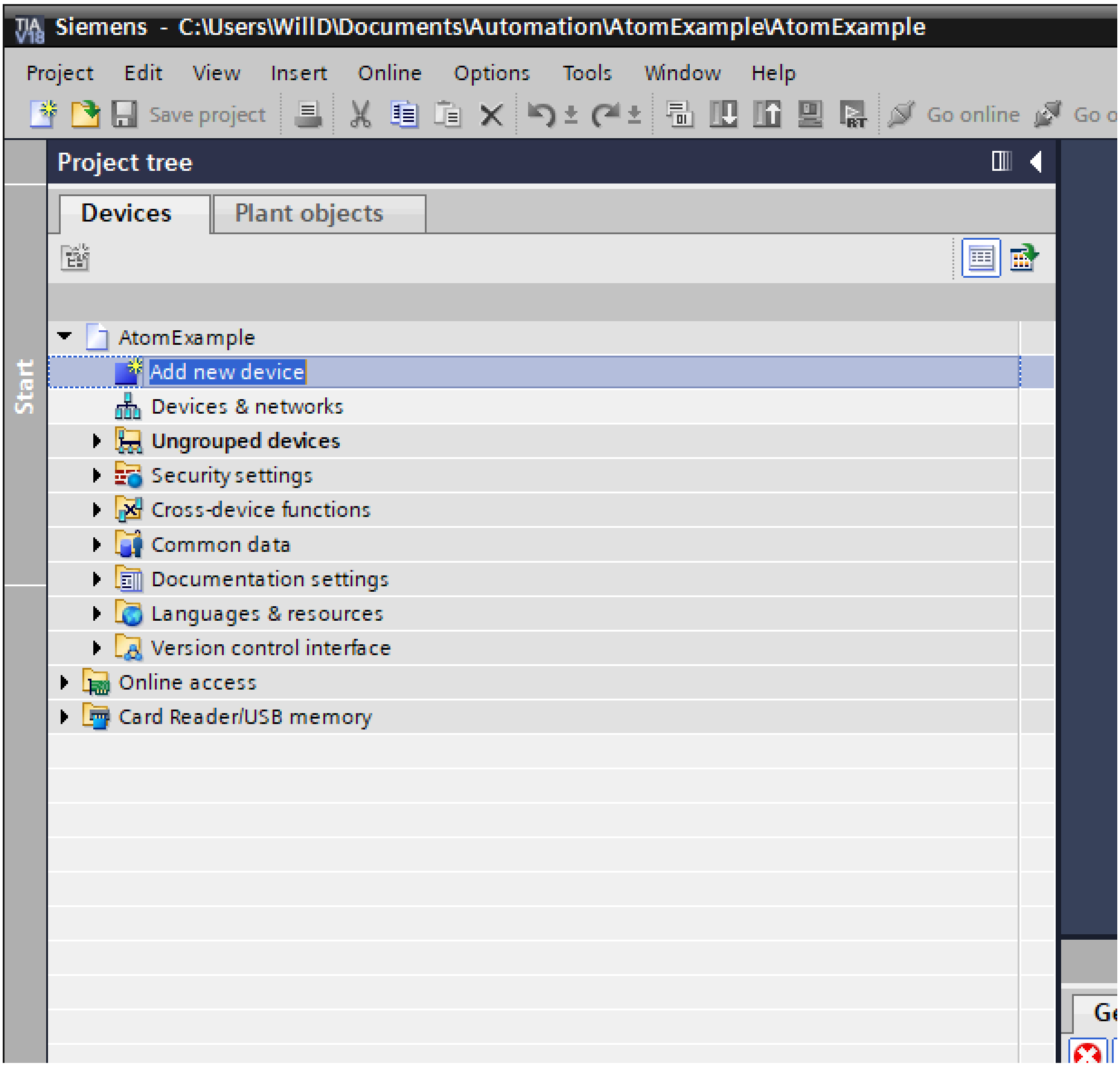


If everything goes well, a dialog should appear "Installation was completed successfully". Click **Close**.

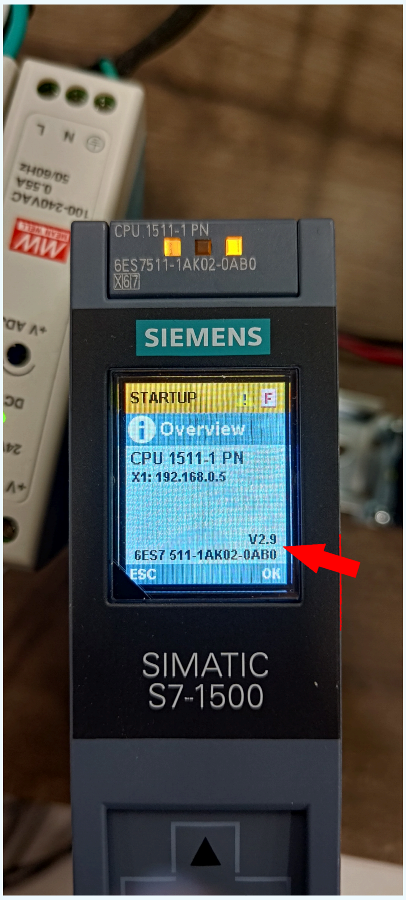
Adding and configuring your PLC

> **INFO**
>
> Even if you're using the PLC simulator, still follow this section and add a Siemens PLC to your project. TIA is capable of simulating the S7-1500 series PLCs.

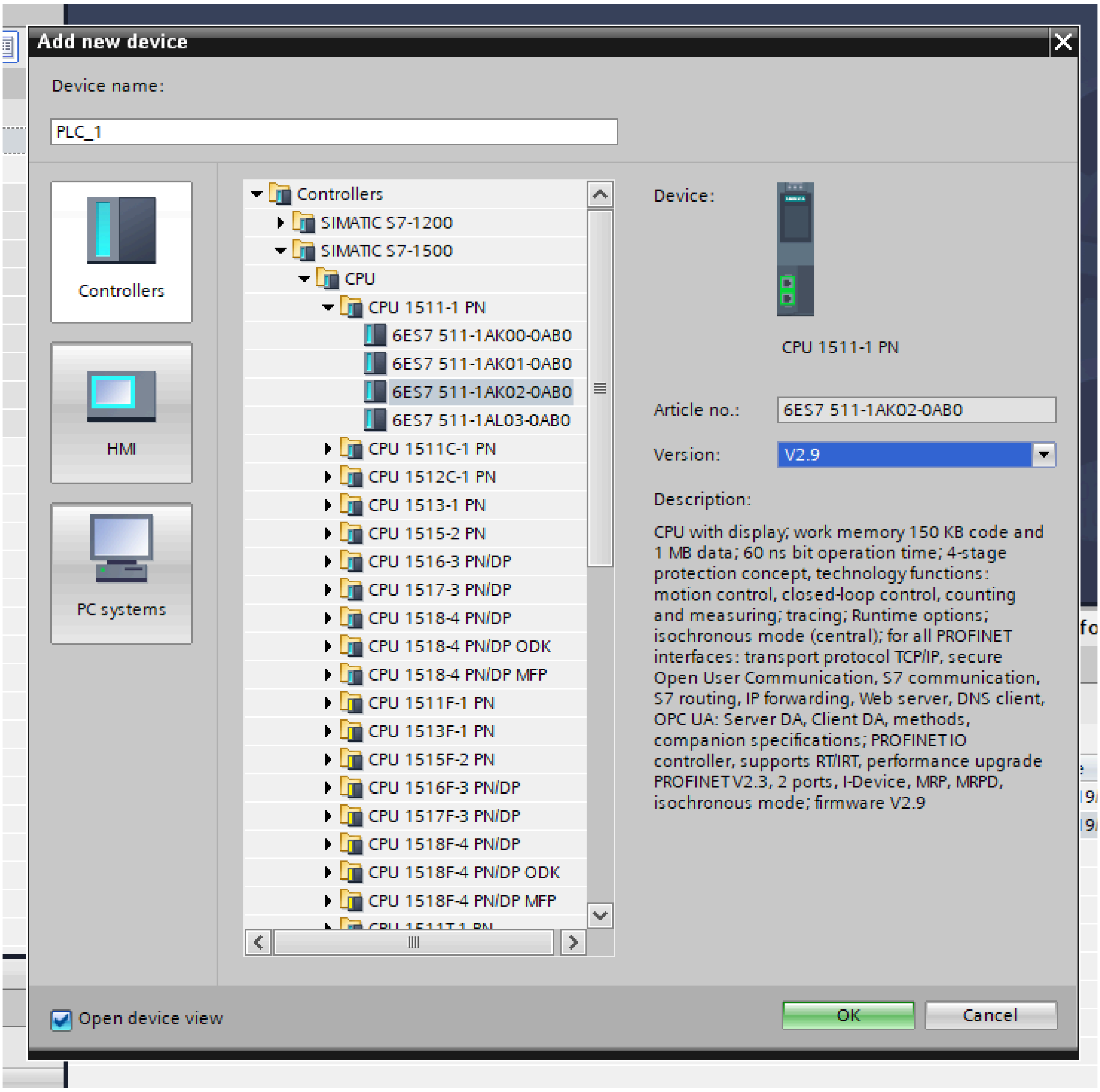
In the **Project tree** pane, within the **Devices** tab, double click **Add new device**:



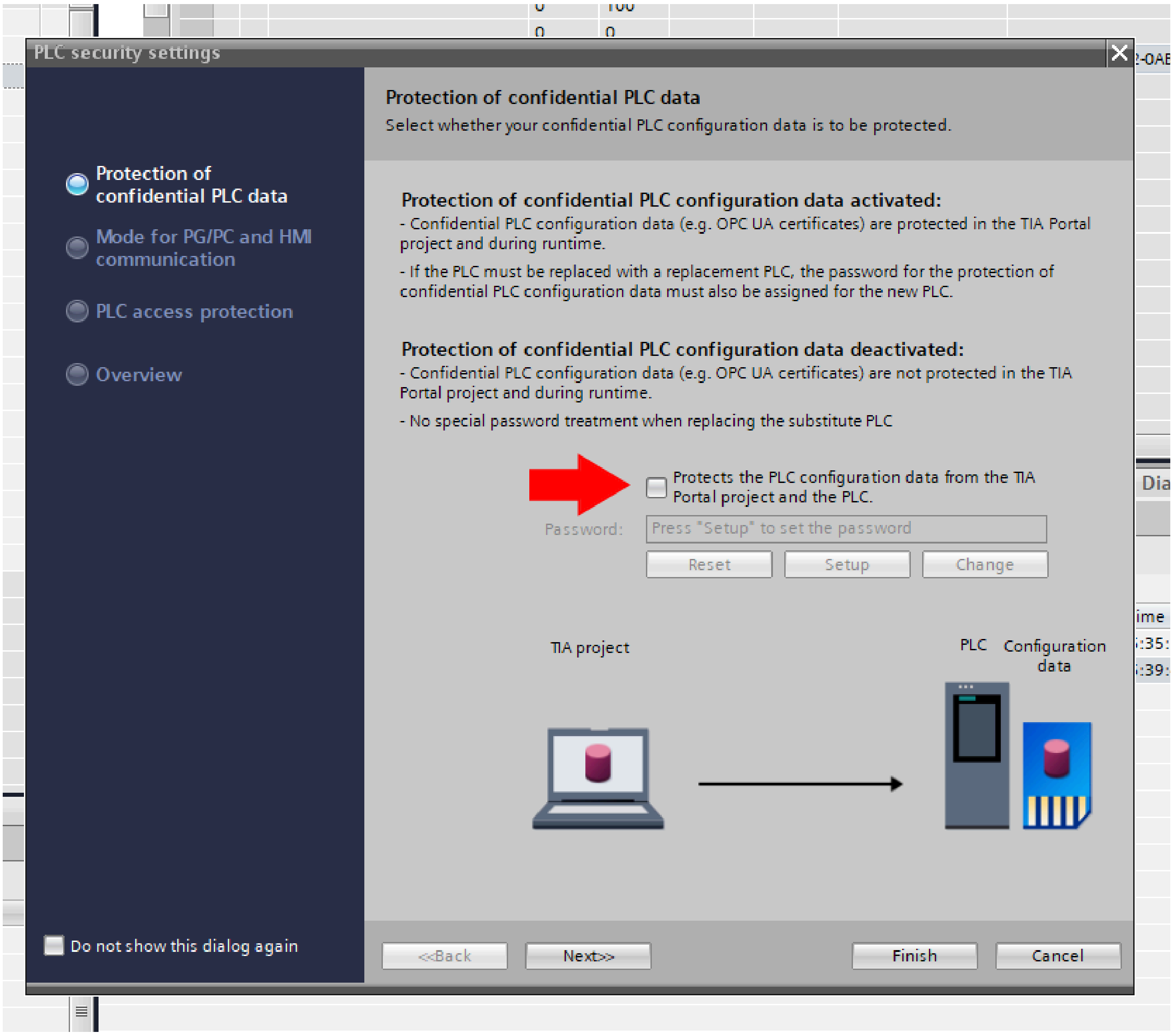
Select the **Controllers** tab, and browse to **Controllers** > **SIMATIC S7-1500** > **CPU** > **CPU 1511-1 PN** > **6ES7 511-1AK02-0AB0** (if you're using a different PLC, select that PLC instead):

> **INFO**
>
> You can generally find your PLC's model number on the frame of your PLC. You can also check the firmware version (and model number) within the **Overview** page on your PLC:
>
> 
>

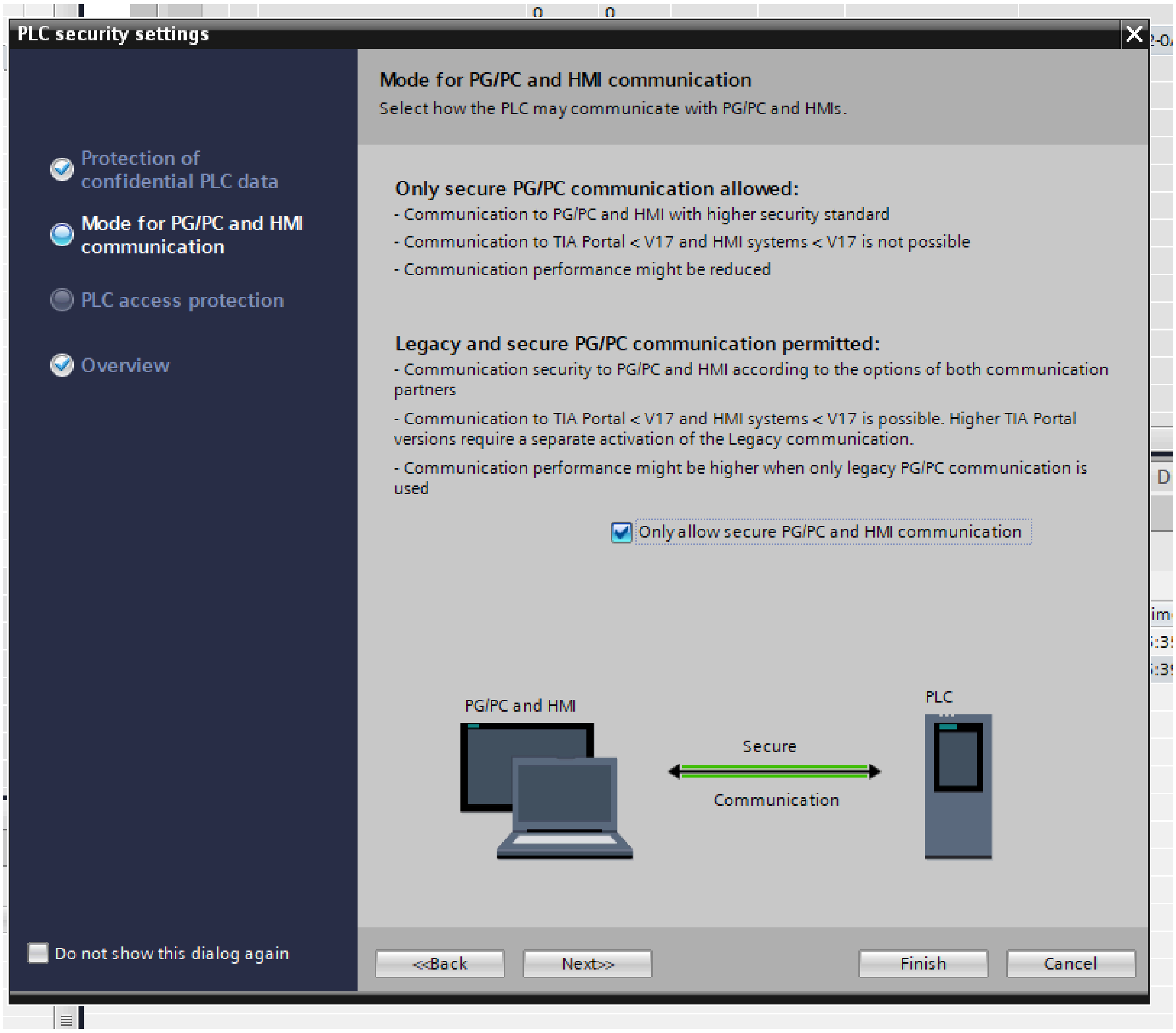
Then, click **Ok**:



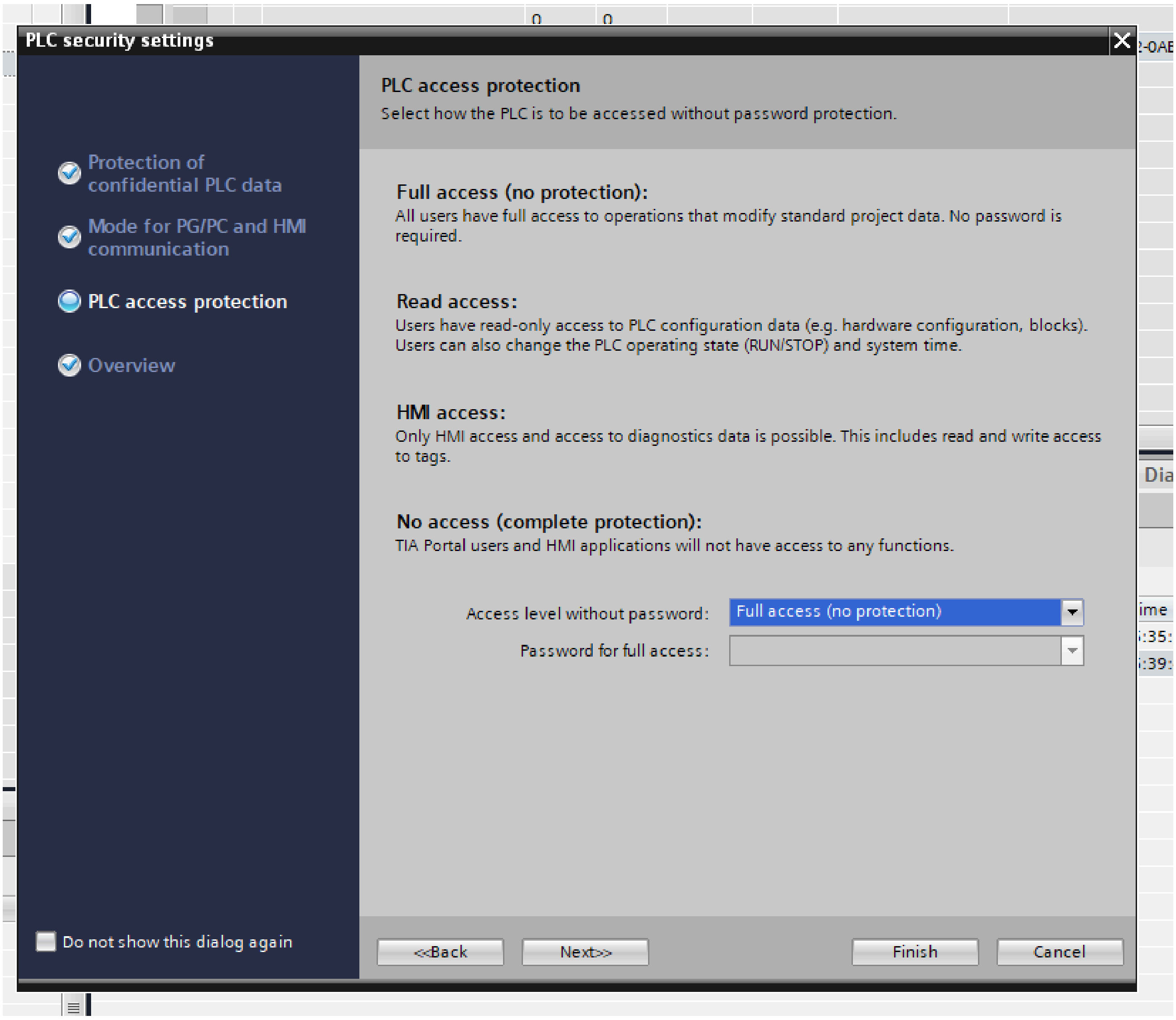
A **PLC security settings** dialog will popup. For this example, we'll disable the PLC password (unchecked):



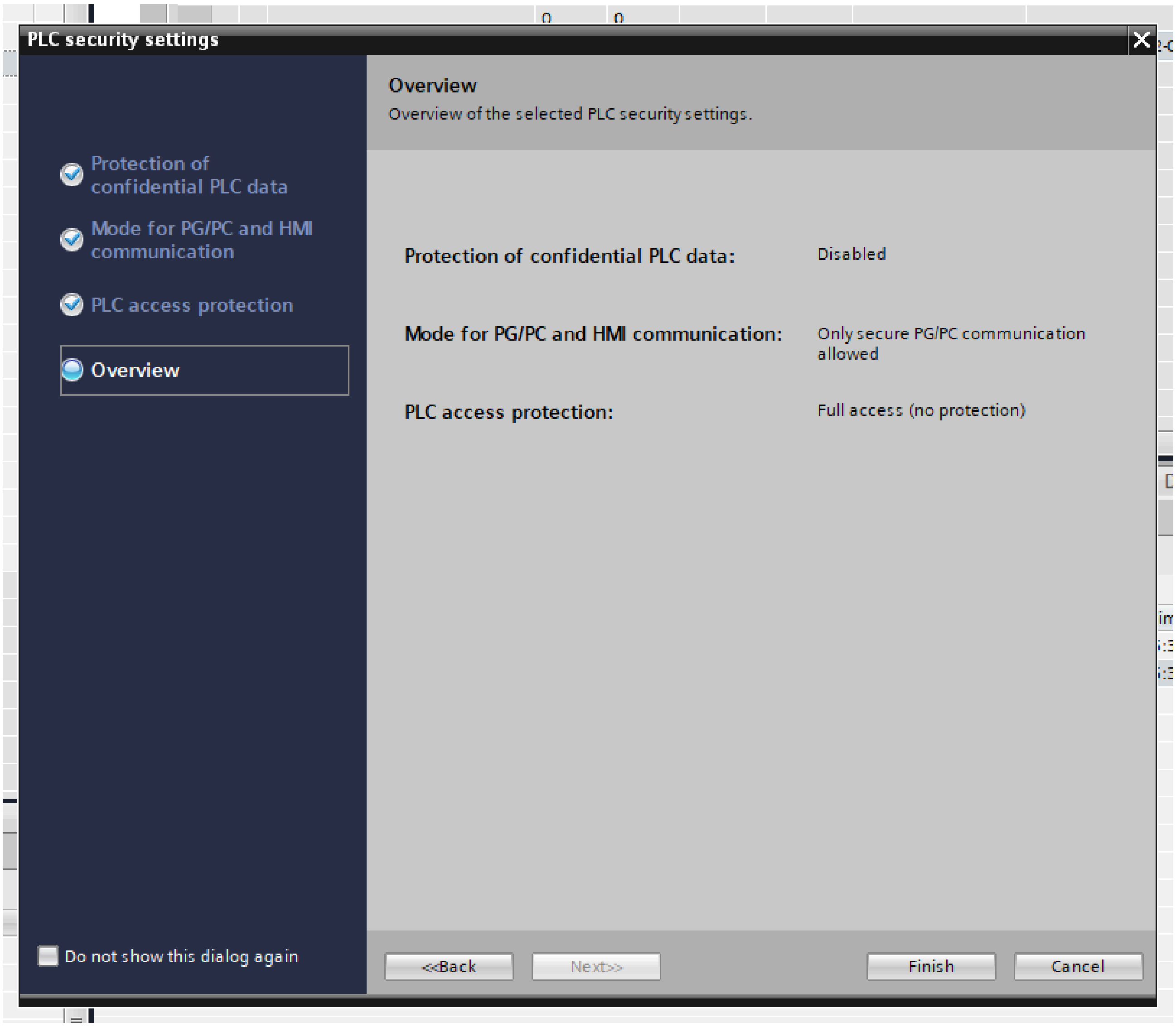
Leave **Only allow secure PG/PC and HMI communications** checked, then click **Next>>**:



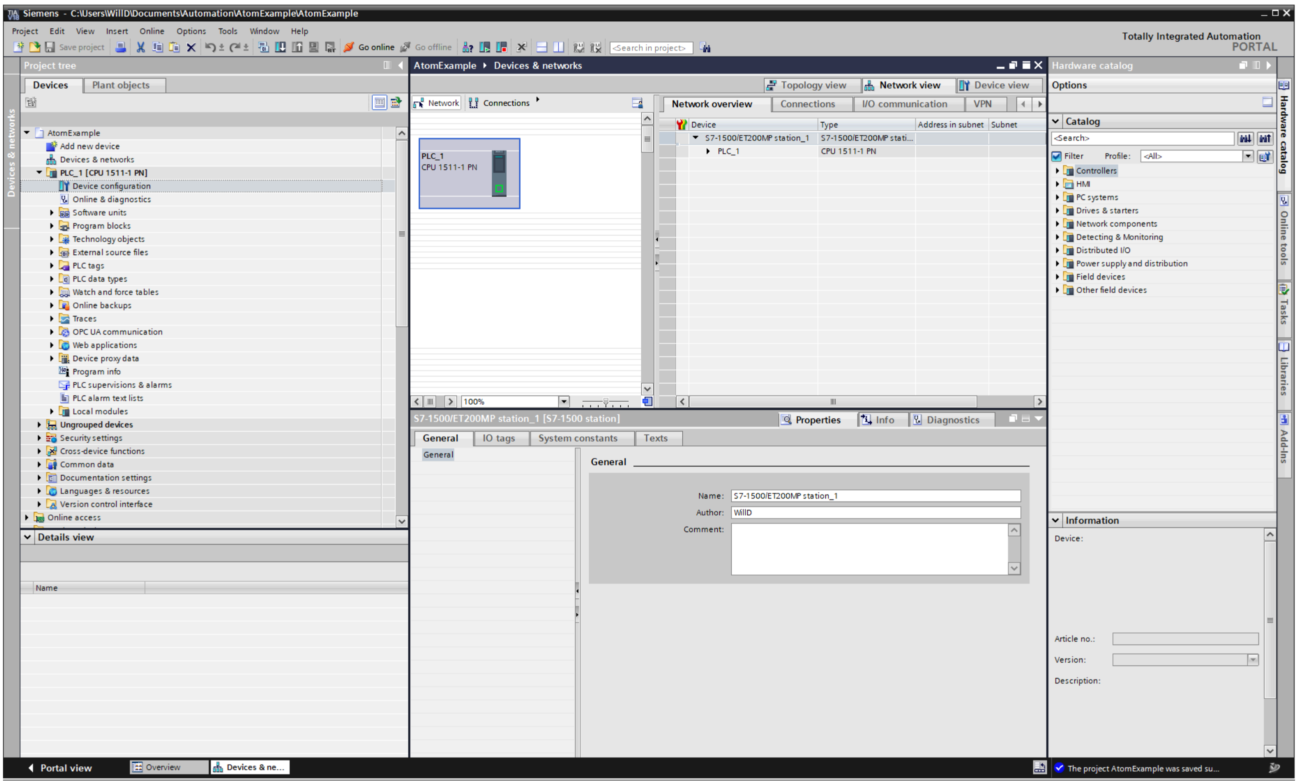
For this example, change **Access level without password** to **Full access (no protection)**, then click **Next>>**:



Then, click **Finish**:

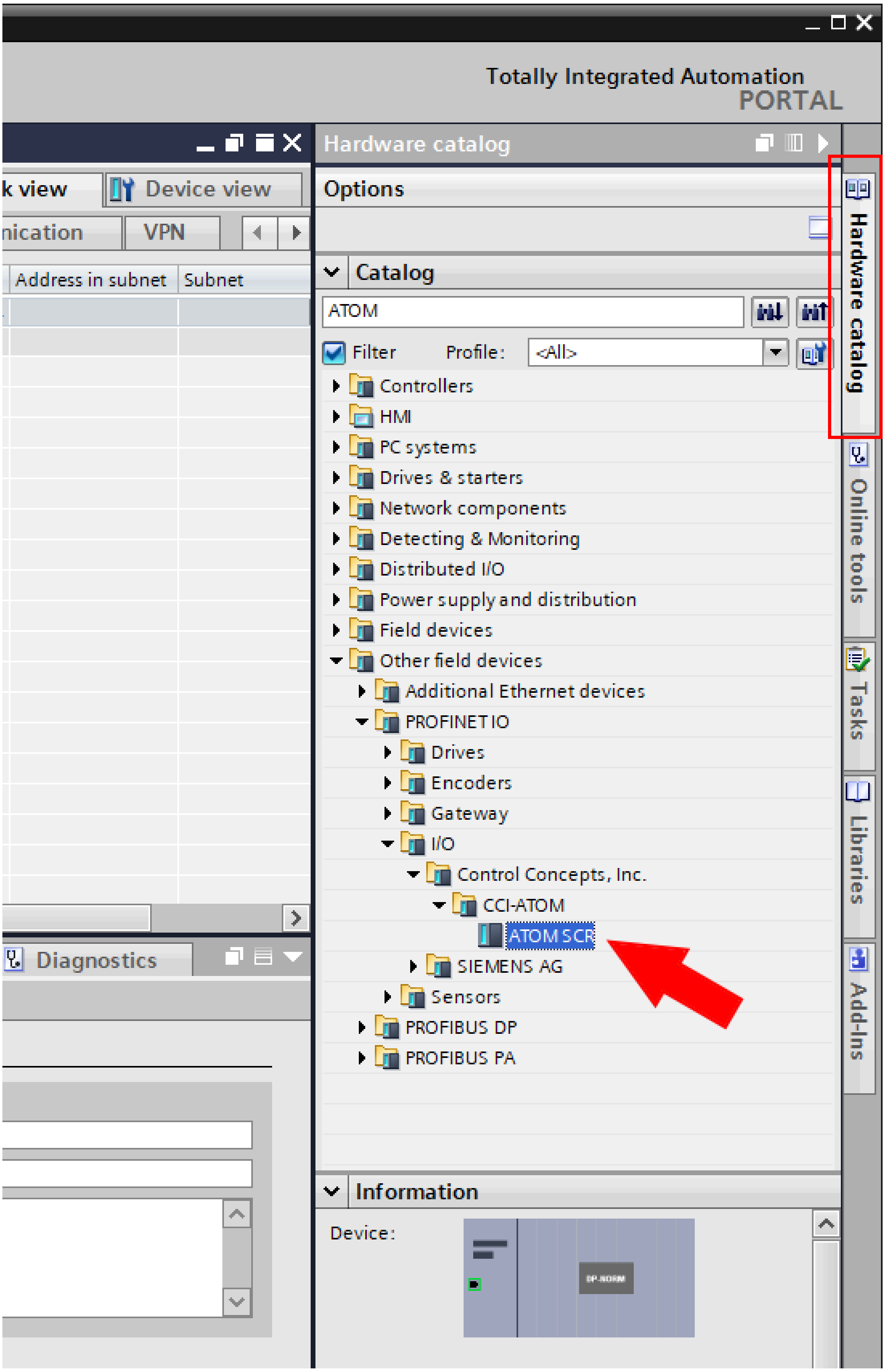


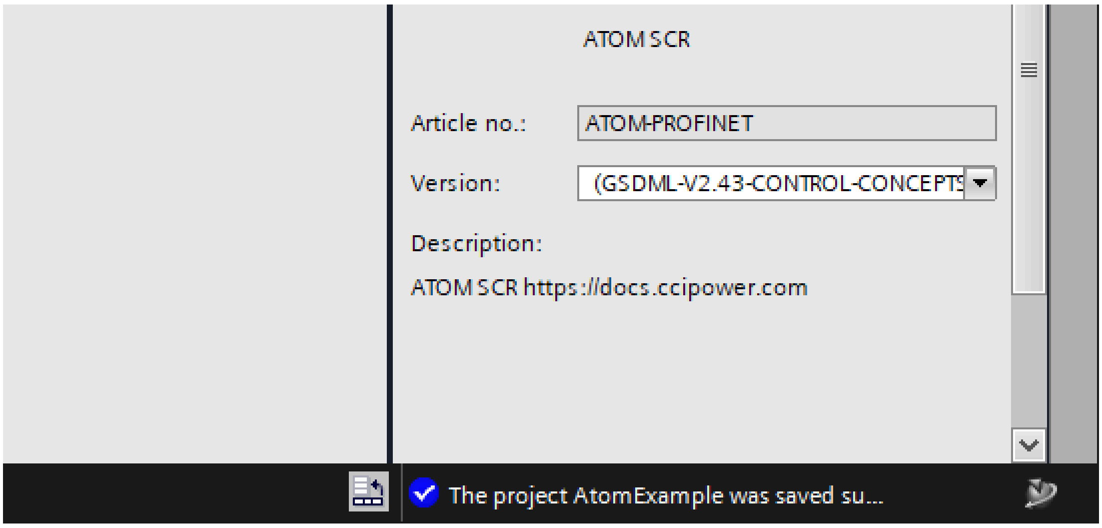
The PLC should appear in the **Network view**:



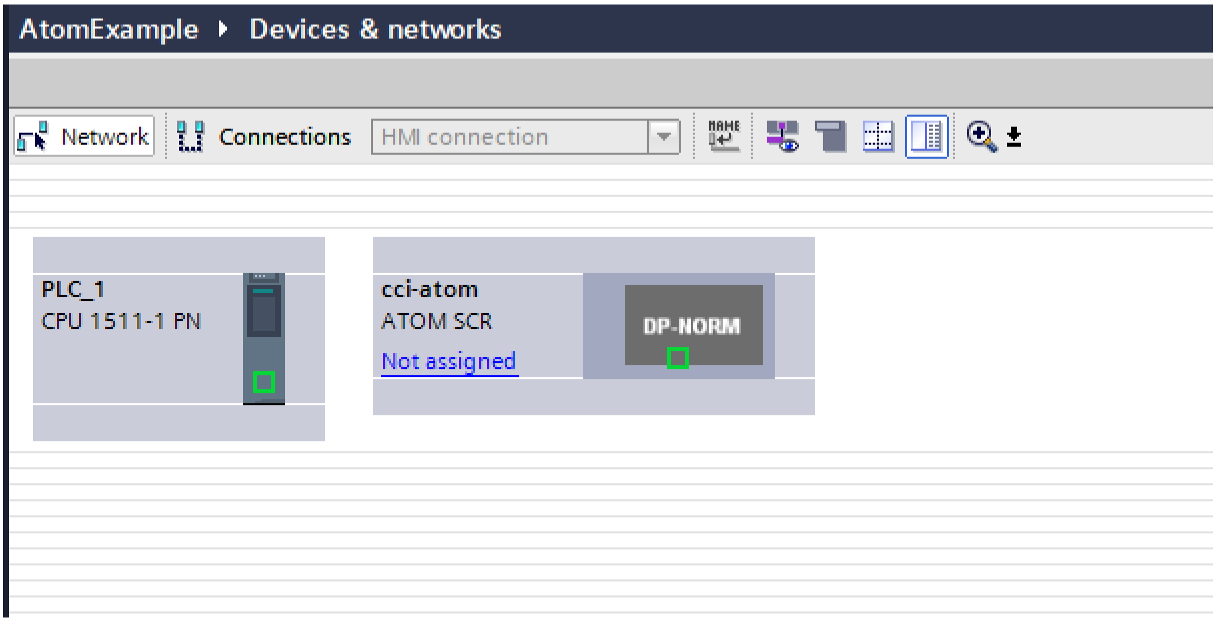
Adding and configuring Atom

To add Atom to your project, select the **Hardware catalog** tab on the right side of the screen. Enter `ATOM` in the search box and double click **ATOM SCR**:

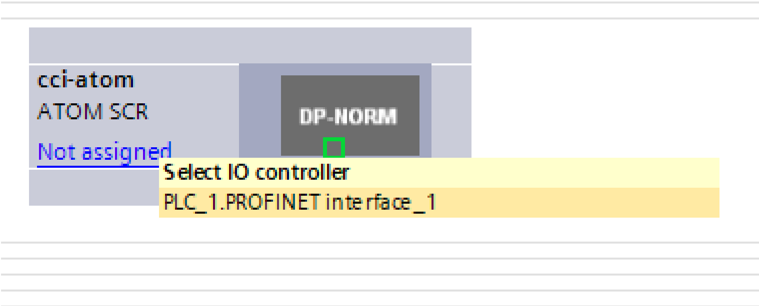

Totally Integrated Automation
PORTAL
Hardware catalog
Options
Catalog
ATOM
Filter
Profile:
<All>
Controllers
HMI
PC systems
Drives & starters
Network components
Detecting & Monitoring
Distributed I/O
Power supply and distribution
Field devices
Other field devices
Additional Ethernet devices
PROFINET IO
Drives
Encoders
Gateway
I/O
Control Concepts, Inc.
CCI-ATOM
ATOM SCR
SIEMENS AG
Sensors
PROFIBUS DP
PROFIBUS PA
Information
Device:
Hardware catalog
Online tools
Tasks
Libraries
Add-ins
Device view
VPN
Address in subnet
Subnet
Diagnostics



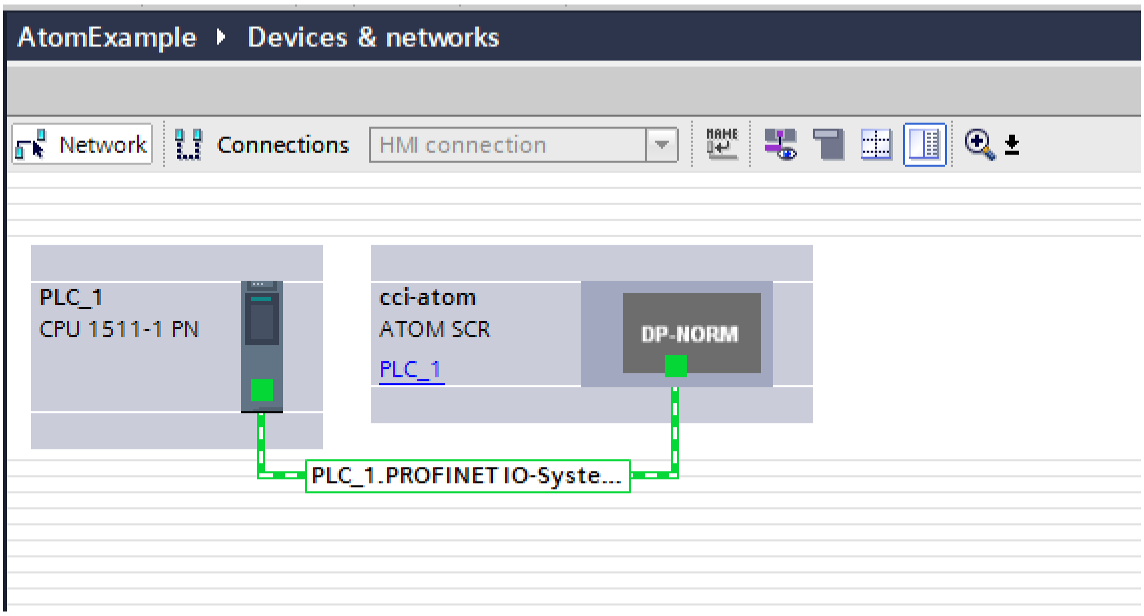
Atom should appear in the **Network view** of your project:



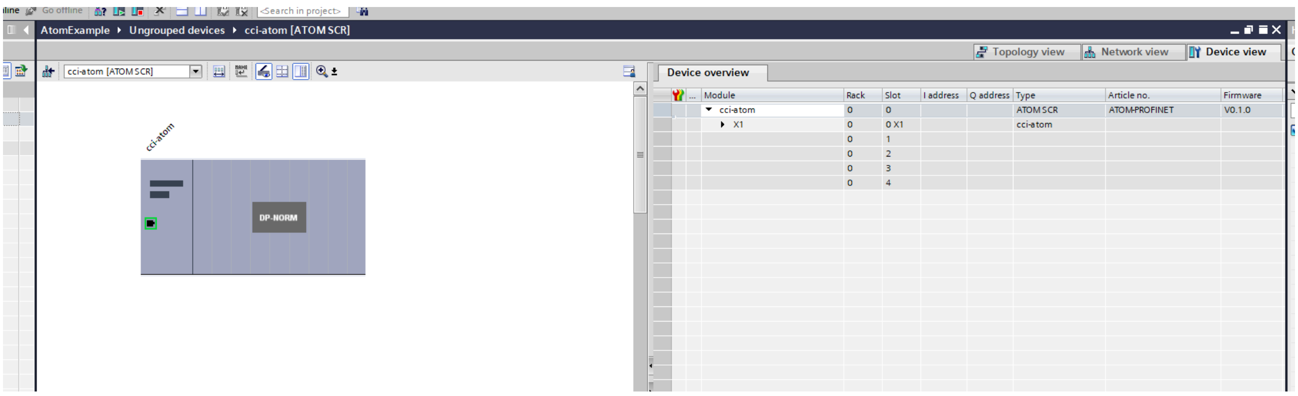
Click **Not assigned** on the Atom block, and select **PLC_1.PROFINET interface_1**:



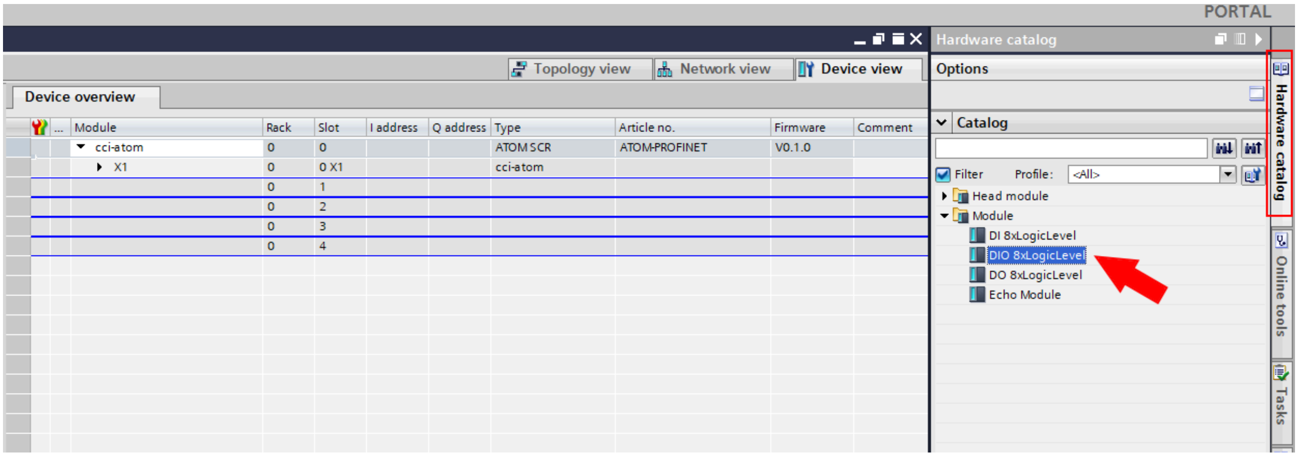
After doing so, a network connection should appear between the PLC and Atom:



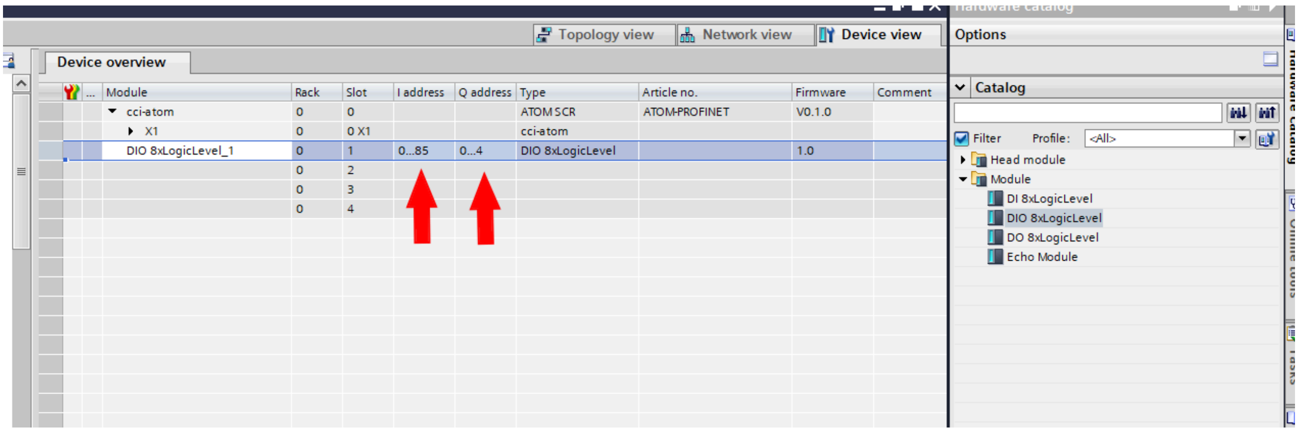
Next, double click on the Atom block to open its **Device view**:



Select the **Hardware catalog** tab on the right side, expand **Module** and select drag the **DIO 8xLogicLevel** module into **Slot 1** of Atom:



Your Atom module configuration should look like this:



INFO

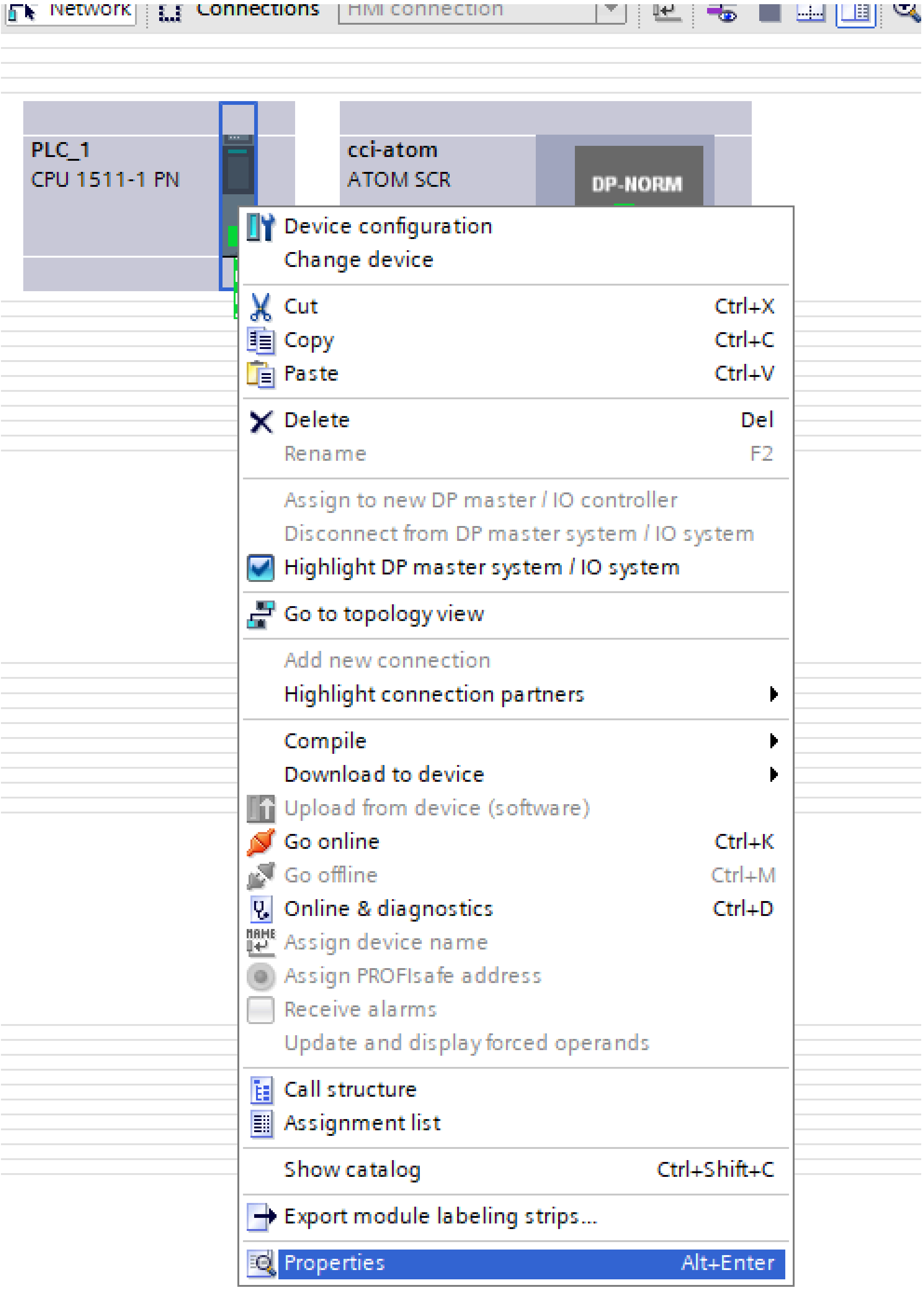
The two red arrows indicate the number of input and output parameters that Atom supports. In this case, the addresses `IW0` to `IW85` are input parameters and `QW0` to `QW85` are output parameters. Check out the Profinet Profile for more information on the available parameters.

Network provisioning

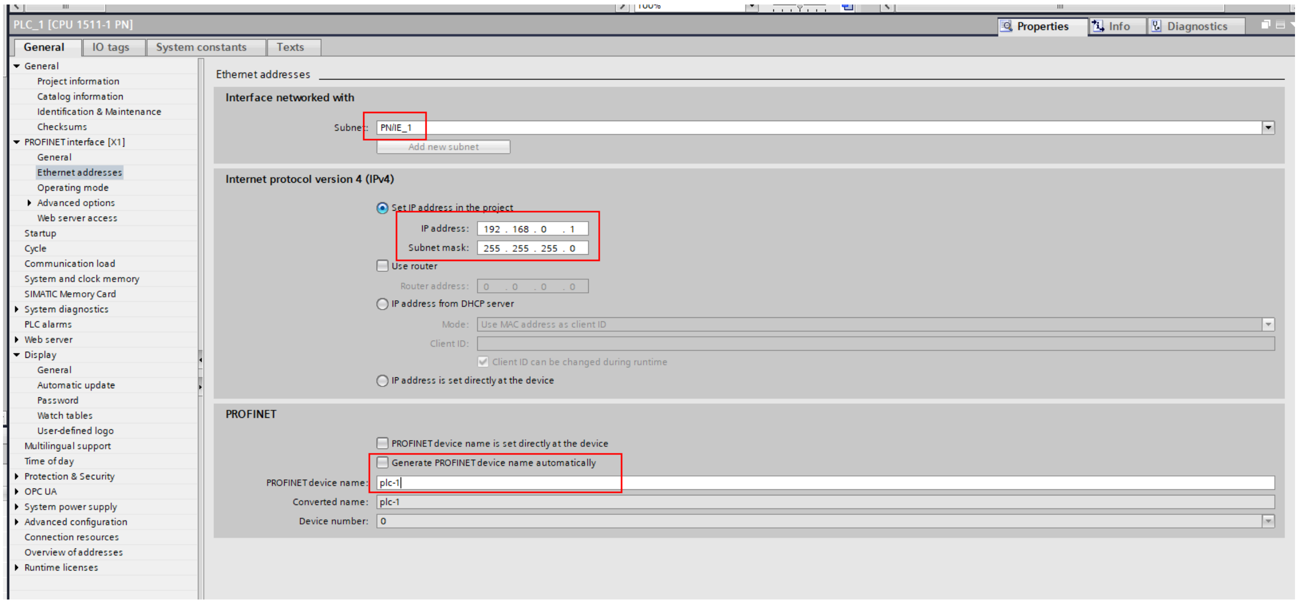
Let's set up a basic Profinet network. We'll provision a basic network like so:

- PC
  - IP address `192.168.0.25`
  - Subnet mask `255.255.255.0`
  - Only used for loading the PLC program
- Siemens PLC
  - Station name `plc-1`
  - IP address `192.168.0.1`
  - Subnet mask `255.255.255.0`
  - Port P1 connected to PC
  - Port P2 connected to Atom
- Atom
  - Station name `atom-1`
  - IP address `192.168.0.2`
  - Subnet mask `255.255.255.0`

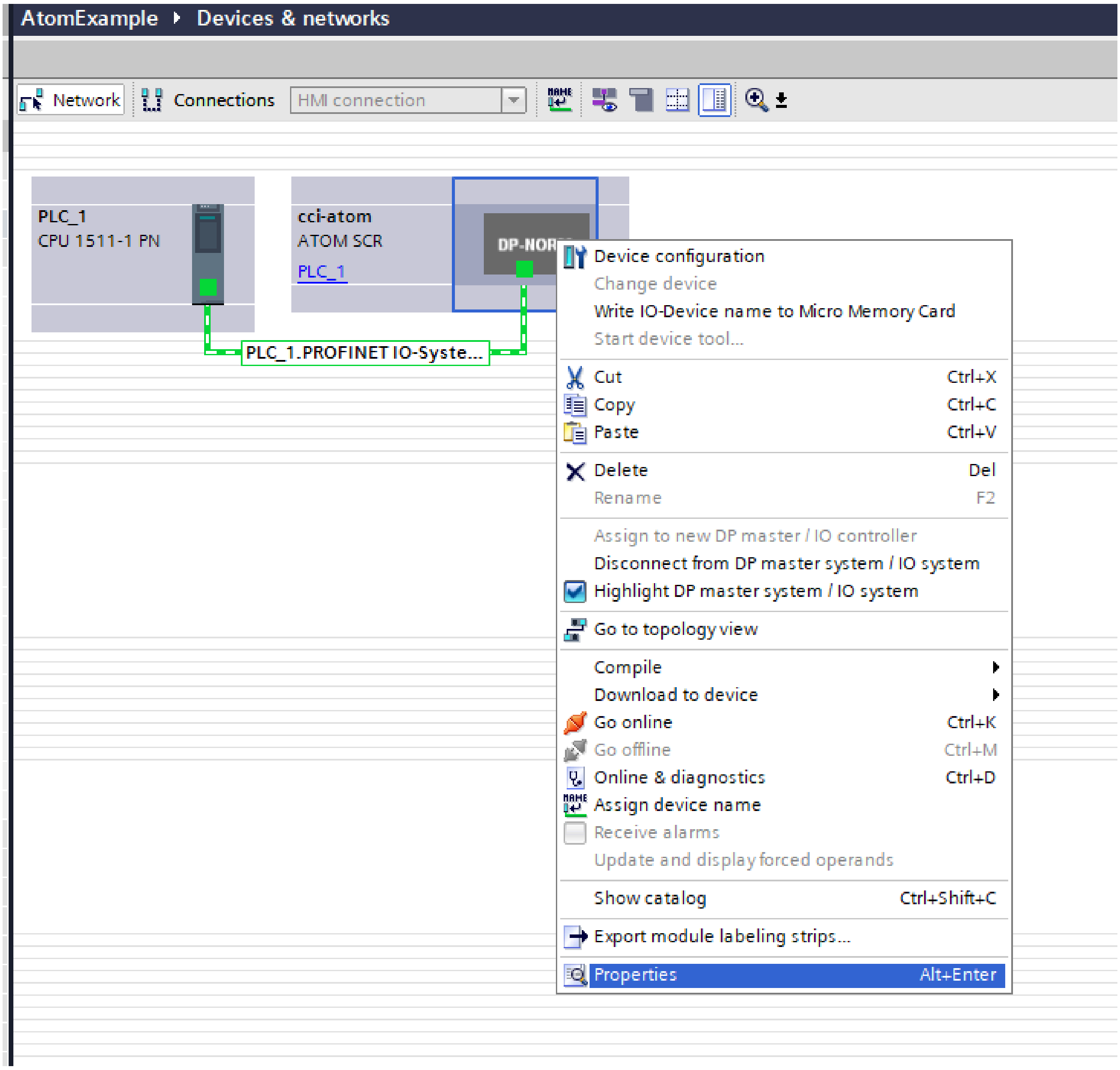
Return to the **Network view**, right click the PLC block (make sure to right click directly on the graphic) and select **Properties**:



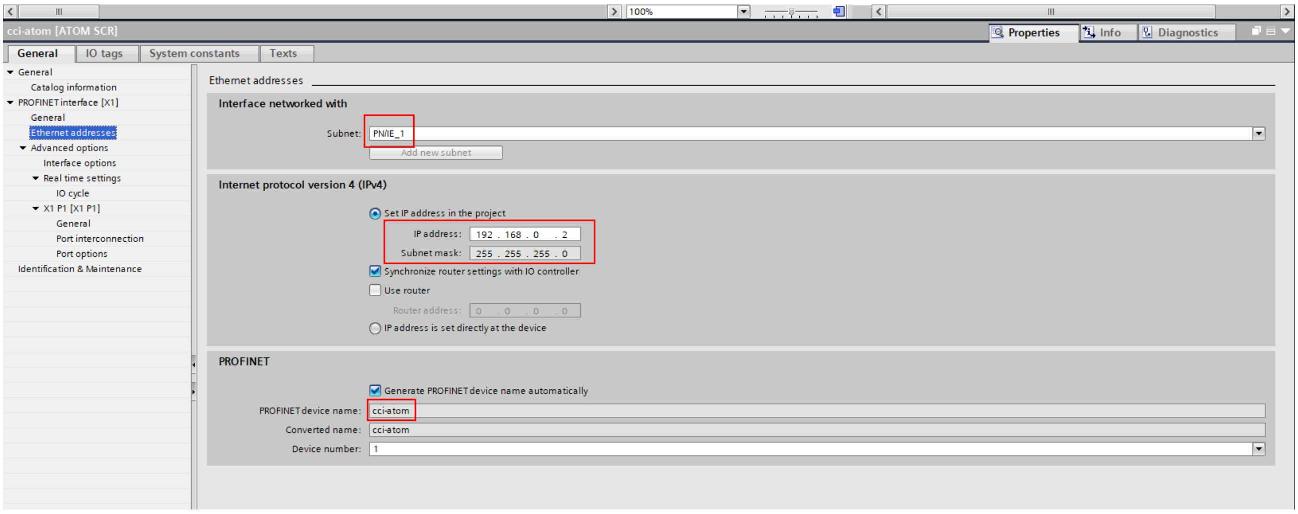
Navigate to **PROFINET interface [X1]** and select **Ethernet addresses**. Change these to match our network provisioning above:



Navigate back to the **Network view**, right click the Atom block (make sure to right click directly on the graphic) and select **Properties**:



Navigate to **PROFINET interface [X1]** and select **Ethernet addresses**. Check these to match our network provisioning above:

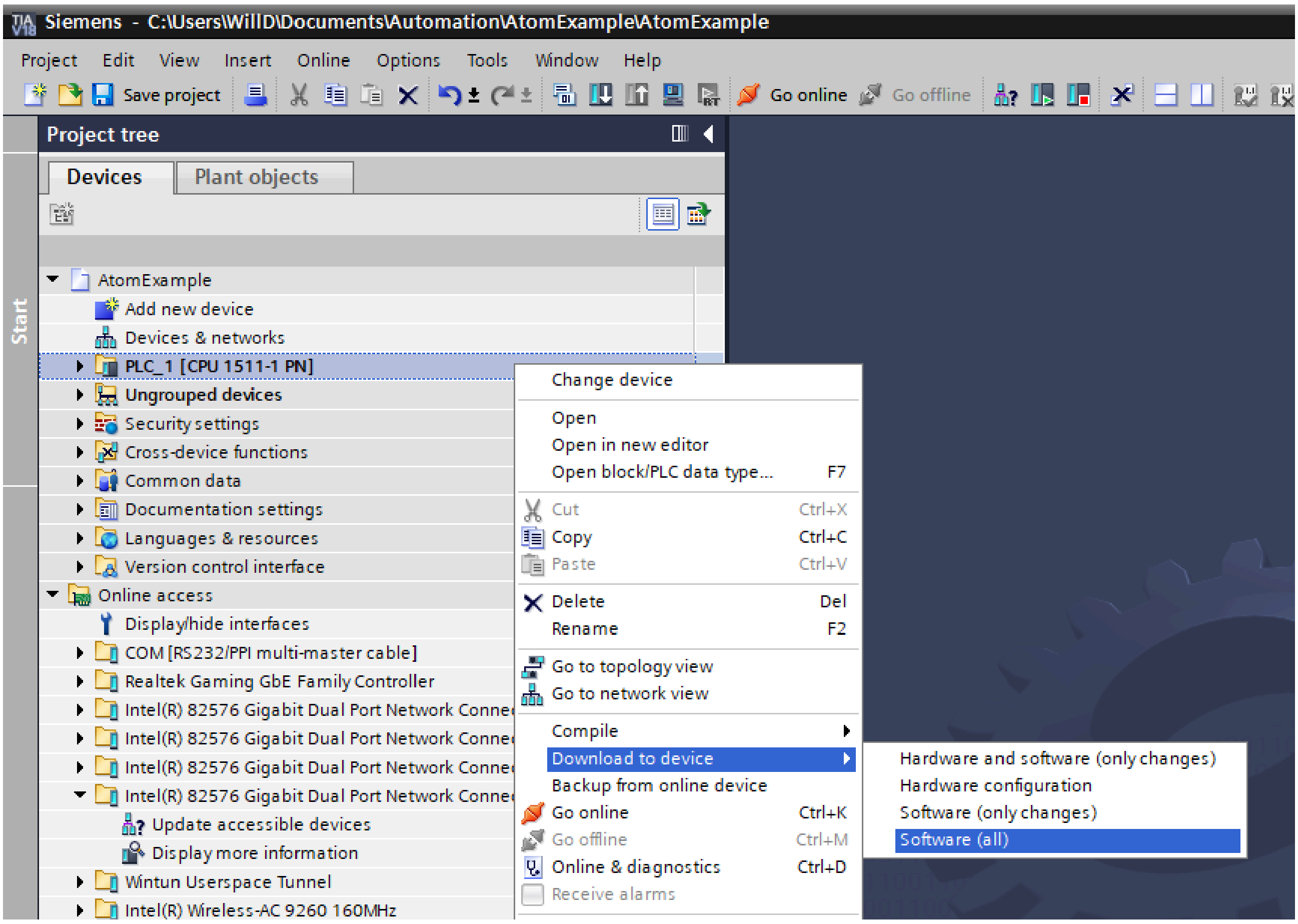


A basic test

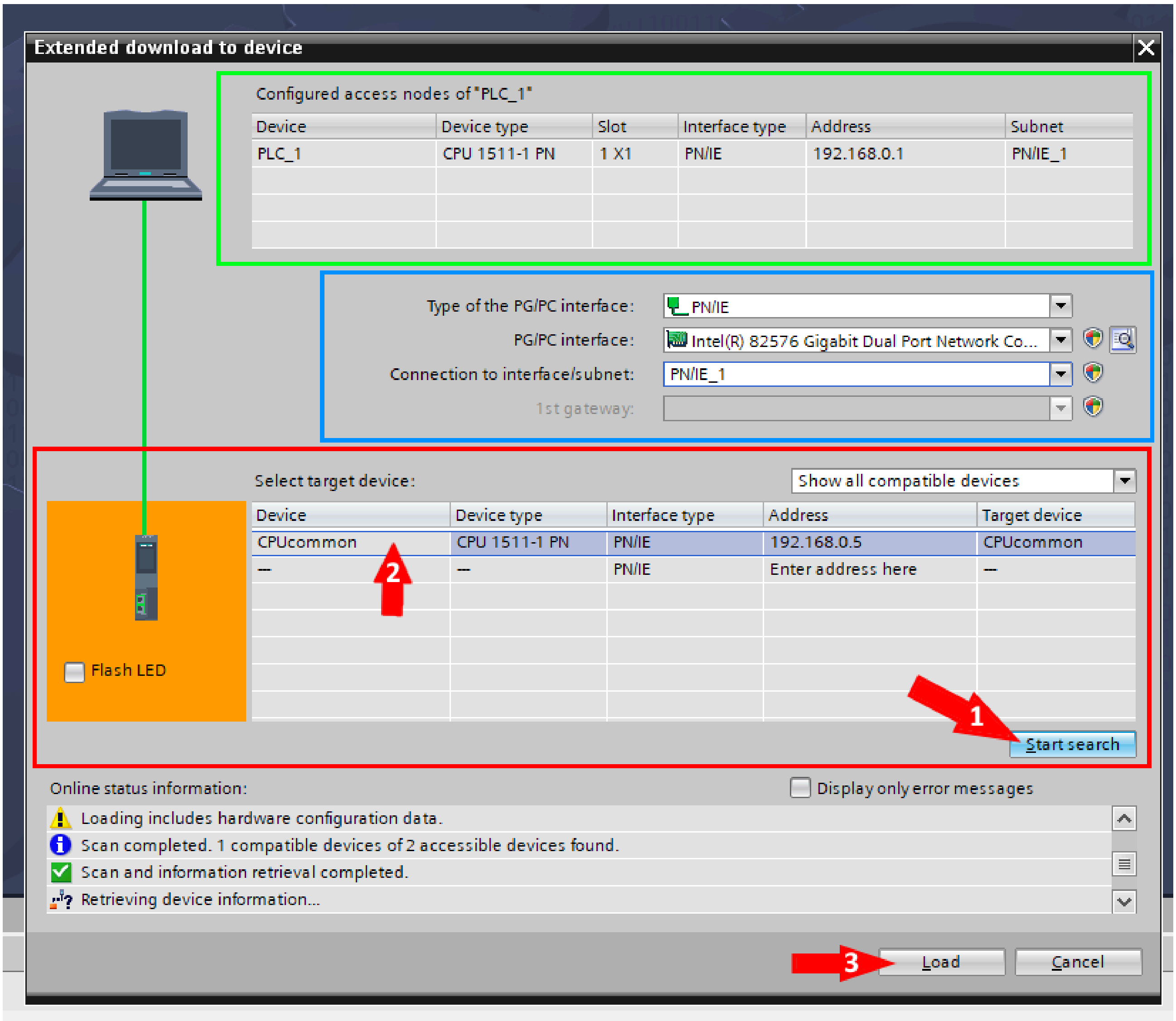
Now that we've configured our network, we should be able to run a simple test to make sure everything is working.

Download to your PLC

Right click on **PLC_1** in the device tree and select **Download to device** > **Software (all)**:



The **Extended download to device dialog** will appear.


Extended download to device
Configured access nodes of "PLC_1"
Device
Device type
Slot
Interface type
Address
Subnet
PLC_1
CPU 1511-1 PN
1 X1
PN/IE
192.168.0.1
PN/IE_1
Type of the PG/PC interface:
PN/IE
PG/PC interface:
Intel(R) 82576 Gigabit Dual Port Network Co...
Connection to interface/subnet:
PN/IE_1
1st gateway:
Select target device:
Show all compatible devices
Device
Device type
Interface type
Address
Target device
CPUcommon
CPU 1511-1 PN
PN/IE
192.168.0.5
CPUcommon
PN/IE
Enter address here
Flash LED
Start search
1
2
Online status information:
Display only error messages
Loading includes hardware configuration data.
Scan completed. 1 compatible devices of 2 accessible devices found.
Scan and information retrieval completed.
Retrieving device information...
3
Load
Cancel

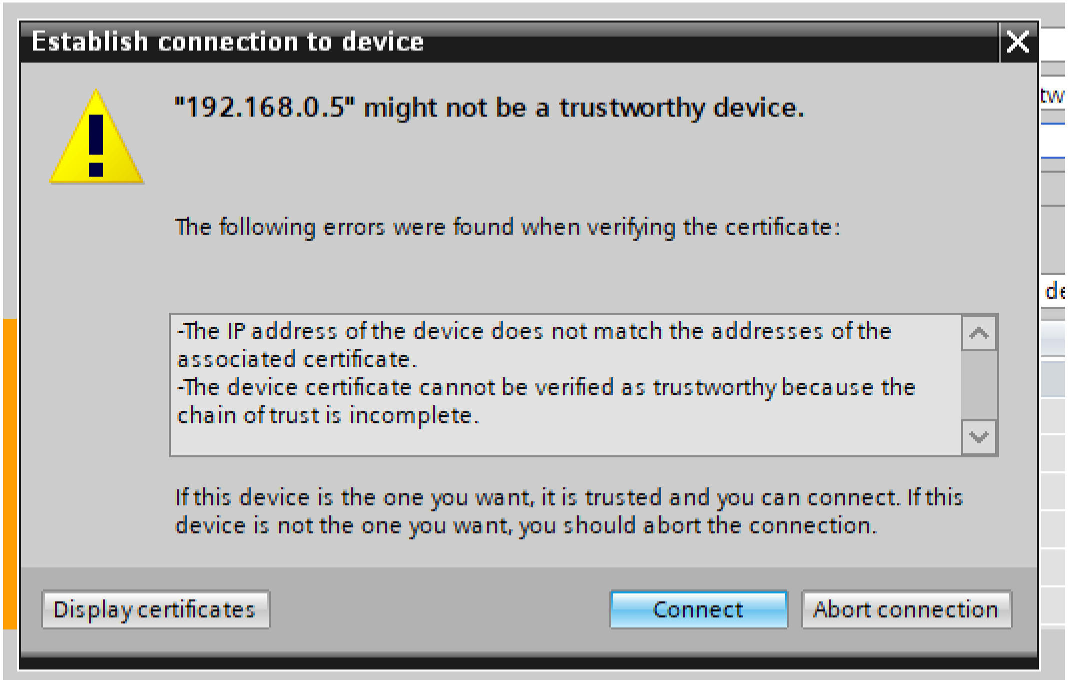
INFO

- The green box shows your provisioned network settings, in other words, what you *want* them to be.
- The red box shows the actual (current) network settings — the actual IP addresses of the PLC, Atom, etc.
- The blue box lets you configure the Ethernet interface on your PC and Profinet subnet to search for your PLC on.

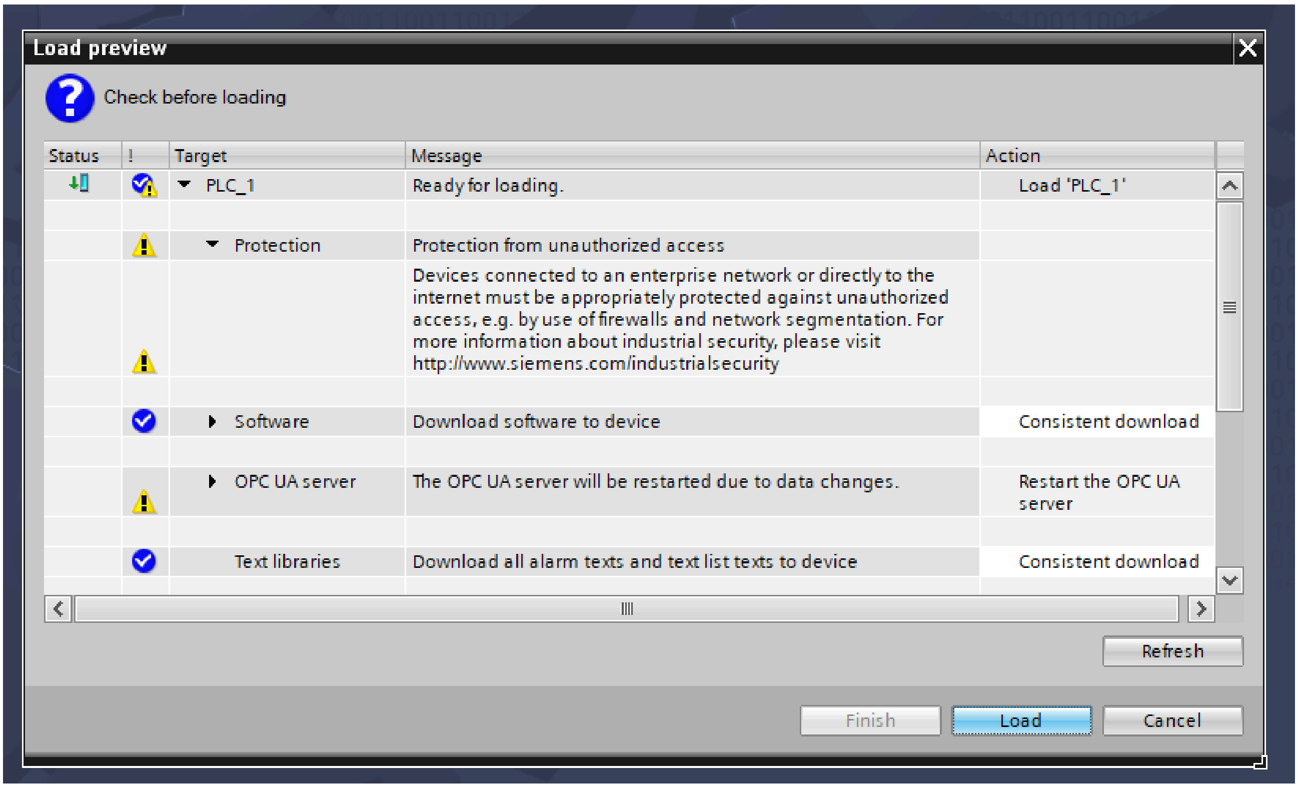
Your PLC will automatically update the station names, IP addresses, and subnet masks that you set up in network provisioning when you start your PLC program, so you don't need to change them manually. The reason this dialog shows up is so that you can tell TIA which PLC to load the program on to.

Make sure **Show compatible devices** is selected, click **Start search**, wait for your PLC to appear, select it, then click **Load**. Notice that our PLC's actual IP address (`192.168.0.5`) isn't the same as the IP address we provisioned for it (`192.168.0.1`). This is fine, because we've linked our provisioned PLC to an actual PLC on the network, TIA knows whichs PLC to program and subsequently will update its network settings along with the network settings of all other Profinet devices on your provisioned network.

You may get a warning dialog like "**X.X.X.X** might not be a trustworthy device", click **Connect**:



The **Load preview** dialog will open, when it finishes preparing, click **Load**.

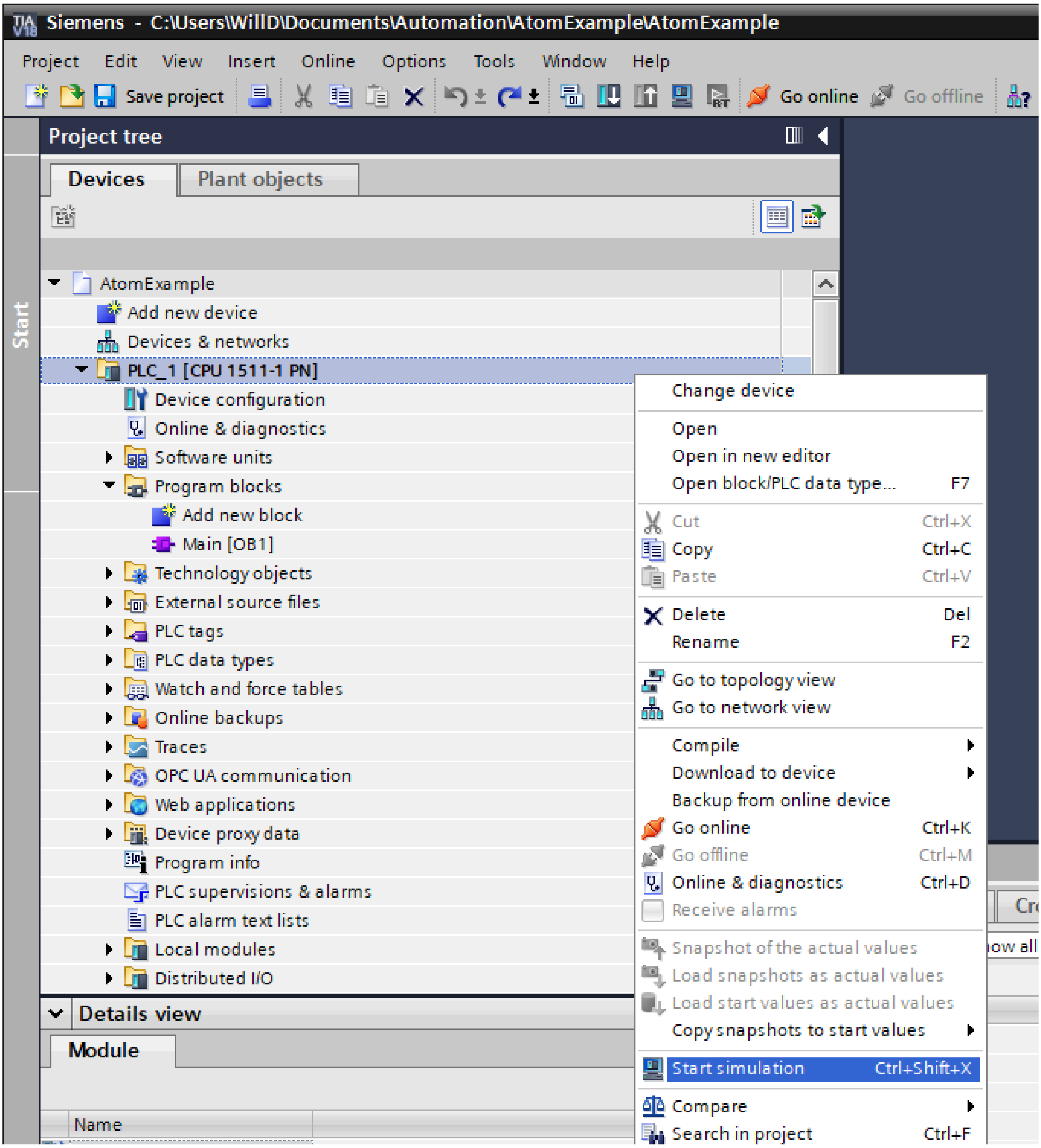


INFO

If the load fails, see Troubleshooting.

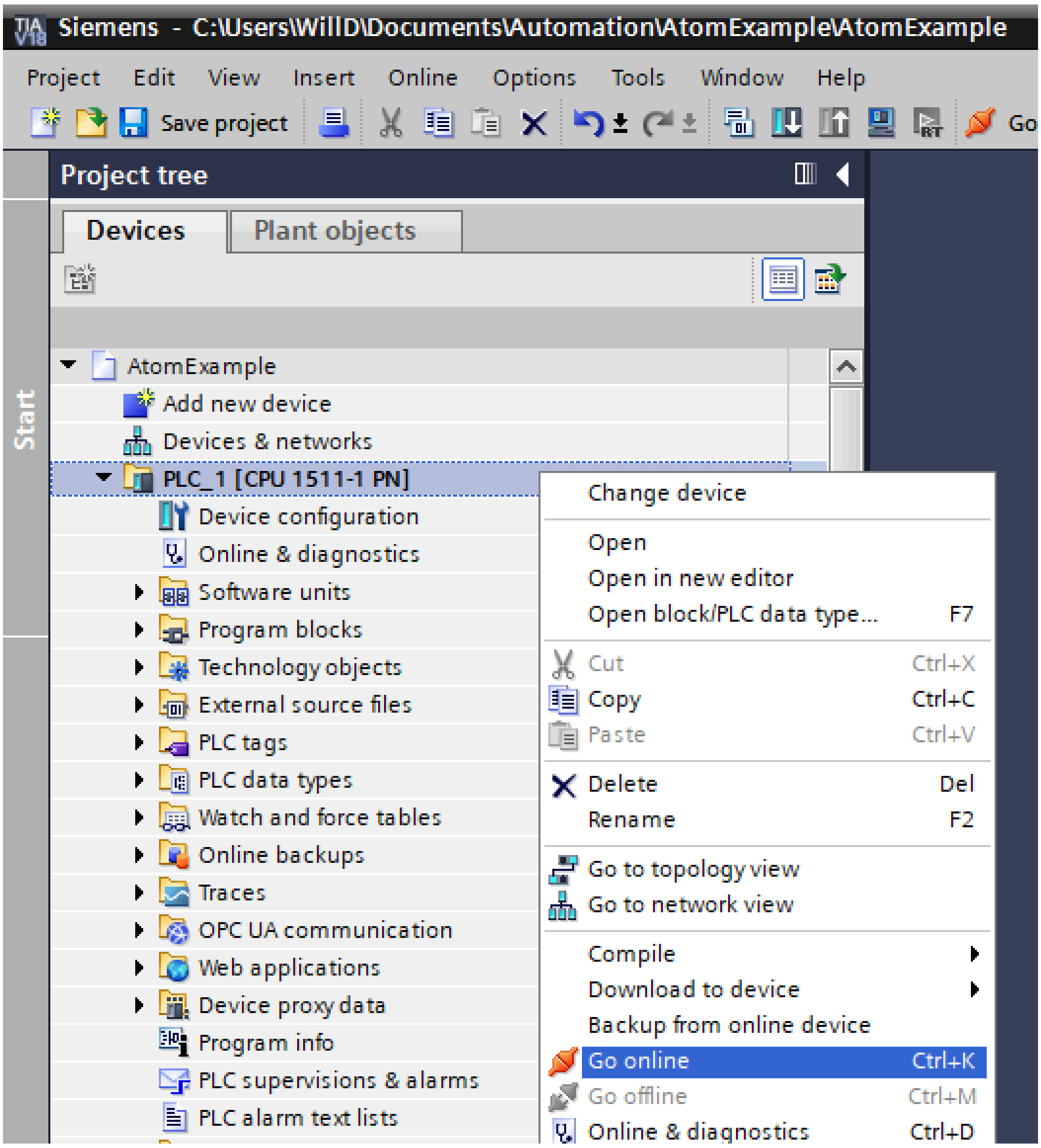
Use a simulator

If you don't have a real PLC handy, you can instead start the PLC simulator by right clicking **PLC_1** in the devices tree and selecting **Start simulation**:

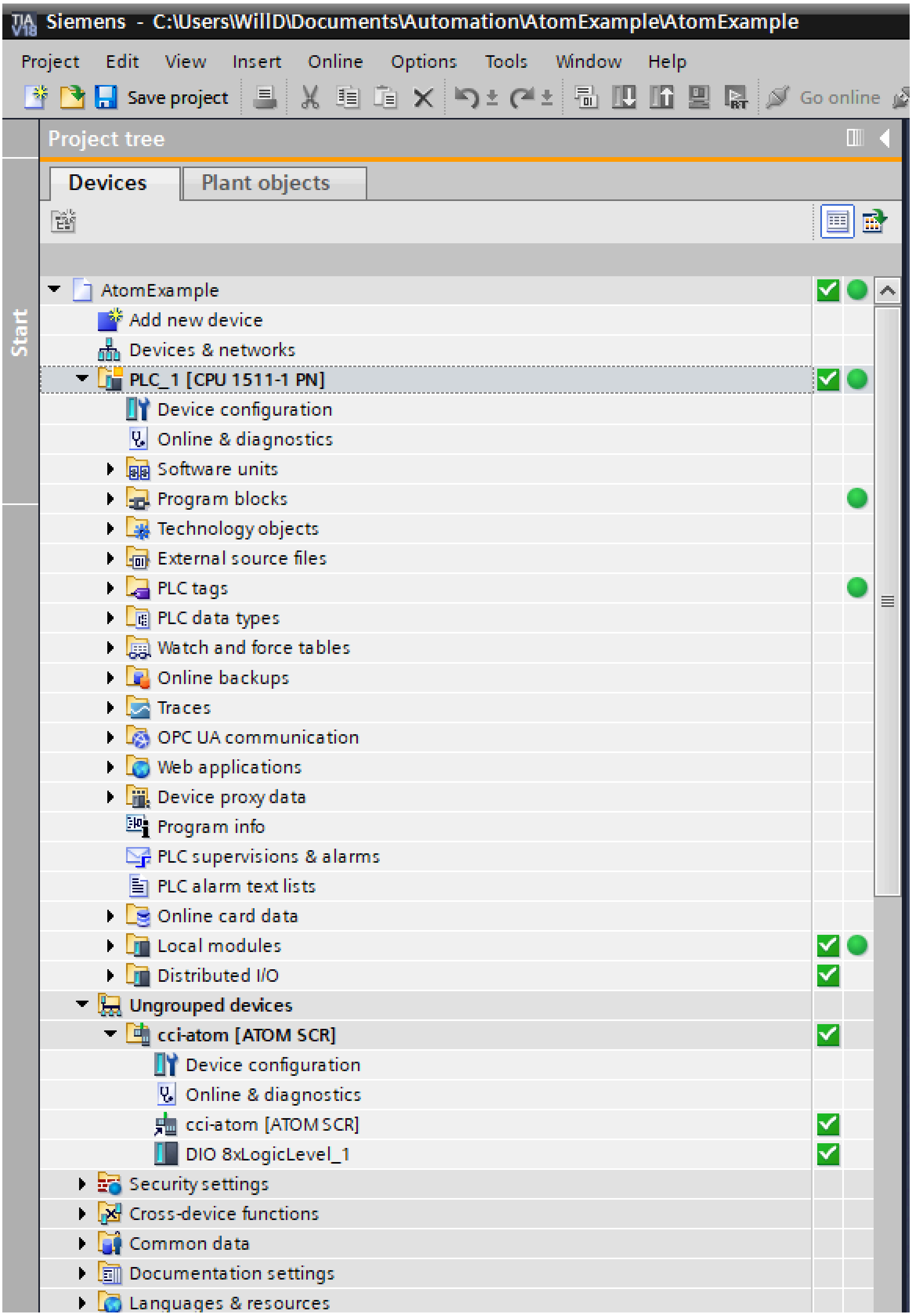


Monitor the heatsink temperature

Next, right click **PLC_1** in the device tree and select **Go online**:

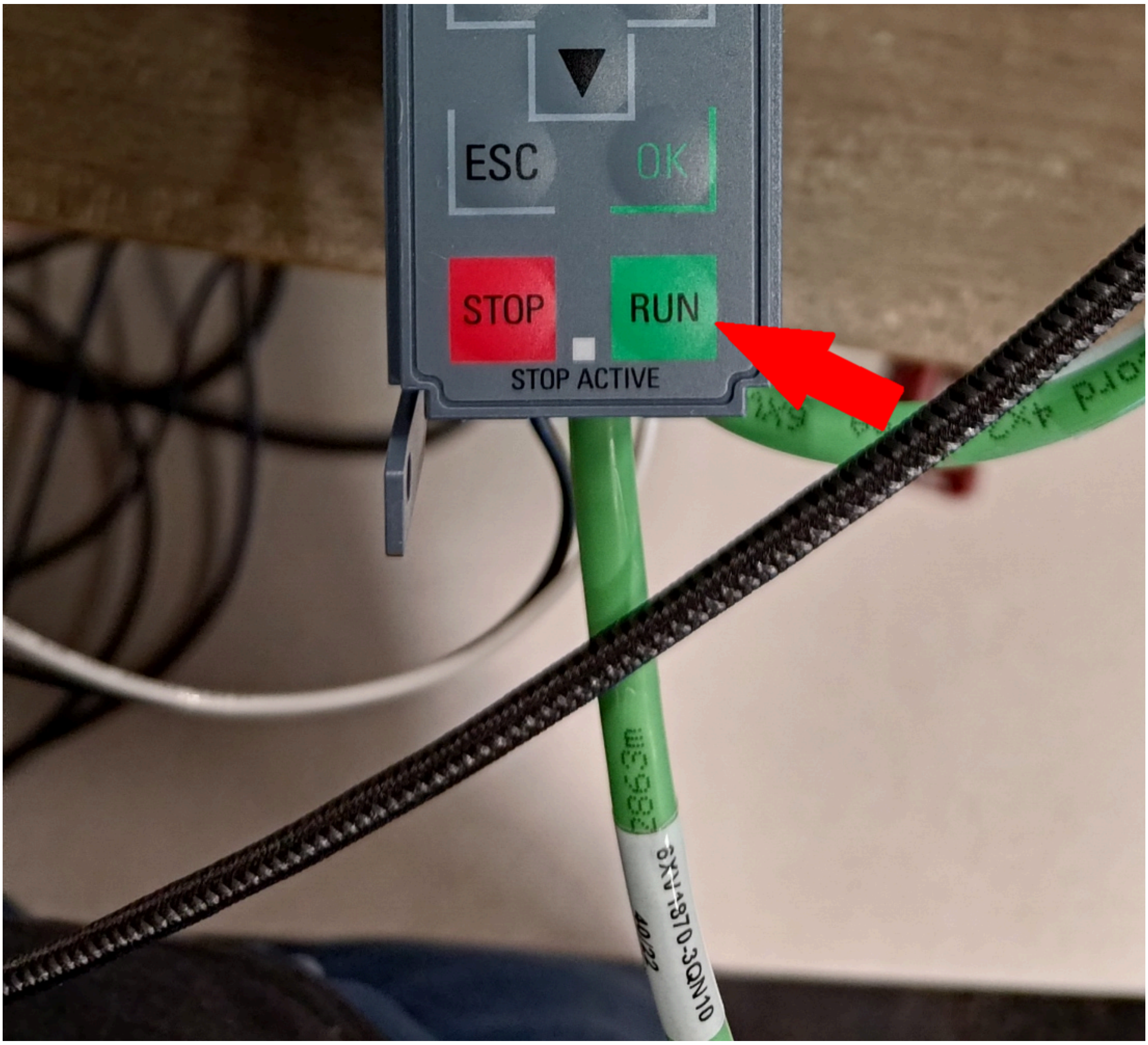


If everything goes well, your device tree should display green checkmarks over each device and their associated modules:

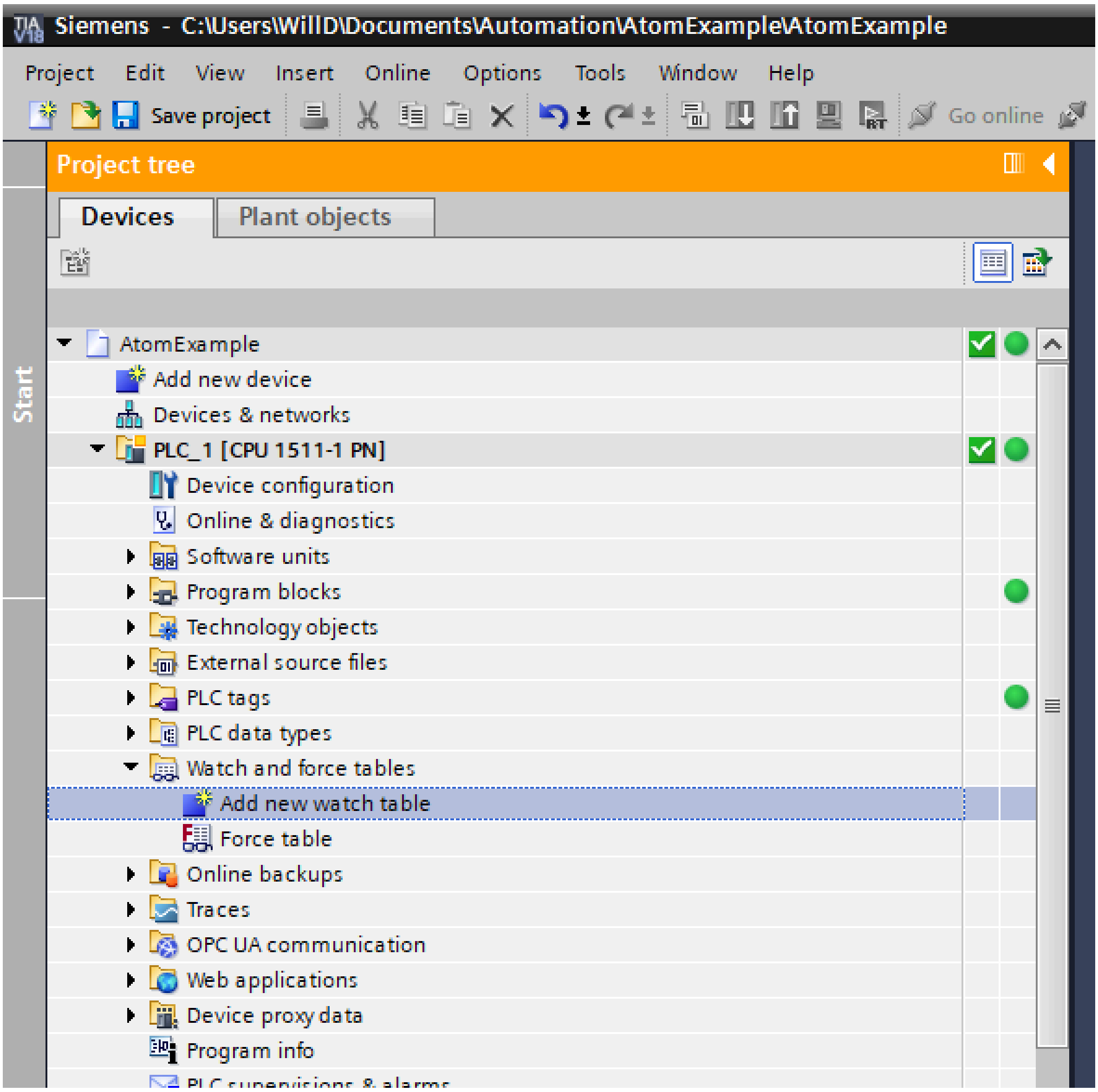

Siemens - C:\Users\WillD\Documents\Automation\AtomExample\AtomExample
Project Edit View Insert Online Options Tools Window Help
Save project
Go online
Start
Project tree
Devices
Plant objects
AtomExample
Add new device
Devices & networks
PLC_1 [CPU 1511-1 PN]
Device configuration
Online & diagnostics
Software units
Program blocks
Technology objects
External source files
PLC tags
PLC data types
Watch and force tables
Online backups
Traces
OPC UA communication
Web applications
Device proxy data
Program info
PLC supervisions & alarms
PLC alarm text lists
Online card data
Local modules
Distributed I/O
Ungrouped devices
cci-atom [ATOM SCR]
Device configuration
Online & diagnostics
cci-atom [ATOM SCR]
DIO 8xLogicLevel_1
Security settings
Cross-device functions
Common data
Documentation settings
Languages & resources

Open the case of your PLC and put it into **RUN** mode (or, if you're using the simulator, click **RUN** in the PLC simulator popup):

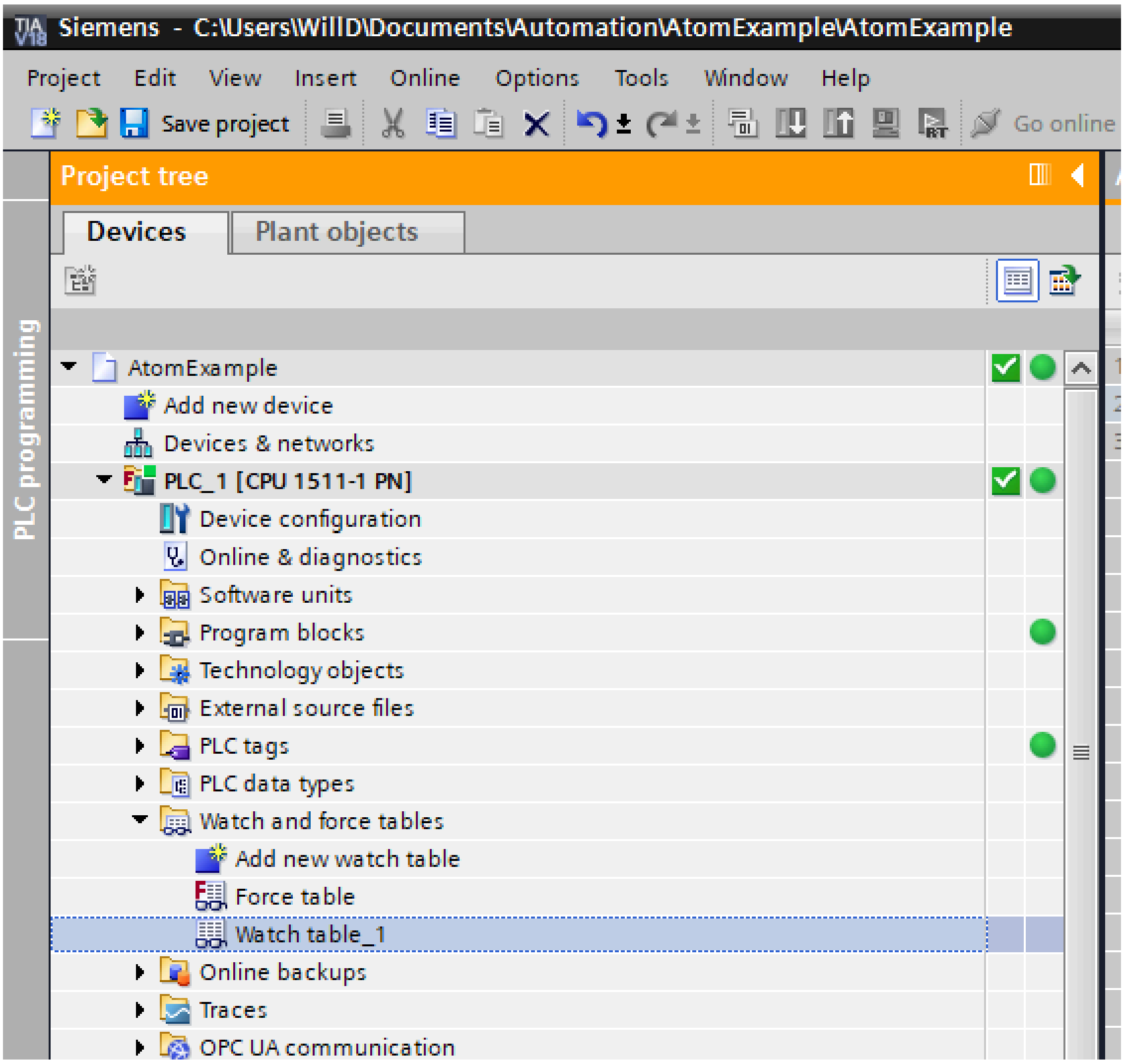

CPU 1511-1 PN
6ES7511-1AK02-0AB0
X67
RUN
F
Overview
OK
X1P1
X1P2
LINK
TX/RX
MAC ADD1
EC-1C-5D-
89-48-C3
MW
MEAN WELL
L
N
50/60Hz
0.55A
DC OK
24V/1.0A
+V ADJ
+V
-V



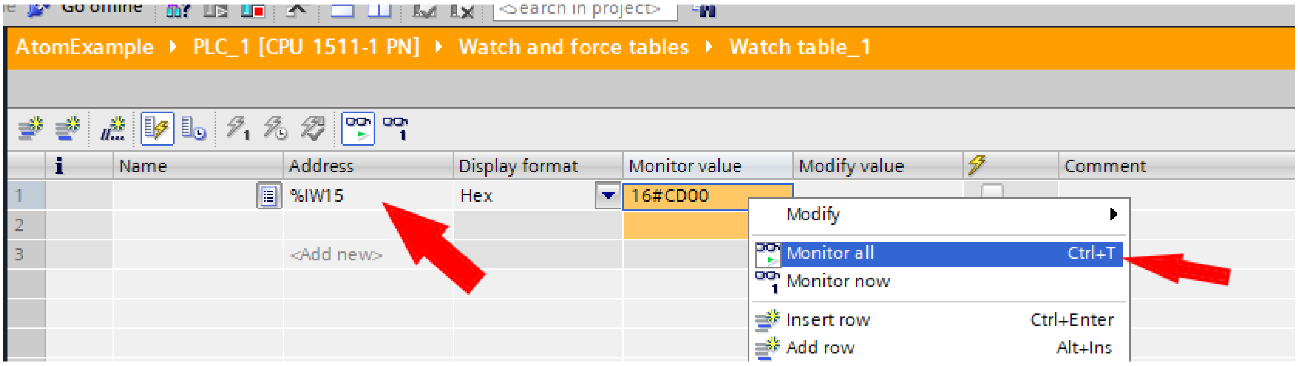
Next, expand **Watch and force tables** under **PLC_1** in the device tree and double click **Add new watch table**:



Double click the newly created watch table **Watch table_1** to open it:



On the first row, enter `%IW15`, then right click the row and select **Monitor all**. `%IW15` is short for input word #15, which corresponds to the low-order word of the heatsink temperature parameter. After clicking **Monitor all**, you should see the value update to a non-zero value. If it does, this means your PLC is successfully talking to Atom over Profinet!

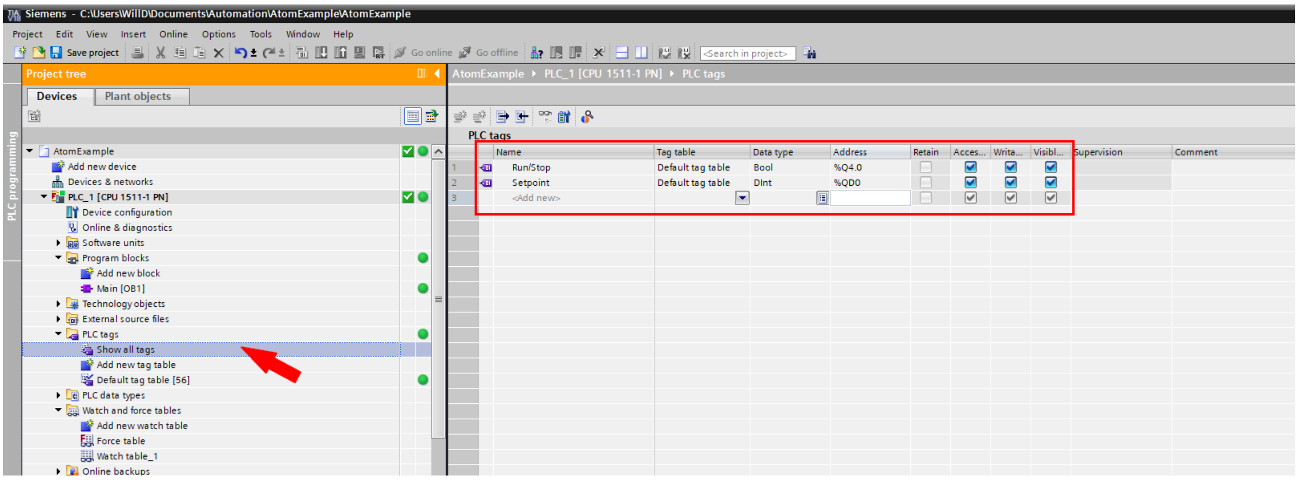


Building a simple PLC program to control Atom

Writing some ladder logic

In the devices tree, expand **PLC_1** > **PLC tags** and double click **Show all tags**. Create two tags:

- Tag #1
  - Name: `Run/Stop`
  - Data type: `Bool`
  - Address: `%Q4.0`
- Tag #2
  - Name: `Setpoint`
  - Data type: `Dint`
  - Address: `%QD0`



Next, in the device tree, expand **PLC_1** > **Program blocks** and double click **Main [OB1]**.

First, create two constants:

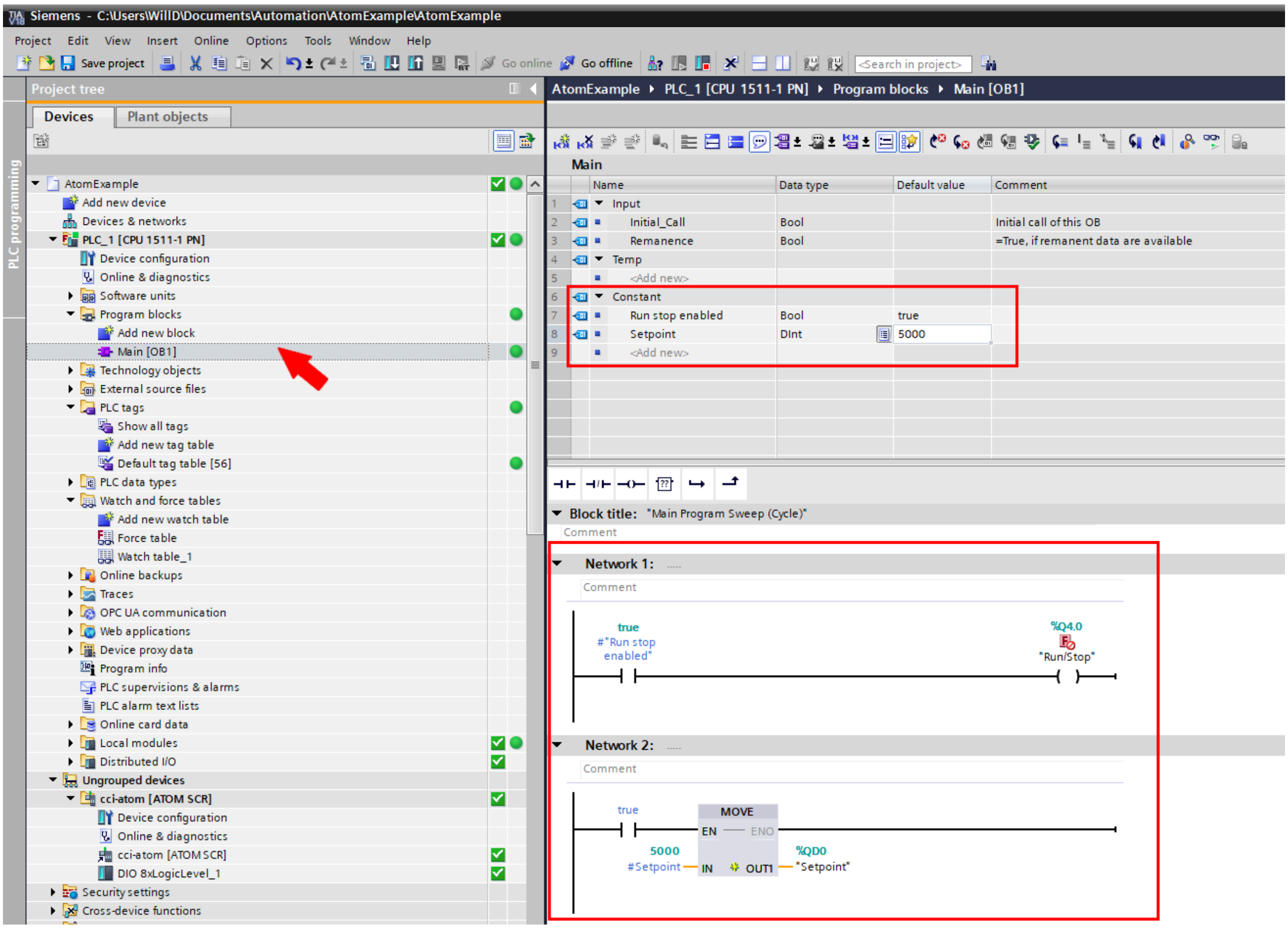
- Name: `Run stop enabled`, Data type: `Bool`, Default value: `true`
- Name: `Setpoint`, Data type: `Dint`, Default value: `5000`

These are the values we will ultimately set to Atom.

Next, create two networks.

- Network #1
  - Insert a **contact** on the left, and drag the `Run stop enabled` constant into it.
  - Insert an **assignment** on the right, and enter `%Q4.0`
- Network #2
  - Insert a **MOVE** block
    - Insert a **contact** on the **EN** input, and set its value to `true`
    - Drag the `Setpoint` constant into the **IN** input
    - Enter `%QD0` into the **OUT1** output

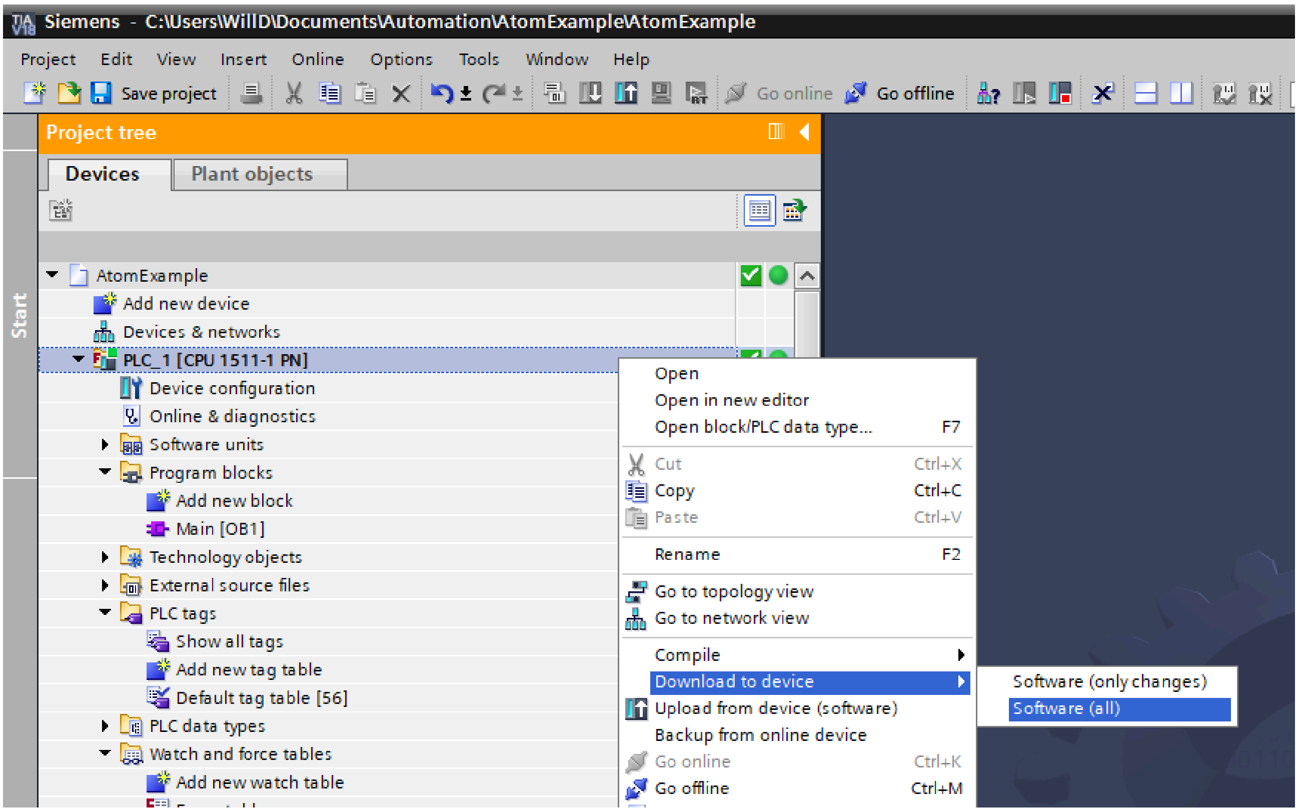
After you're done, your PLC program should look like this (you can also download the example project with the completed program):



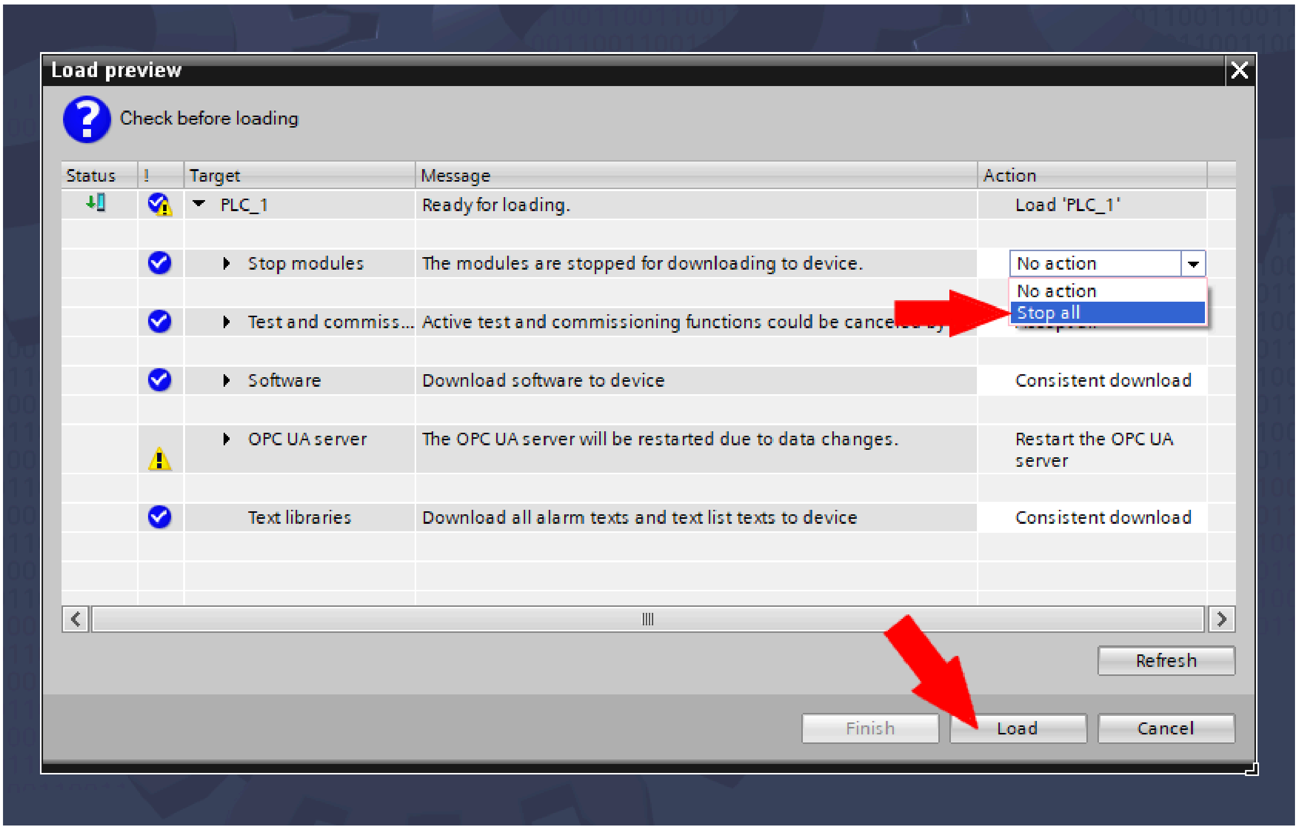
Running on your PLC

> **INFO**
>
> This will still work if you're running a simulator, make sure you started the simulation as shown in Use a simulator.

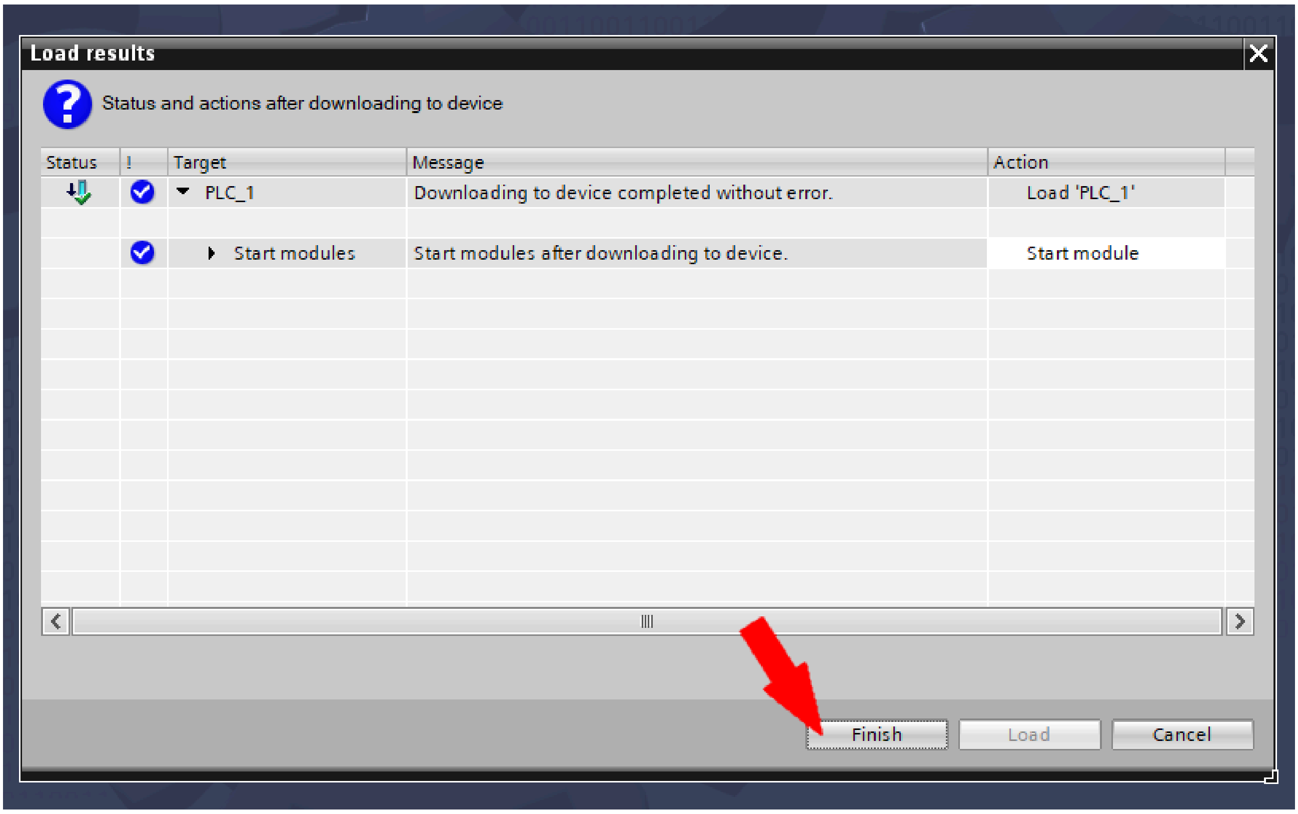
Right click **PLC_1** in the devices tree, select **Download to device** > **Software (all)**:



A **Load preview** dialog will show up, if your PLC was previously running, you may have to set the **Stop modules** action to **Stop all**. Then, click **Load**:

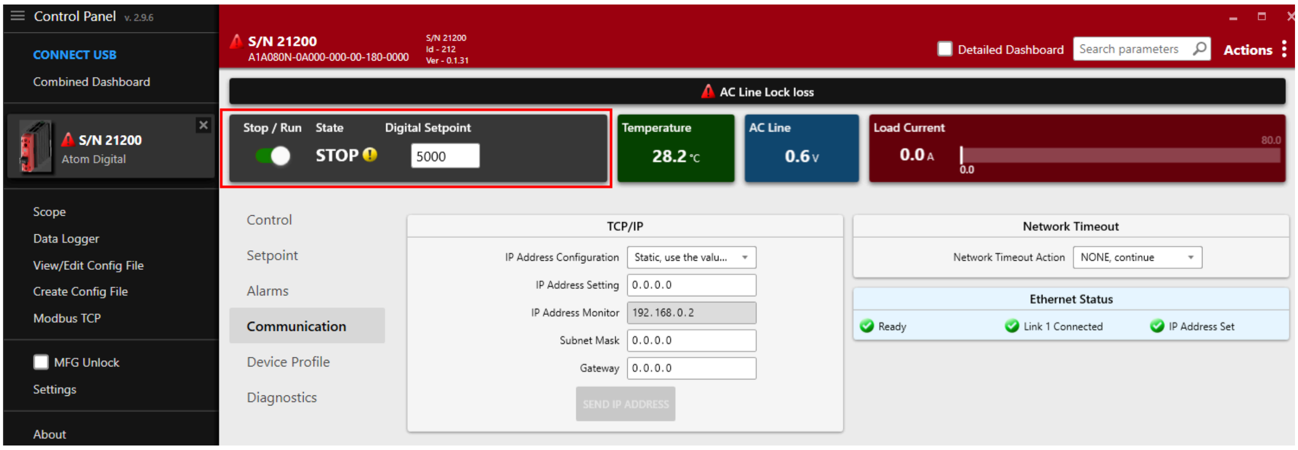


If the load fails, check out Troubleshooting. Otherwise, click **Finish**:



Make sure your PLC is set to RUN. If everything worked, the PLC will put Atom into RUN with a setpoint of `5000`. If you connect a USB cable to your Atom and look in Control Panel, you should see the setpoint and run/stop parameters update.

> **INFO**
>
> Try switching the **Stop / Run** switch *off* in Control Panel or updating the setpoint. Notice that the PLC immediately sets the run/stop and setpoint parameters back to their original values.

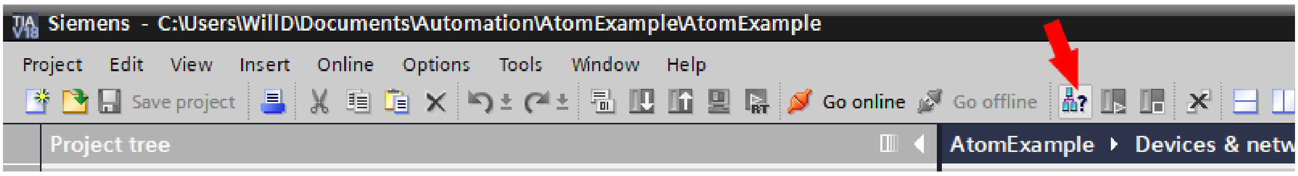


Troubleshooting

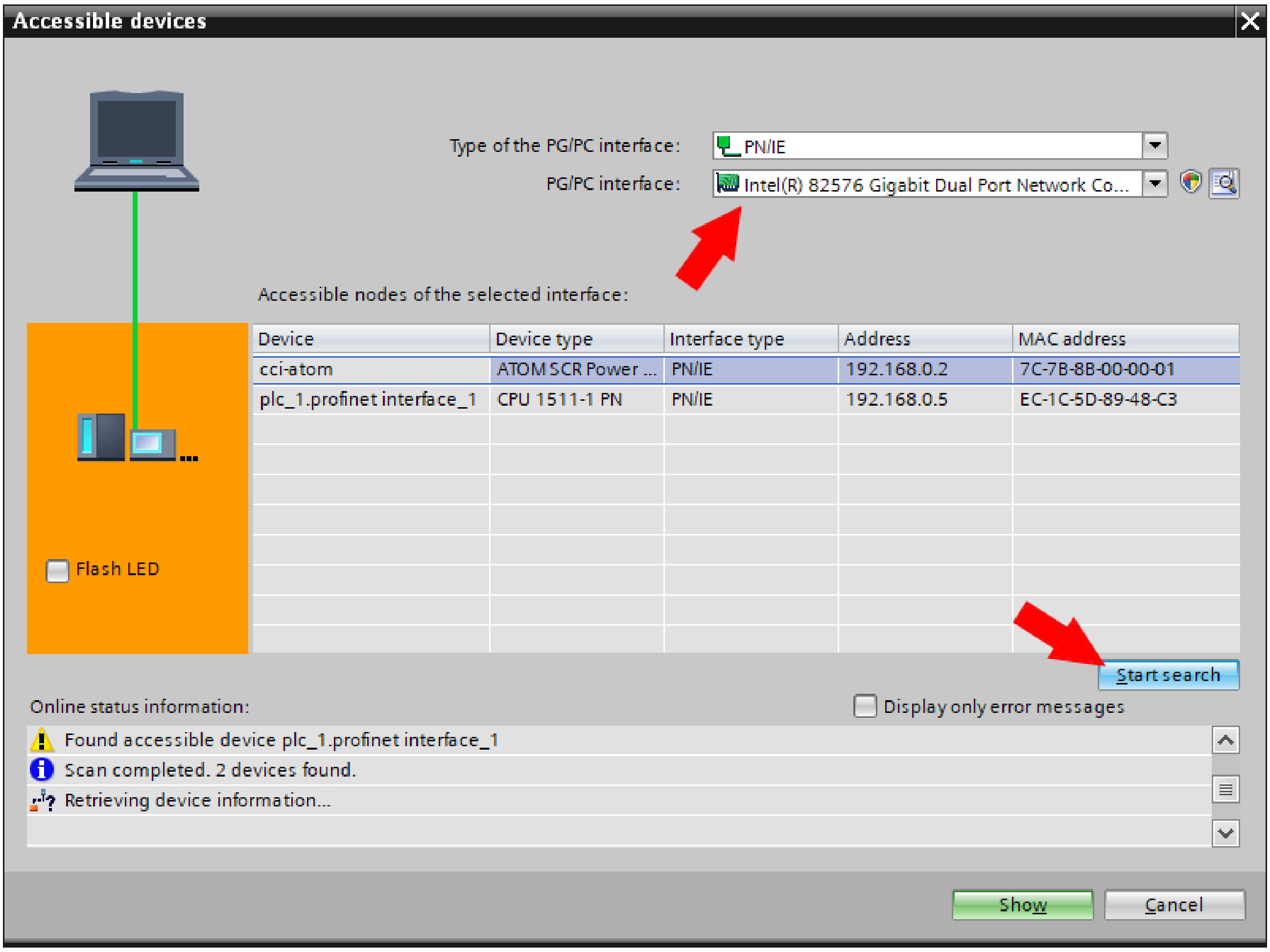
Download to PLC fails

If the download to your PLC fails, you can try resetting your PLC.

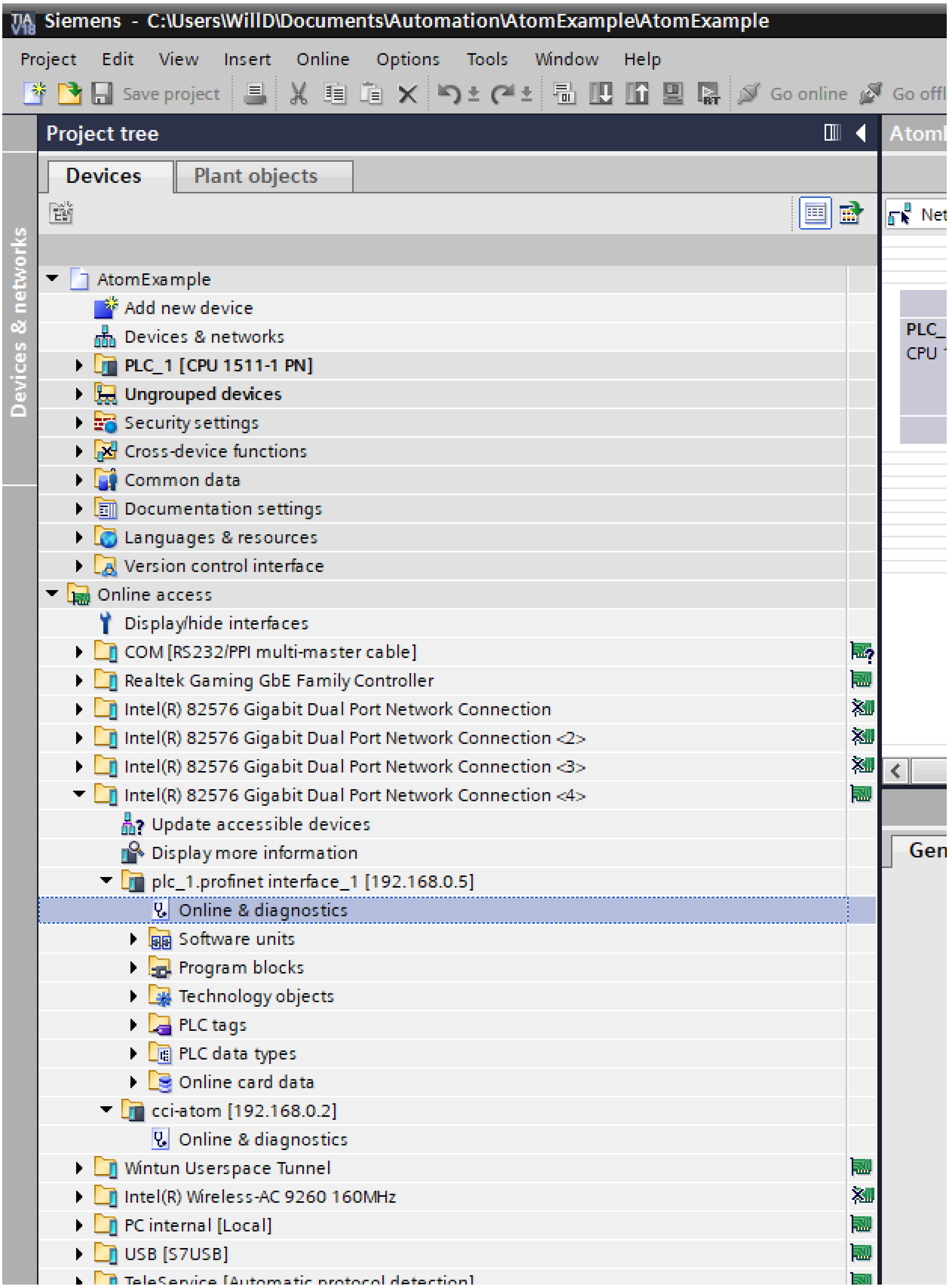
First, click the **Accessible devices** icon in the menu bar:

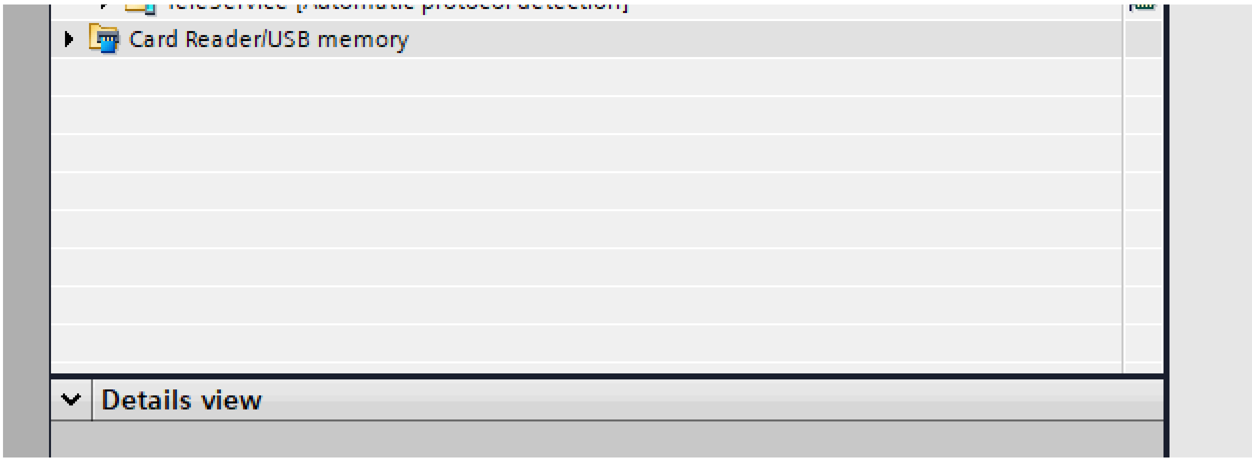


When the **Accessible devices** dialog appears, select the network adapter on your PC that is connected to your PLC and click **Start search**:

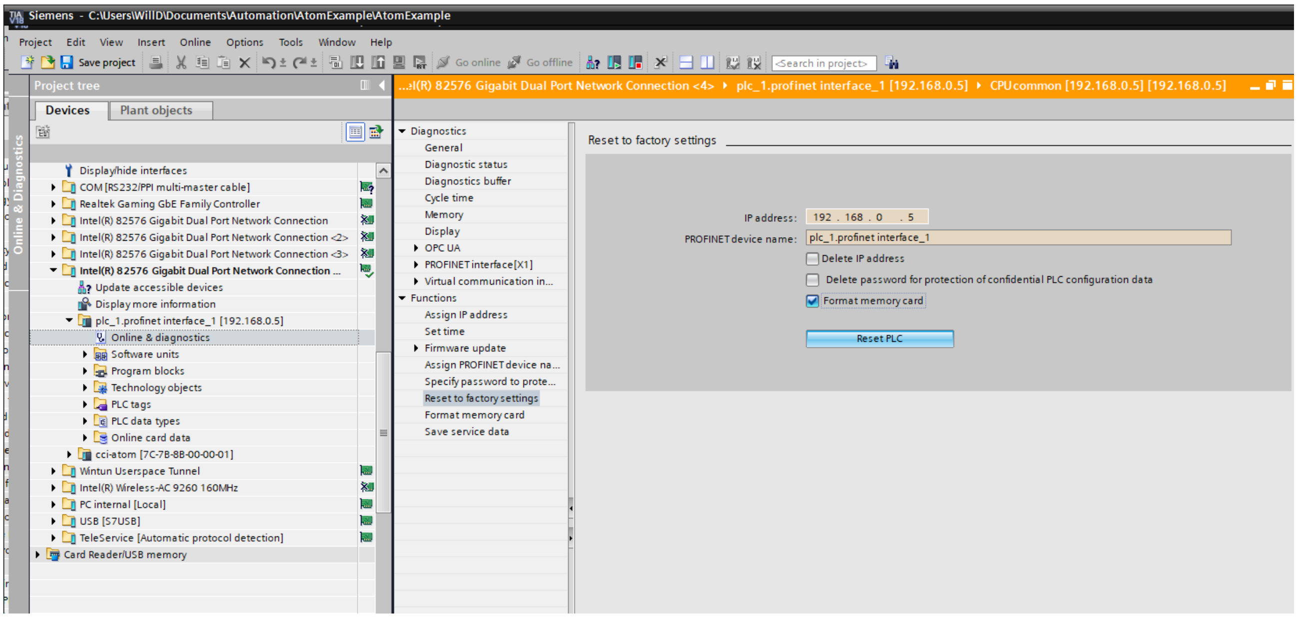


In the devices tree, expand **Online access** > Your PC's network adapter connected to your PLC > **plc_1.profinet interface_1 (192.168.0.05)** (may be different for you) > **Online & diagnostics**:


Siemens - C:\Users\WillD\Documents\Automation\AtomExample\AtomExample
Project Edit View Insert Online Options Tools Window Help
Save project
Go online
Go offl
Project tree
Devices
Plant objects
Devices & networks
AtomExample
Add new device
Devices & networks
PLC_1 [CPU 1511-1 PN]
Ungrouped devices
Security settings
Cross-device functions
Common data
Documentation settings
Languages & resources
Version control interface
Online access
Display/hide interfaces
COM [RS232/PPI multi-master cable]
Realtek Gaming GbE Family Controller
Intel(R) 82576 Gigabit Dual Port Network Connection
Intel(R) 82576 Gigabit Dual Port Network Connection <2>
Intel(R) 82576 Gigabit Dual Port Network Connection <3>
Intel(R) 82576 Gigabit Dual Port Network Connection <4>
Update accessible devices
Display more information
plc_1.profinet interface_1 [192.168.0.5]
Online & diagnostics
Software units
Program blocks
Technology objects
PLC tags
PLC data types
Online card data
cci-atom [192.168.0.2]
Online & diagnostics
Wintun Userspace Tunnel
Intel(R) Wireless-AC 9260 160MHz
PC internal [Local]
USB [S7USB]
Atom
Net
PLC_
CPU
Gen



Expand **Functions** > **Reset to factory settings**, check **Format memory card**, then click **Reset PLC**:



ATOM / Fieldbus / PROFINET / Codesys

In this tutorial, you'll learn how to use Codesys with the SoftPLC emulator to connect to ATOM using Profinet and perform some basic operations and monitor data. You can follow along using the SoftPLC emulator or your own PLC.

We provide examples for both ladder logic and structured text.

If you haven't yet, please review ATOM's Profinet Profile.

If you'd like to skip the tutorial, you can download a completed example project:

- Download ATOM_Codesys_Profinet_LadderLogic_Example.zip
- Download ATOM_Codesys_Profinet_StructuredText_Example.zip

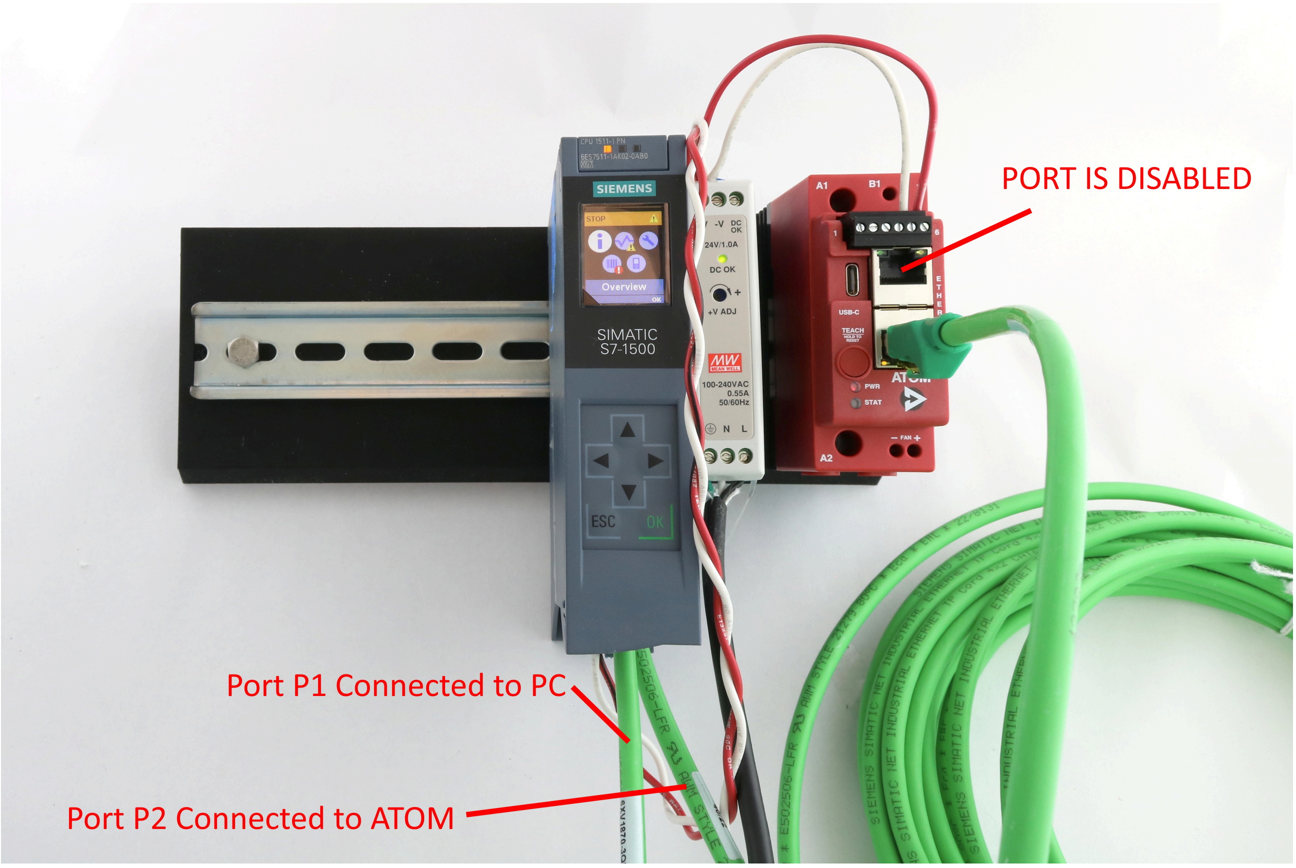
Prerequisites

1. Install Codesys
2. Download ATOM's GSDML file

Hardware setup

> ⚠ **IMPORTANT**
>
> When Atom is configured for Profinet, the Ethernet port closest to the 24V power connector is **disabled**. You must use opposing Ethernet port nearest the reset button as shown below or the PLC won't be able to connect to Atom.

Connect 24V to your PLC and Atom unit with the provided power cable. Connect Atom to your PC with an Ethernet cable.



INFO

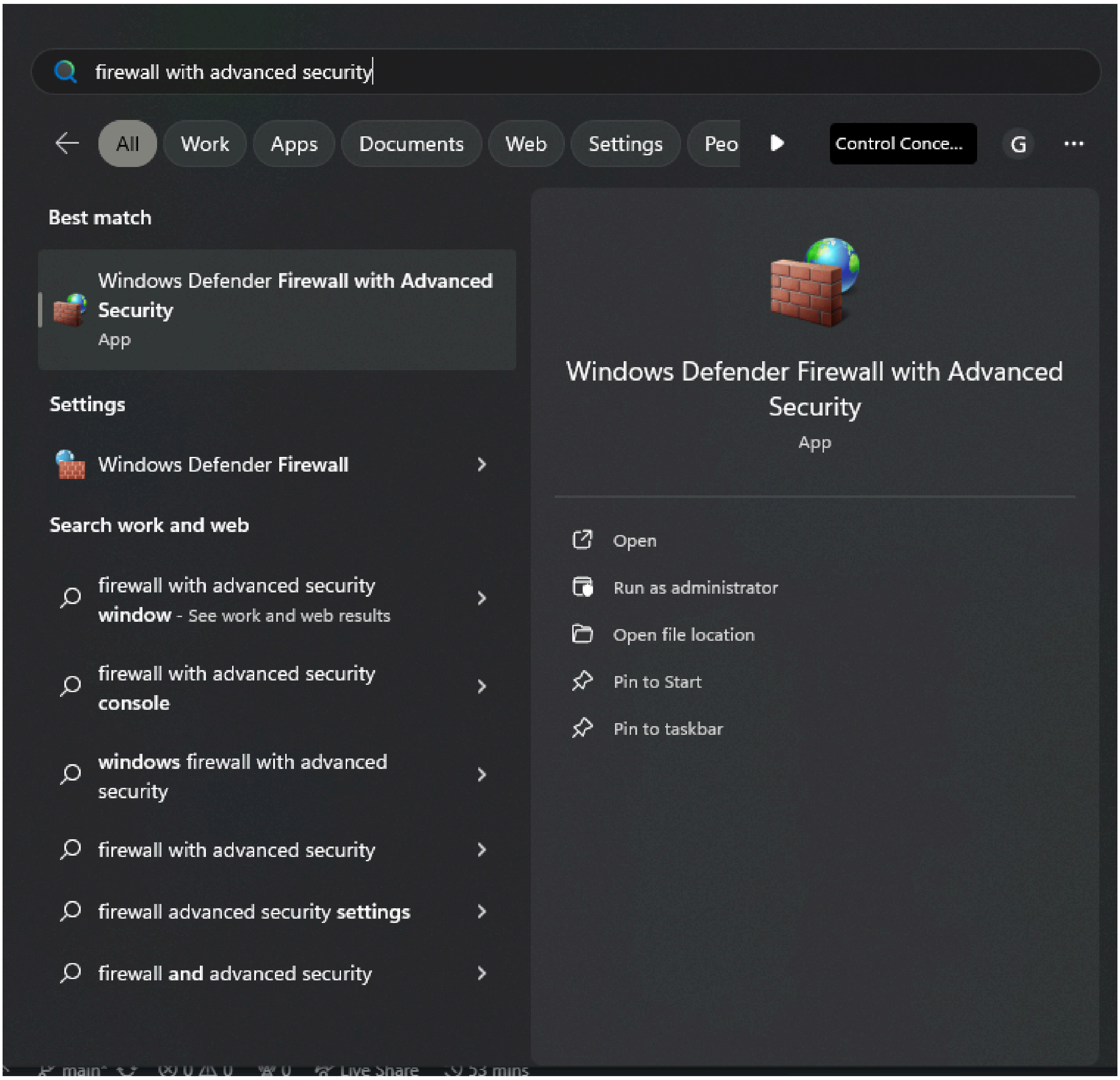
To simplify this diagram, we have not connected a load to Atom. You may connect a load or leave it disconnected, either way is fine for the purposes of this tutorial.

If you do not connect a load, you can still verify your PLC is working by connecting a USB cable to Atom and using Control Panel to watch the parameters change/verify the PLC is receiving the correct monitor data.

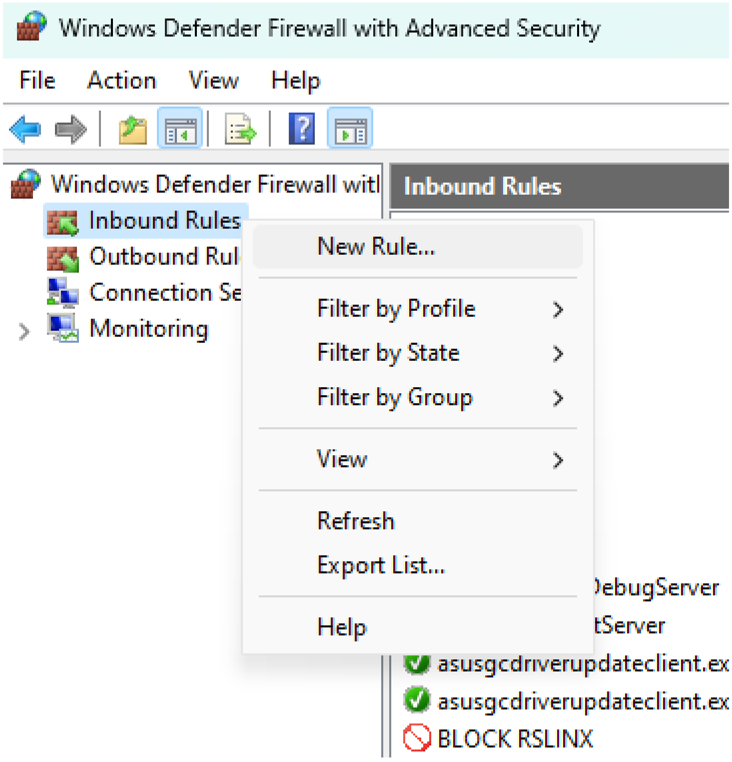
Configure Windows firewall

Codesys requires you to allow incoming Profinet UDP packets through the Windows firewall so that the SoftPLC is able to receive UDP Profinet requests from Atom.

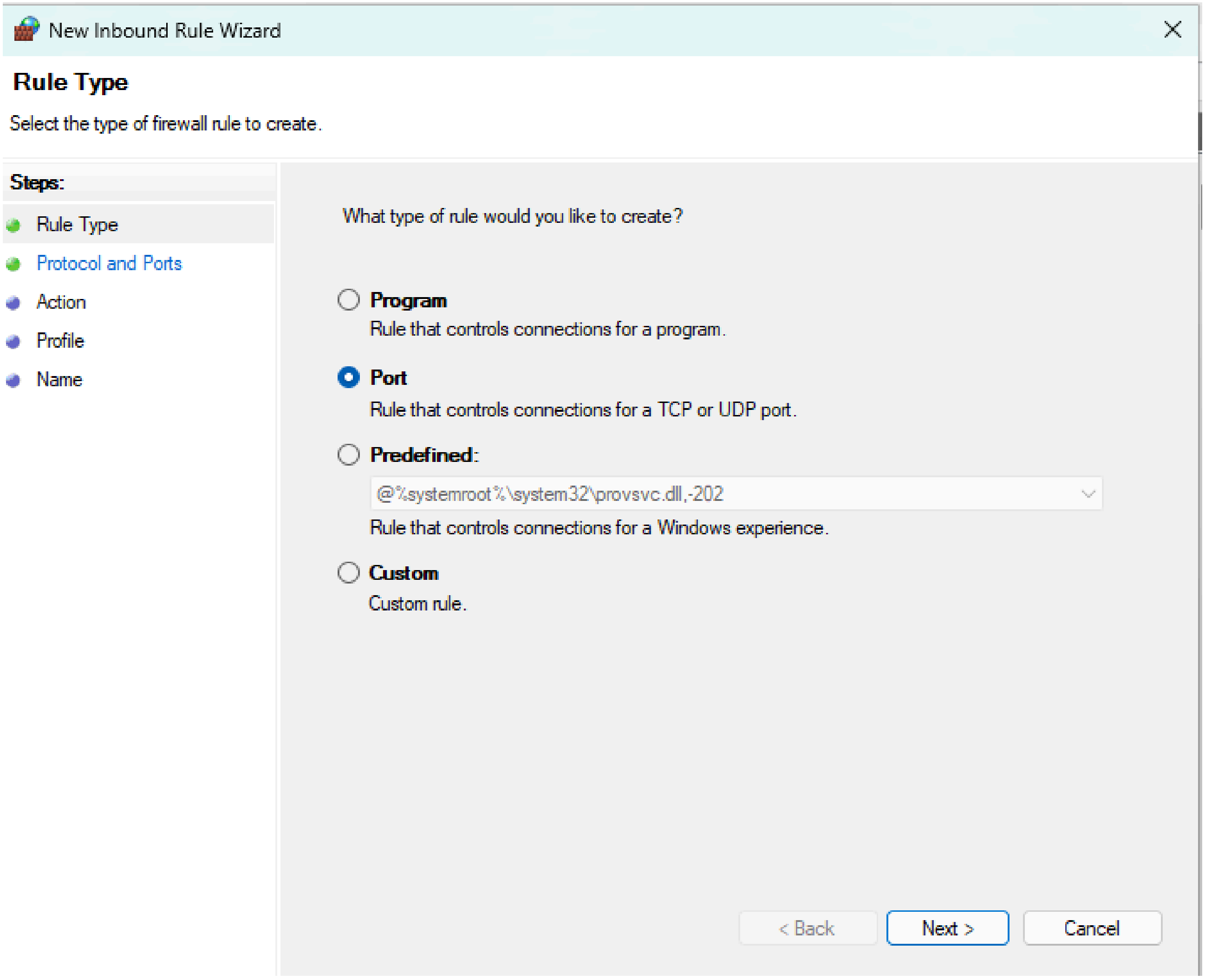
1. First, search for **Windows Defender Firewall with Advanced Security** and open it:



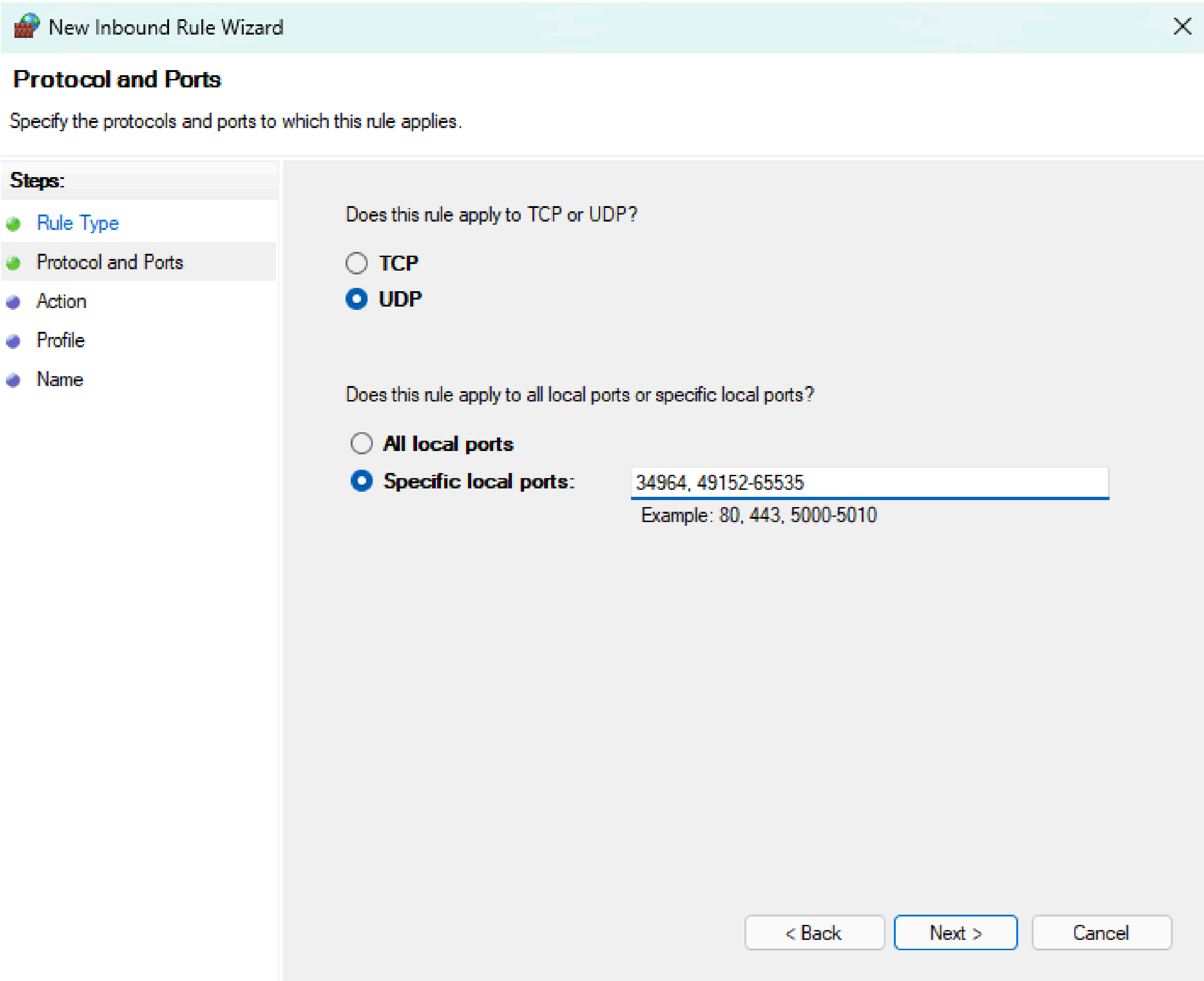
2. Right click on **Inbound Rules** and select **New Rule**:



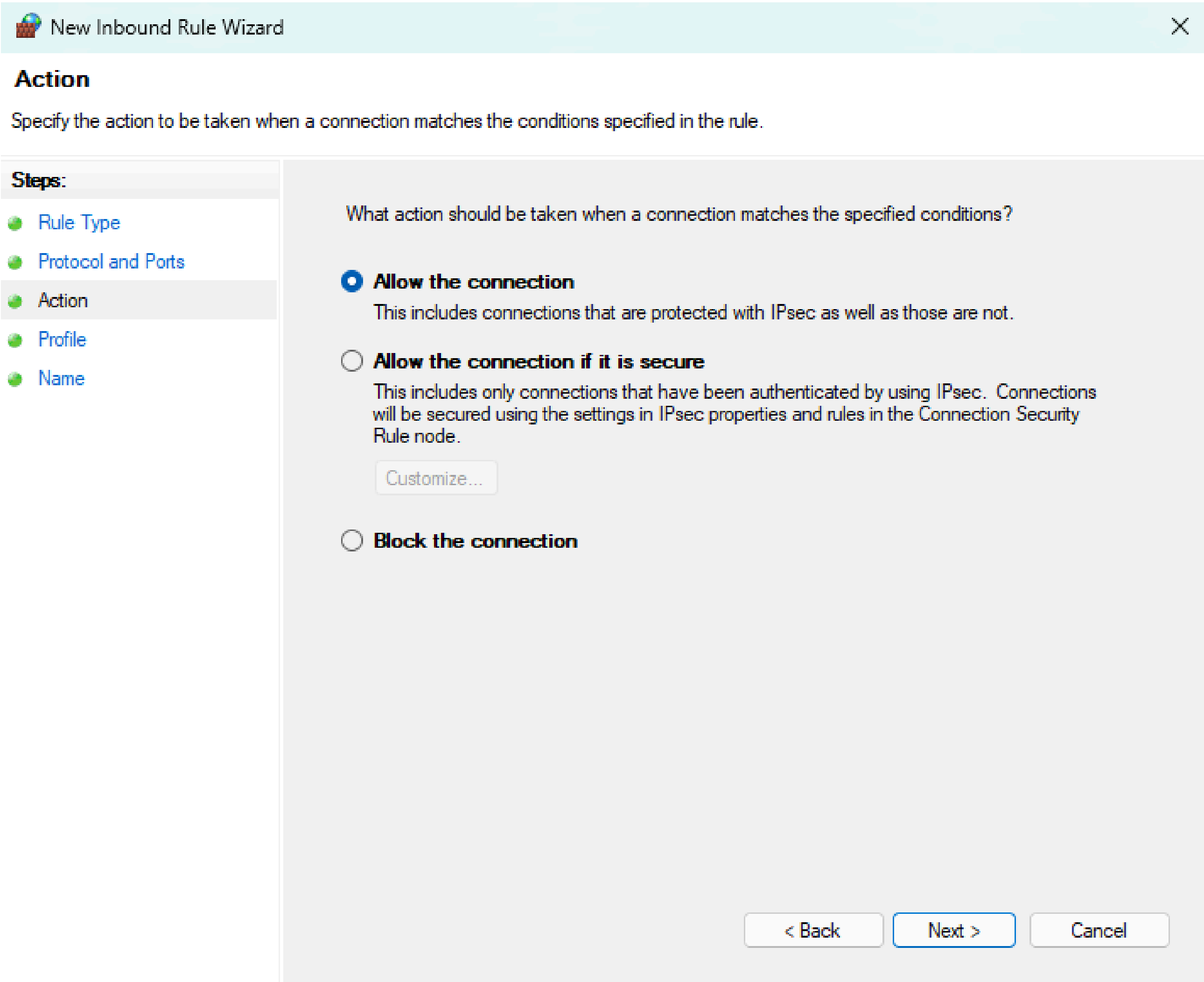
3. Select **Port**, then click **Next**:



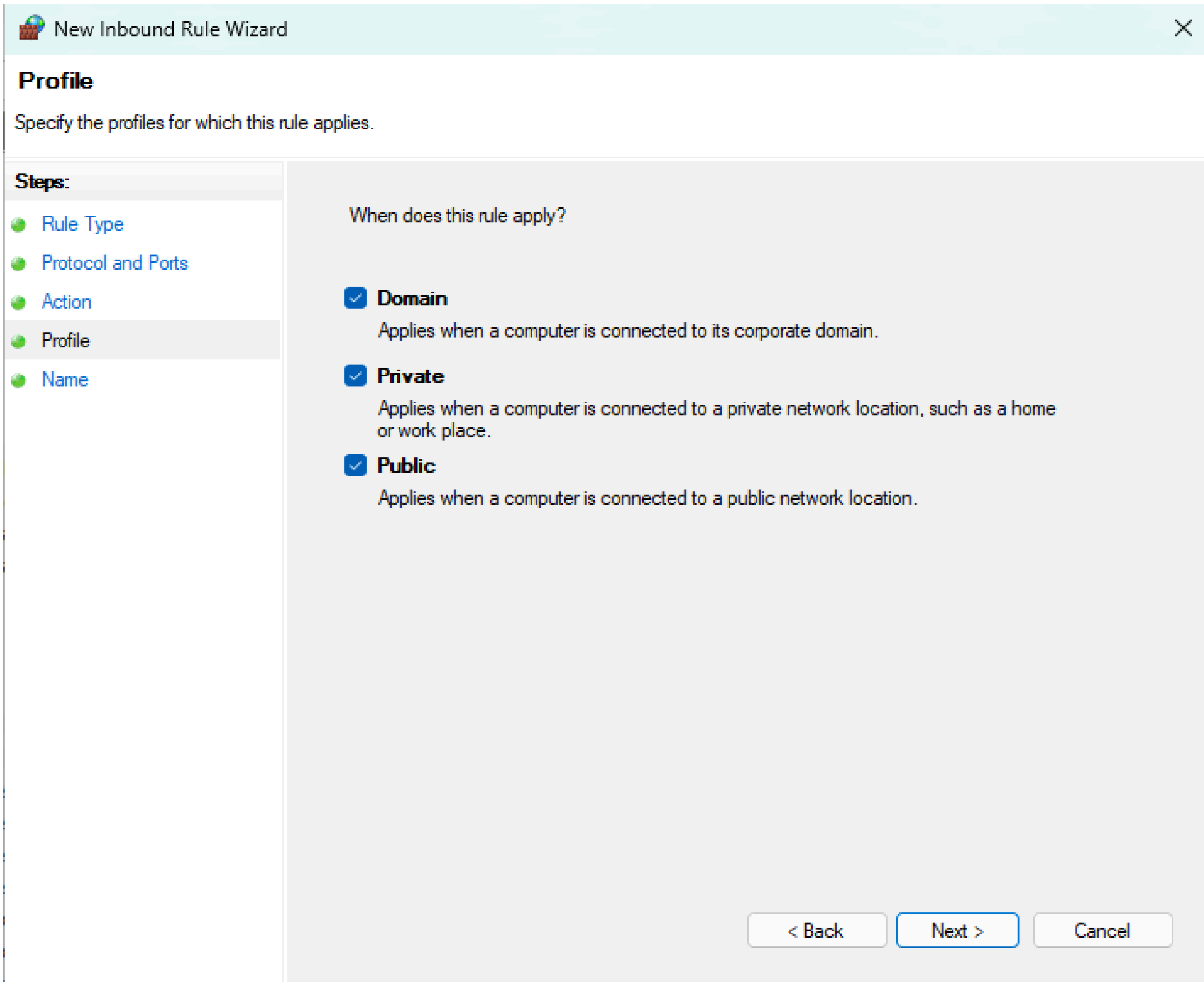
4. Select **UDP**, then select **Specific local ports** and enter `34964, 49152-65535`, then click **Next**:



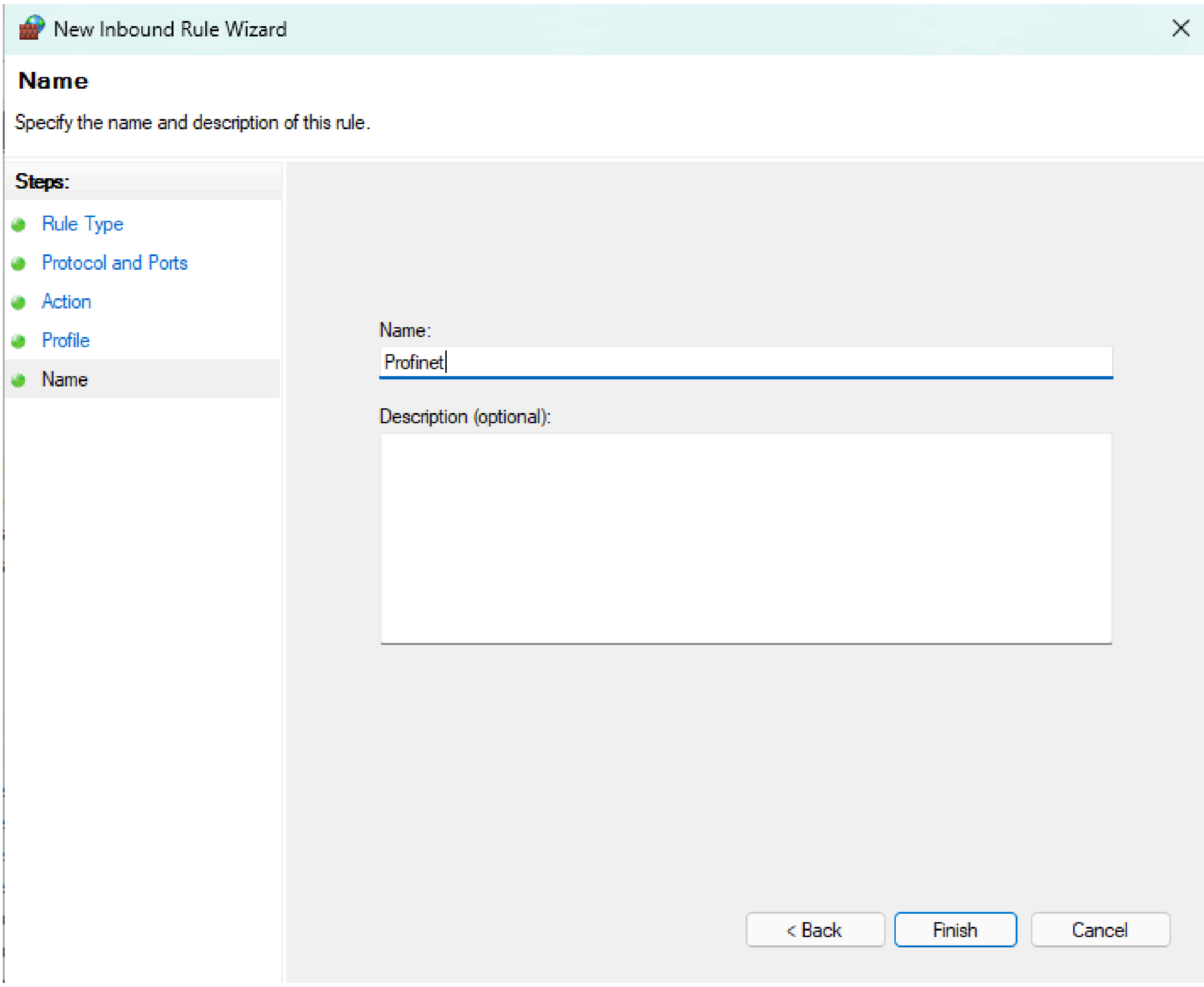
5. Select **Allow the connection**, then click **Next**:



6. Select which network types this rule applies to, then click **Next**:

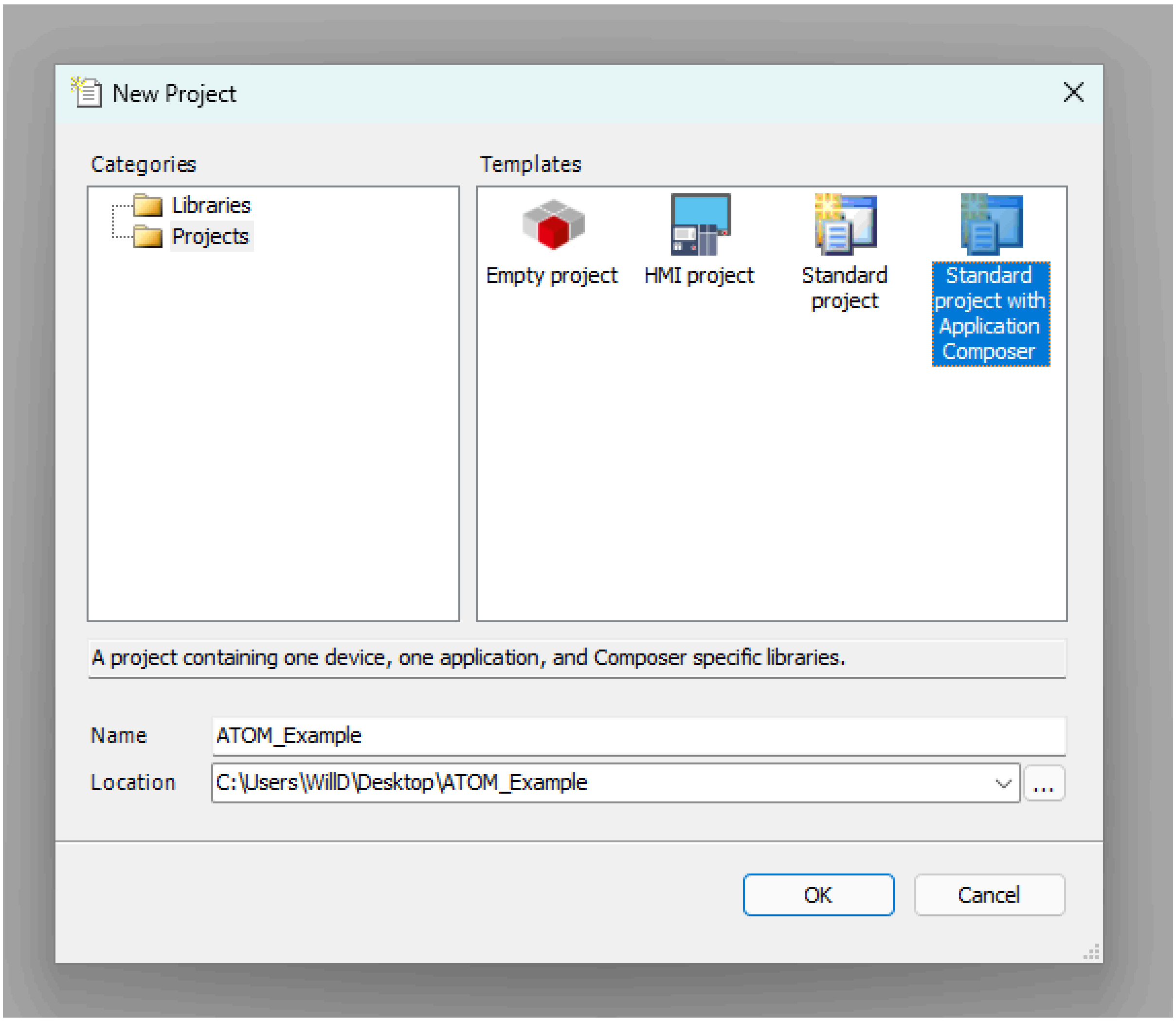


7. Name the rule **Profinet**, then click **Finish**:

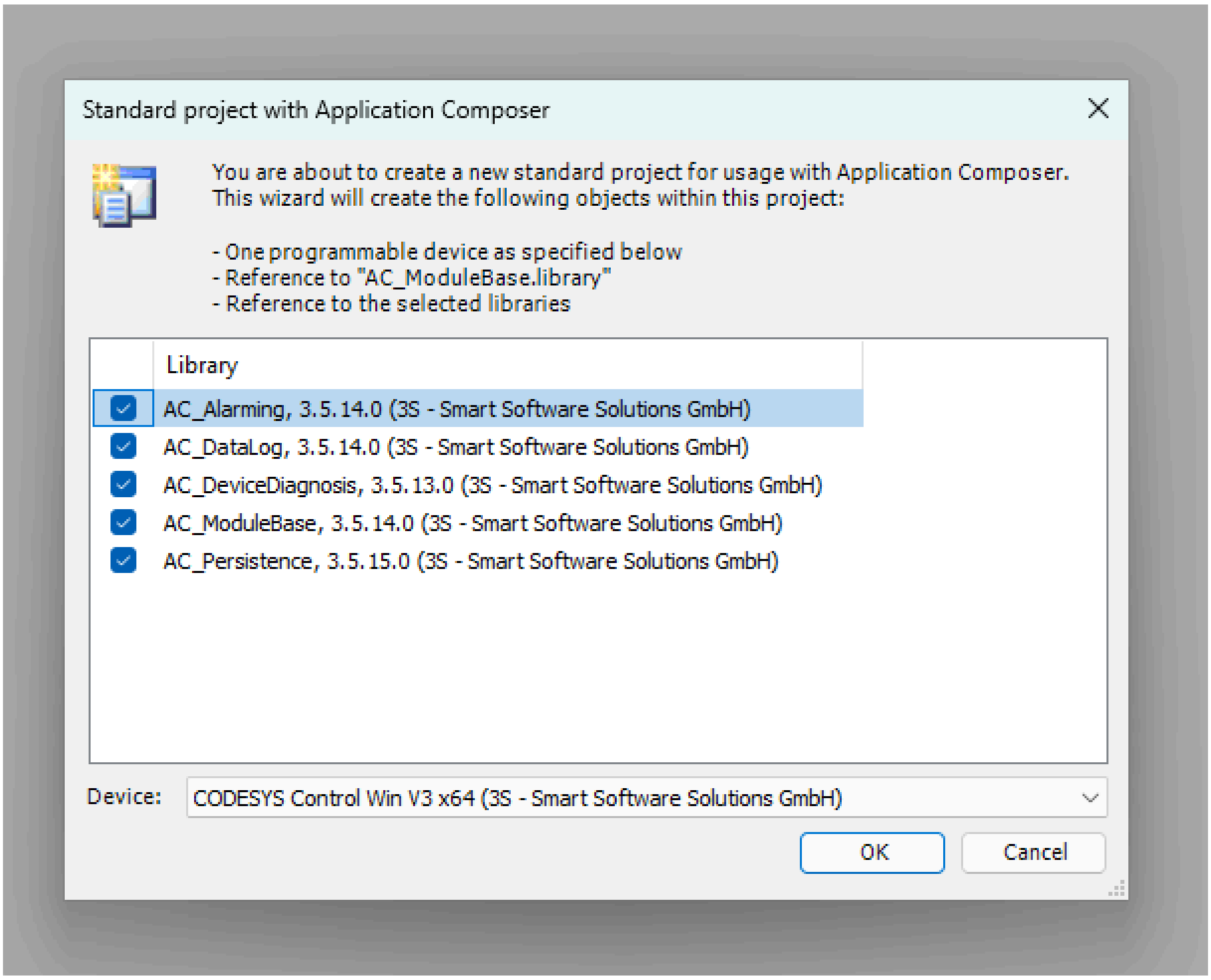


Create a Codesys project

Create a new Codesys project using the **Standard project with Application Composer** template:



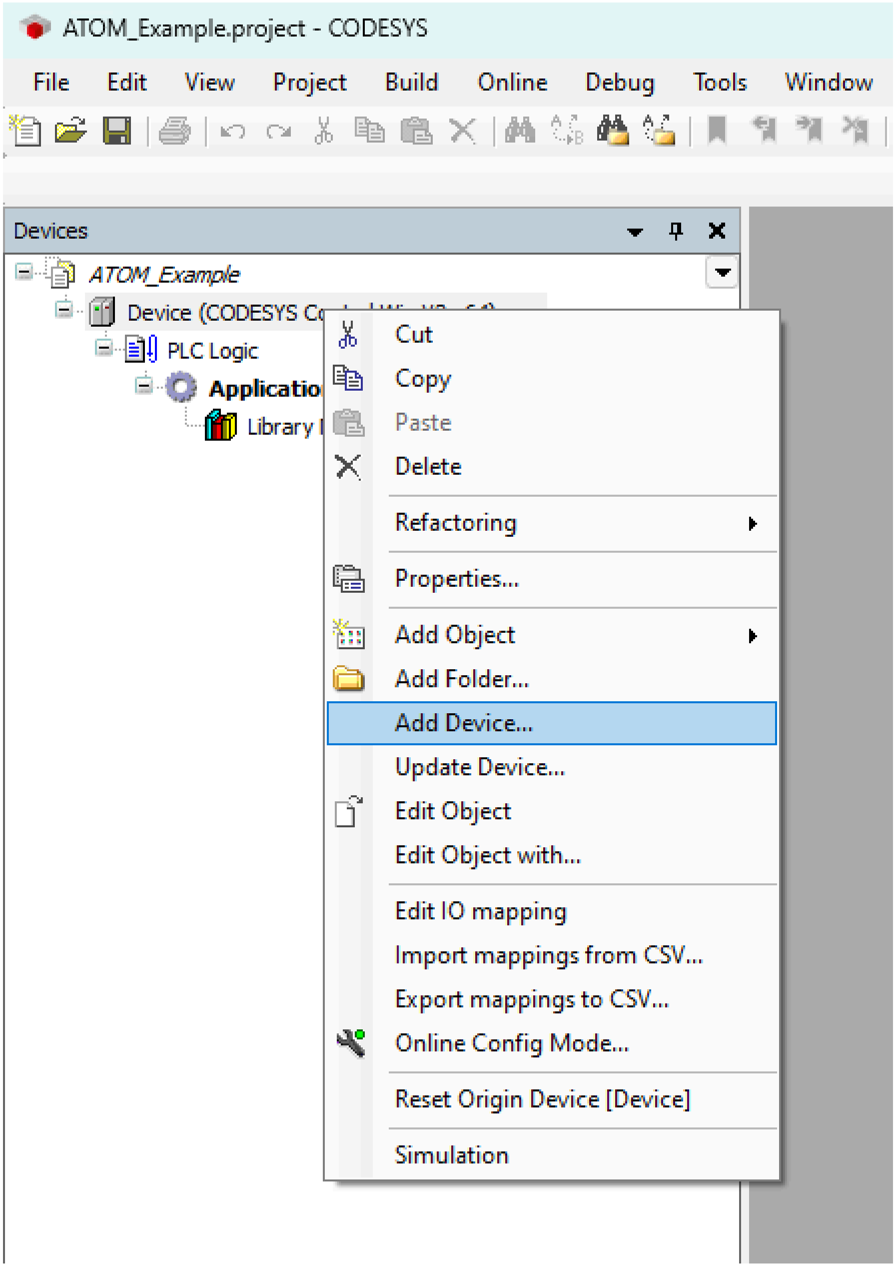
Check each library to include it in the project and select **CODESYS Control WIN V3 x64** as the device:



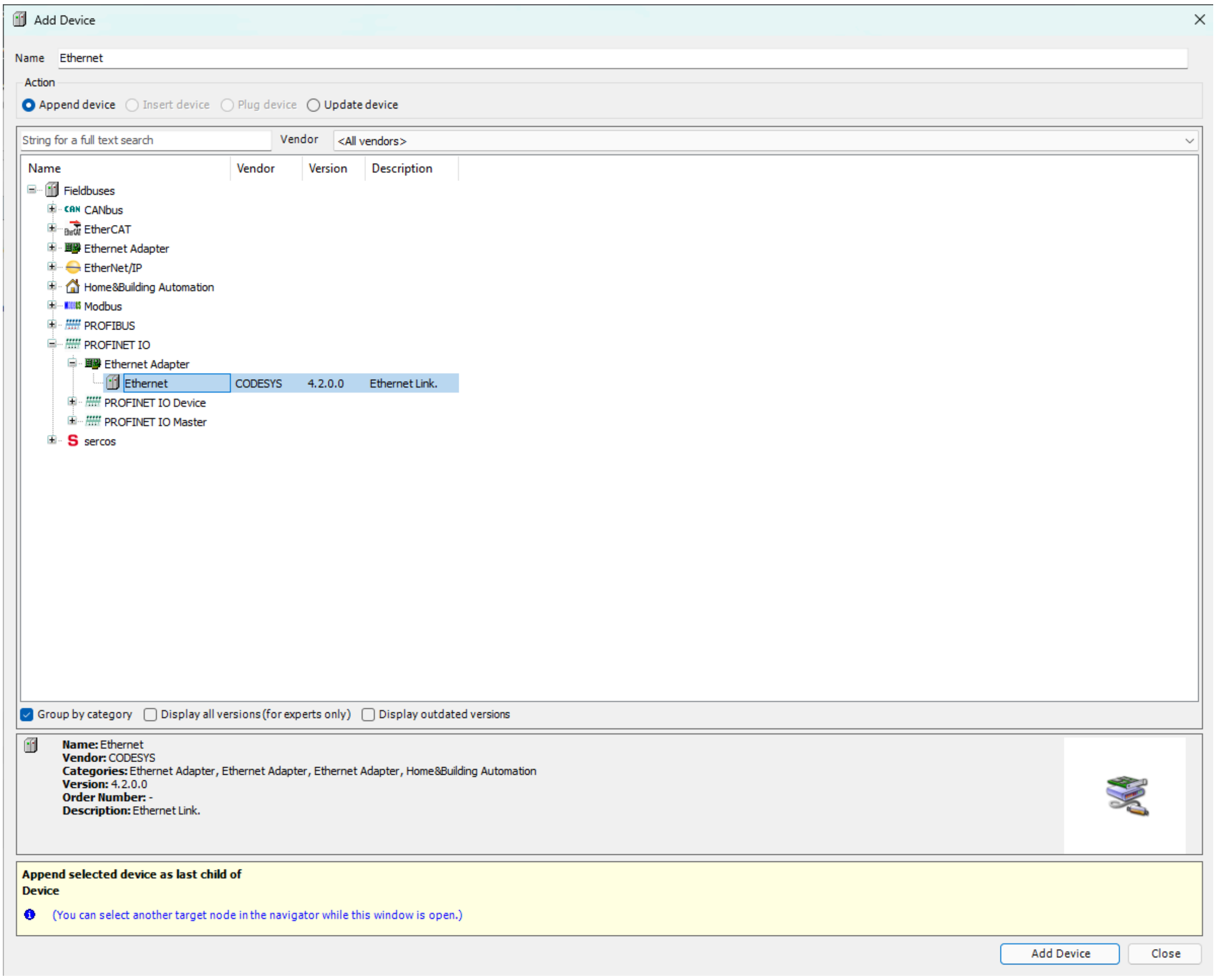
Adding a Profinet Controller

Next we'll add a Profinet Controller device. This allows the SoftPLC to discover Profinet I/O devices on the network (in our case, ATOM) and establish a connection with them.

First, right click **Device** and select **Add Device**:



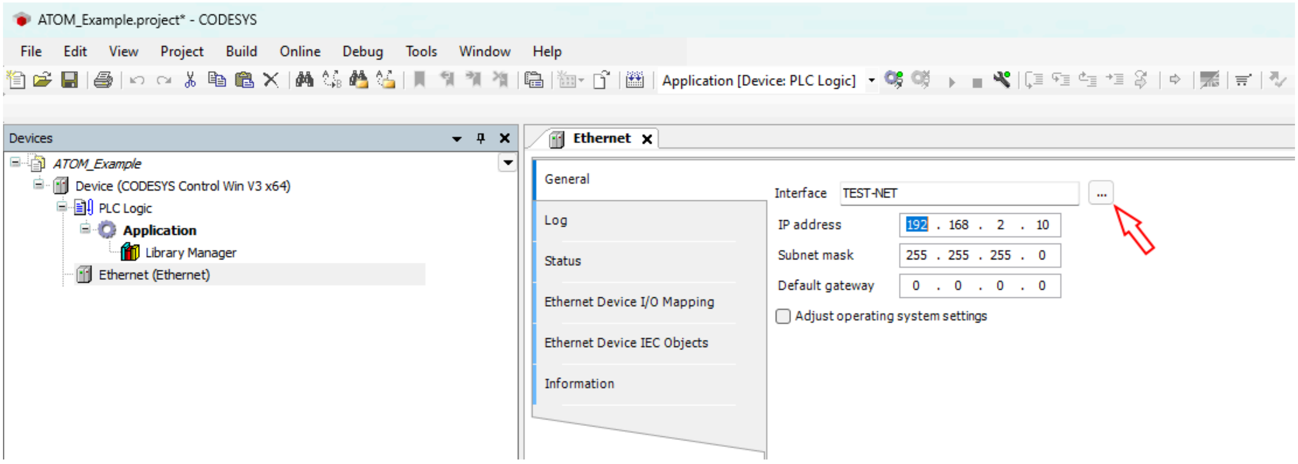
Next, expand **PROFINET IO > Ethernet Adapter** and select **Ethernet**, then click **Add Device**:



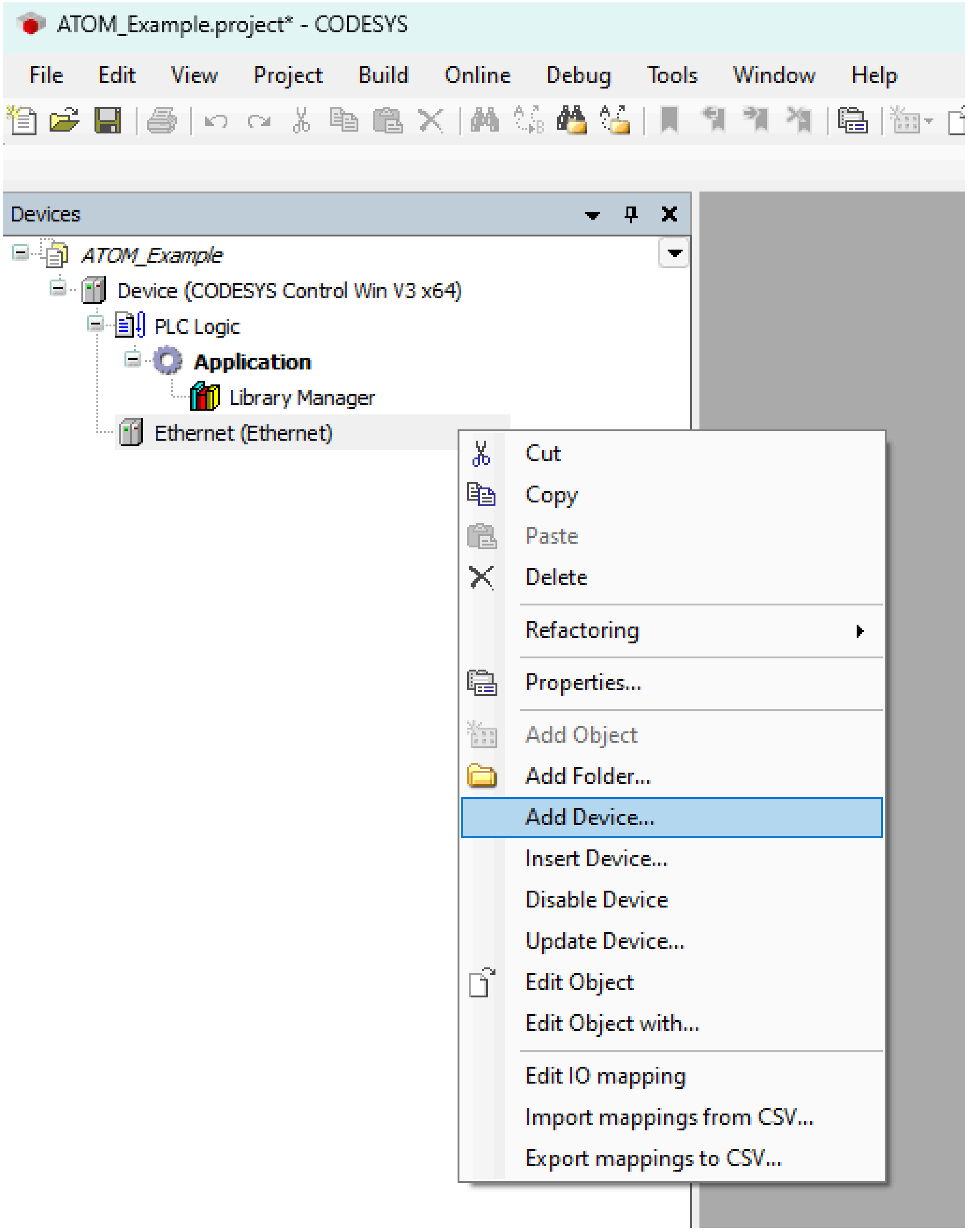
The newly added **Ethernet** device will now appear in the device tree. Double click **Ethernet (Ethernet)** to open its configuration tab. Within the **General** configuration tab, use the button indicated by the red arrow to select the network interface of the host machine that will be used to communicate with ATOM. In our case, we have a `TEST-NET` interface but this will be different for you.

INFO

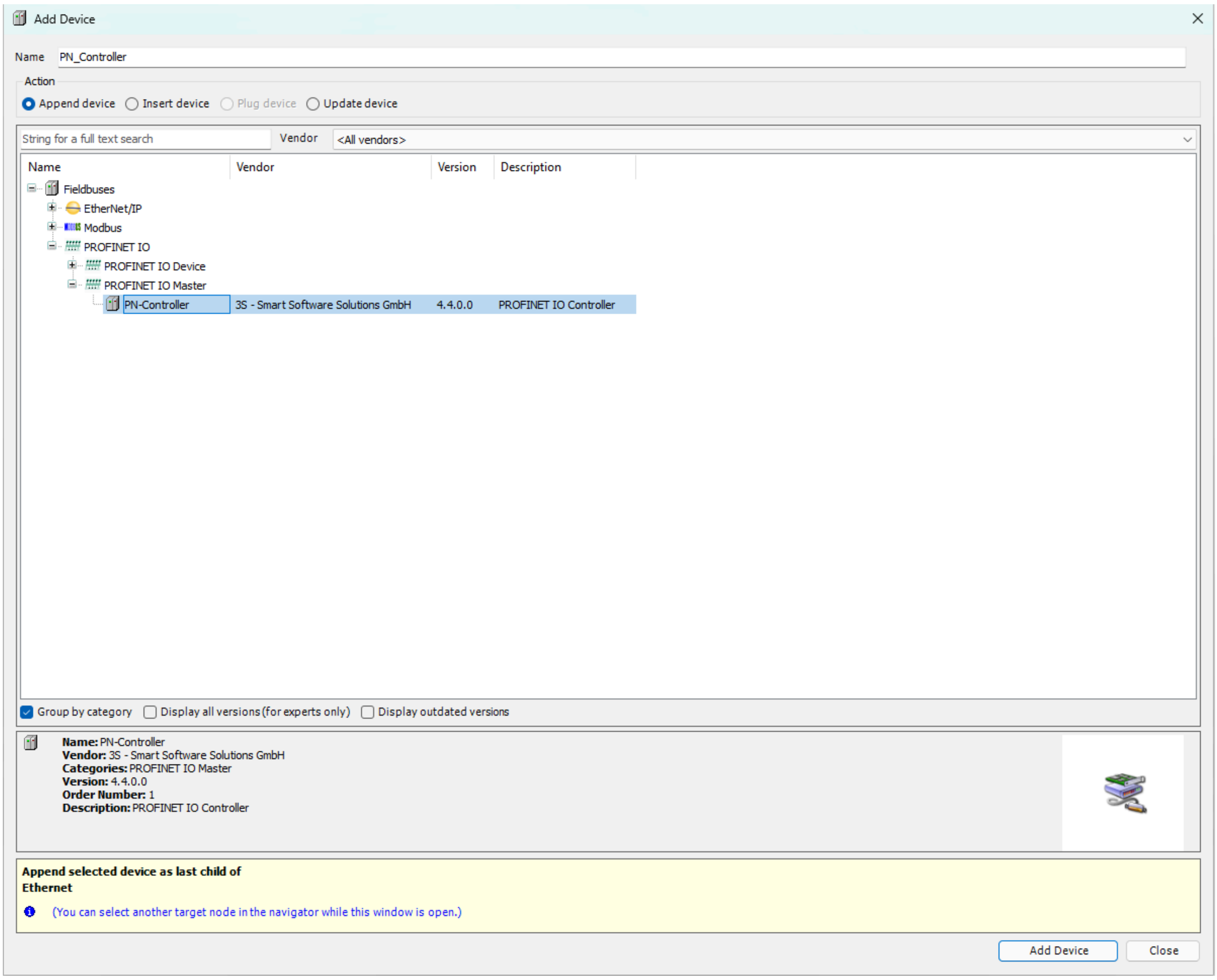
Note, you may get an error dialog displaying "Gateway not configured properly". If this is the case, make sure your SoftPLC is online by right-clicking the Codesys Win SysTray icon and starting the PLC. Navigate to the CODESYS Control Win V3 device in Codesys and use **Scan Network** to make sure the gateway is detected. Then, you can select the network interface.



Next, right click **Ethernet (Ethernet)** and select **Add Device**:



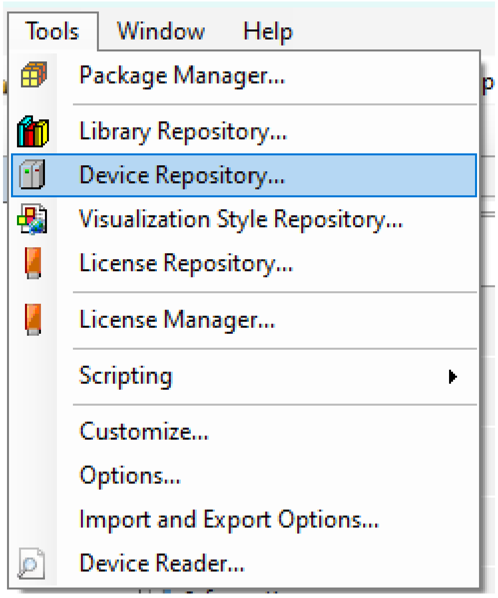
Expand **PROFINET IO > PROFINET IO Master**, select **PN-Controller** then click **Add Device**:



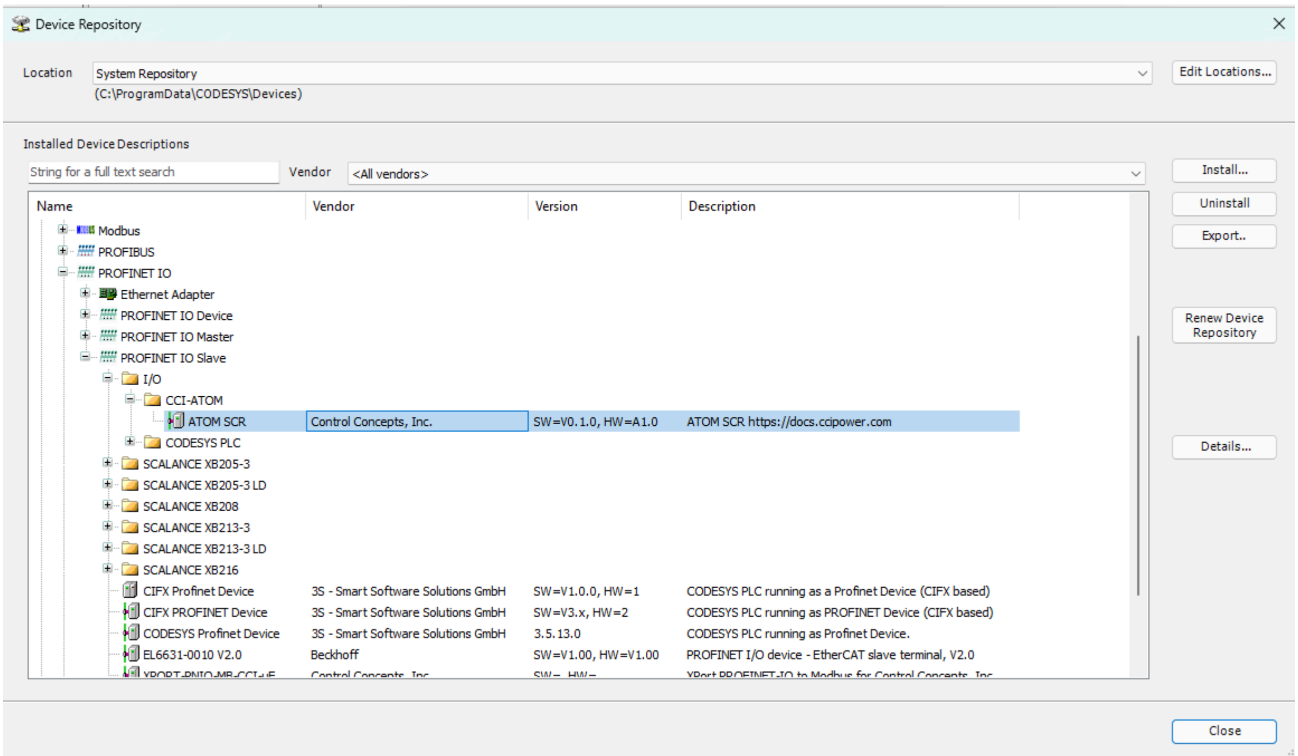
Your device tree should update to include the **PN-Controller** device.

Adding ATOM to the controller

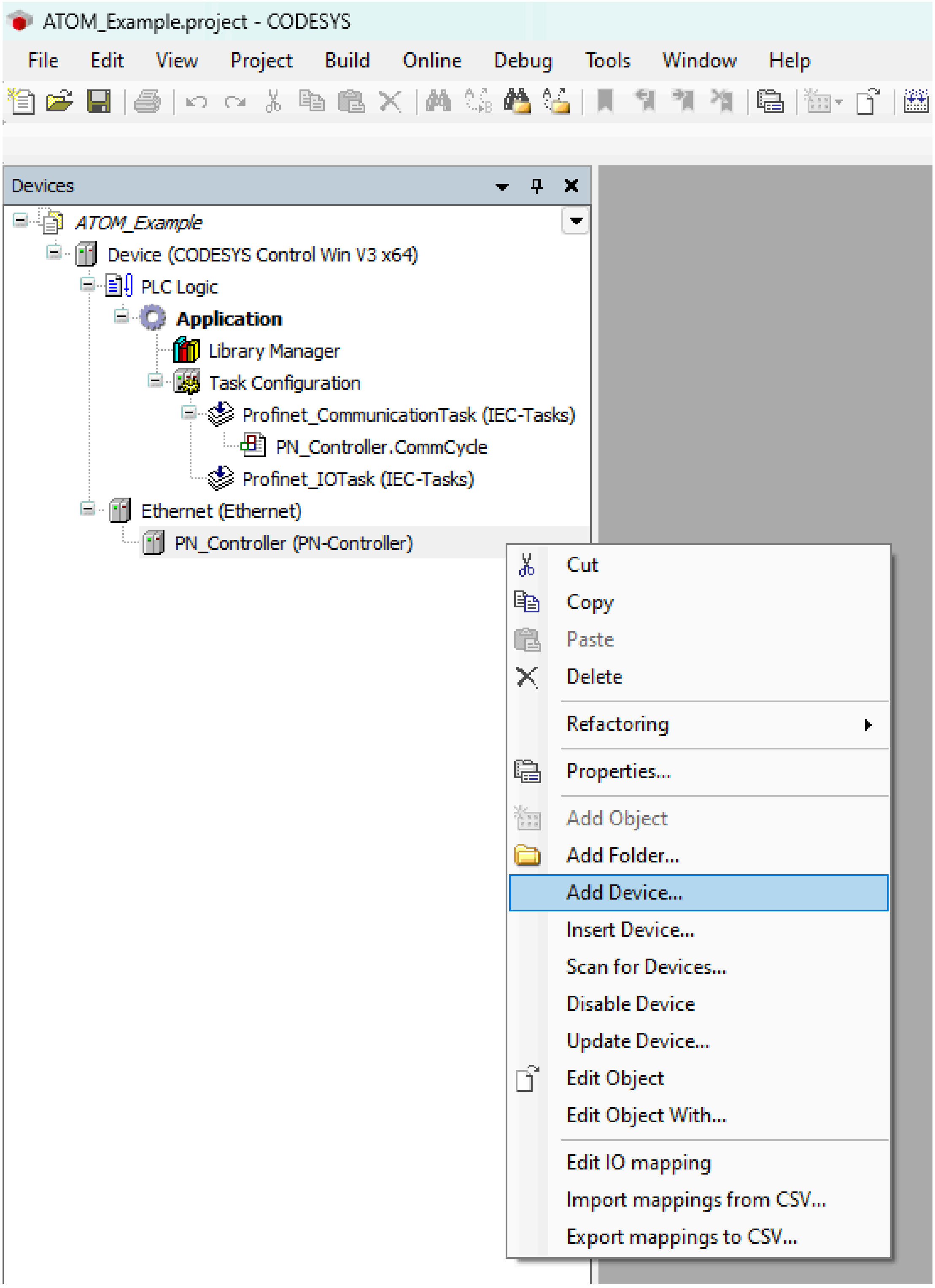
First, we'll import ATOM's GSDML file you downloaded earlier into our Codesys device library. Open the tools menu and select **Device repository**:



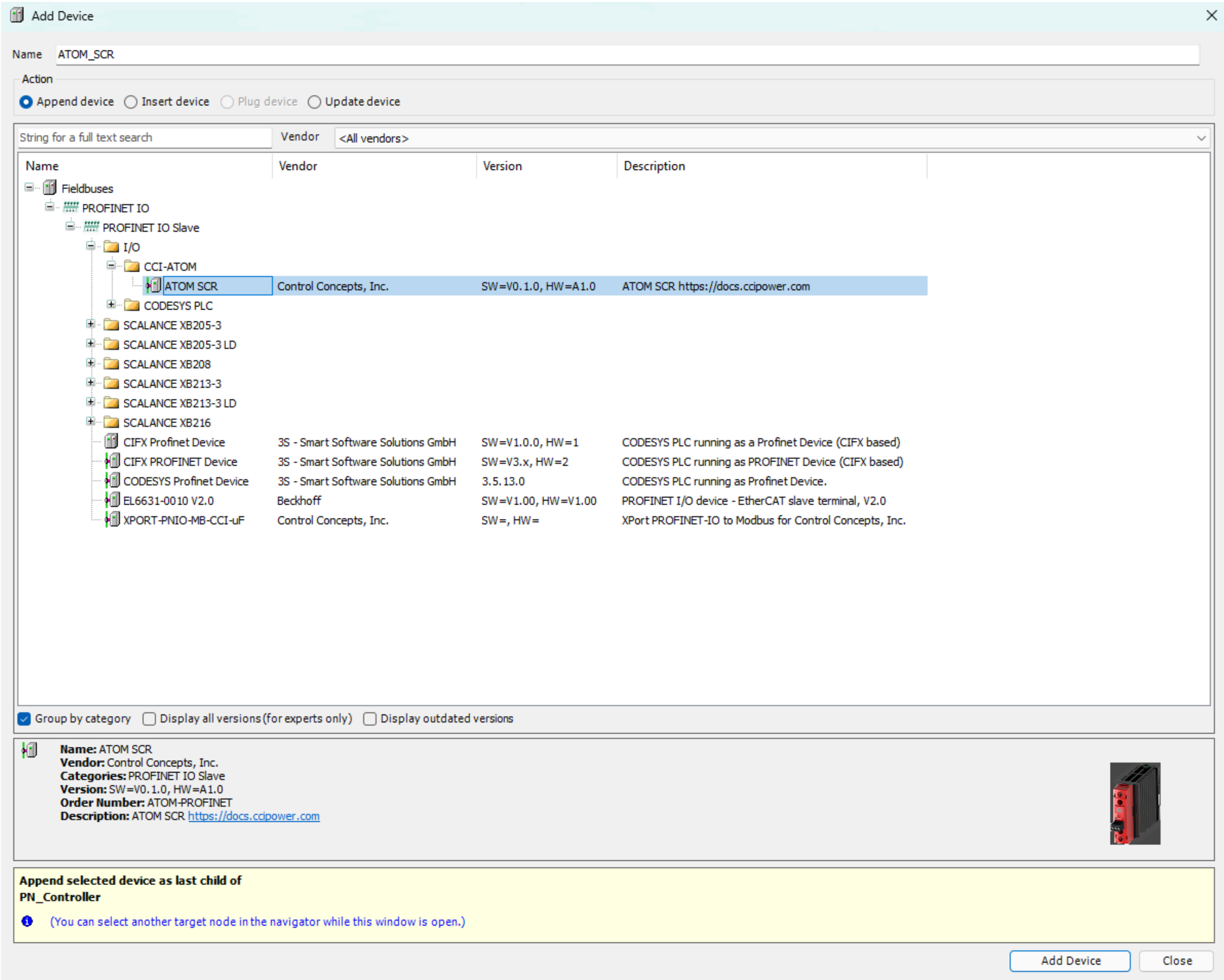
Next, click **Install** and select the `GSDML-V2.43-Control-Concepts-ATOM_20231108.xml` file. After you click install, **Atom** will appear under the **PROFINET IO > PROFINET IO Slave > I/O** category. Click **Close** to dismiss the dialog:



Now, we'll add ATOM to the PN-Controller. Right click **EtherNet/IP Scanner (EtherNet/IP Scanner)** and select **Add Device**:



Expand **PROFINET IO** > **PROFINET IO Slave** > **I/O** > **CCI-ATOM** and select **ATOM SCR**, then click **Add Device**:



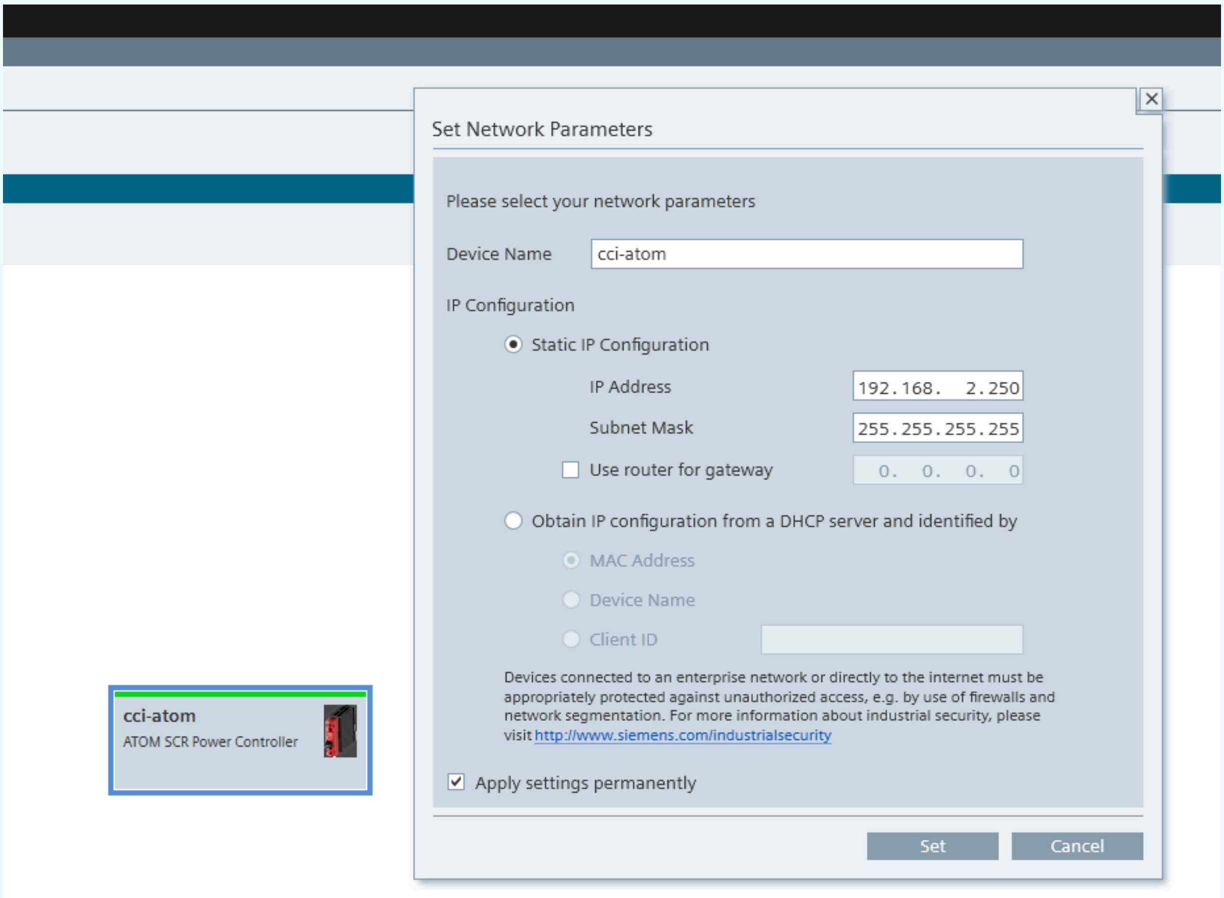
Finally, double click **Atom (Atom)** to open its configuration tab. In the **General** tab, set the **Station name**, **IP Address**, and **Subnet mask** for your ATOM SCR:

INFO

You can find or change these parameters in Control Panel, or using a tool like Proneta. Make sure your station name and IP settings on Atom are properly set to the same values you enter here so that Codesys can connect to ATOM.

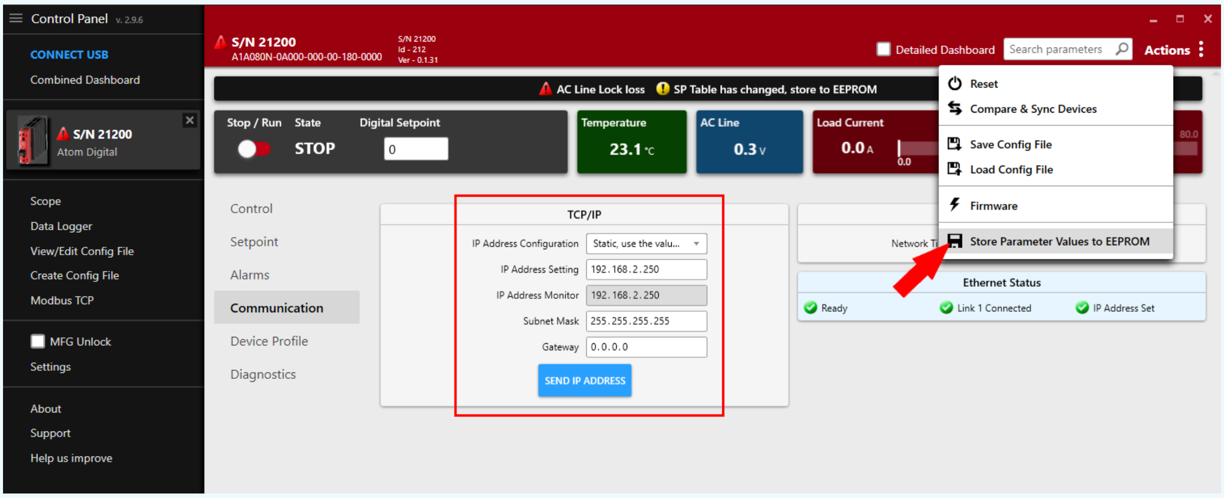
Proneta

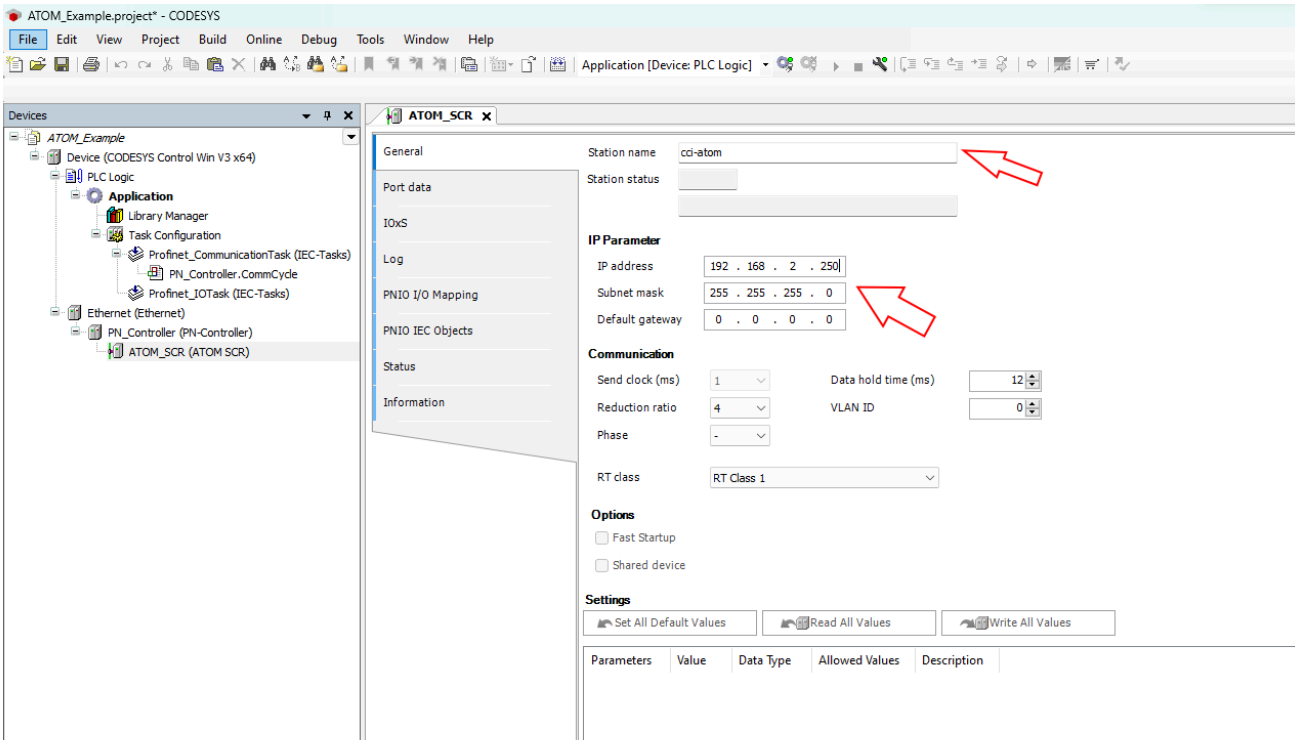
If you're using Proneta, make sure to change the IP settings with **Store permanently** checked.



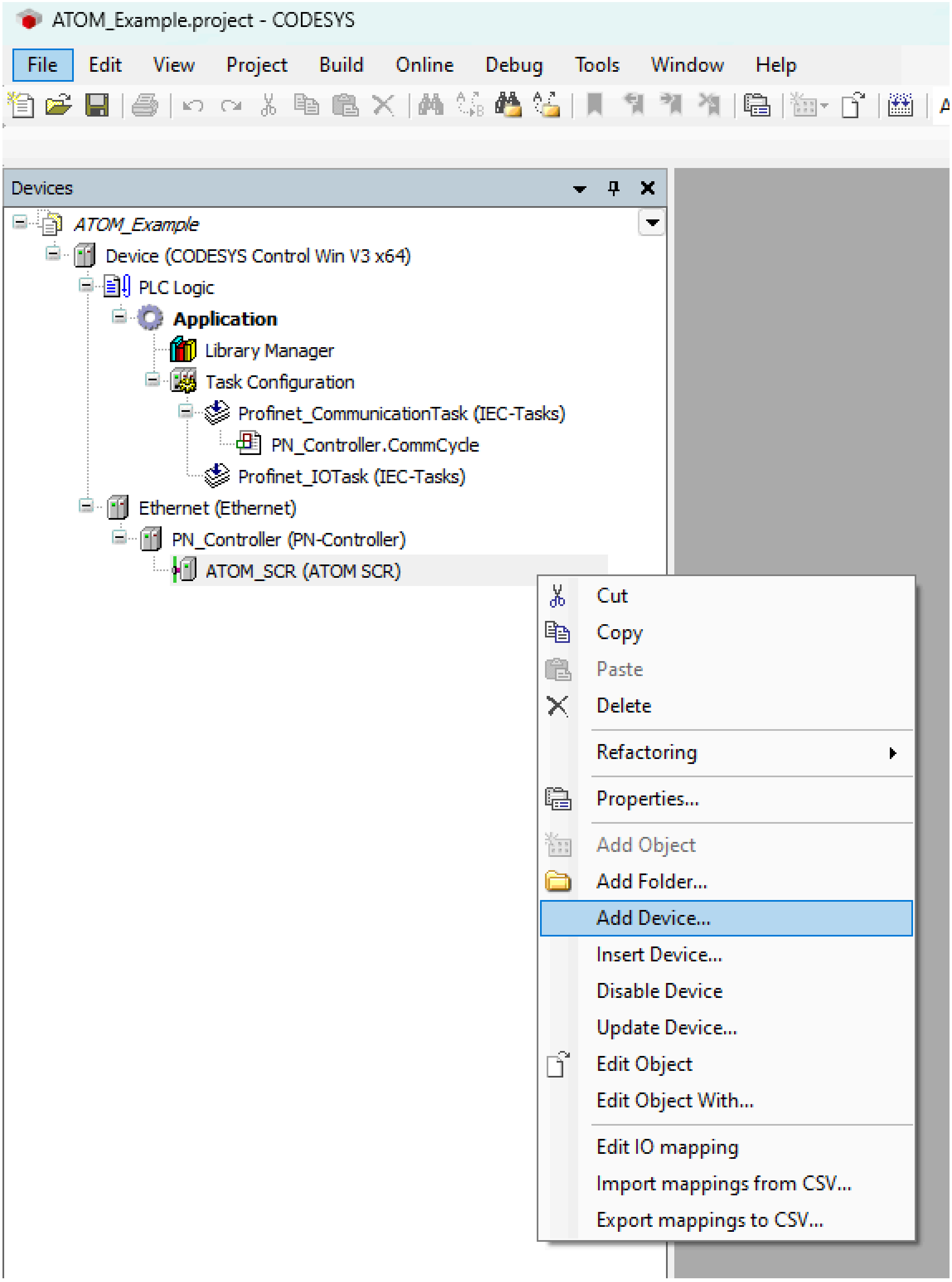
Control Panel

Connect your Atom unit to your PC using a USB cable. Open Control Panel and update your Atom's communication parameters. When you're finished, click **Send IP**

> **Address**, then go to **Actions** in the upper right and select **Store Parameter Values to EEPROM**:
>
> 
>

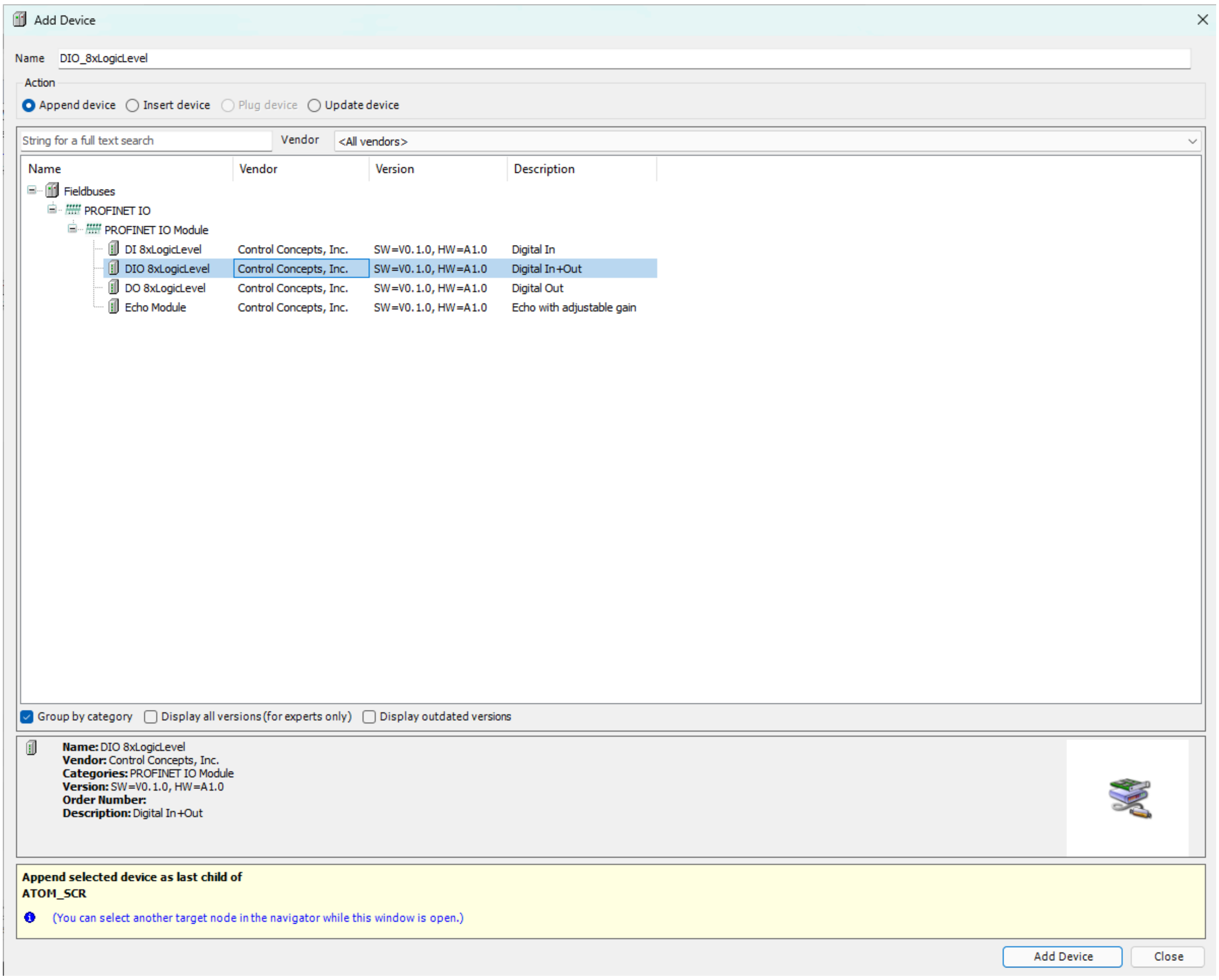


Next, right click **ATOM_SCR (ATOM SCR)** and select **Add Device**:



Here, you can choose which Profinet I/O modules to enable for your Atom. Select **DIO 8xLogicLevel** which allows both input and output of data to/from Atom. You can add other

I/O modules if needed. Then, click **Add Device**:



Create a program

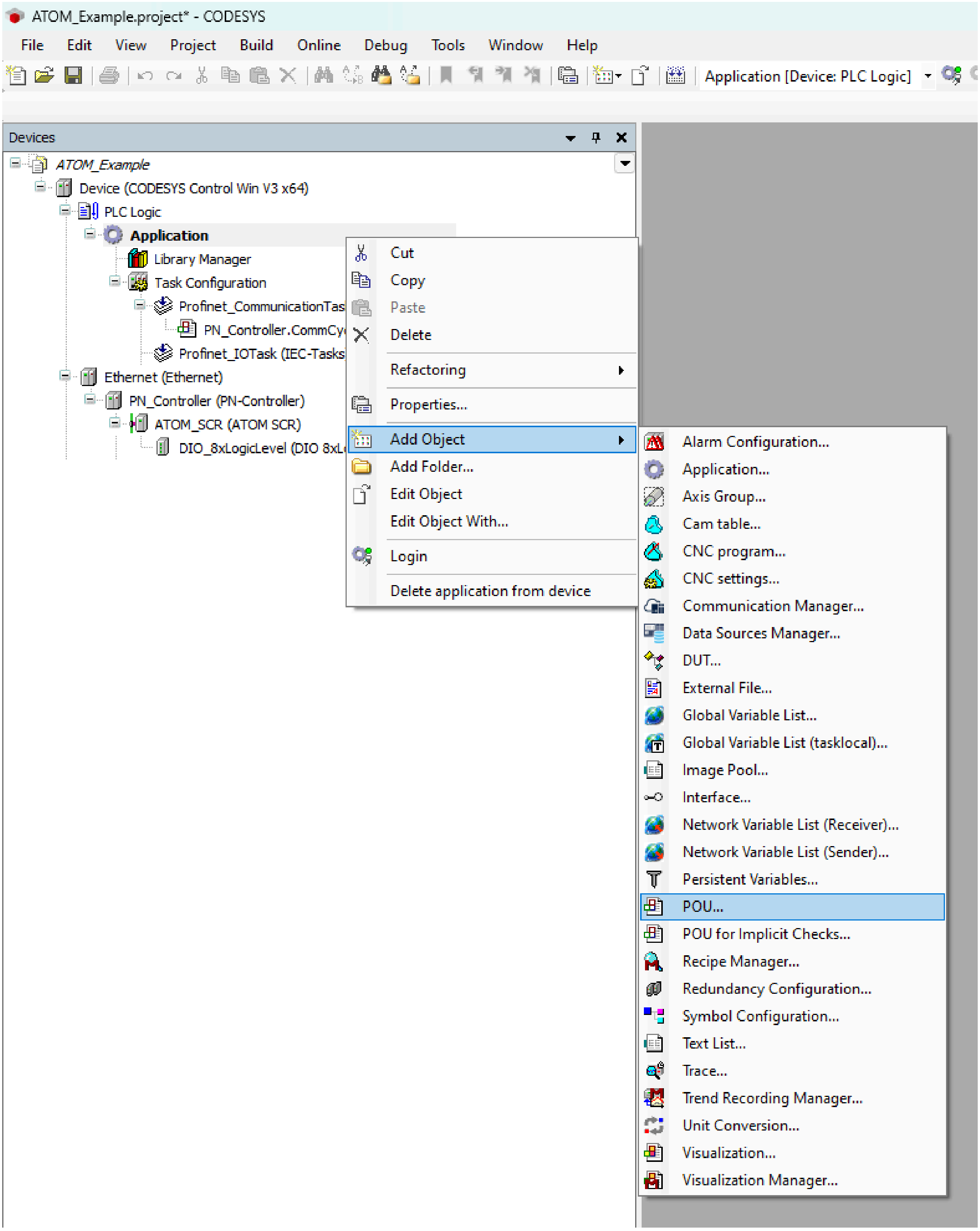
Next, we'll create a PLC program. We provide examples for both ladder logic and structured text:

- Program with ladder logic
- Program with structured text

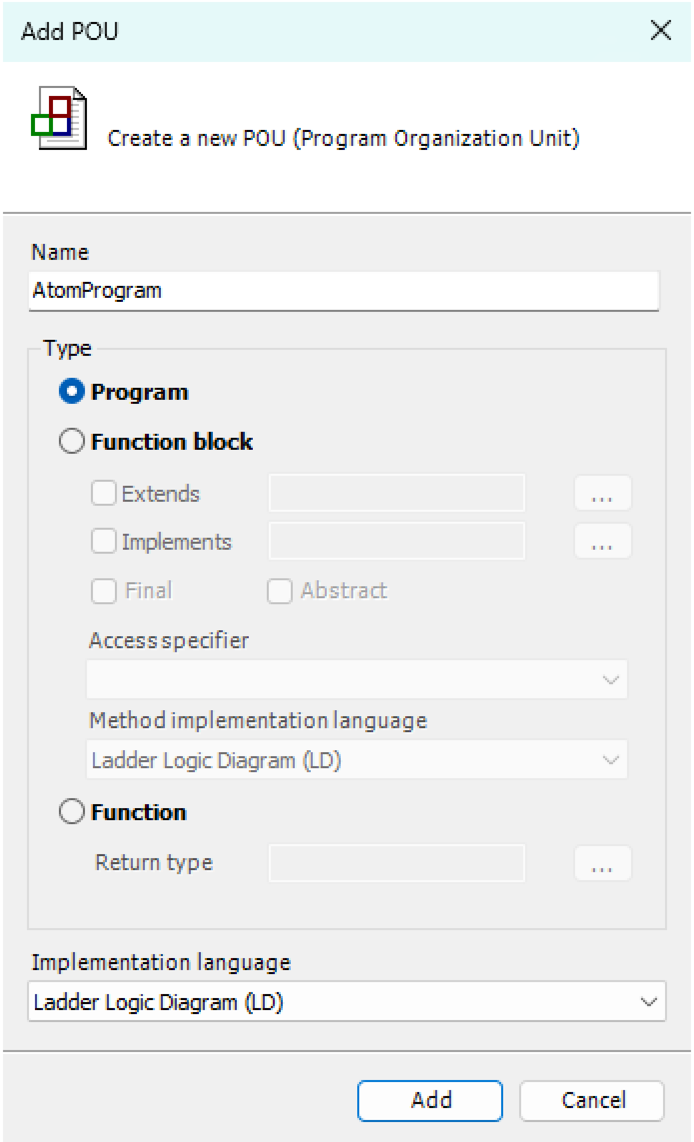
Example: Ladder logic

Creating the program

Right click **Application** and select **Add Object > POU**:


ATOM_Example.project* - CODESYS
File
Edit
View
Project
Build
Online
Debug
Tools
Window
Help
Application [Device: PLC Logic]
Devices
ATOM_Example
Device (CODESYS Control Win V3 x64)
PLC Logic
Application
Library Manager
Task Configuration
Profinet_CommunicationTas
PN_Controller.CommCy
Profinet_IOTask (IEC-Tasks
Ethernet (Ethernet)
PN_Controller (PN-Controller)
ATOM_SCR (ATOM SCR)
DIO_8xLogicLevel (DIO 8xL
Cut
Copy
Paste
Delete
Refactoring
Properties...
Add Object
Add Folder...
Edit Object
Edit Object With...
Login
Delete application from device
Alarm Configuration...
Application...
Axis Group...
Cam table...
CNC program...
CNC settings...
Communication Manager...
Data Sources Manager...
DUT...
External File...
Global Variable List...
Global Variable List (tasklocal)...
Image Pool...
Interface...
Network Variable List (Receiver)...
Network Variable List (Sender)...
Persistent Variables...
POU...
POU for Implicit Checks...
Recipe Manager...
Redundancy Configuration...
Symbol Configuration...
Text List...
Trace...
Trend Recording Manager...
Unit Conversion...
Visualization...
Visualization Manager...

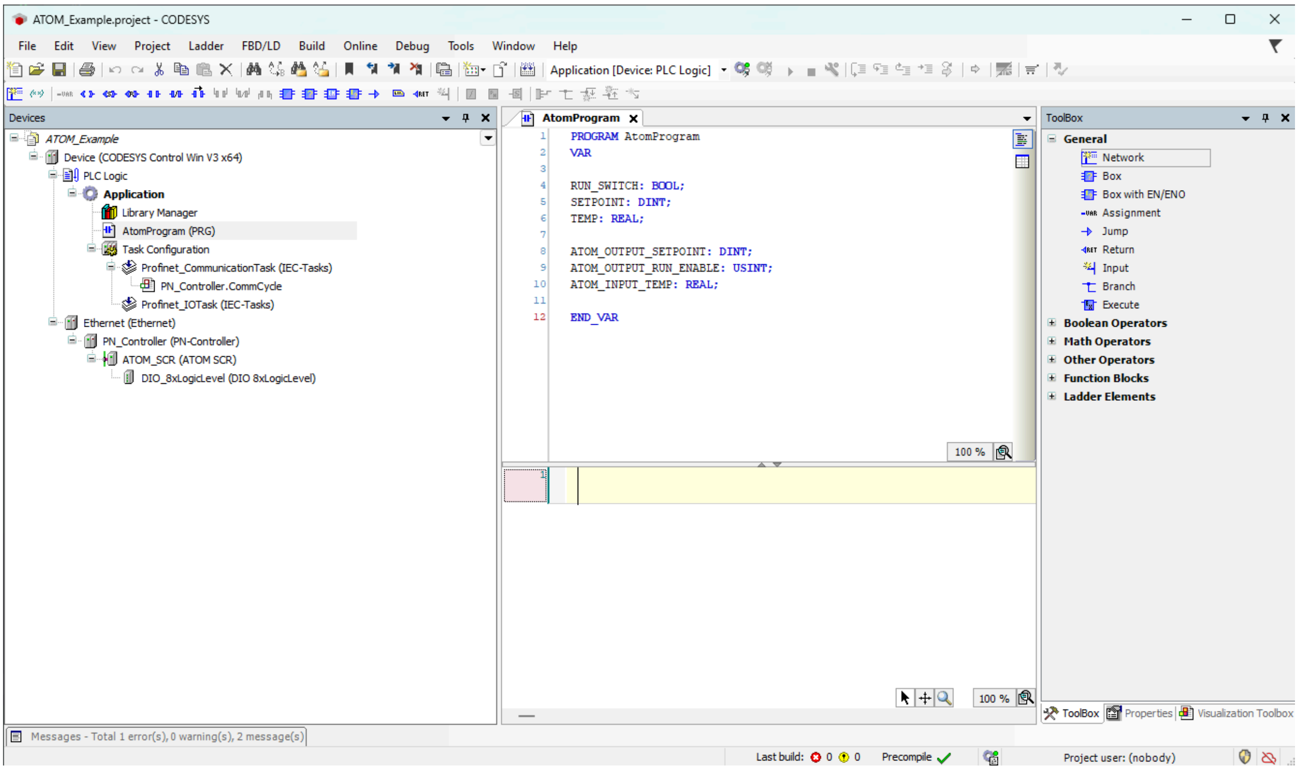
Set the name to `AtomProgram` and select **Ladder Diagram (LD)** as the Implementation language:



Copy the following code into the top panel of the **AtomProgram** editor:

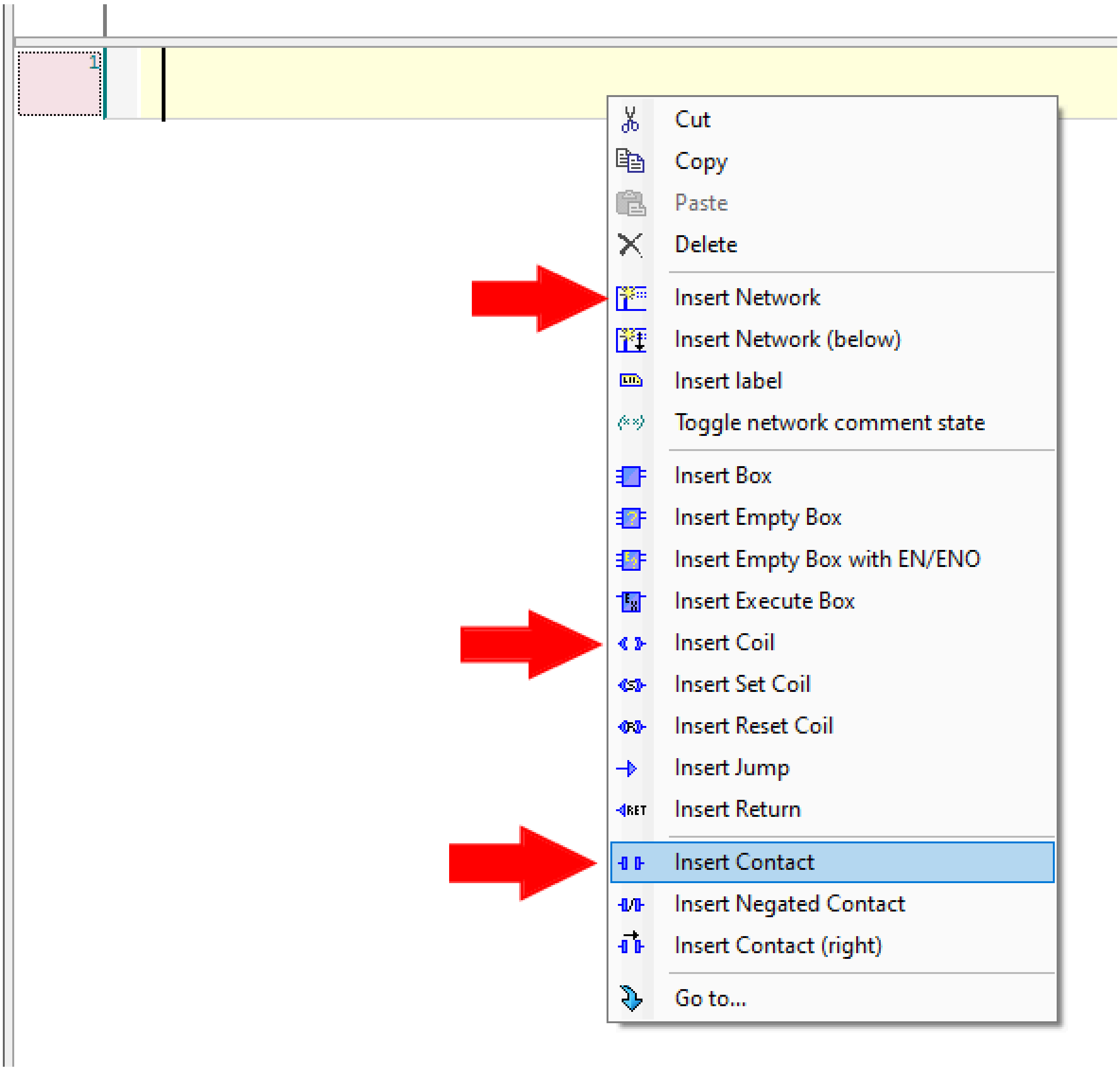
```
PROGRAM AtomProgram
VAR

RUN_SWITCH: BOOL;
SETPOINT: DINT;
TEMP: REAL;

ATOM_OUTPUT_SETPOINT: DINT;
ATOM_OUTPUT_RUN_ENABLE: USINT;
ATOM_INPUT_TEMP: REAL;

END_VAR
```

After you've copied the code over, the editor for **AtomProgram** should look like this:

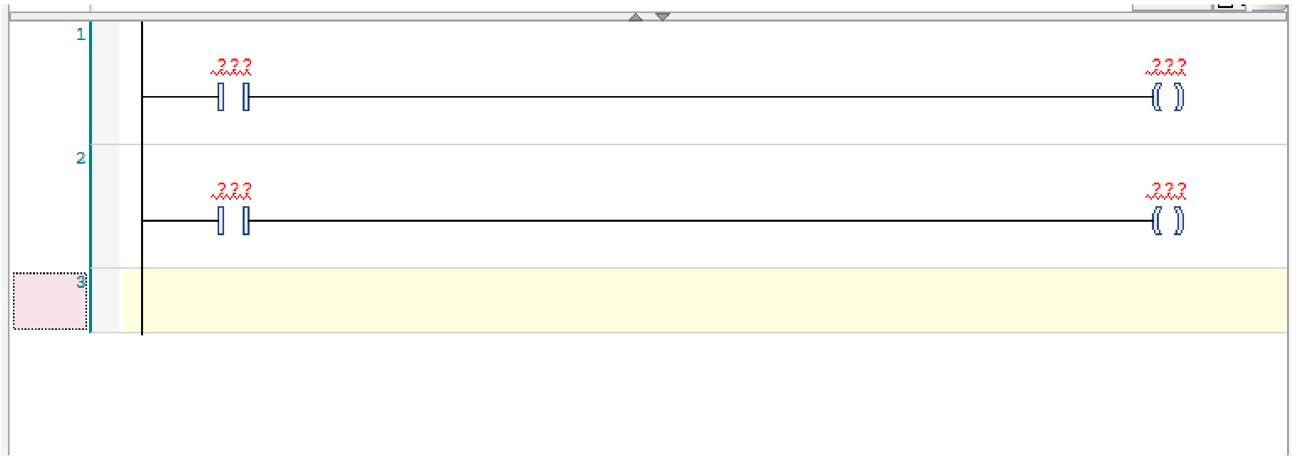


In the bottom panel of the editor, we'll create a simple ladder logic program using the variables we just added above.

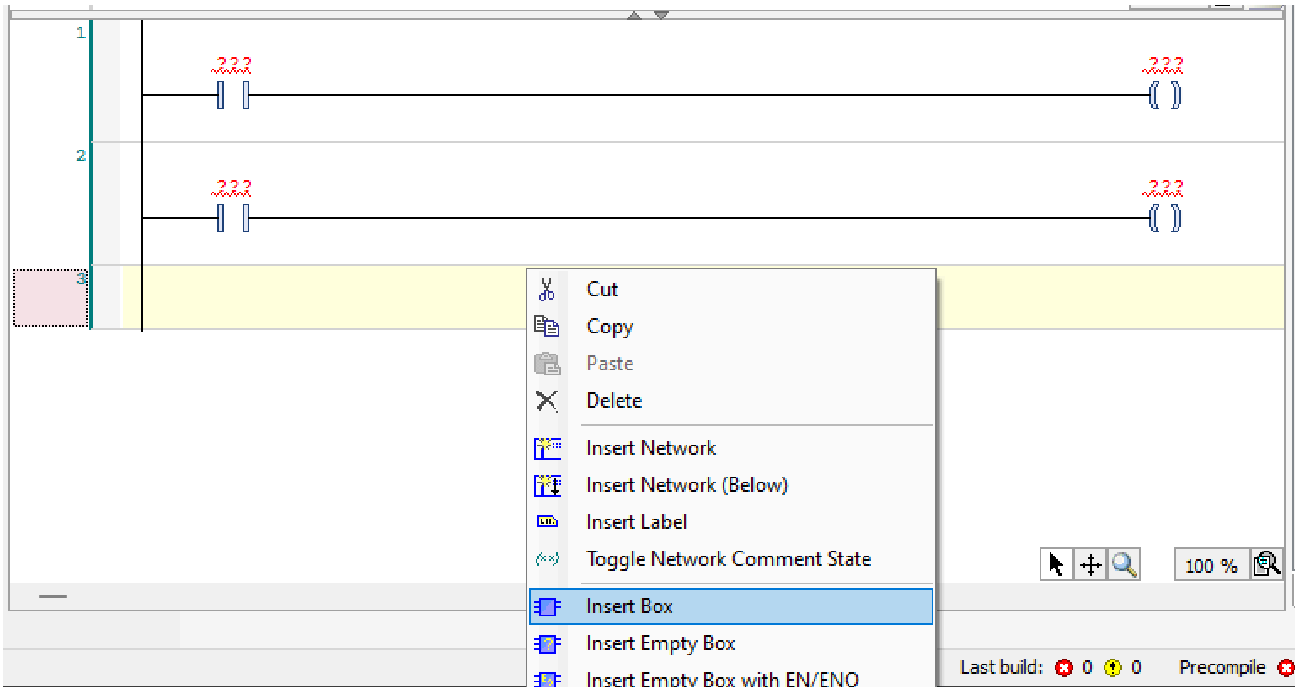
1. Create **3** networks total by right-clicking and selecting **Insert Network** three times.
2. For the first *two* rungs (networks), insert a contact and a coil.



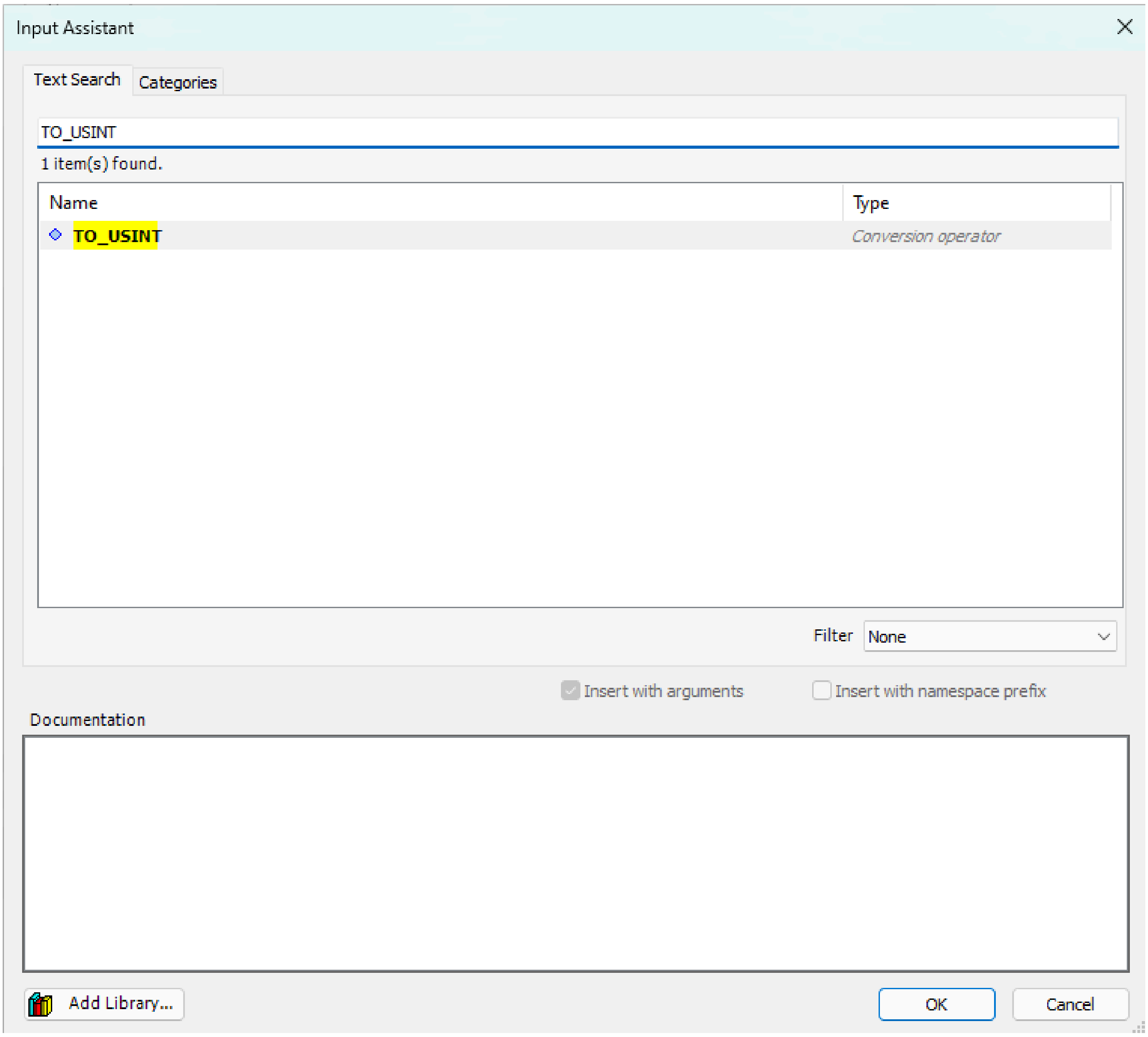
After you're finished, your ladder logic program should look like:



On the third rung, right click and select **Insert Box**:



Add a **TO_USINT** box:



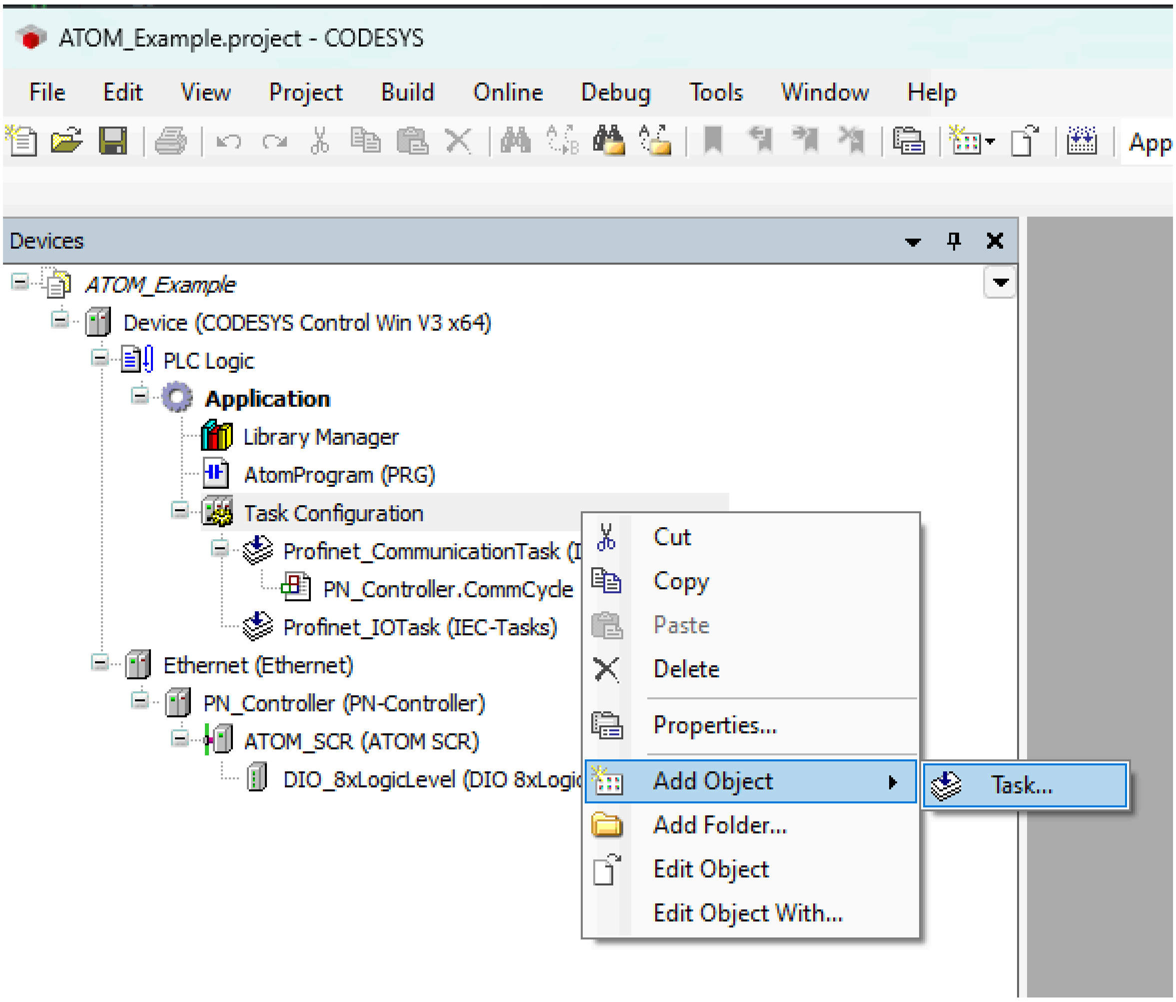
For the first two rungs, replace the `???` with the corresponding variables:

1. **Rung #1** - `ATOM_INPUT_TEMP` and `TEMP`
2. **Rung #2** - `SETPOINT` and `ATOM_OUTPUT_SETPOINT`

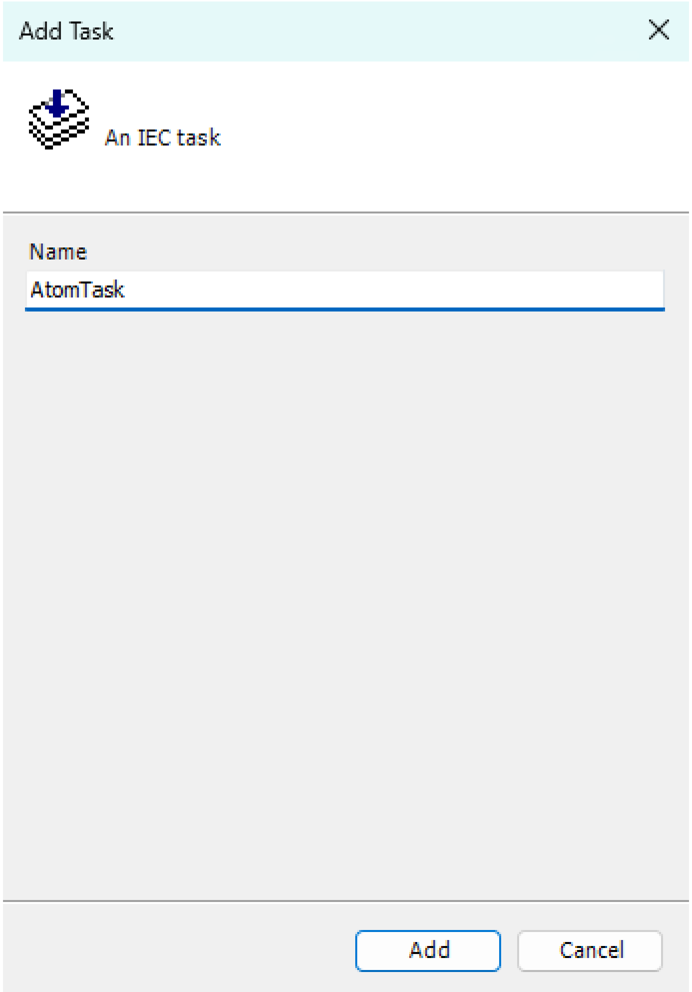
On the third rung, set the input to `EN` to `TRUE` and set the input parameter to `RUN_SWTICH` and output parameter to `ATOM_OUTPUT_RUN_ENABLE`. After you're finished, your ladder logic program should look like:

Finally, we'll add a task to call **AtomProgram** from the PLC's control loop:

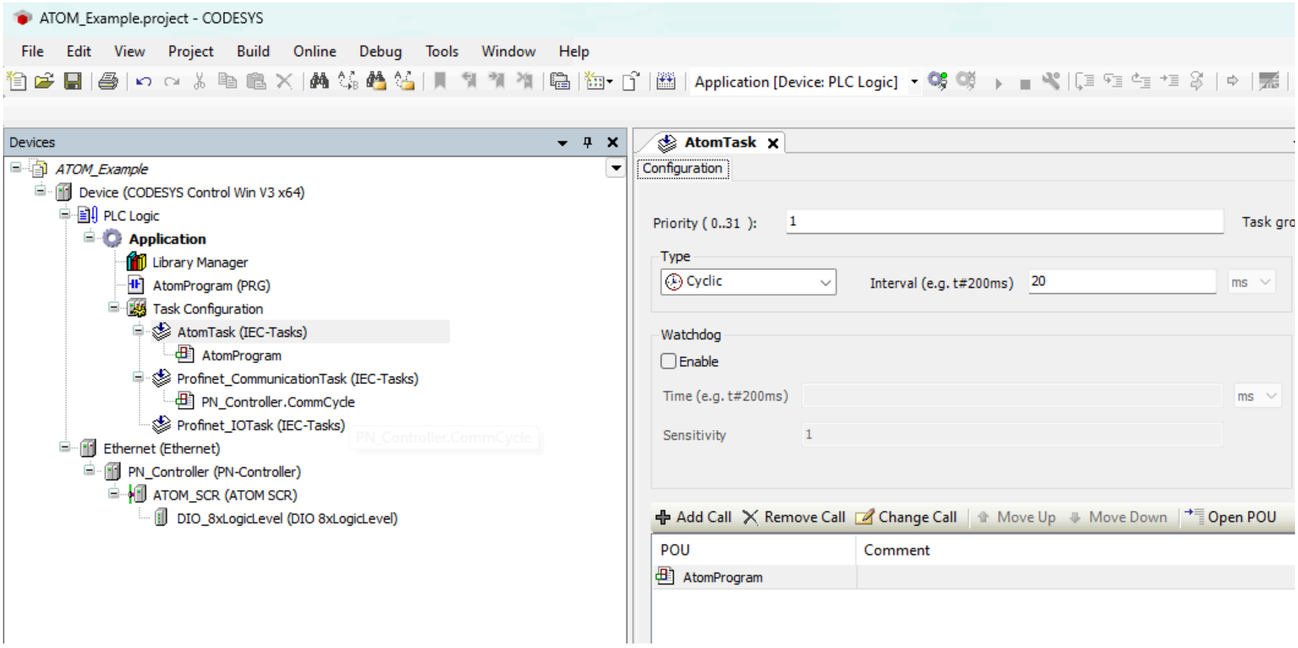
Right click **Task Configuration** and select **Add Object > Task**:



Name your task `AtomTask` and click **OK**:



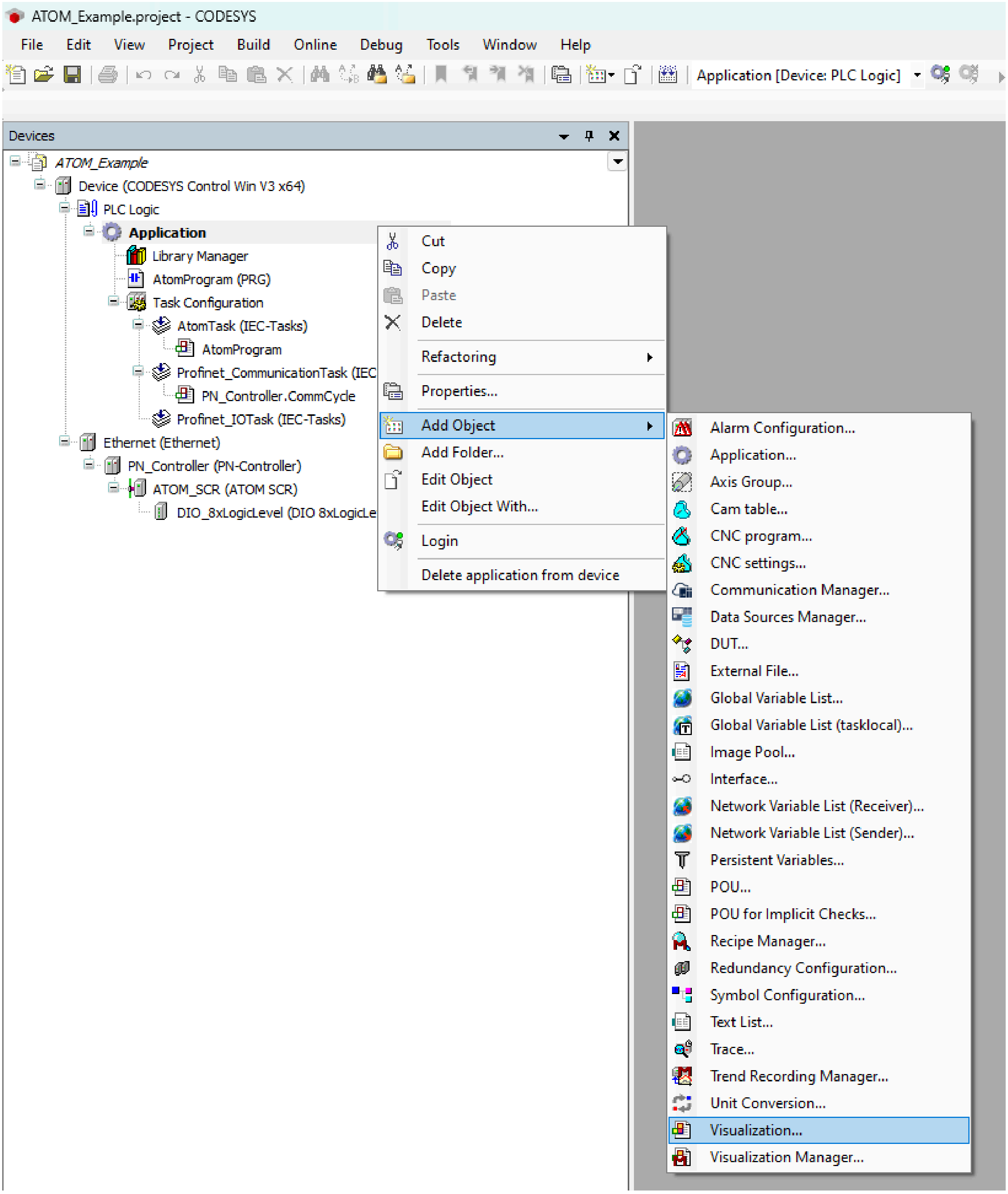
Next, double click **AtomTask (IEC-Tasks)** to open its configuration tab. Click **Add Call** and select **Application > AtomProgram**. After doing so, AtomTask's configuration should look like:



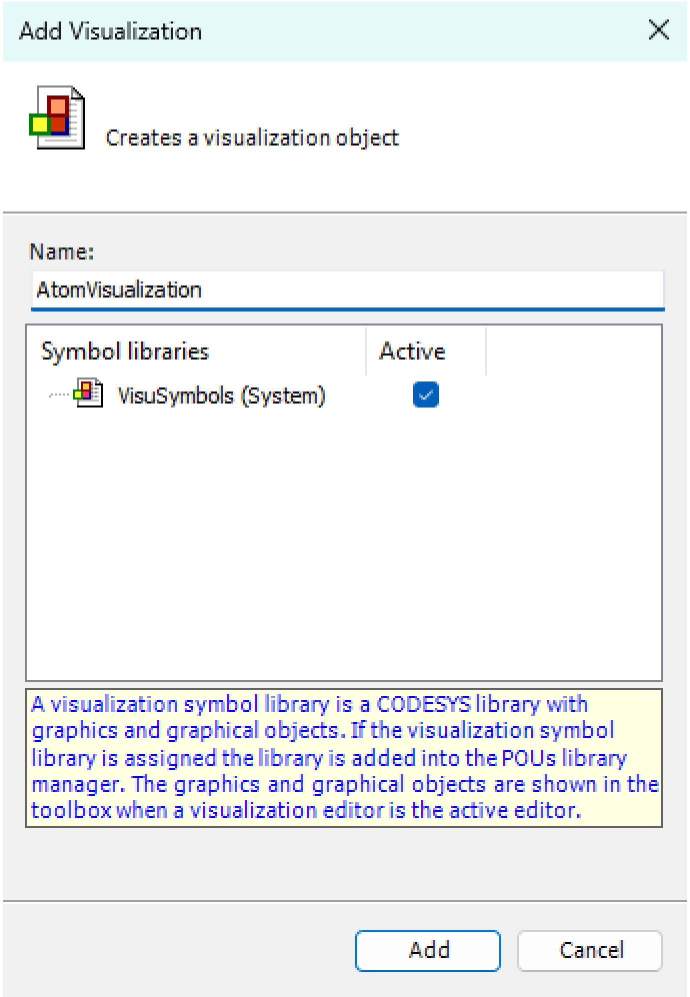
Setting up visualization

Next, we'll set up a simple visualization display to control and monitor ATOM.

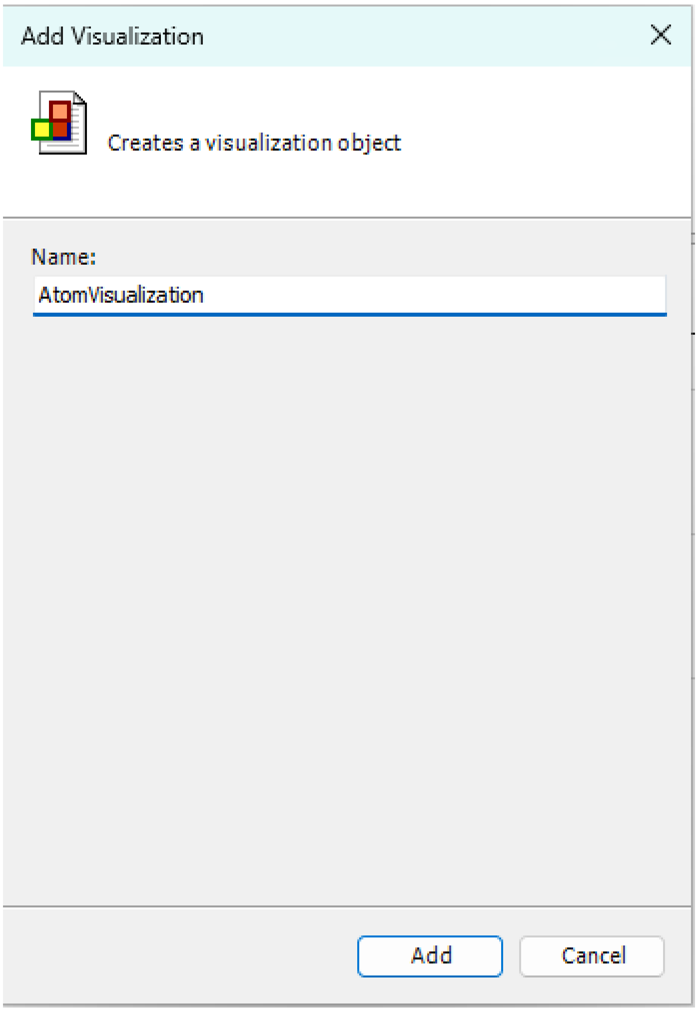
Right click **Application** and select **Add Object > Visualization**:



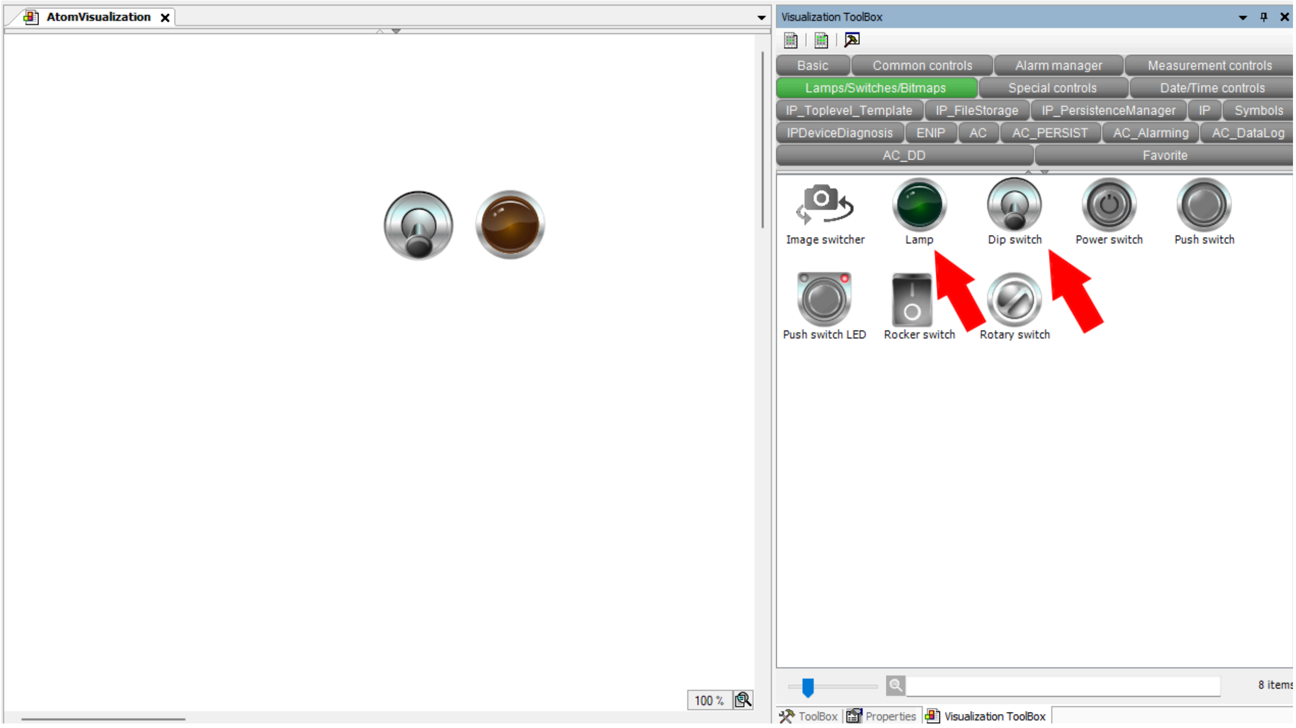
Make sure to check **Active** for **VisuSymbols (System)**, then click **Add**:



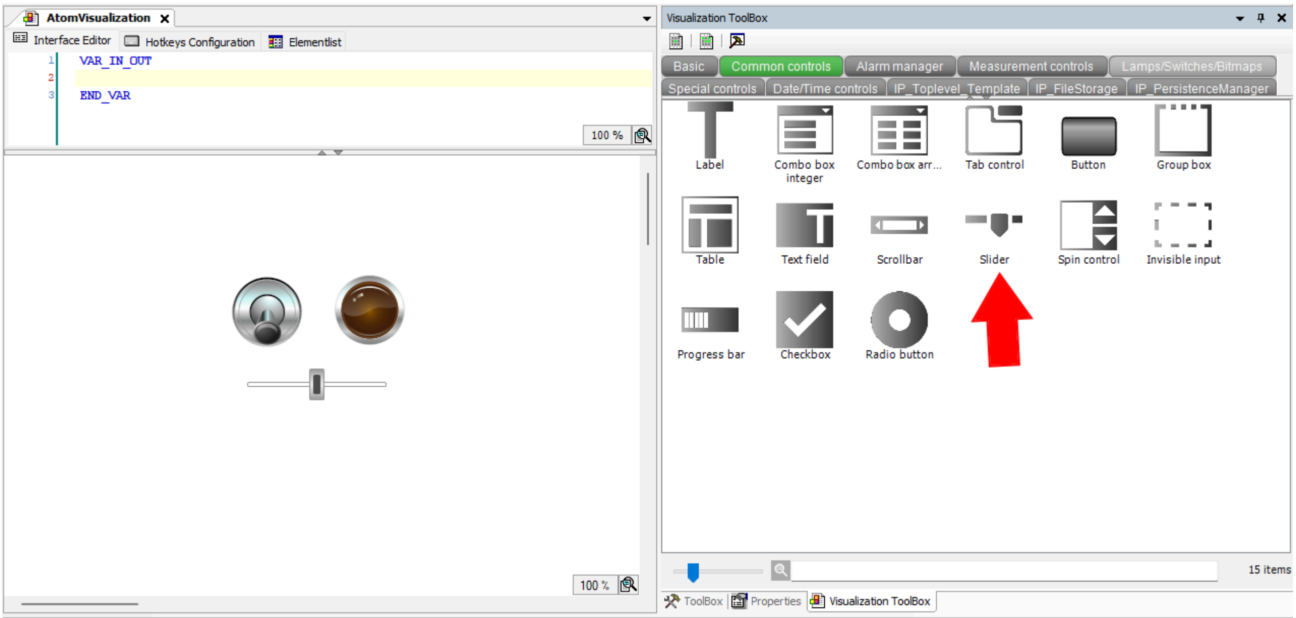
Name your visualization `AtomVisualization` and click **Add**:



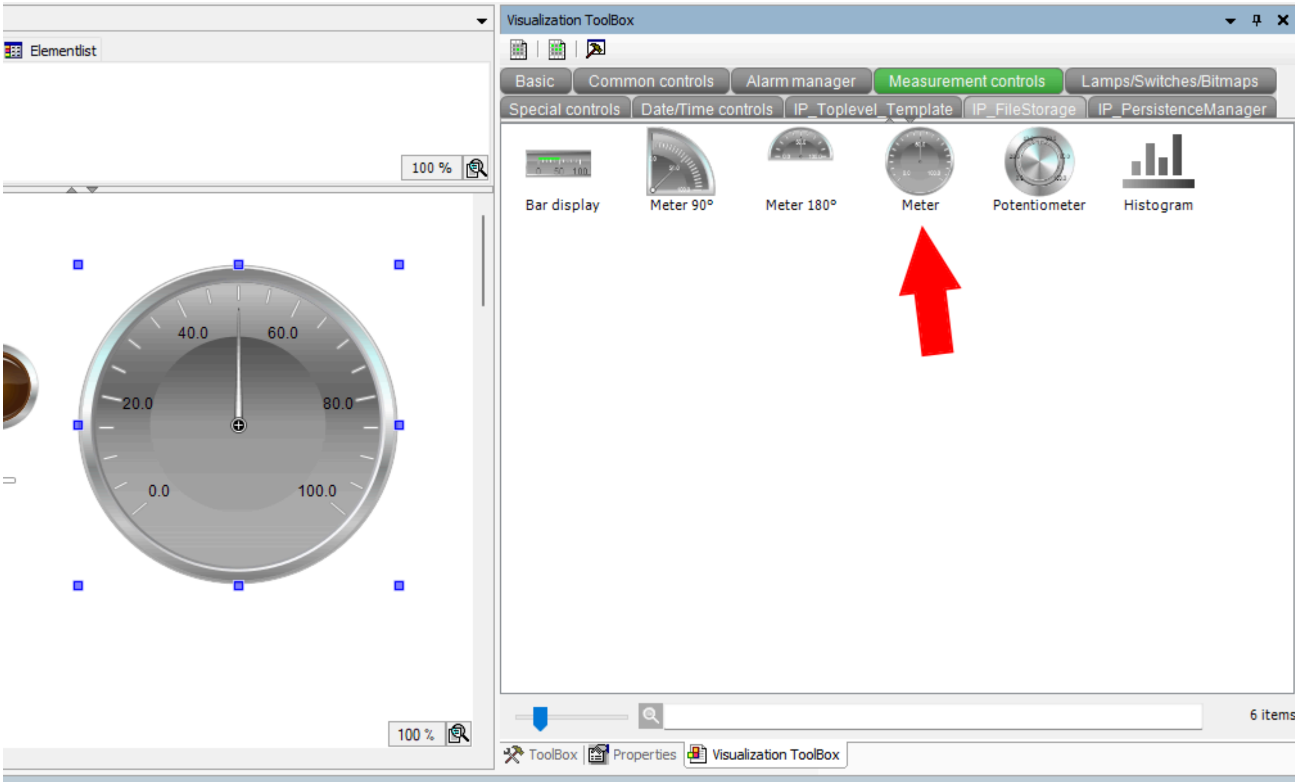
Double click **AtomVisualization** to open its configuration editor. From the **Visualization ToolBox** panel on the right, select the **Lamps/Switches/Bitmaps** category and add a lamp and a dip switch:



Next, in the **Common controls** category, add a slider:

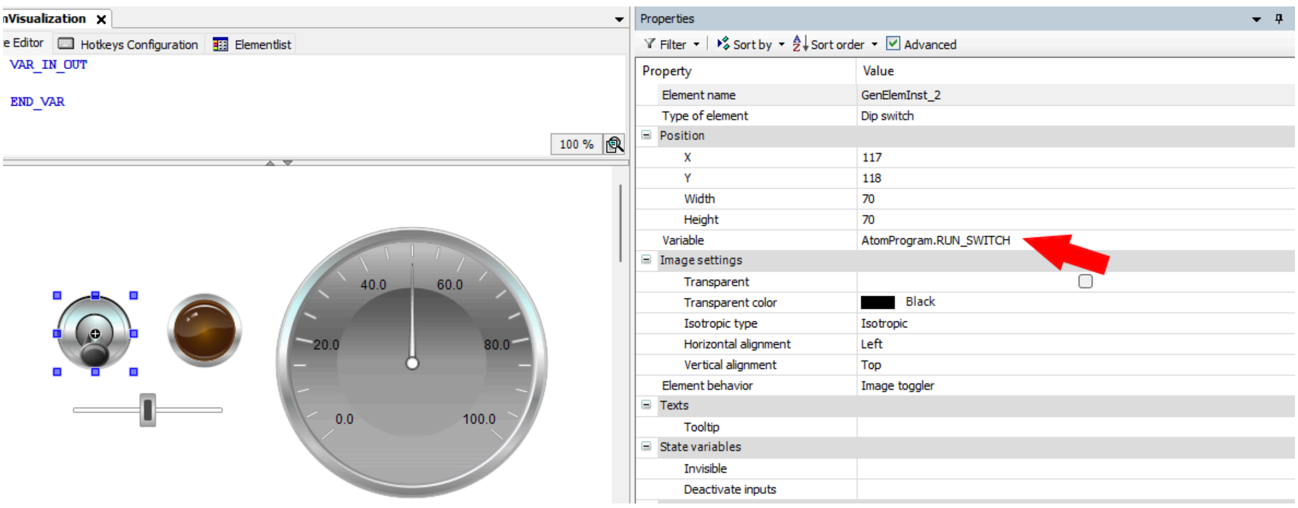


Finally, in the **Measurement controls** category, add a meter:

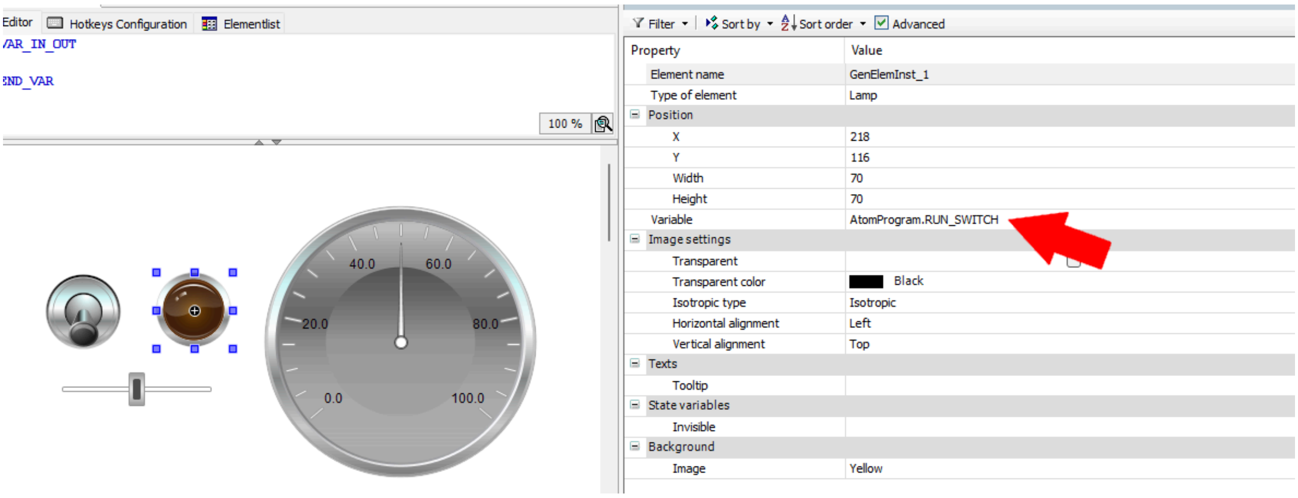


Wiring up the controls

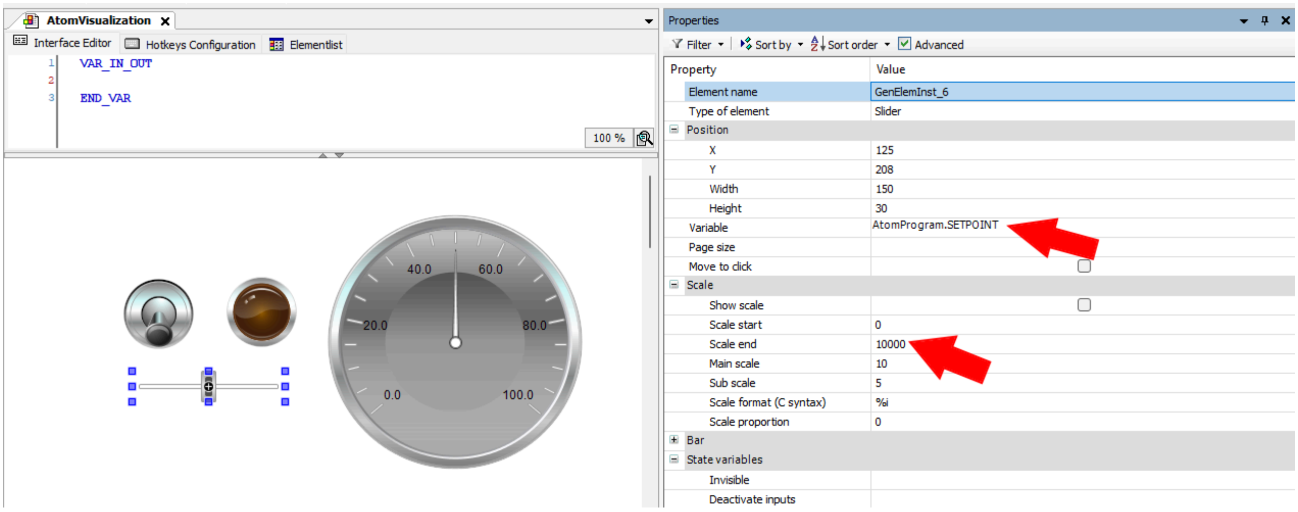
Next, we'll connect the controls to our PLC program. Select the dip switch and set the **Variable** field to `AtomProgram.RUN_SWITCH` as indicated by the red arrow:



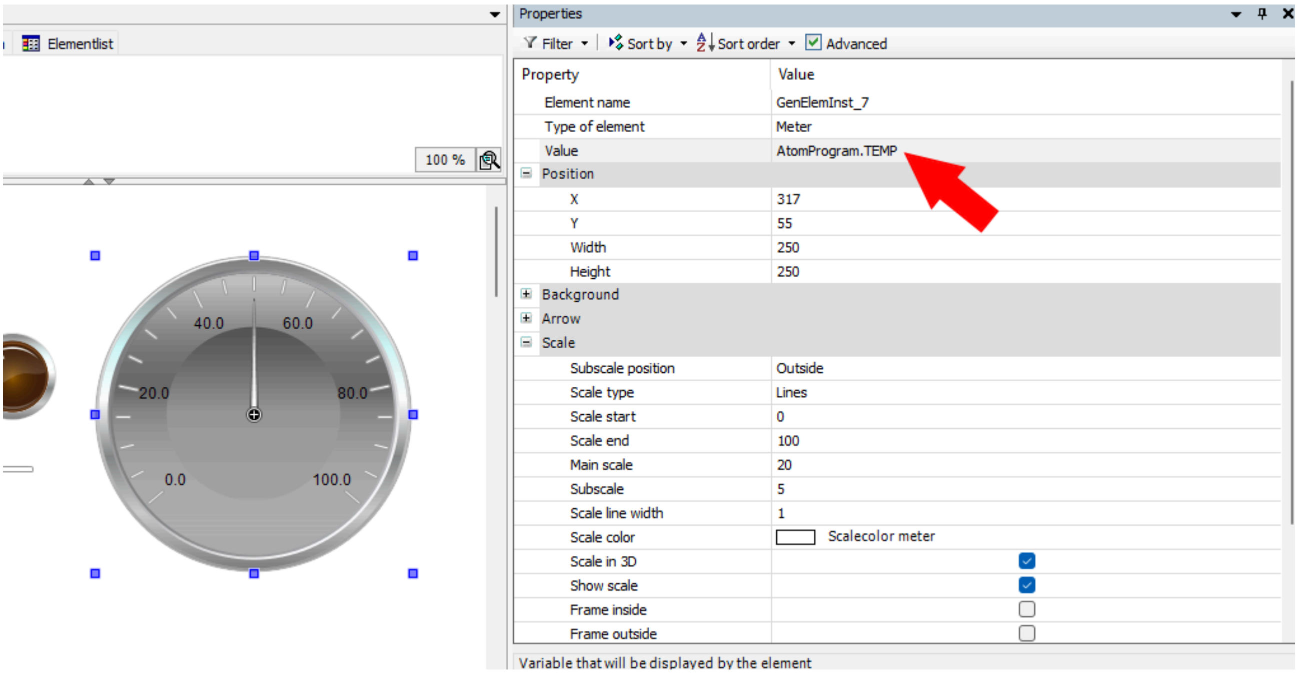
Select the lamp and set the **Variable** field to `AtomProgram.RUN_SWITCH` as indicated by the red arrow:



Select the slider and set the **Variable** field to `AtomProgram.SETPOINT` and set **Scale end** to `10000`:

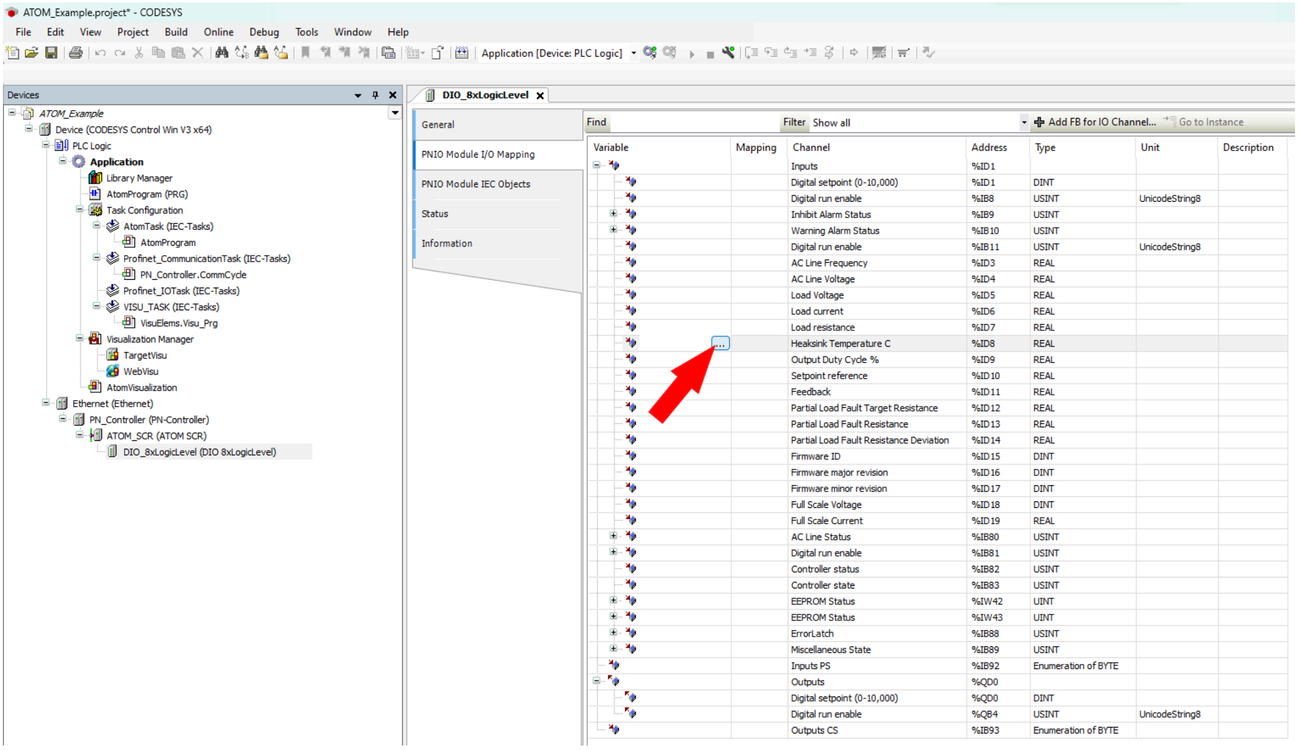


Select the meter and set the **Variable** field to `AtomProgram.TEMP`:

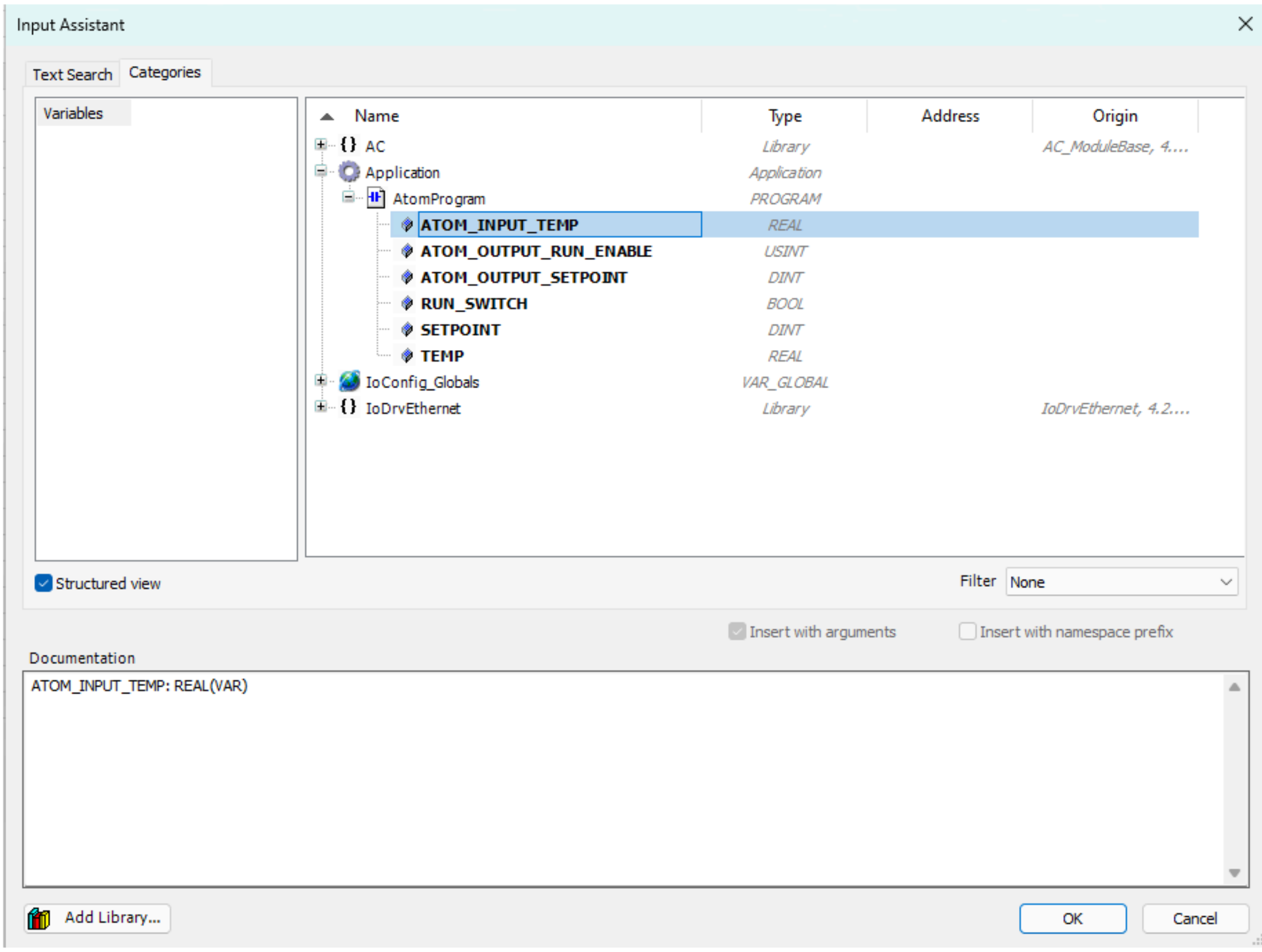


Mapping variables

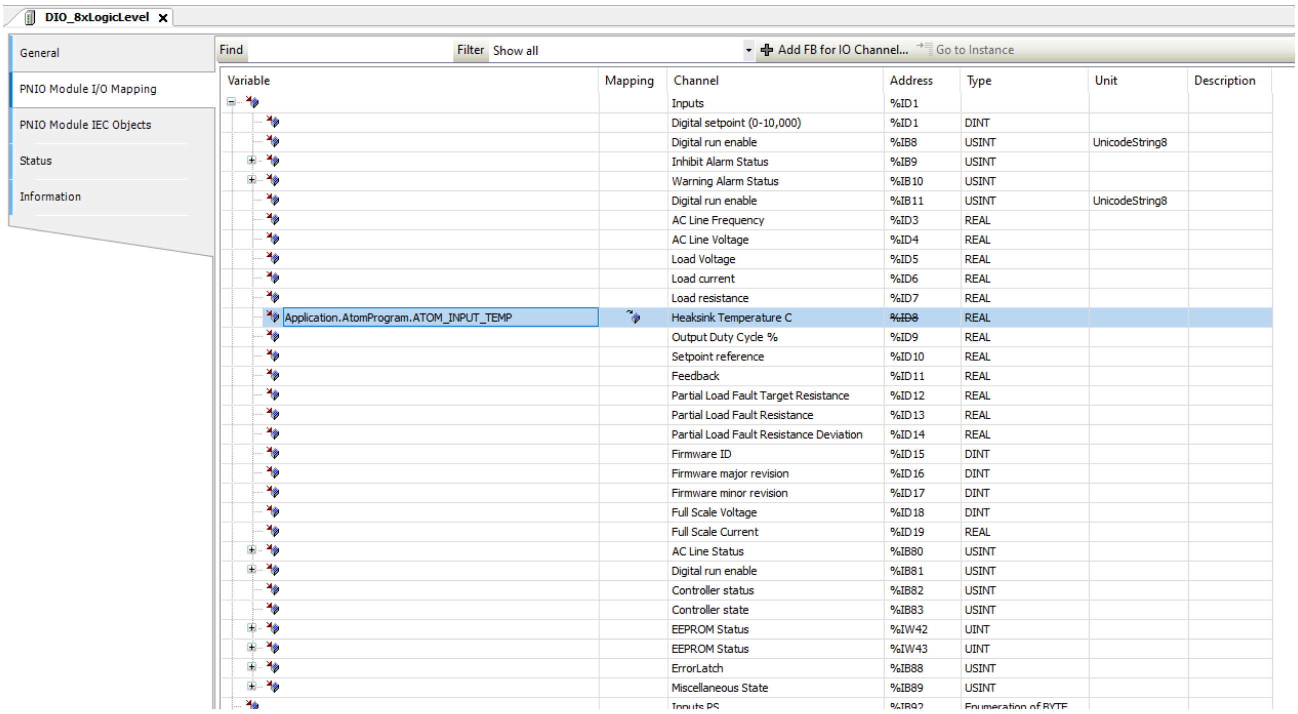
Finally, we'll map our PLC variables to ATOM. Double click **DIO_8xLogicLevel (DIO 8xLogicLevel)** in the device tree to open its configuration window. Select the **PNIO Module I/O Mapping** tab:



Above, select the button indicated by the red arrow. This will open the **Input Assistant** dialog. Select **Application > AtomProgram > ATOM_INPUT_TEMP** and click **Add**:



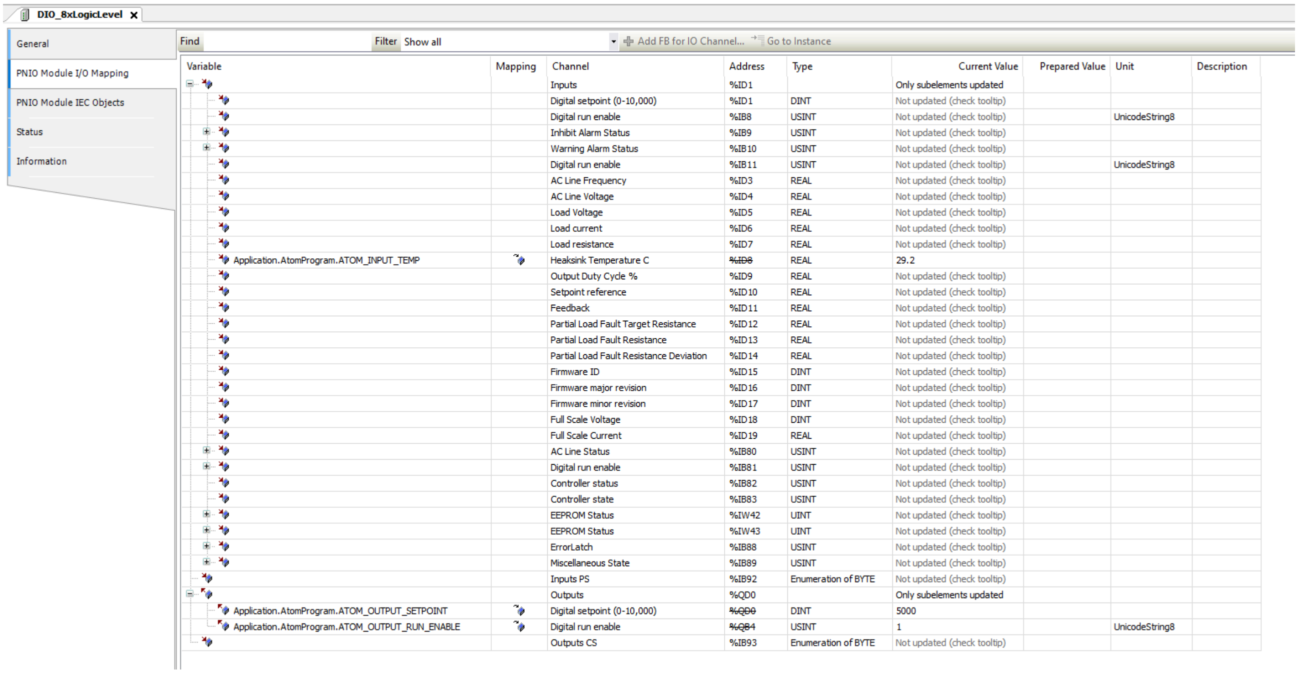
After doing so, your input I/O mappings should look like:



Repeat this for your output I/O mappings:

1. Map **Digital setpoint** to `Application.AtomProgram.ATOM_OUTPUT_SETPOINT`
2. Map **Digital run enable** to `Application.AtomProgram.ATOM_OUTPUT_RUN_ENABLE`

Change the **Filter** to **Show only outputs** and repeat the process for the outputs. Map **Digital setpoint** to `Application.AtomProgram.ATOM_OUTPUT_SETPOINT` and **Digital RUN Enable** to `Application.AtomProgram.ATOM_OUTPUT_RUN_ENABLE`.



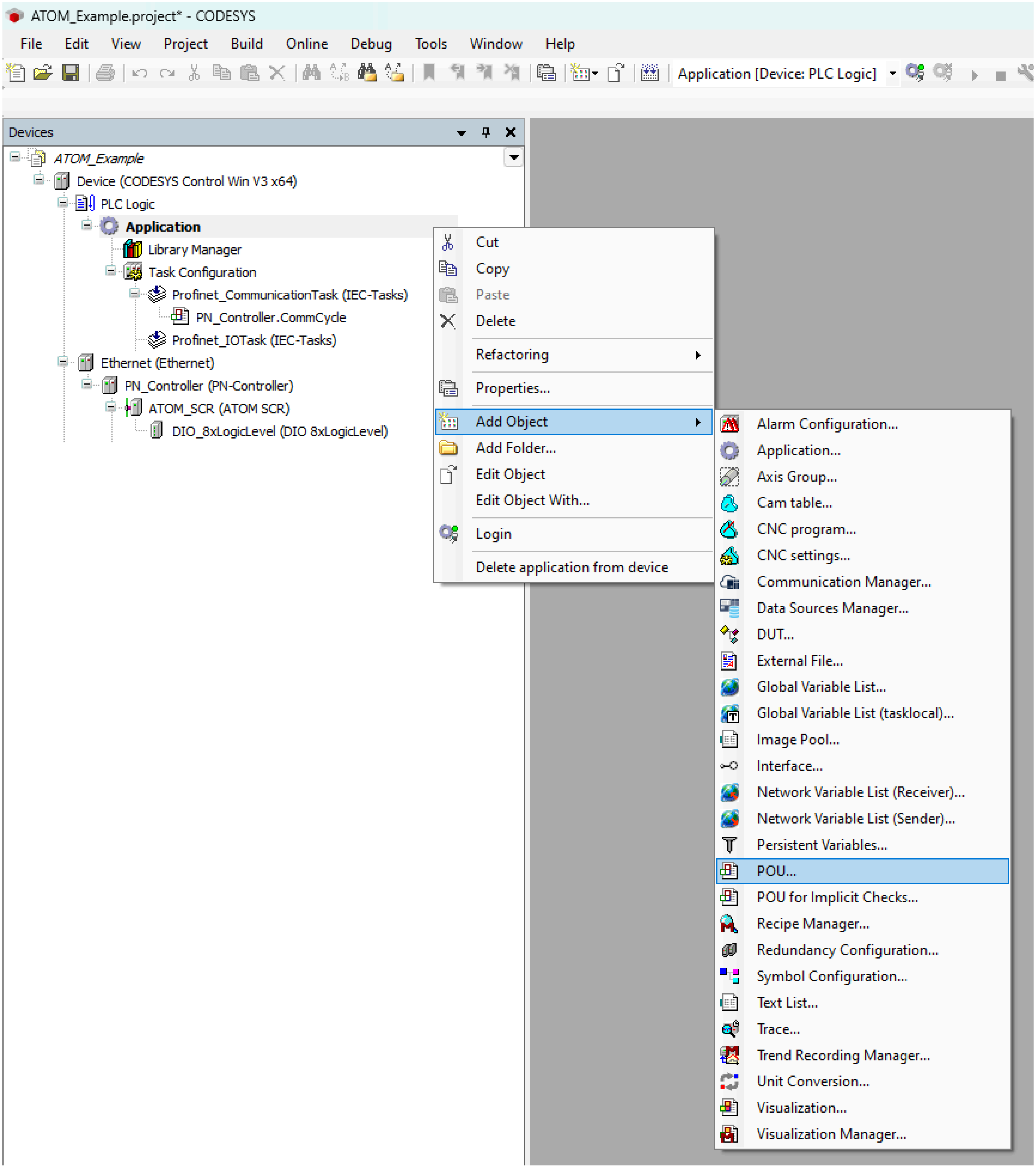
Variable	Mapping	Channel	Address	Type	Current Value	Prepared Value	Unit	Description
		Inputs	%ID1		Only subelements updated			
		Digital setpoint (0-10,000)	%ID1	DINT	Not updated (check tooltip)			
		Digital run enable	%IB8	USINT	Not updated (check tooltip)		UnicodeString8	
		Inhibit Alarm Status	%IB9	USINT	Not updated (check tooltip)			
		Warning Alarm Status	%IB10	USINT	Not updated (check tooltip)			
		Digital run enable	%IB11	USINT	Not updated (check tooltip)		UnicodeString8	
		AC Line Frequency	%ID3	REAL	Not updated (check tooltip)			
		AC Line Voltage	%ID4	REAL	Not updated (check tooltip)			
		Load Voltage	%ID5	REAL	Not updated (check tooltip)			
		Load current	%ID6	REAL	Not updated (check tooltip)			
		Load resistance	%ID7	REAL	Not updated (check tooltip)			
Application.AtomProgram.ATOM_INPUT_TEMP		Heaksink Temperature C	~~%ID8~~	REAL	29.2			
		Output Duty Cycle %	%ID9	REAL	Not updated (check tooltip)			
		Setpoint reference	%ID10	REAL	Not updated (check tooltip)			
		Feedback	%ID11	REAL	Not updated (check tooltip)			
		Partial Load Fault Target Resistance	%ID12	REAL	Not updated (check tooltip)			
		Partial Load Fault Resistance	%ID13	REAL	Not updated (check tooltip)			
		Partial Load Fault Resistance Deviation	%ID14	REAL	Not updated (check tooltip)			
		Firmware ID	%ID15	DINT	Not updated (check tooltip)			
		Firmware major revision	%ID16	DINT	Not updated (check tooltip)			
		Firmware minor revision	%ID17	DINT	Not updated (check tooltip)			
		Full Scale Voltage	%ID18	DINT	Not updated (check tooltip)			
		Full Scale Current	%ID19	REAL	Not updated (check tooltip)			
		AC Line Status	%IB80	USINT	Not updated (check tooltip)			
		Digital run enable	%IB81	USINT	Not updated (check tooltip)			
		Controller status	%IB82	USINT	Not updated (check tooltip)			
		Controller state	%IB83	USINT	Not updated (check tooltip)			
		EEPROM Status	%IW42	UINT	Not updated (check tooltip)			
		EEPROM Status	%IW43	UINT	Not updated (check tooltip)			
		ErrorLatch	%IB88	USINT	Not updated (check tooltip)			
		Miscellaneous State	%IB89	USINT	Not updated (check tooltip)			
		Inputs PS	%IB92	Enumeration of BYTE	Not updated (check tooltip)			
		Outputs	%QD0		Only subelements updated			
Application.AtomProgram.ATOM_OUTPUT_SETPOINT		Digital setpoint (0-10,000)	~~%QD0~~	DINT	5000			
Application.AtomProgram.ATOM_OUTPUT_RUN_ENABLE		Digital run enable	~~%QB4~~	USINT	1		UnicodeString8	
		Outputs CS	%IB93	Enumeration of BYTE	Not updated (check tooltip)			

You're all set! Go to the Running the program with SoftPLC section to run your program.

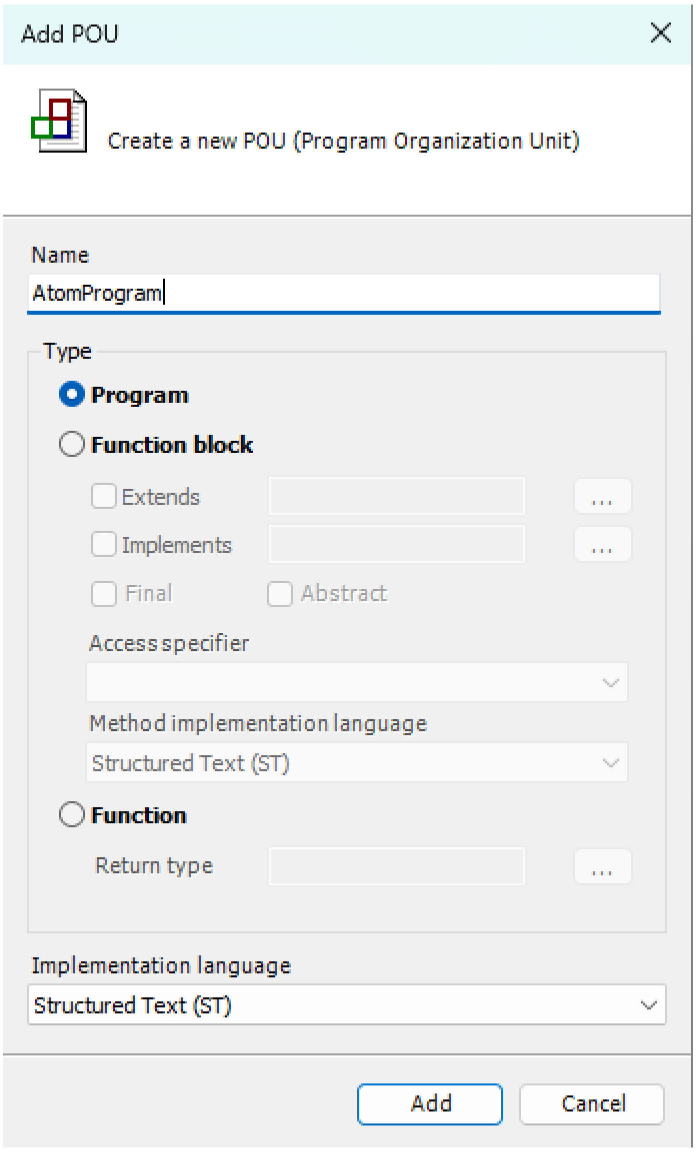
Example: Structured text

Creating the program

Right click **Application** and select **Add Object > POU**:



Name your **POU** `AtomProgram` and select **Structured Text (ST)** as the language:



Next, let's create a basic program. We'll check to make sure no alarms are active and then write a setpoint value of `8000` and set run enable to `true`.

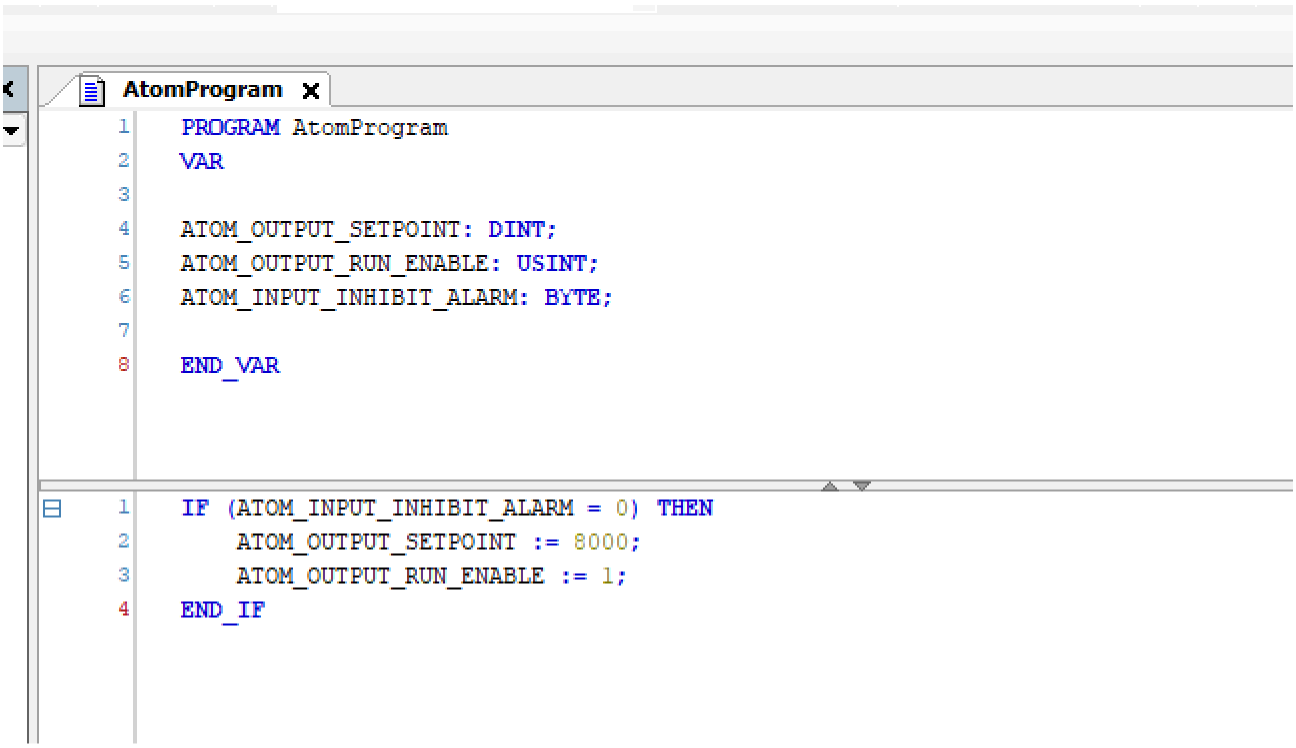
Copy the following code into the top panel of the **AtomProgram** editor:

```
PROGRAM AtomProgram
VAR

ATOM_OUTPUT_SETPOINT: DINT;
ATOM_OUTPUT_RUN_ENABLE: USINT;
ATOM_INPUT_INHIBIT_ALARM: BYTE;

END_VAR
```

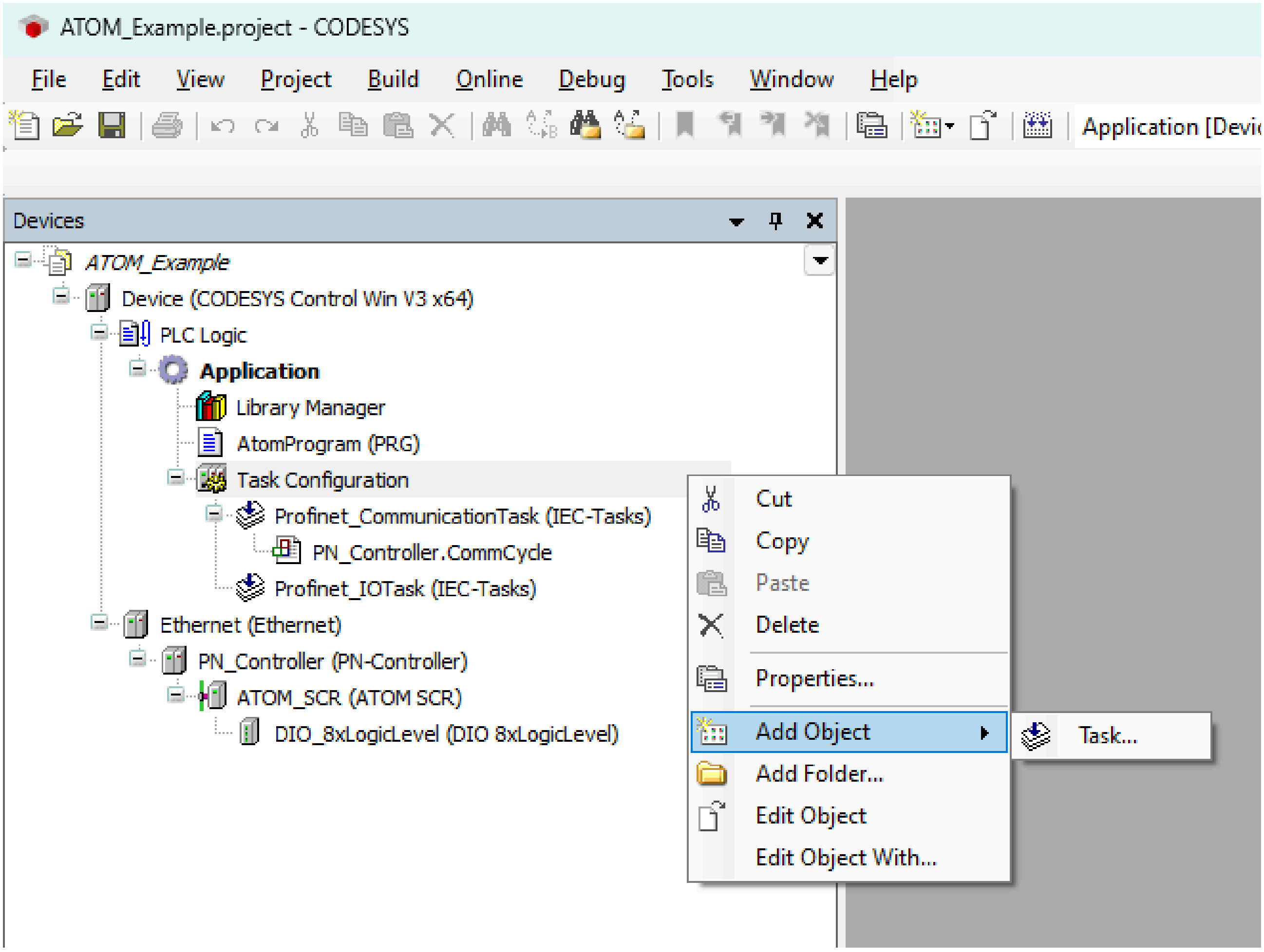
Copy the following code into the main program section:

```
IF (ATOM_INPUT_INHIBIT_ALARM = 0) THEN
        ATOM_OUTPUT_SETPOINT := 8000;
        ATOM_OUTPUT_RUN_ENABLE := 1;
END_IF
```

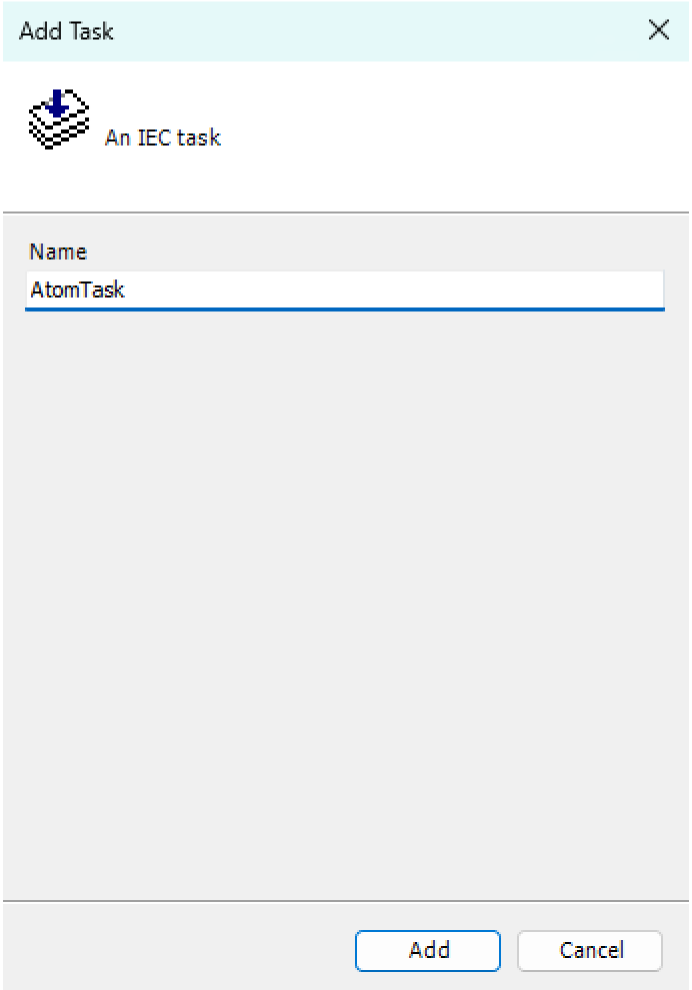
Your editor should look like:



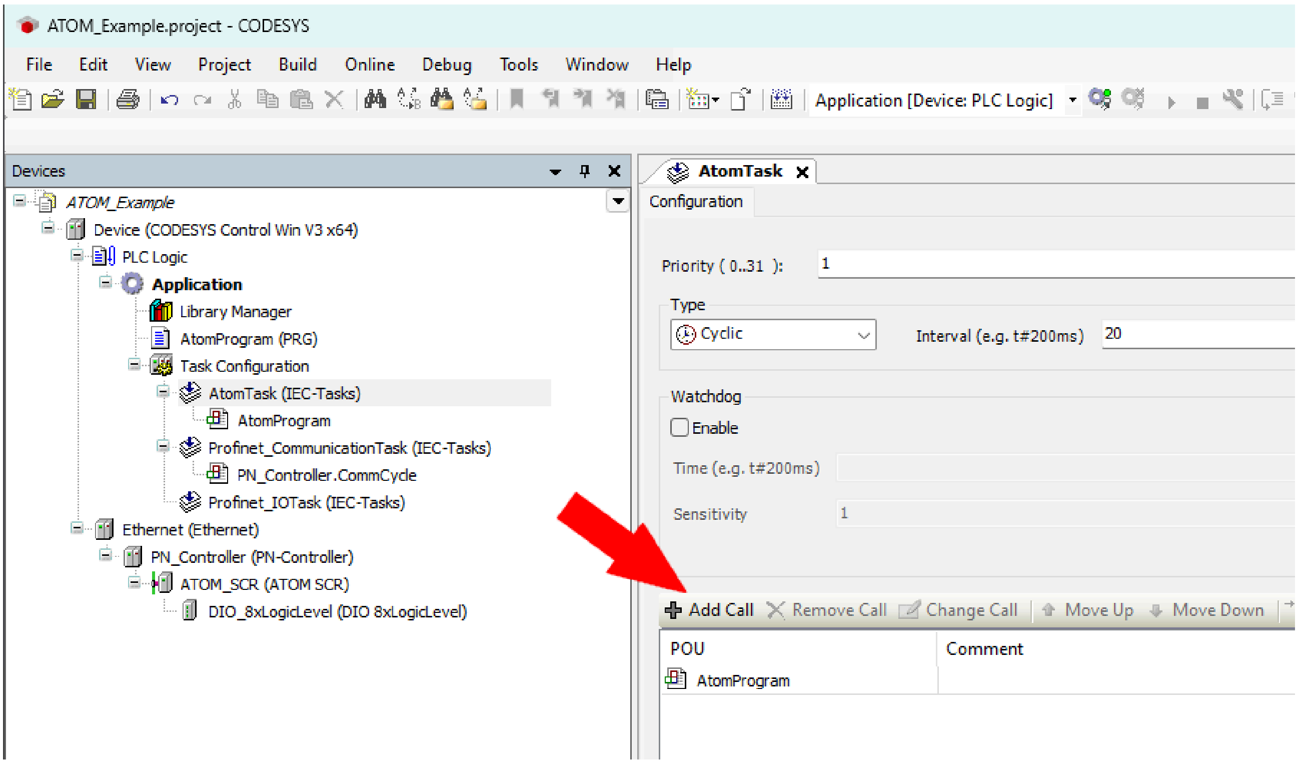
Next, we'll add a new task to call our program. Right click **Task Configuration** and Select **Add Object > Task**:



Name your task `AtomTask` and click **Add**:

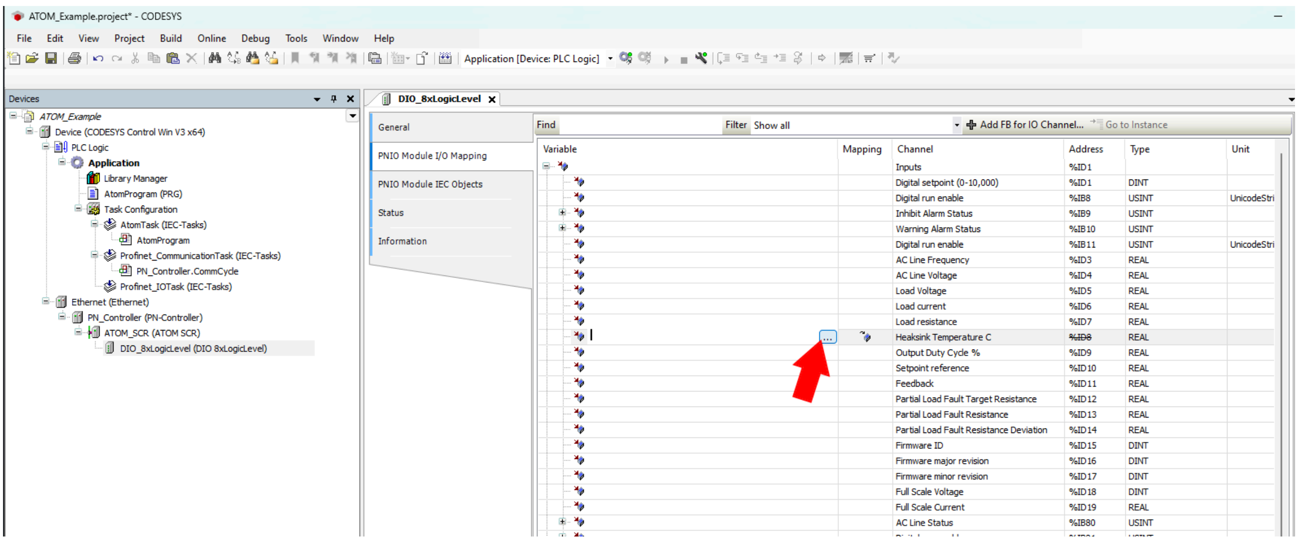


Next, double click **AtomTask (IEC-Tasks)** to open its configuration tab. Click **Add Call** and select **Application > AtomProgram**. After doing so, **AtomTask**'s configuration should look like:

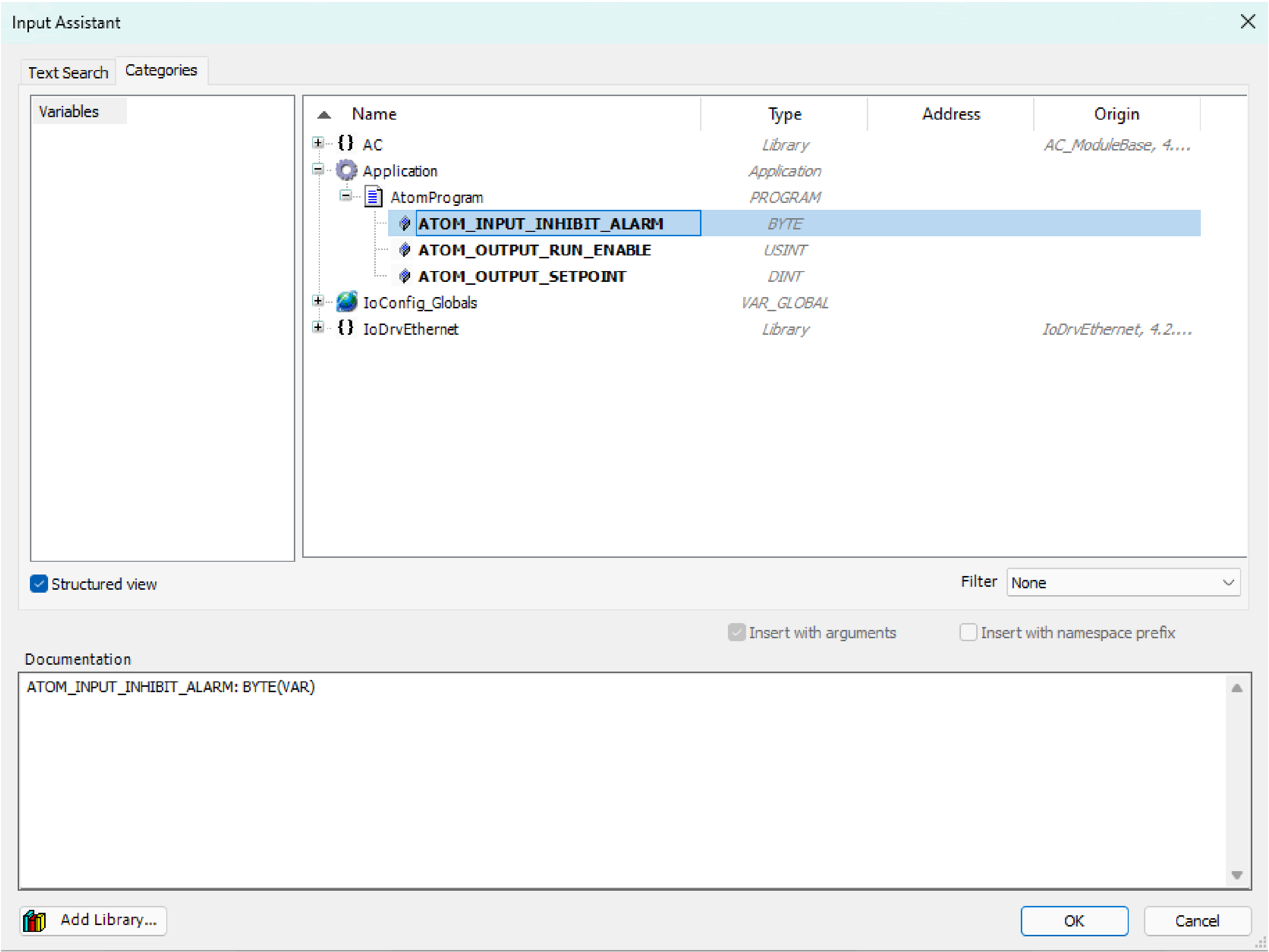


Mapping variables

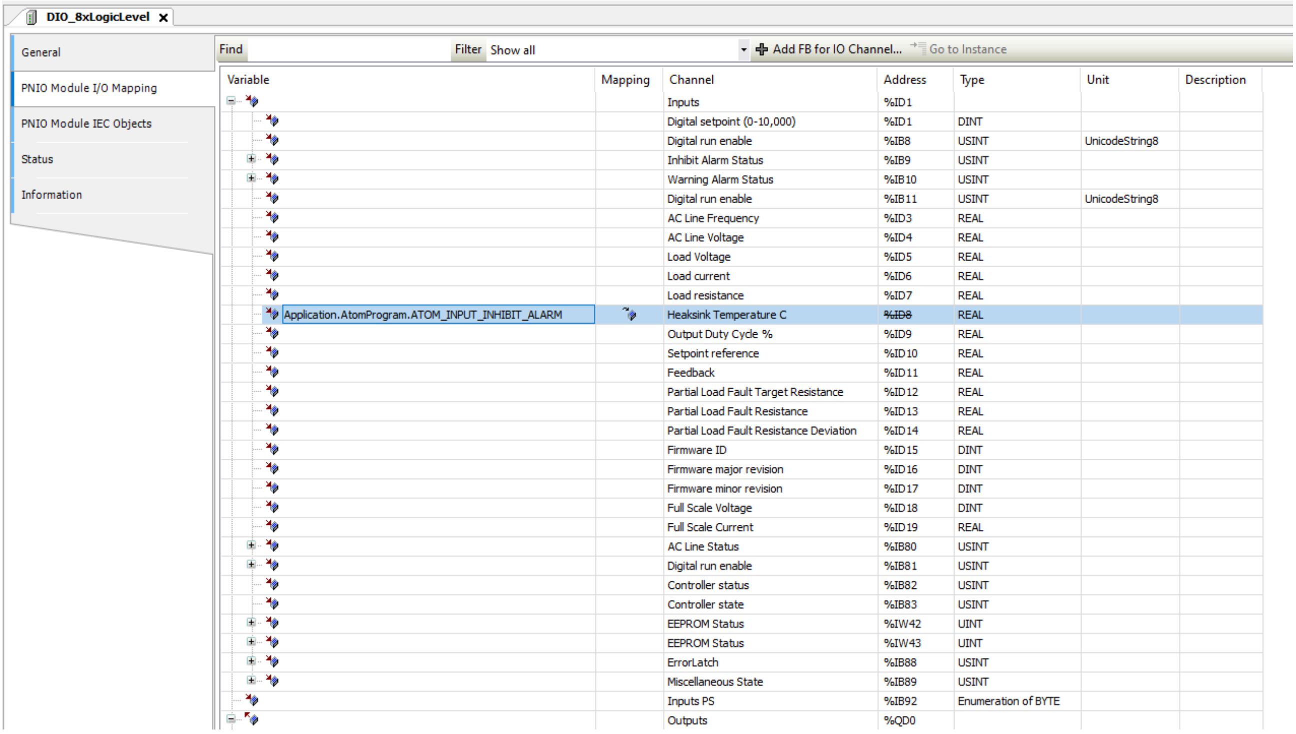
Finally, we'll map our PLC variables to ATOM. Double click **DIO_8xLogicLevel (DIO 8xLogicLevel)** in the device tree to open its configuration window. Select the **PNIO Module I/O Mapping** tab:



Above, select the button indicated by the red arrow. This will open the **Input Assistant** dialog. Select **Application > AtomProgram > ATOM_INPUT_INHIBIT_ALARM** and click **Add**:



After doing so, your input I/O mappings should look like:

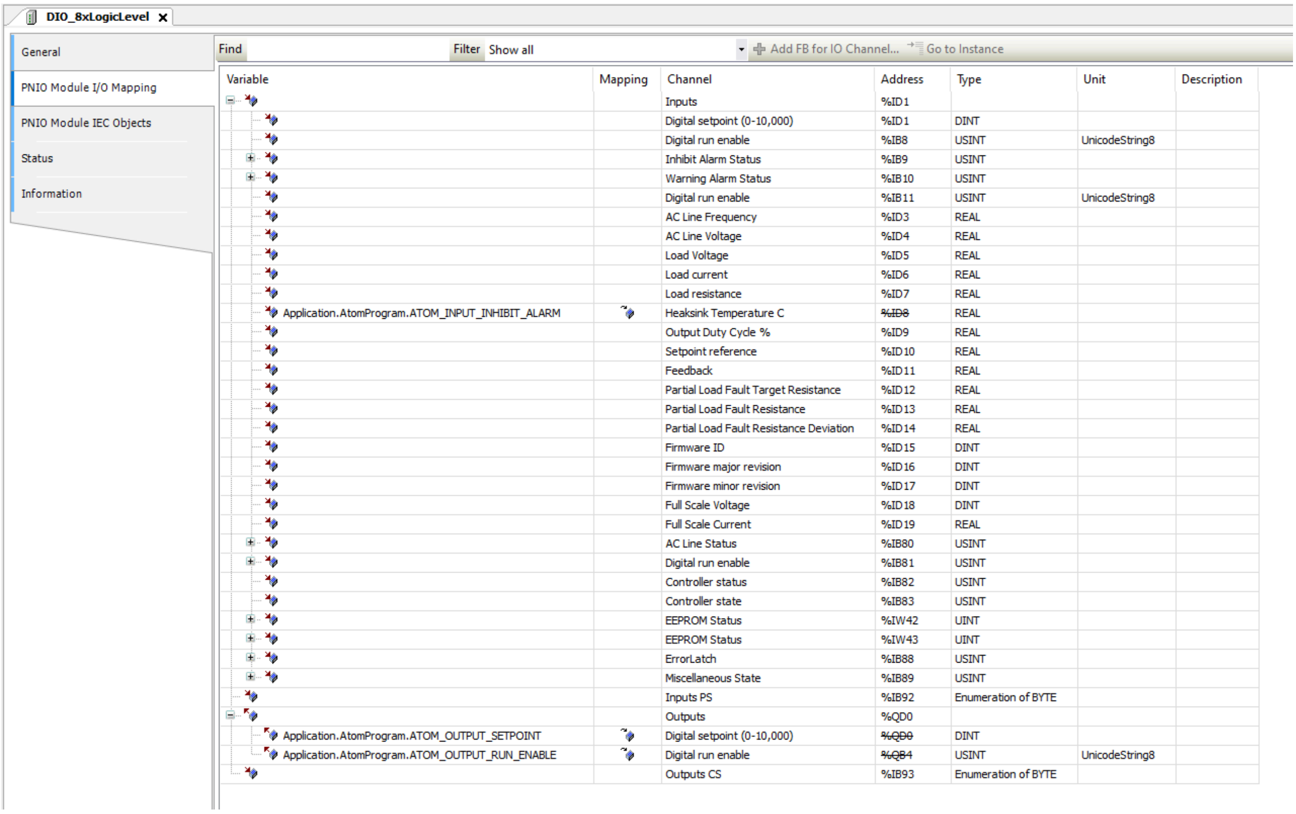


Variable	Mapping	Channel	Address	Type	Unit	Description
		Inputs	%ID1			
		Digital setpoint (0-10,000)	%ID1	DINT		
		Digital run enable	%IB8	USINT	UnicodeString8	
		Inhibit Alarm Status	%IB9	USINT		
		Warning Alarm Status	%IB10	USINT		
		Digital run enable	%IB11	USINT	UnicodeString8	
		AC Line Frequency	%ID3	REAL		
		AC Line Voltage	%ID4	REAL		
		Load Voltage	%ID5	REAL		
		Load current	%ID6	REAL		
		Load resistance	%ID7	REAL		
Application.AtomProgram.ATOM_INPUT_INHIBIT_ALARM		Heaksink Temperature C	~~%ID8~~	REAL		
		Output Duty Cycle %	%ID9	REAL		
		Setpoint reference	%ID10	REAL		
		Feedback	%ID11	REAL		
		Partial Load Fault Target Resistance	%ID12	REAL		
		Partial Load Fault Resistance	%ID13	REAL		
		Partial Load Fault Resistance Deviation	%ID14	REAL		
		Firmware ID	%ID15	DINT		
		Firmware major revision	%ID16	DINT		
		Firmware minor revision	%ID17	DINT		
		Full Scale Voltage	%ID18	DINT		
		Full Scale Current	%ID19	REAL		
		AC Line Status	%IB80	USINT		
		Digital run enable	%IB81	USINT		
		Controller status	%IB82	USINT		
		Controller state	%IB83	USINT		
		EEPROM Status	%IW42	UINT		
		EEPROM Status	%IW43	UINT		
		ErrorLatch	%IB88	USINT		
		Miscellaneous State	%IB89	USINT		
		Inputs PS	%IB92	Enumeration of BYTE		
		Outputs	%QD0			

Repeat this for your output I/O mappings:

1. Map **Digital setpoint** to `Application.AtomProgram.ATOM_OUTPUT_SETPOINT`
2. Map **Digital run enable** to `Application.AtomProgram.ATOM_OUTPUT_RUN_ENABLE`

Change the **Filter** to **Show only outputs** and repeat the process for the outputs. Map **Digital setpoint** to `Application.AtomProgram.ATOM_OUTPUT_SETPOINT` and **Digital RUN Enable** to `Application.AtomProgram.ATOM_OUTPUT_RUN_ENABLE`.



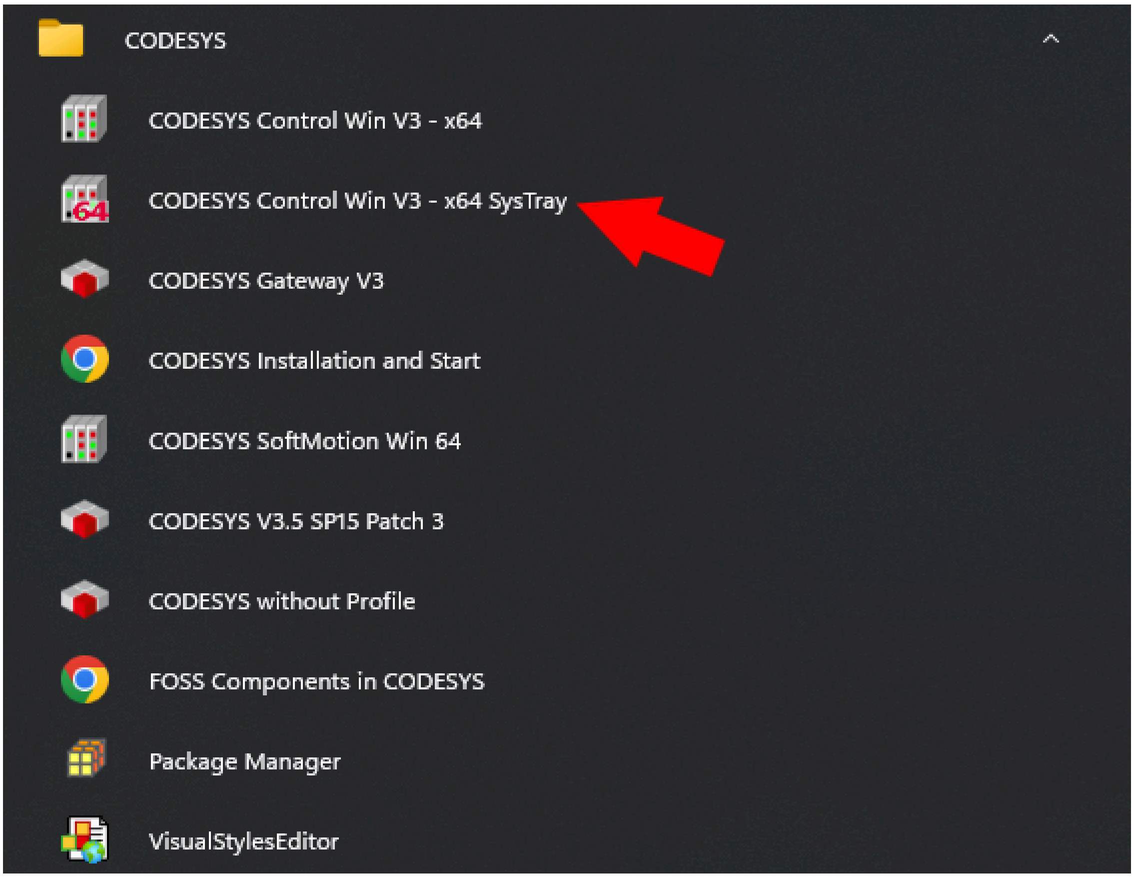
Variable	Mapping	Channel	Address	Type	Unit	Description
		Inputs	%ID1			
		Digital setpoint (0-10,000)	%ID1	DINT		
		Digital run enable	%IB8	USINT	UnicodeString8	
		Inhibit Alarm Status	%IB9	USINT		
		Warning Alarm Status	%IB10	USINT		
		Digital run enable	%IB11	USINT	UnicodeString8	
		AC Line Frequency	%ID3	REAL		
		AC Line Voltage	%ID4	REAL		
		Load Voltage	%ID5	REAL		
		Load current	%ID6	REAL		
		Load resistance	%ID7	REAL		
Application.AtomProgram.ATOM_INPUT_INHIBIT_ALARM		Heaksink Temperature C	~~%ID8~~	REAL		
		Output Duty Cycle %	%ID9	REAL		
		Setpoint reference	%ID10	REAL		
		Feedback	%ID11	REAL		
		Partial Load Fault Target Resistance	%ID12	REAL		
		Partial Load Fault Resistance	%ID13	REAL		
		Partial Load Fault Resistance Deviation	%ID14	REAL		
		Firmware ID	%ID15	DINT		
		Firmware major revision	%ID16	DINT		
		Firmware minor revision	%ID17	DINT		
		Full Scale Voltage	%ID18	DINT		
		Full Scale Current	%ID19	REAL		
		AC Line Status	%IB80	USINT		
		Digital run enable	%IB81	USINT		
		Controller status	%IB82	USINT		
		Controller state	%IB83	USINT		
		EEPROM Status	%IW42	UINT		
		EEPROM Status	%IW43	UINT		
		ErrorLatch	%IB88	USINT		
		Miscellaneous State	%IB89	USINT		
		Inputs PS	%IB92	Enumeration of BYTE		
		Outputs	%QD0			
Application.AtomProgram.ATOM_OUTPUT_SETPOINT		Digital setpoint (0-10,000)	~~%QD0~~	DINT		
Application.AtomProgram.ATOM_OUTPUT_RUN_ENABLE		Digital run enable	~~%QB4~~	USINT	UnicodeString8	
		Outputs CS	%IB93	Enumeration of BYTE		

You're all set! Go to the Running the program with SoftPLC section to run your program.

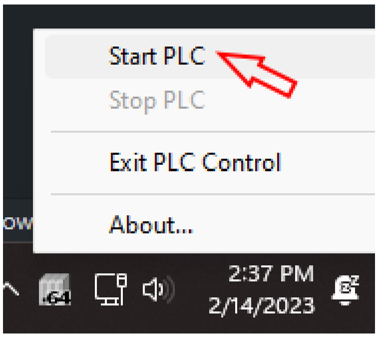
Running the program with SoftPLC

> **INFO**
>
> The instructions to run your program are the same regardless of whether you are using ladder logic or structured text.
>
> The only difference is that in the ladder logic example, a visualization window will open that allows you to control ATOM.

To debug the program, first make sure you start **Codesys WIN Control V3 - x64 SysTray**

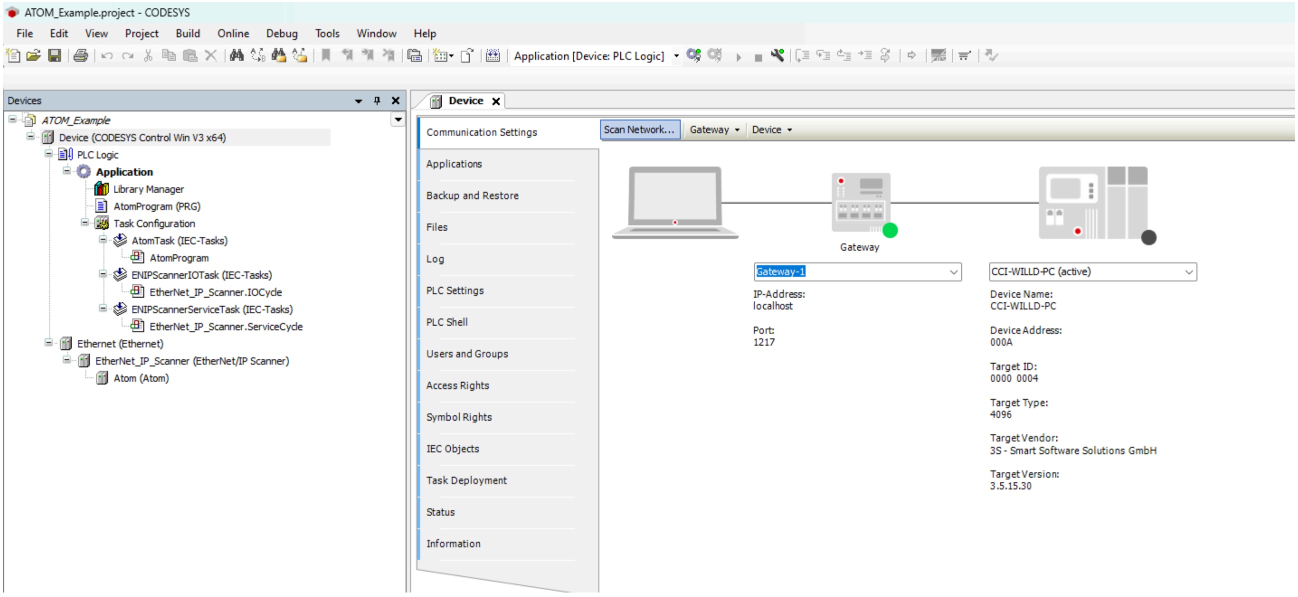


This will launch the Codesys SoftPLC. You should see an icon appear in your systray and you can right click it and select **Start PLC** to start the SoftPLC:

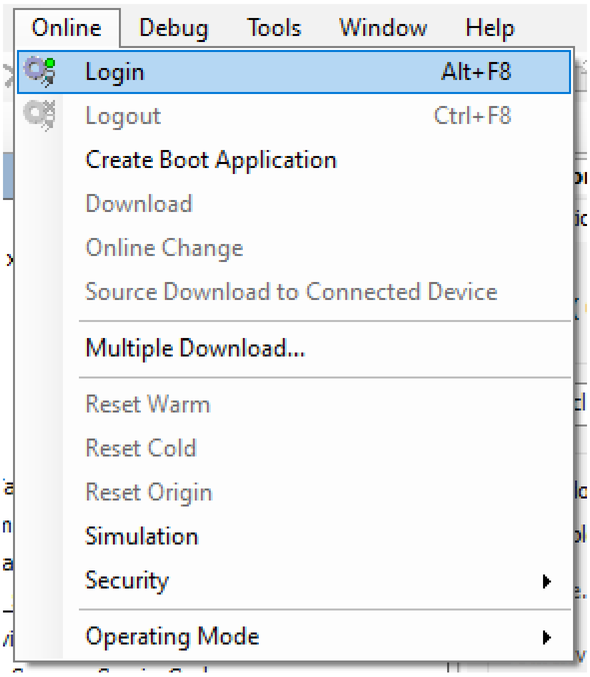


Next, connect your Atom to your PC via an Ethernet cable, ensuring to use the network interface you specified in the Adding a Profinet controller section.

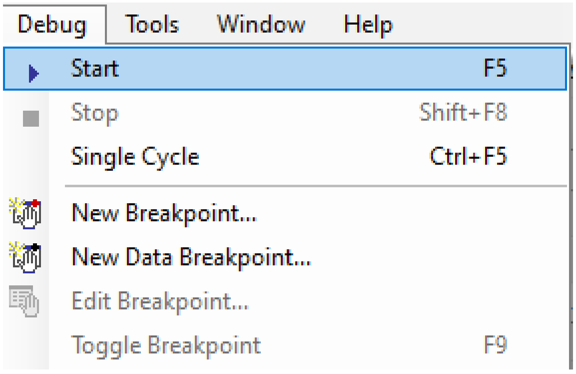
Next, in Codesys double click **Application** to open its configuration window. Here you can select **Scan Network** to discover your SoftPLC:



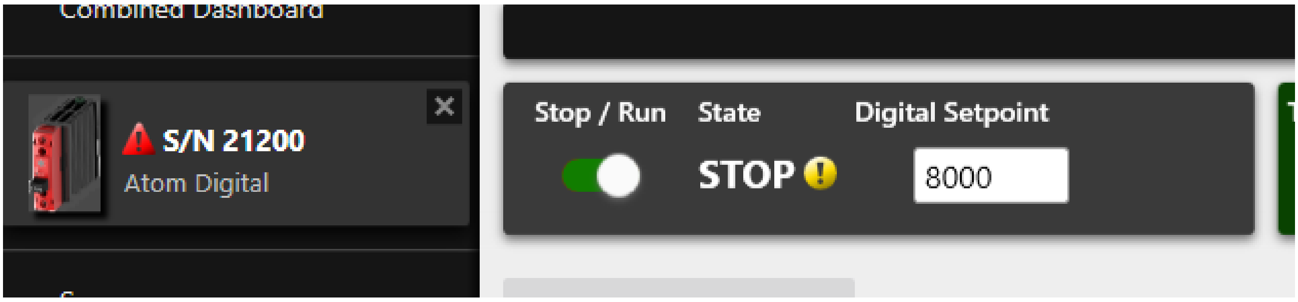
Finally, **Login** to your SoftPLC:



Then you can start debugging the program:



If you use Control Panel to monitor ATOM, you should see the **Stop / Run** state and the **Digital Setpoint** values change to reflect the PLC program's instructions. If you followed the structured text example, the values will change once and remain fixed. If you followed the ladder logic example, a visualization control panel will appear. Flipping the dip switch or adjusting the slider will immediately update ATOM and the changes should reflect in real-time:



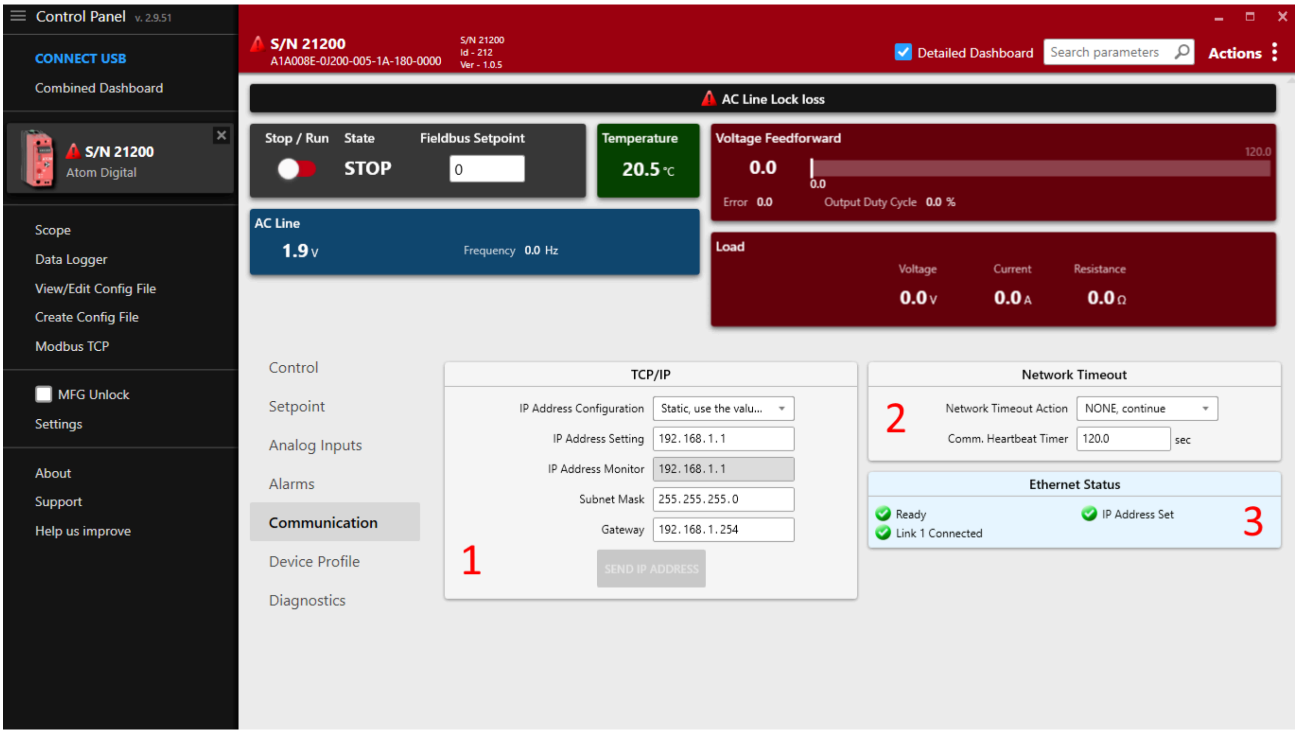
ATOM / Fieldbus / ModbusTCP / Overview

Atom supports the following ModbusTCP operations:

- Read Holding Registers (Function code `3`)
- Write Single Holding Register (Function code `6`)
- Write single coil/command (Function code `5`)

> **INFO**
>
> ModbusTCP is always available and running, even if you're using Profinet or EtherNet/IP. ModbusTCP exposes more parameters than the other fieldbus protocols and may be useful for more advanced configuration.

Control Panel Communication Settings



Some communication settings can be configured in the **Communication** tab in Control Panel.

- Section `1`: TCP/IP settings
  - **IP Address Configuration**
    - `Static`: Use the IP address, subnet mask, and gateway specified below.
    - `DHCP`: Use DHCP to obtain an IP address.
  - **IP Address Setting**: The IP address of the ATOM controller.
  - **IP Address Monitor**: The current IP address of the ATOM controller.
  - **Subnet Mask**: The subnet mask of the ATOM controller.
  - **Gateway**: The gateway address for the ATOM controller.
- Section `2`: Network Timeout
  - The EtherNet/IP heartbeat timeout (Encapsulation Inactivity Timeout) in seconds.
  - You can configure a network timeout action to perform when the device loses communication with the PLC:
    - `None`: Do nothing

        - `STOP, fault shutdown`: STOP the controller, disabling output
        - `Use network timeout setpoint`: Configure an alternative setpoint to use when the controller loses communication with the PLC.
- Section `3`: Ethernet status
    - Indicates the status of both RJ45 ports, IP address configuration, conflict detection, and any other errors with the EtherNet/IP connection.

> **INFO**
>
> ### Control Panel and PLC software
>
> These settings are synchronized with your PLC environment. You do not have to use Control Panel to change these settings - you can stay in your PLC software. Control Panel merely provides them as an altnerative way to configure ATOM's EtherNet/IP settings.
>
> You can use Control Panel simultaneously with your PLC software without issues.

> **WARNING**
>
> ### IP Address Conflict Detection
>
> ATOM uses **IP Address Conflict Detection** to detect IP address conflicts on the network. If ATOM detects another device using the same IP address, it will disable all network communication until the conflict is resolved.
>
> Please ensure all devices on the network are assigned unique a IP address.

Hardware considerations

⚠ **WARNING**

Daisy chaining

As ATOM has two RJ45 ports, it can be easily daisy-chained. When daisy-chaining ATOM, take care to avoid a loop in the network. In some loop configurations, ATOM is susceptible to network broadcast storms, which can cause the controller to become unresponsive. If you are daisy-chaining ATOM, ensure that the network is loop-free.

ATOM works with both unmanaged and managed switches. We recommend a managed switch for larger networks to give you more control over the network topology.

Registers

Register #	Name	Default	Min	Max	Scale	Unit
2	Feedback Type	1	1	2		
3	Firing Mode	5	1	6		
4	Slew Rate	10	1	100		
5	Control Loop	1	0	1		
6	Full Scale Voltage	480.0	10.0	600.0	10	v

Register #	Name	Default	Min	Max	Scale	Unit
7	Full Scale Current	80.0	2.0	100.0	10	A
8	Voltage Limit	700	10	700		v
9	Current Limit	84.0	1.0	105.0	10	A
10	Current Trip	240	5	245	1	A
11	Analog Setpoint Zero threshold	0	0	0		
12	Analog Setpoint Type	0	1	2		
13	Analog Setpoint Low Cmd	0.00	-5.00	25.00	100	v, ma
14	Analog Setpoint Low Out	0.00	0.00	125.00	100	%
15	Analog Setpoint High Cmd	0.00	-5.00	25.00	100	v,ma
16	Analog Setpoint High Out	0.00	0.00	125.00	100	%
17	Partial Load	0	0	1		

Register #	Name	Default	Min	Max	Scale	Unit
	Fault Enable					
18	PLF Tolerance	8.0	0.0	100.0	10	%
19	Partial Load Fault Resistance	8.00	0.10	655.35	100	ohm
20	PLF Alarm Delay time	10	1	120		sec
21	Relay Alarm Mask	384	0	65535		
22	Shorted SCR detect enable	0	0	2		
23	Open Load detect enable	0	0	1		
24	Digital Setpoint 1 (EEPROM)	0	0	10000		
25	Digital Setpoint 2 (RAM) 1	0	0	10000		
26	Digital RUN Enable	0	0	1		
27	Setpoint Select	2	2	2		

Register #	Name	Default	Min	Max	Scale	Unit
28	Digital RUN Enable power-up default	0	0	1		
29	PLF Teach Enable	0	0	1		
30	Communications Heartbeat Time	0	0	65535		
31	Network Timeout Action	0	0	2		
32	Network Timeout Setpoint	0	0	10000		
33	IP Address Configuration method	1	0	1		
34	IP Address, OCTET 1	192	0	255		
35	IP Address, OCTET 2	168	0	255		
36	IP Address,	71	0	255		

Register #	Name	Default	Min	Max	Scale	Unit
	OCTET 3					
37	IP Address, OCTET 4	250	0	255		
38	Subnet Mask, OCTET 1	255	0	255		
39	Subnet Mask, OCTET 2	255	0	255		
40	Subnet Mask, OCTET 3	255	0	255		
41	Subnet Mask, OCTET 4	0	0	255		
42	Gateway IP Address, OCTET 1	192	0	255		
43	Gateway IP Address, OCTET 2	168	0	255		
44	Gateway IP Address, OCTET 3	0	0	255		

Register #	Name	Default	Min	Max	Scale	Unit
45	Gateway IP Address, OCTET 4	100	0	255		
46	Relay Normal State	0	0	1		sec
		Format				
201	Active Setpoint	X				
202	Analog Setpoint %	SXXX.X			10	%
203	Analog Setpoint Cmd	SXXX.X			10	V,A
204	Analog Setpoint Signal	SXX.XX			10	v,ma
205	Inhibit Alarm Status	XXXXXXXX				
206	Controller Status	X				
207	AC Line Frequency	XX.X			10	Hz

Register #	Name	Default	Min	Max	Scale	Unit
208	Line Voltage	XXX.X			10	V
209	Load Voltage	XXX.X			10	V
210	Load Current	XXX.X			10	A
211	Load Resistance	XXXX.X			10	ohm
212	Heatsink temp	XXX.X			10	C
213	Controller State	X				
214	Output Duty Cycle %	XXX.X			10	%
215	Setpoint Reference	XXX.X			10	V,A
216	Feedback	XXX.X			10	V,A
217	Control Loop Error	SXXX.X			10	V,A
218	Warning Alarm Status	XXXXXXXX				100
219	Partial Load	XXX.XX			100	ohm

Register #	Name	Default	Min	Max	Scale	Unit
	Fault Target Res					
220	Partial Load Fault Resistance	XXX.XX			100	ohm
221	PLF Resistance Deviation	SXXX.X			10	%
222	Partial Load Fault Status	XXXXXXXX				
310	In Service Time HI	XXXXXXXXXX				
311	In Service Time LO					
312	Processor Temperature	XXX.X			10	C
330	EE Calibration bits	XXXX				
331	Calibration ADC bits In	XXXX				
332	Firmware ID	XXXXX				

Register #	Name	Default	Min	Max	Scale	Unit
333	Firmware Revision	XX.XX			100	
334	Minor Revision	XX				
335	Feedback Read status	X				
336	Misc Status	XXXXXXXX				
337	EEPROM Status	XXXXXXXXXXXXXXXX				
338	AC Line Status	XXXXXXXX				
339	Load Status	XXXXXXXX				
340	Error Latch	XXXXXXXX				
341	Ethernet status	XXXXXXXXXXXXXXXX				
342	Network Heartbeat Timer	XXXXX				

Register #	Name	Default	Min	Max	Scale	Unit
343	IP Address in use, OCTET 1	XXX				
344	IP Address in use, OCTET 2	XXX				
345	IP Address in use, OCTET 3	XXX				
346	IP Address in use, OCTET 4	XXX				

Additional parameter descriptions

Inhibit Alarm Status

Inhibit alarm status is a 8-bit bitfield:

7	6	5	4	3	2	1
Reserved	Reserved	Reserved	Reserved	Feedback Loss	Over Temperature	Over Current Trip

If any bit is set to 1, the controller will *not* be allowed to run.

Warning Alarm Status

Warning alarm status is a 8-bit bitfield:

7	6	5	4	3	2	1	0
Reserved	Reserved	High temperature	Shorted SCR	Open Load	Partial Load Fault	Current Limit	Voltage Limit

Warning alarms are not considered critical and will not prevent the controller from running.

Feedback Read Status

Feedback status is a 8-bit bitfield:

7	6	5	4	3	2	1	
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Ti

Indicates whether the controller has acquired feedback on the line. If any bit is set to 1, then the controller has lost feedback.

AC Line Status

AC Line status is a 8-bit bitfield:

7	6	5	4	3	2	1	0
Reserved	Reserved	Sync-Locked (to AC Line)	Pre-Lock 2	Pre-Lock 1	Reserved	AC Line B OK	AC Line A OK

Bits 5 must be set to 1 before the controller can provide power to the load.

Load Status

Load status is a 8-bit bitfield:

7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	Open Load	Reserved	Reserved	Reserved	Short SCR

Controller Status

Controller status is one of:

Value	Description
0	Disabled
1	Initialization
2	Normal, operating
3	Calibration
4	Diagnostic

Controller State

Controller state is one of:

Value	State	Description
0	STOP	The state the controller is in when AC Line voltage is not present.
1	RUN	The state the controller is in when AC Line voltage is present and the controller is synchronized to the AC line.
2	FAULT	A latching state of output shutdown caused by over current or over temperature alarms. A power cycle or processor reset is required to clear this state.
3	FAULT RESET	Used as a temporary state to transition from FAULT to RUN once again.

EEPROM Status

EEPROM status is an 16-bit bitfield. EEPROM is used to store controller configuration and calibration data. Any errors in EEPROM may indicate that the firmware is corrupted.

Bit	Description
0	EEPROM Initialization
1	SP Table Error
2	MFG CP Table Error
3	Calibration Table Error
4	Reserved

Bit	Description
5	Reserved
6	Backup Calibration Table Error
7	Bottom Board Calibration Table Error
8	SP Definition Table needs updating
9	Bottom Board Calibration Backup Error
10	Reserved
11	Reserved
12	EEPROM is write protected
13	Reserved
14	Reserved
15	Feedback Calibration Table has changed, store to EEPROM

Error Latch

Error latch is a 8-bit bitfield:

7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	Feedback loss	SCR timing loss	Line Frequency failure	Phase loss or missing cycle	Line Lock Loss

Error latch is provided as a diagnostic troubleshooting aid.

Miscellaneous Status

Miscellaneous status is an 8-bit bitfield:

7	6	5	4	3	2	1
Reserved	Initialization in progress	Reserved	Reserved	Waiting for ENTER key during initialization	Reserved	USB Powered

Data types

All Modbus registers are 2-bytes (WORD).

Registers may be an unsigned integer (most commonly), bitfield, integer, bool, or decimal.

Most of these are straightforward. Decimals are stored as integers with a scale in our controllers.

The **Scale** column indicates how much you should *divide* the value by when reading it, and how much you should *multiply* it by when writing it.

Commands

CMD #	Name	Description
6	Factory reset	Reset to factory settings
13	Reset parameter	Resets user parameters to defaults
24	Store to EEPOM	Saves all parameters to permanent storage
198	Identify	Flashes LEDs on controller
248	Reset	Effectively restarts the controller

Miscallaneous

IMPORTANT

You may notice that ModbusTCP parameter numbers are one less than the same parameters in other ATOM fieldbus profiles. This is because ModbusTCP uses zero-based addressing and subtracts 1 from all register numbers. The table above lists the actual register numbers you should use in your PLC project.

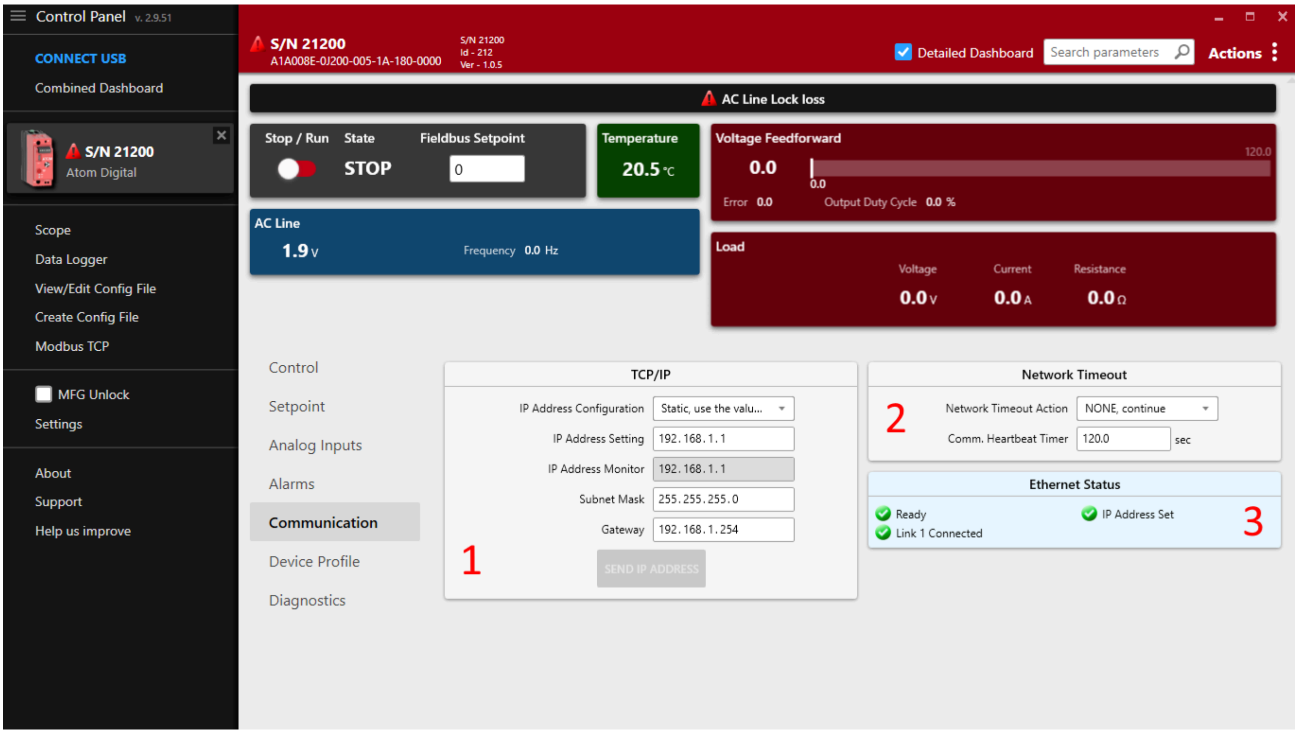
ATOM / Fieldbus / EtherCAT / Overview

> NOTE
>
> Control Concepts has run the EtherCat Conformance Test Tool to verify that ATOM is fully compliant with the EtherCAT standard.

ESI file

> INFO
>
> Download ATOM's ESI file: Atom.xml.

Control Panel Communication Settings



Some communication settings can be configured in the **Communication** tab in Control Panel.

- Section `1`: TCP/IP settings
  - **IP Address Configuration**
    - `Static`: Use the IP address, subnet mask, and gateway specified below.
    - `DHCP`: Use DHCP to obtain an IP address.
  - **IP Address Setting**: The IP address of the ATOM controller.
  - **IP Address Monitor**: The current IP address of the ATOM controller.
  - **Subnet Mask**: The subnet mask of the ATOM controller.
  - **Gateway**: The gateway address for the ATOM controller.
- Section `2`: Network Timeout
  - The EtherNet/IP heartbeat timeout (Encapsulation Inactivity Timeout) in seconds.
  - You can configure a network timeout action to perform when the device loses communication with the PLC:
    - `None`: Do nothing

    - `STOP, fault shutdown`: STOP the controller, disabling output
    - `Use network timeout setpoint`: Configure an alternative setpoint to use when the controller loses communication with the PLC.
- Section `3`: Ethernet status
  - Indicates the status of both RJ45 ports, IP address configuration, conflict detection, and any other errors with the EtherNet/IP connection.

> **INFO**
>
> ### Control Panel and PLC software
>
> These settings are synchronized with your PLC environment. You do not have to use Control Panel to change these settings - you can stay in your PLC software. Control Panel merely provides them as an altnerative way to configure ATOM's EtherNet/IP settings.
>
> You can use Control Panel simultaneously with your PLC software without issues.

> **WARNING**
>
> ### IP Address Conflict Detection
>
> ATOM uses **IP Address Conflict Detection** to detect IP address conflicts on the network. If ATOM detects another device using the same IP address, it will disable all network communication until the conflict is resolved.
>
> Please ensure all devices on the network are assigned unique a IP address.

Hardware considerations

WARNING

Daisy chaining

As ATOM has two RJ45 ports, it can be easily daisy-chained. When daisy-chaining ATOM, take care to avoid a loop in the network. In some loop configurations, ATOM is susceptible to network broadcast storms, which can cause the controller to become unresponsive. If you are daisy-chaining ATOM, ensure that the network is loop-free.

ATOM works with both unmanaged and managed switches. We recommend a managed switch for larger networks to give you more control over the network topology.

Parameters

Overview

The following is an overview of the parameters available over EtherCAT. These parameters can be accessed and modified through TwinCAT or other EtherCAT master software.

Inputs (`DT6000`)

Index	Name	Type	Description
0x6000:01	Line Voltage	UINT	Input AC line voltage in tenths of a volt (i.e. `800` = `80.0 V`)
0x6000:02	Load Voltage	UINT	Load Voltage in tenths of a volt (i.e. `800` = `80.0 V`)

Index	Name	Type	Description
0x6000:03	Load Current	UINT	Load Current in tenths of an amp (i.e. `800` = `80.0 A`)
0x6000:04	Load Resistance	UINT	Load resistance in tenths of an Ohm (i.e. `800` = `80.0 Ohms`)
0x6000:05	Heatsink Temp	UINT	Heatsink temperature in tenths of a Celsius (i.e. `800` = `80.0 C`)
0x6000:06	AC Line Frequency	UINT	AC Line Frequency in tenths of a Hertz (i.e. `800` = `80.0 Hz`)
0x6000:07	Controller State	UINT	See controller state description
0x6000:08	Output Duty Cycle	UINT	Indicates the amount, in tenths of a percent (`800` = `80.0%`), that the output of the controller is ON
0x6000:09	Setpoint Reference	UINT	The command reference input to the control compensation loop in `V`, or `A` (per the feedback parameter)
0x6000:10	Feedback	UINT	The control output supplied to the load in units determined by the ?Feedback? selection
0x6000:11	Setpoint selected	UINT	Active setpoint. `1` = Analog setpoint, `2` = Digital setpoint, `3` = Fieldbus setpoint

Index	Name	Type	Description
0x6000:12	Inhibit Alarm Status	UINT	Indication of alarms that cause the controller to be shut OFF and not allowed to RUN. See inhibit alarm status description
0x6000:13	Controller Status	UINT	Indicates the operational status of the controller. See controller status description
0x6000:14	Warning Alarm	UINT	Indication of conditions that cause specific warning alarms. See warning alarm description

Inhibit Alarm Status

Inhibit alarm status is a 8-bit bitfield:

7	6	5	4	3	2	1
Reserved	Reserved	Reserved	Reserved	Feedback Loss	Over Temperature	Over Current Trip

If any bit is set to 1, the controller will *not* be allowed to run.

Warning Alarm Status

Warning alarm status is a 8-bit bitfield:

7	6	5	4	3	2	1	0
Reserved	Reserved	High Temperature	Shorted SCR	Open Load	Partial Load Fault	Current Limit	Voltage Limit

Warning alarms are not considered critical and will not prevent the controller from running.

Controller Status

Controller status is one of:

Value	Description
0	Disabled
1	Initialization
2	Normal, operating
3	Calibration
4	Diagnostic

Controller State

Controller state is one of:

Value	State	Description
0	STOP	The state the controller is in when AC Line voltage is not present.

Value	State	Description
1	RUN	The state the controller is in when AC Line voltage is present and the controller is synchronized to the AC line.
2	FAULT	A latching state of output shutdown caused by over current or over temperature alarms. A power cycle or processor reset is required to clear this state.
3	FAULT RESET	Used as a temporary state to transition from FAULT to RUN once again.

Outputs (`DT7000`)

Index	Name	Type	Description
0x7000:01	Fieldbus setpoint	UINT	A value between 0 and 10,000 indicating the desired output current. The value is scaled to the output range of ATOM. For example, if the output range is 0-100A, a value of 5000 would set the output to 50A. `0 = 0%`, `10,000 = 100%`
0x7000:02	Digital Run Enable	UINT	`0` = Disable output, `1` = Enable output. When disabled, the output current is set to 0A.

Configuration (`DT8000`)

Index	Name	Type	Description
0x8000:01	Feedback Type	UINT	Sets the signal type used for feedback by the control loop. `1` = Voltage Feedforward, `2` = Load Current.
0x8000:02	Firing mode	UINT	Selects the desired type of firing mode.
0x8000:03	Slew rate	UINT	`1-100`: Sets the control loop response for Phase Angle and Half-Cycle DC firing modes. Higher value = slower response, Lower value = faster response.
0x8000:04	Control Loop	UINT	Closed loop compares the feedback with the setpoint to achieve the correct output. Open loop adjusts the output duty cycle of the controller directly without adjusting for feedback.
0x8000:05	Full Scale Voltage	UINT	Set to the expected output voltage when the controller output is fully ON 100%. This equates to the voltage output command when feedback type is set to Voltage feedforward and the setpoint is at 100% (maximum)
0x8000:06	Full Scale Current	UINT	Set to the expected output current when the controller output is fully ON 100%. This equates to the current output command when feedback type is set to Load Current and the setpoint is at 100% (maximum)

Index	Name	Type	Description
0x8000:07	Voltage Limit	UINT	`10 - 700`: Sets the maximum output voltage allowed by the controller.
0x8000:08	Current Limit	UINT	`1.0 - 84.0`: Sets the maximum output current allowed by the controller.
0x8000:09	Partial Load Fault Enable	UINT	`0` = Disable partial load fault detection & alarm, `1` = Enable partial load fault detection & alarm.
0x8000:10	Partial Load Fault Tolerance	UINT	`0.0 - 100.0` Sets the maximum percent load resistance deviation from the ?Partial Load Fault Resistance? value. Deviations outside this band will trigger a Partial Load Fault alarm.
0x8000:11	Partial Load Fault Resistance	UINT	`0.10 - 655.35` - Sets the nominal resistance of the load. This is used for comparison in determination of a partial load fault alarm condition.
0x8000:12	Partial Load Fault Alarm Delay Time	UINT	Sets the delay time, in seconds, after detection of a partial load fault until the alarm is indicated.
0x8000:13	Relay Alarm Mask	UINT	See relay mask bitfield
0x8000:14	Shorted SCR Check Enable	UINT	Enables and disables shorted SCR detection and alarm indication. Shorted SCR detection is

Index	Name	Type	Description
			always performed when AC Line is ON and the controller?s output is OFF.
0x8000:15	Open Load Detect Enable	UINT	Enables and disables open load detection and alarm indication.

Relay mask

Relay mask is an 16-bit bitfield:

15-9	8	7	6	5	4	3	
Reserved	Over Current Trip	Over Temperature	Partial Load Fault/Open Load	Shorted SCR	Current Limit	Voltage Limit	A Li Lo

ATOM / Fieldbus / EtherCAT / TwinCAT 3

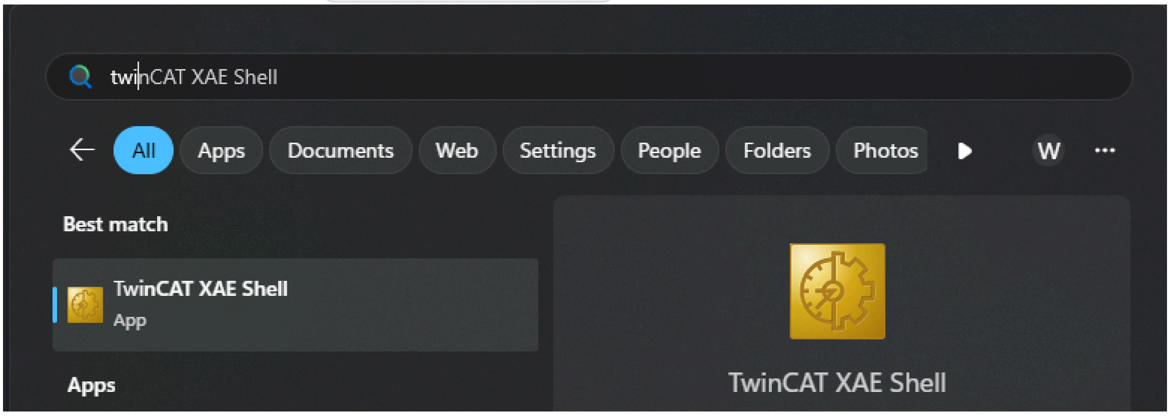
In this tutorial, you'll learn how to control ATOM over EtherCAT with TwinCAT 3.

> **NOTE**
>
> If you'd like to skip this tutorial, download the completed example project: AtomExampleTwinCat.zip.

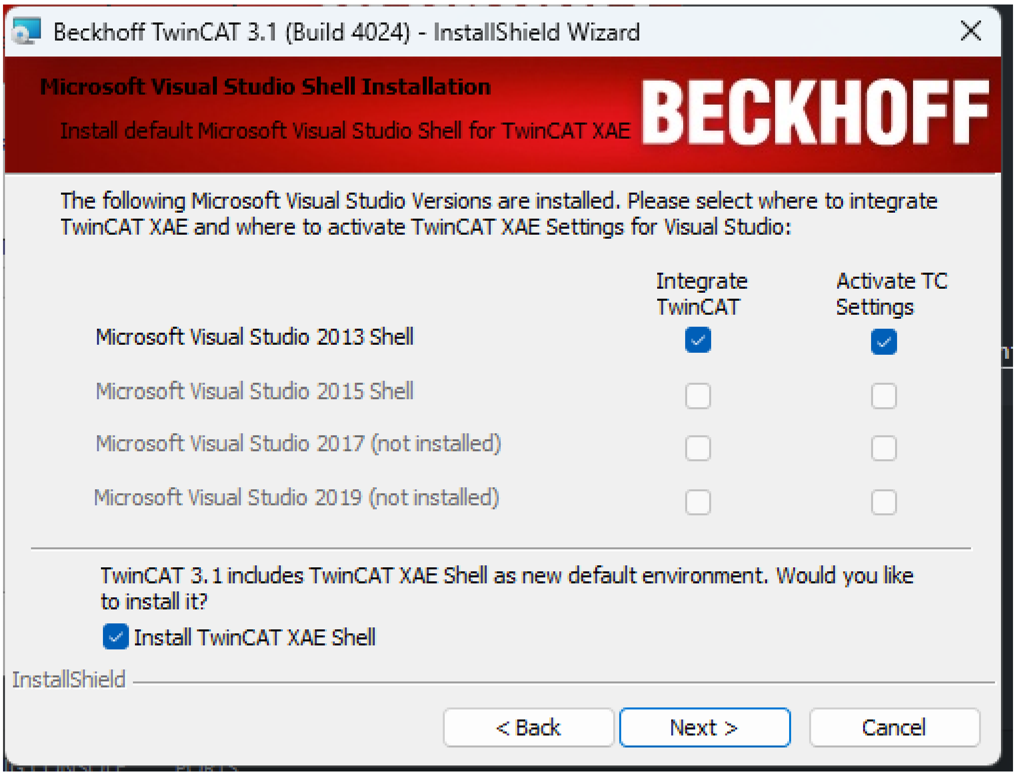
> **NOTE**
>
> If you are unfamiliar with TwinCAT 3, check out PLC programming using TwinCAT 3 video series by Jakob Sagatowski.

Prerequisites

1. A PC with TwinCAT 3 Engineering installed
   - We recommend using `TwinCat XAE Shell`:



- This can be installed during the installation process:



- An Intel-based network interface:

  TwinCAT 3 requires an Intel-based network adapter to work properly.

2. *(Optional)* A Beckhoff EtherCAT PLC (e.g., CX9020, CX5130)
   - TwinCAT 3 includes a built-in soft PLC simulator that you can follow along with.

     This tutorial will cover both options - using a PLC and using the soft PLC simulator.
3. Download ATOM's ESI file: Atom.xml

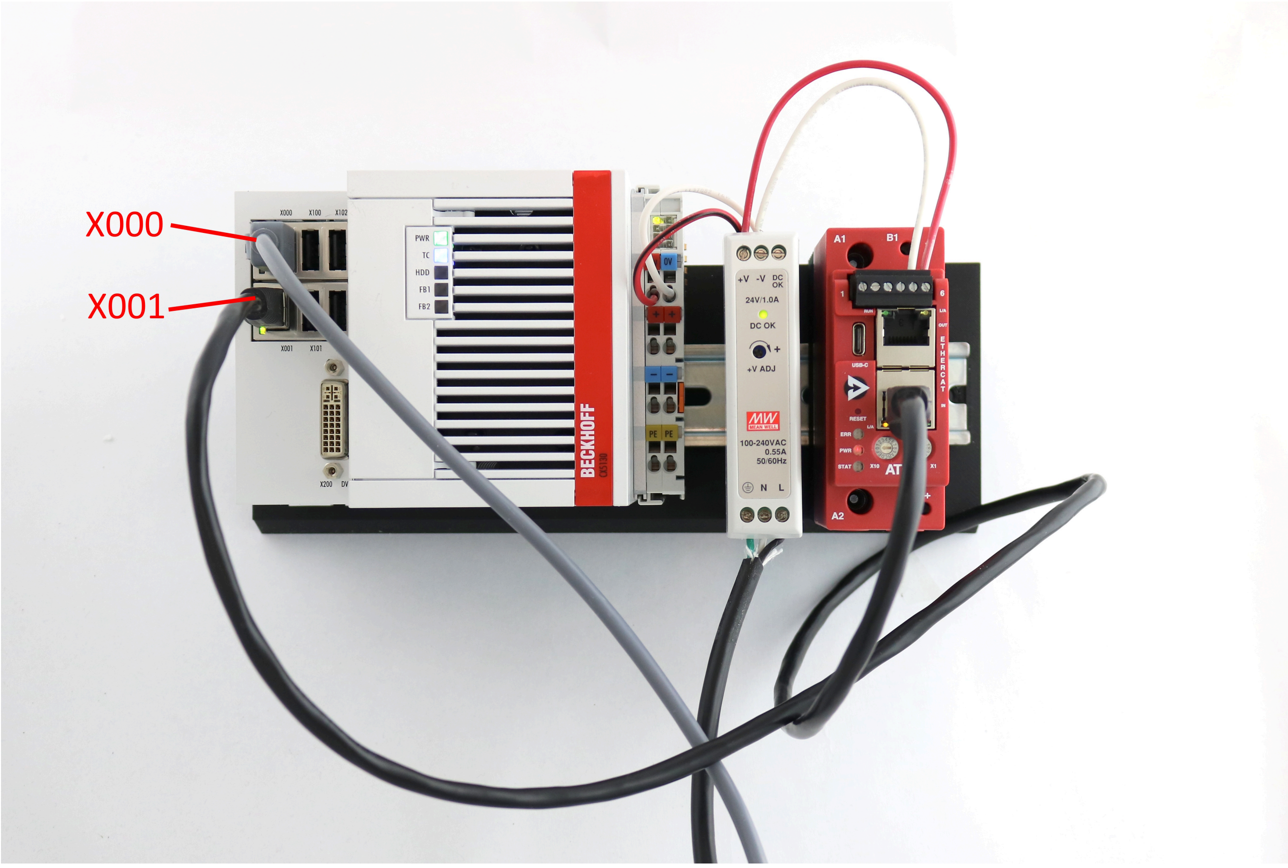
Hardware setup

> **INFO**
>
> To simplify this tutorial, we skip connecting a load to ATOM. The fieldbus configuration remains the same regardless of whether you connect a load or not.

> **INFO**
>
> ## EtherCAT ID switches
>
> ATOM has two rotary switches that together set the EtherCAT ID. The EtherCAT ID is a two byte ID that unqiuely identifies ATOM on an EtherCAT network.
>
> The `X1` rotary switch sets the low byte of the EtherCAT ID, and the `X10` rotary switch sets the high byte of the EtherCAT ID.
>
> For example, when `X1` is set to `1` and `X10` is set to `2`, the EtherCAT ID is `0x21` (33 in decimal).
>
> The EtherCAT ID is used for manual addressing of EtherCAT devices which is useful for seamless device replacement and consistent identification across restarts. In this example, we use TwinCAT's built in automatic addressing, so the EtherCAT ID switch positions don't matter.

Connections:

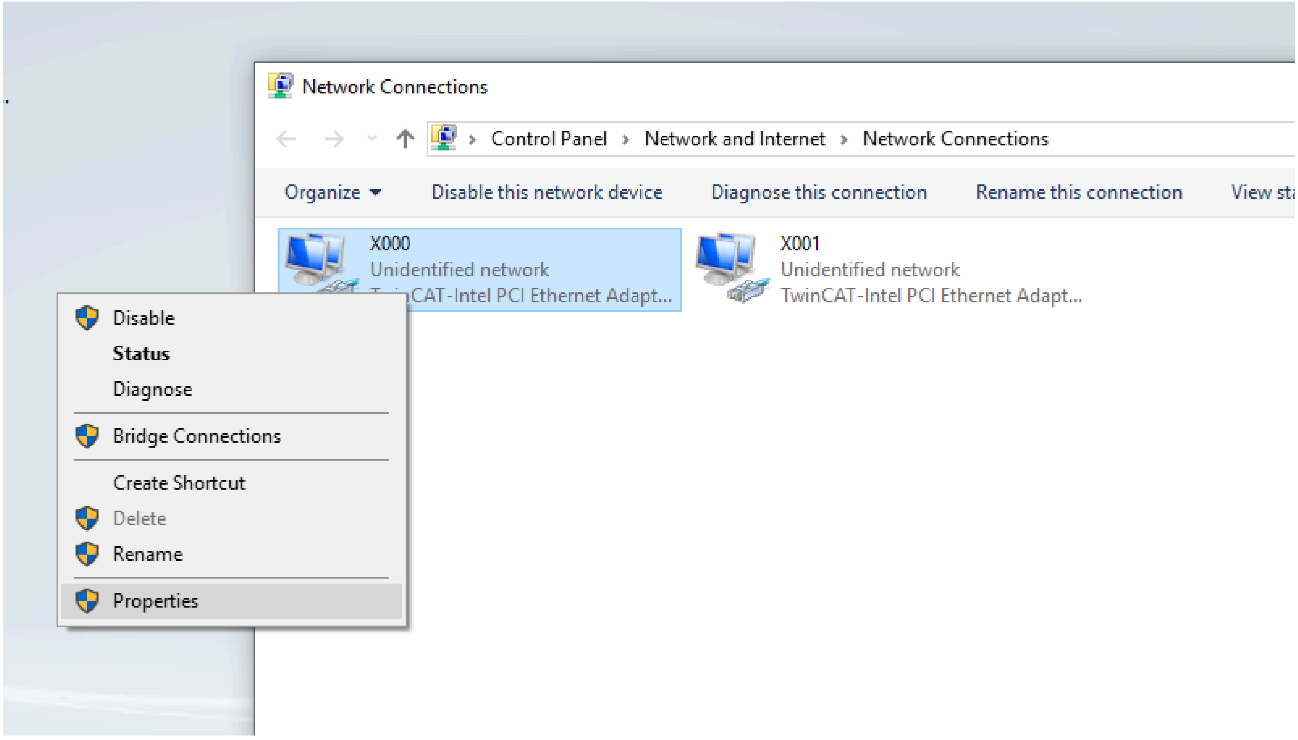
- Connect port `X000` on your PLC to your PC with an Ethernet cable.
- Connect port `X001` on your PLC to the `IN` port on ATOM with an Ethernet cable.
- Connect 24V DC power to ATOM and your PLC.



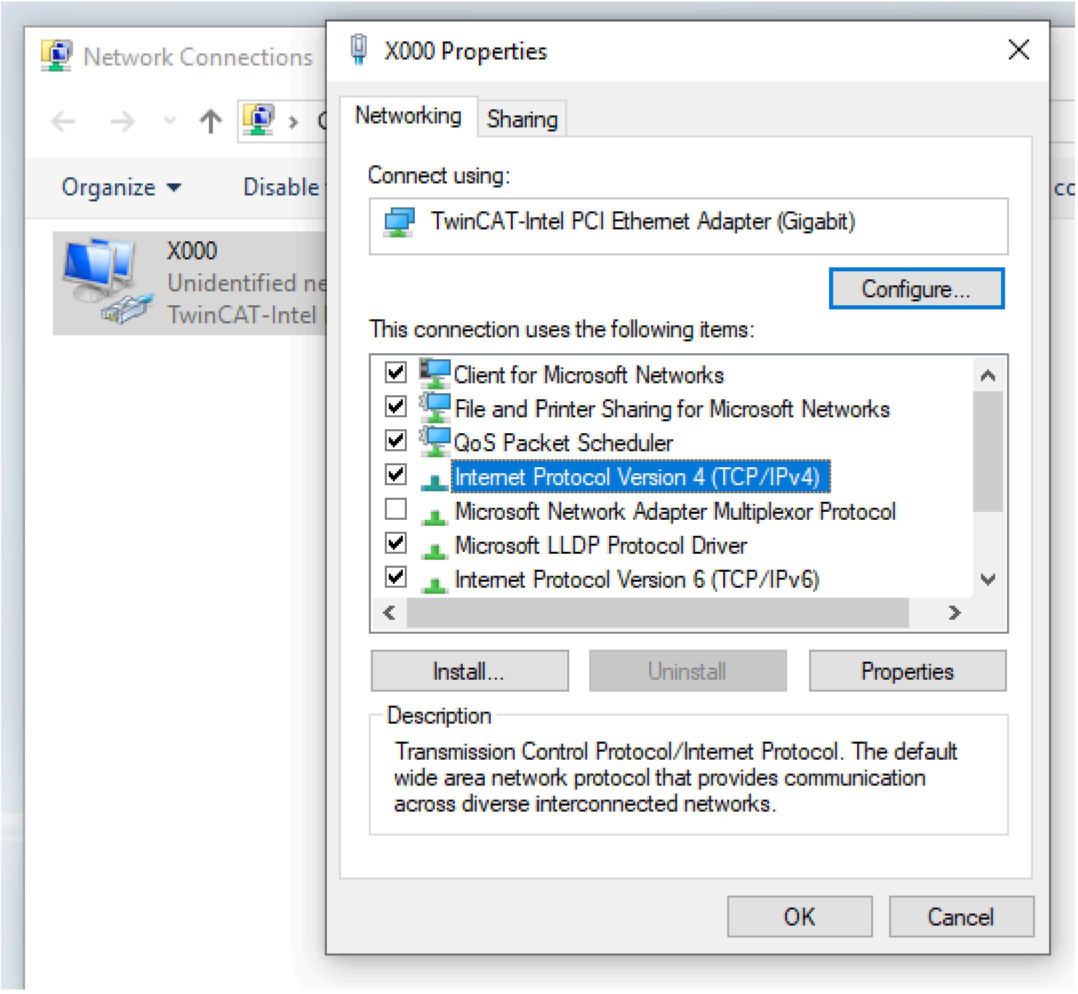
PLC configuration

> **NOTE**
>
> You can skip this section if you are using the TwinCAT soft PLC simulator.

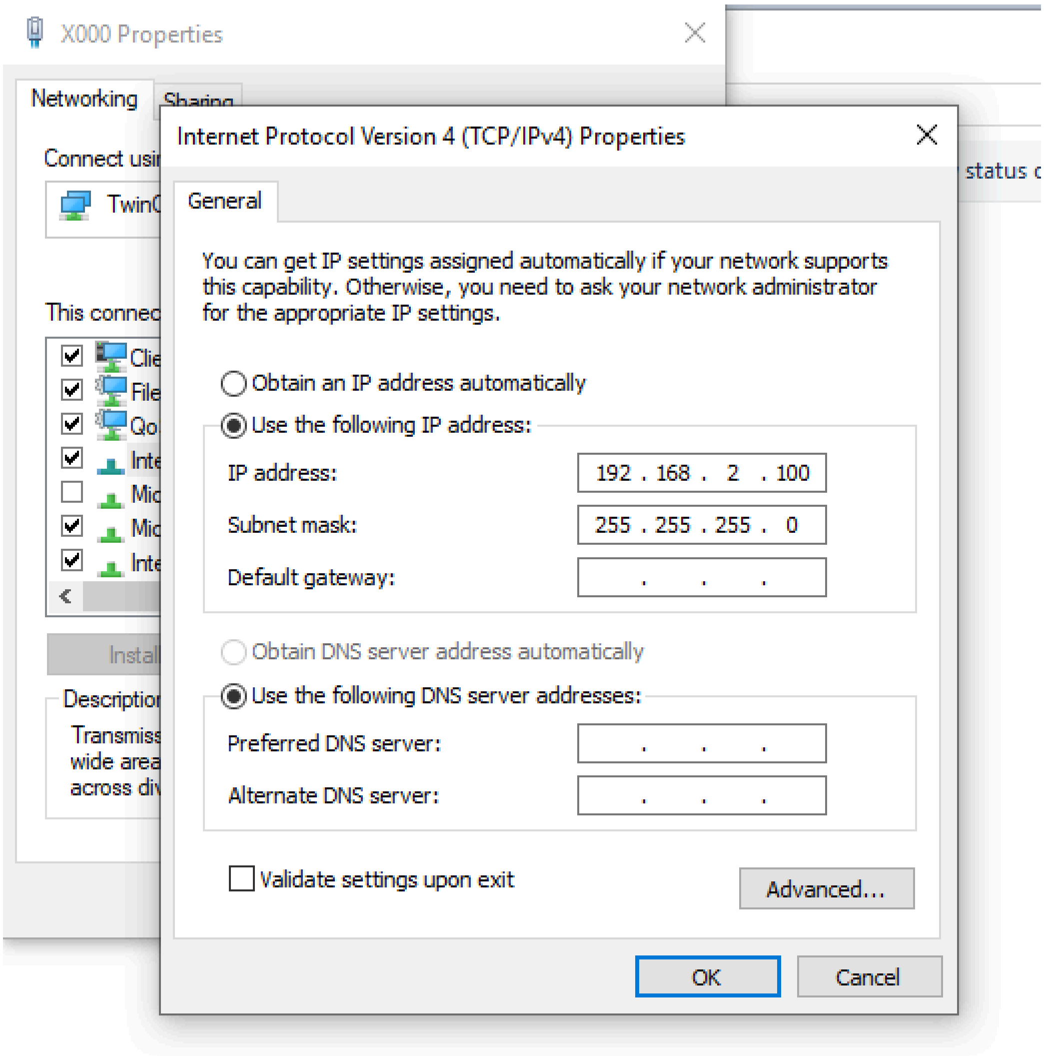
1. If you are using a PLC, connect a monitor and keyboard to it and power it on.
2. Open Network Connection Manager, right-click the `X000` network interface, and select **Properties**:



3. In the **X000 Properties** dialog, select **Internet Protocol Version 4 (TCP/IPv4)** and click **Properties**:



4. In the **Internet Protocol Version 4 (TCP/IPv4) Properties** dialog, select **Use the following IP address** and enter the following values. Then click **OK**:
    - IP address: `192.168.2.100`
    - Subnet mask: `255.255.255.0`

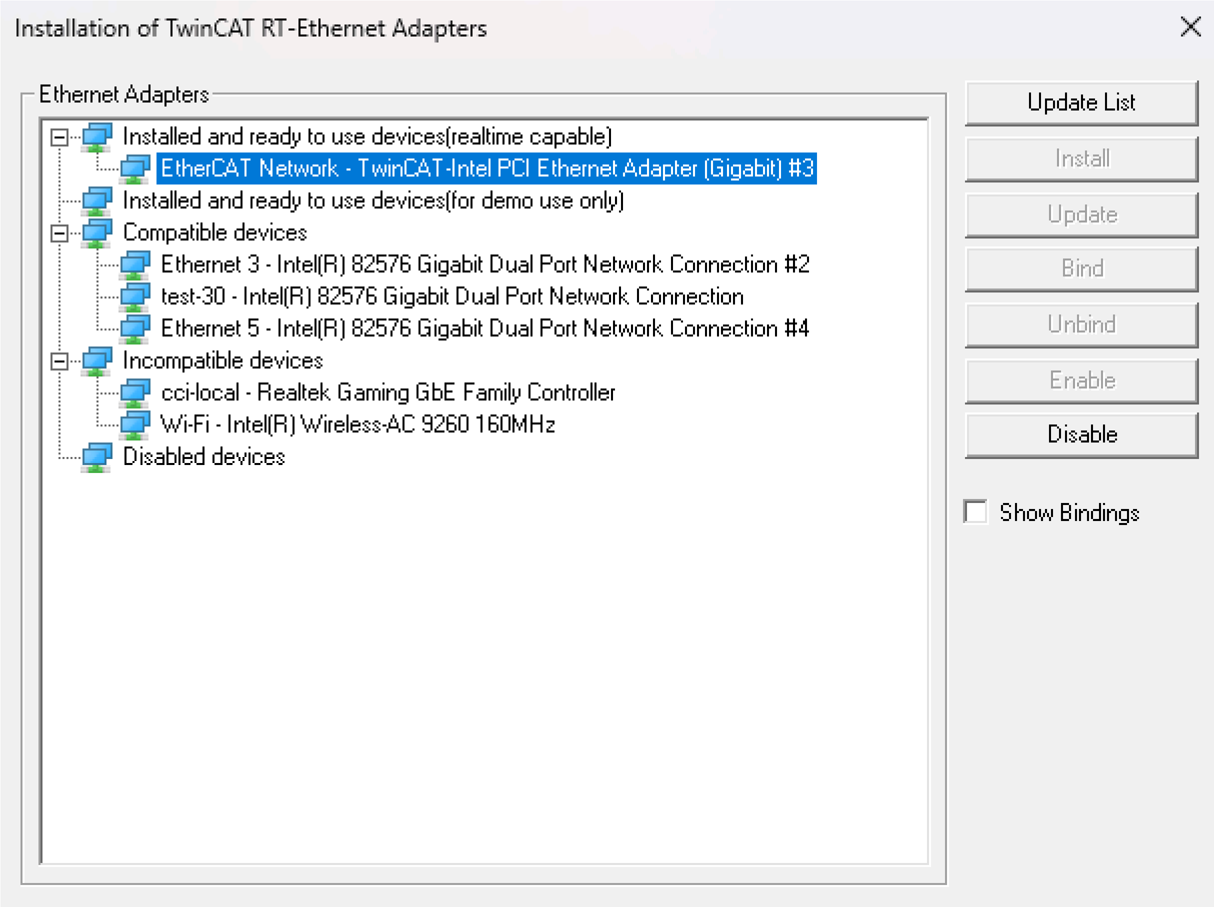


PC configuration

Back on your PC, follow these steps:

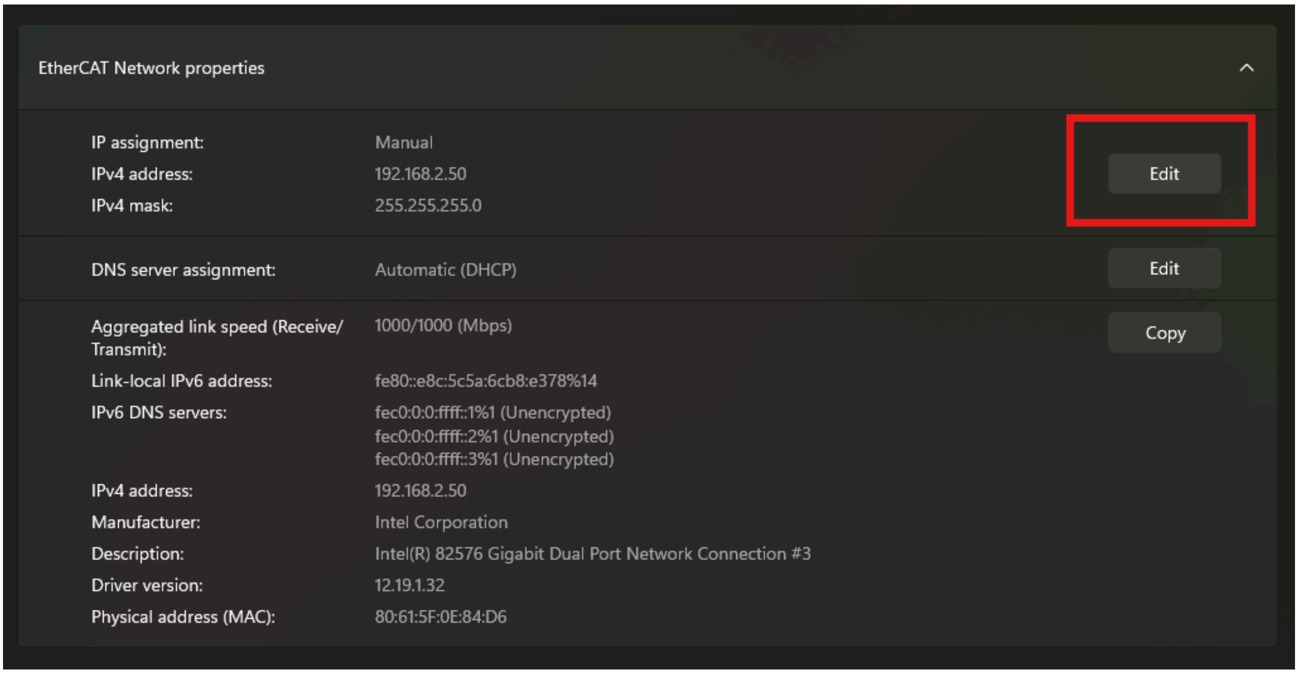
> **INFO**
>
> This uses a network adapter named `EtherCAT Network` to make it easy to remember and follow along. If your network adapter has a different name, use that instead.

1. Navigate to your TwinCAT 3 installation directory (typically `C:\TwinCAT\3.1\System`) and execute `TcRteInstall.exe`. Ensure that the network adapter you intend to use appears under the **Installed and ready to use devices(realtime capable)** category. If it does not, select the network adapter and click **Install**:

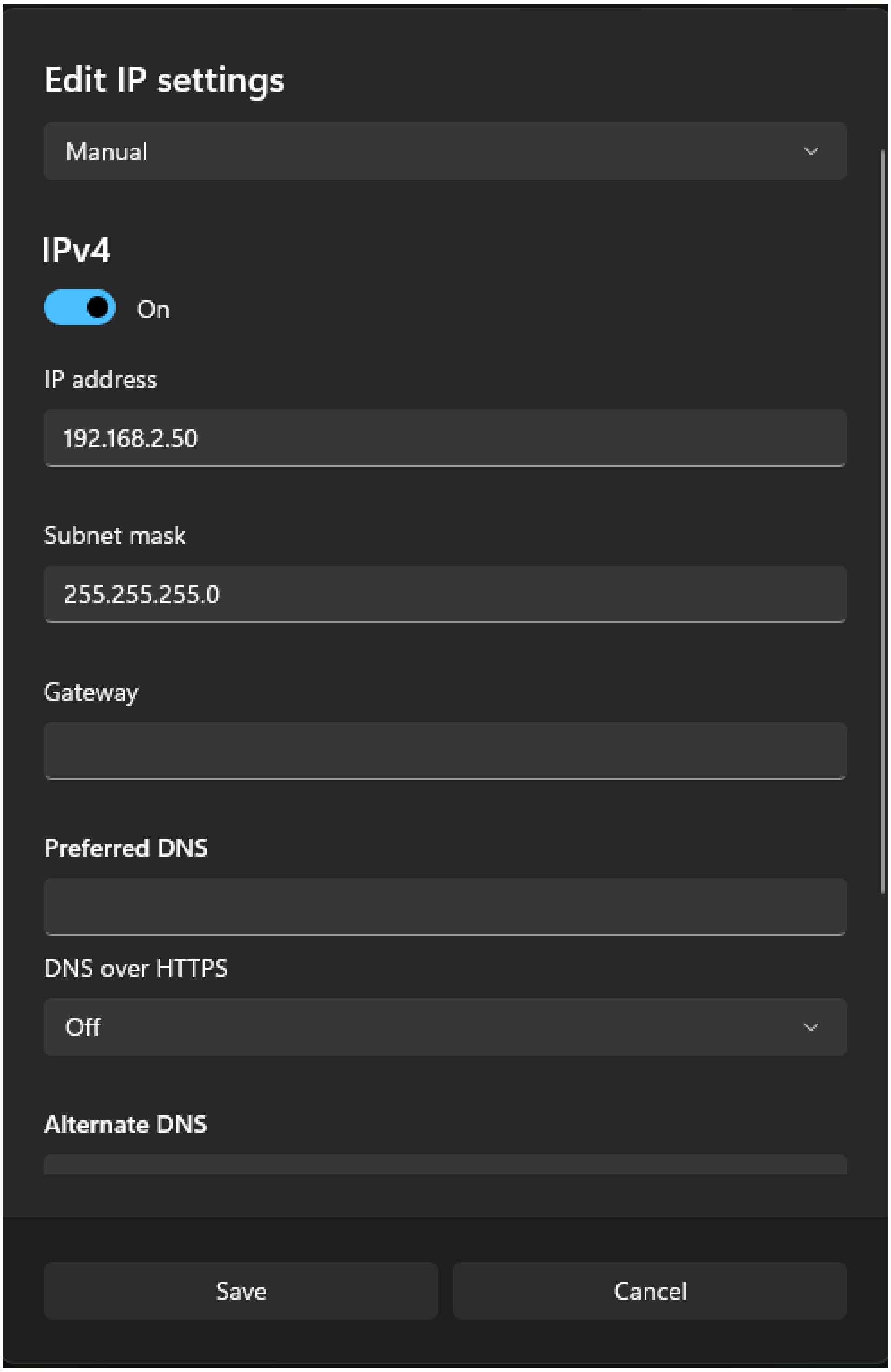


2. Place `Atom.xml` (from the requirements section) in the `C:\TwinCAT\3.1\Config\Io\EtherCAT` directory. This is where TwinCAT looks for ESI files.

3. Navigate to your Settings app and locate your `EtherCAT Network` adapter. Right-click it and select **Edit**:

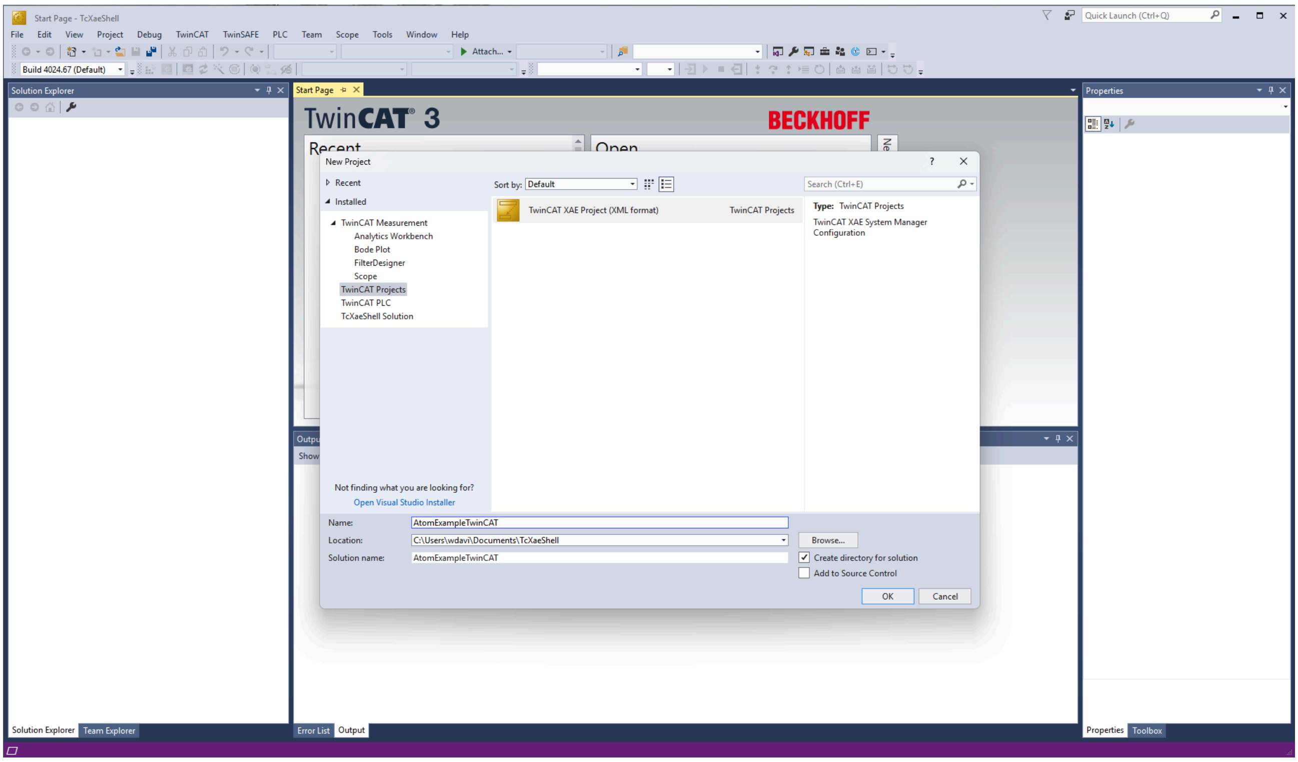


4. In the **EtherCAT Network Properties** dialog, set the `IP address` and `Subnet mask` to the following values. Then, hit **Save**.
    - IP address: `192.168.2.50`
    - Subnet mask: `255.255.255.0`



Creating a TwinCAT 3 project

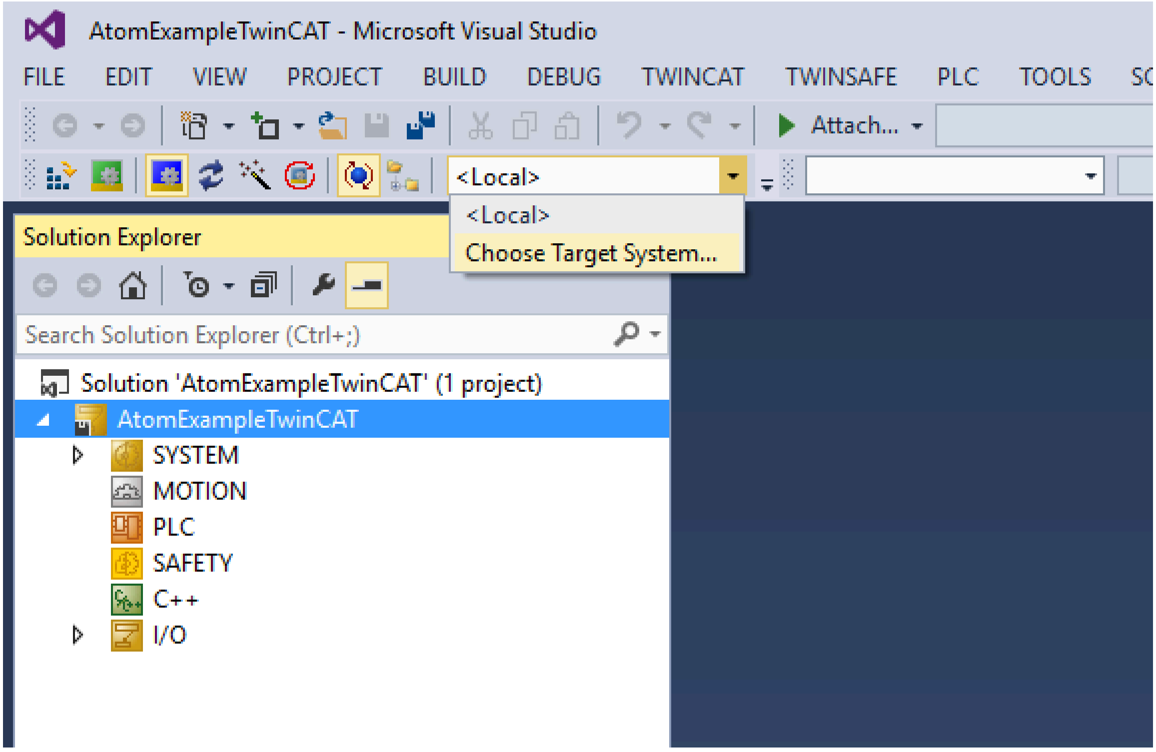
1. Open TwinCAT 3 and select **File** > **New Project**. Name it `AtomExampleTwinCAT` and click **OK**:



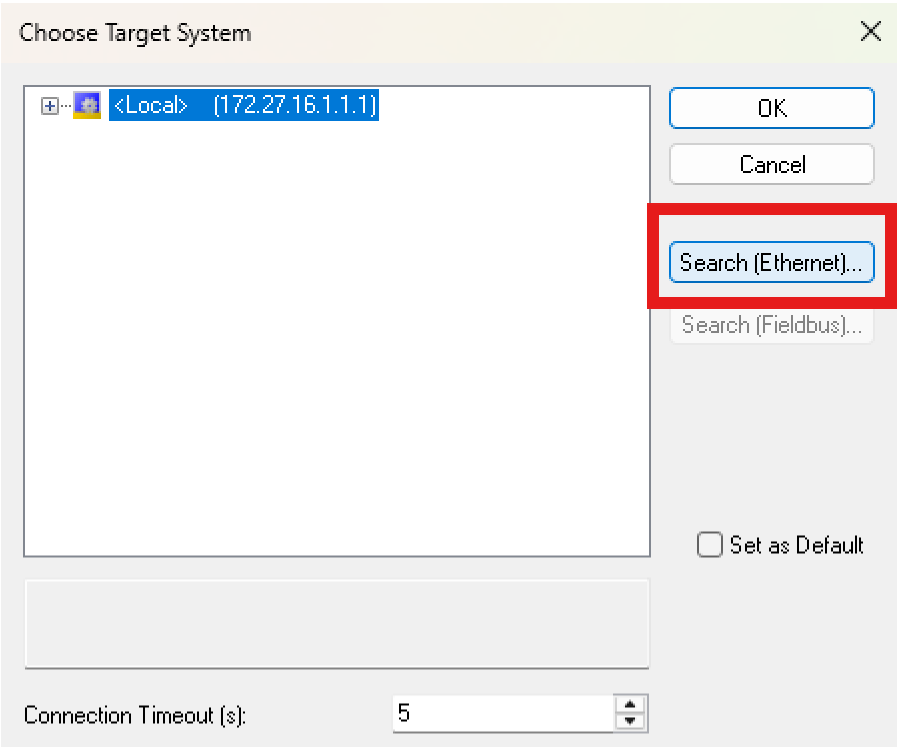
Connecting to your PLC

> ⓘ **NOTE**
>
> If you are using the TwinCAT soft PLC simulator, you can skip this section.

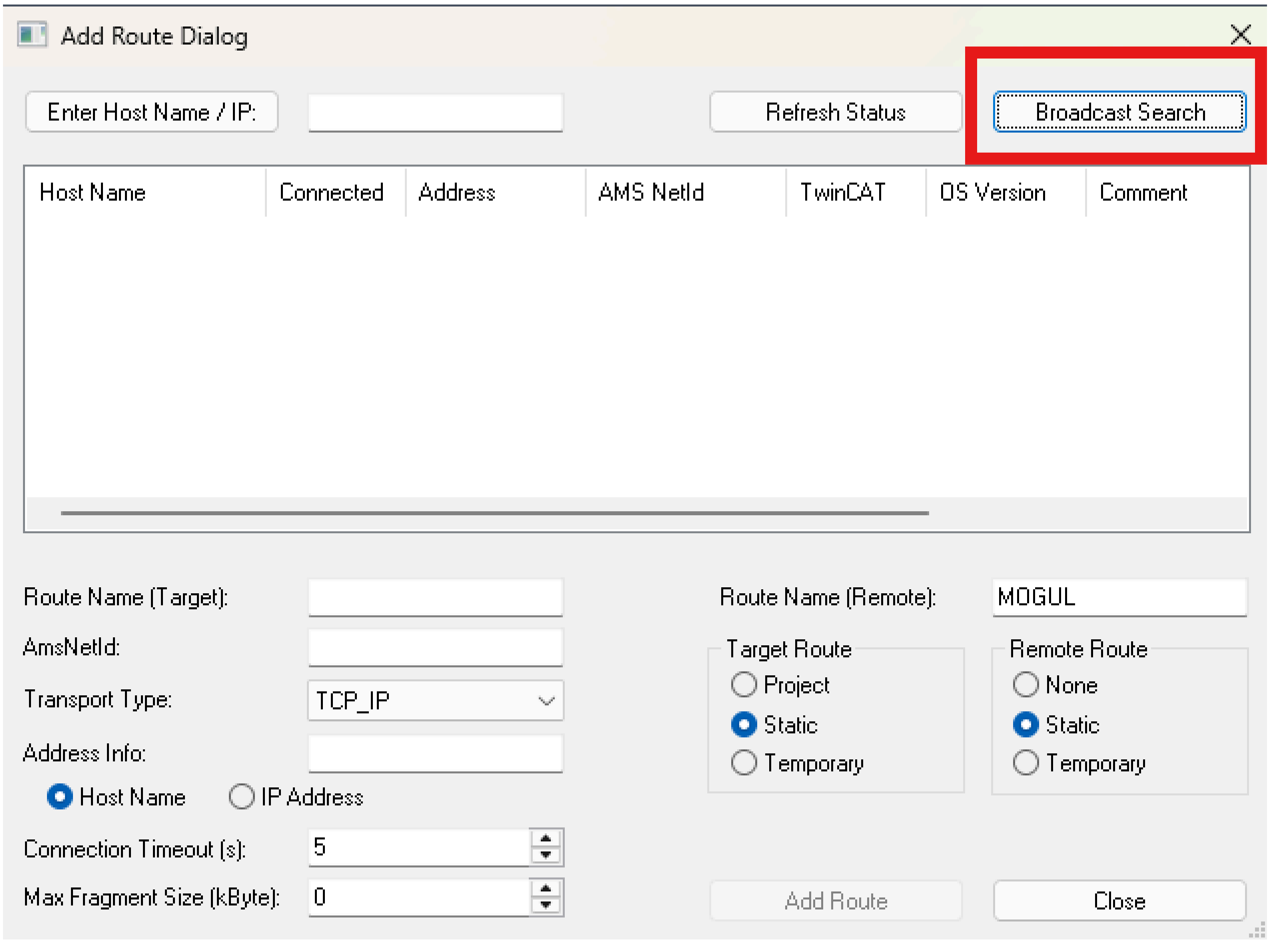
1. Select the target dropdown and click **Choose Target System...**:



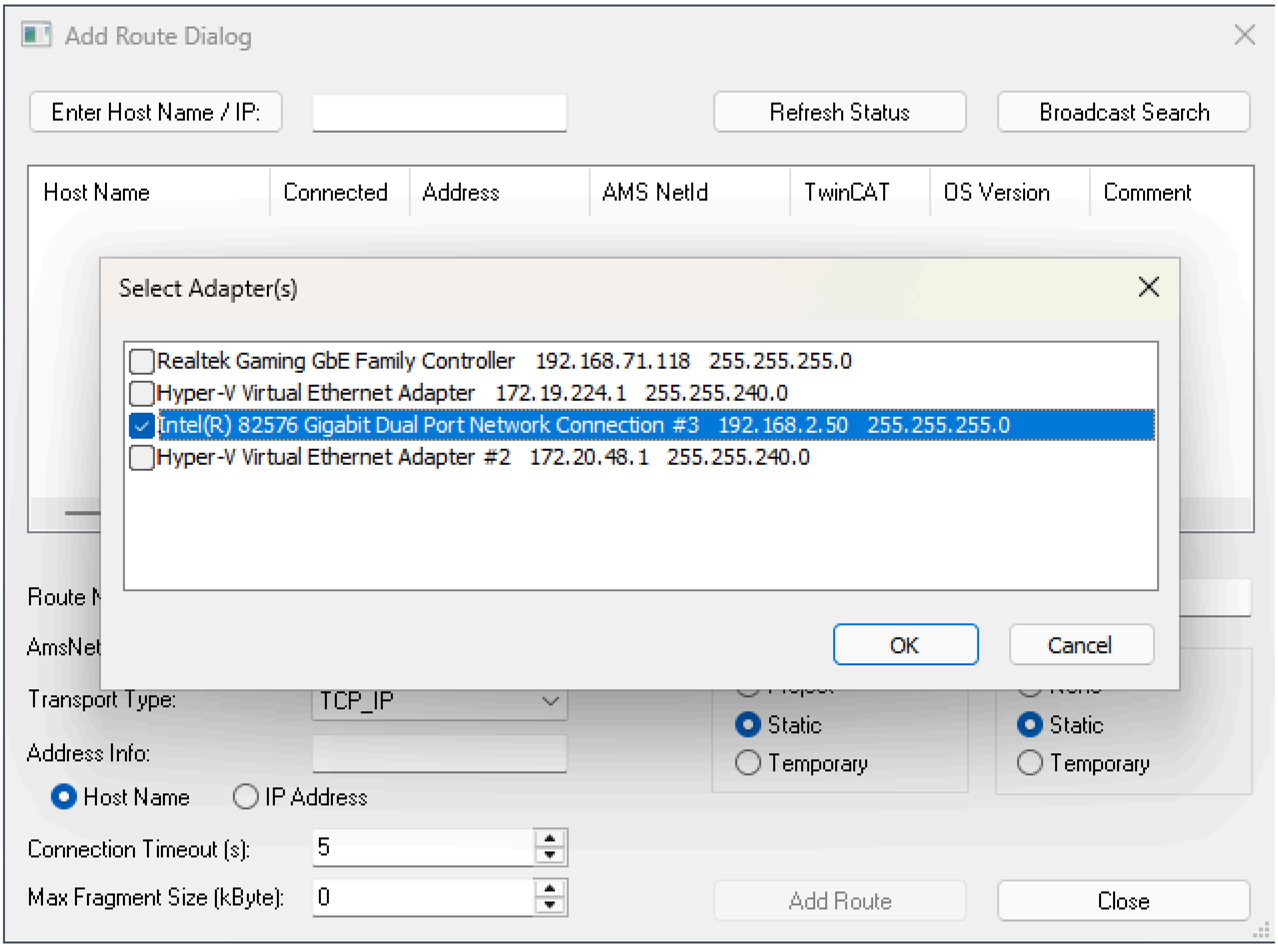
2. Click **Search (Ethernet)**:



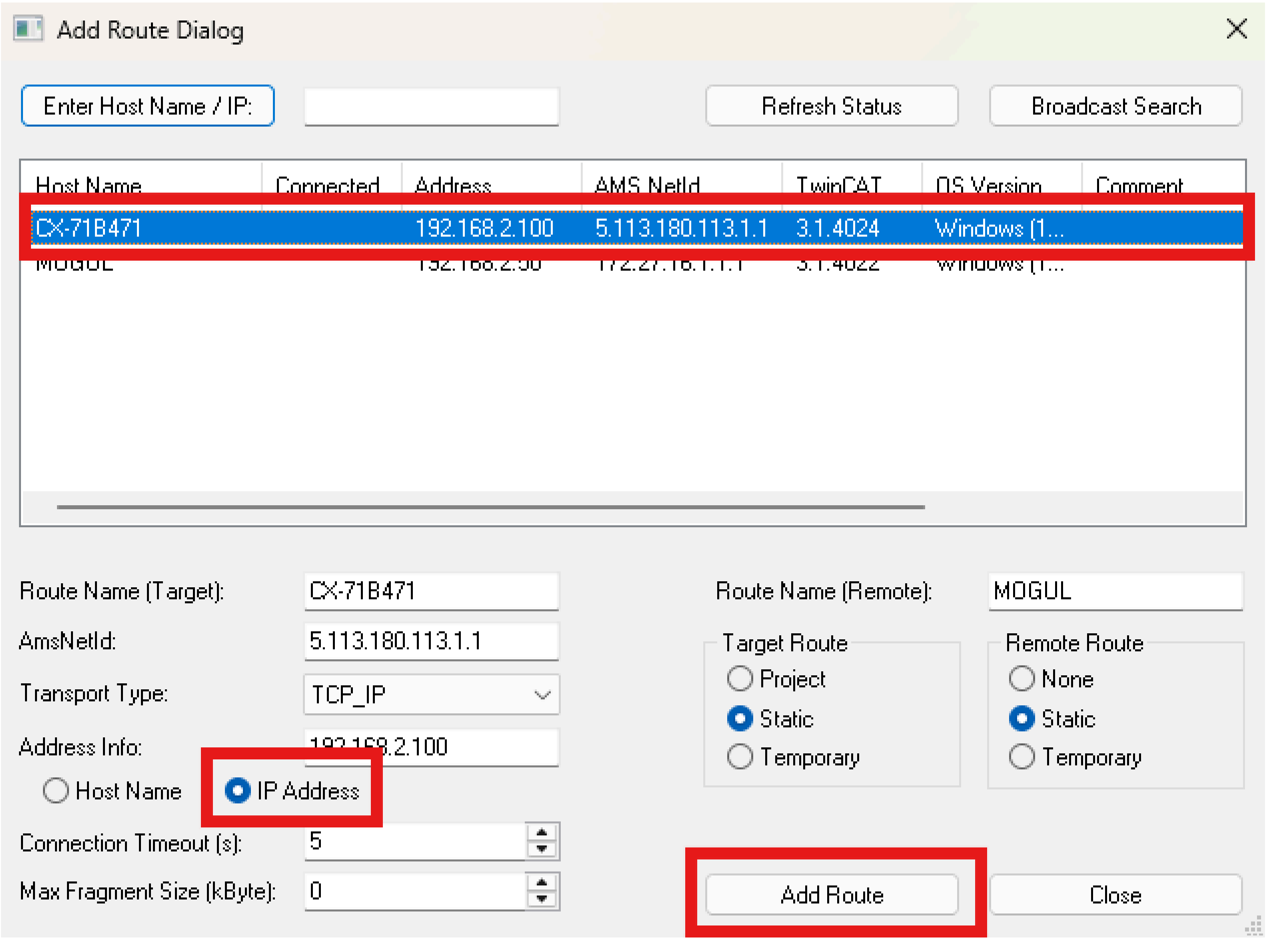
3. Select **Broadcast Search**:



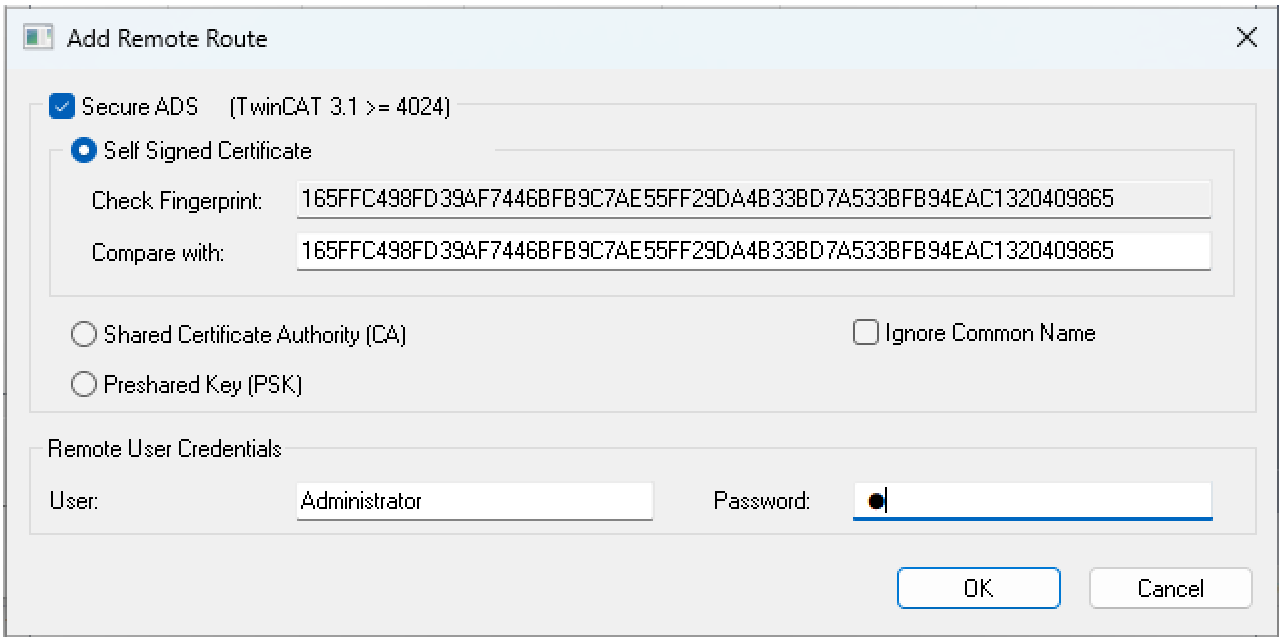
4. Select the network adapter to search - pick the one labeled `192.168.2.50`:



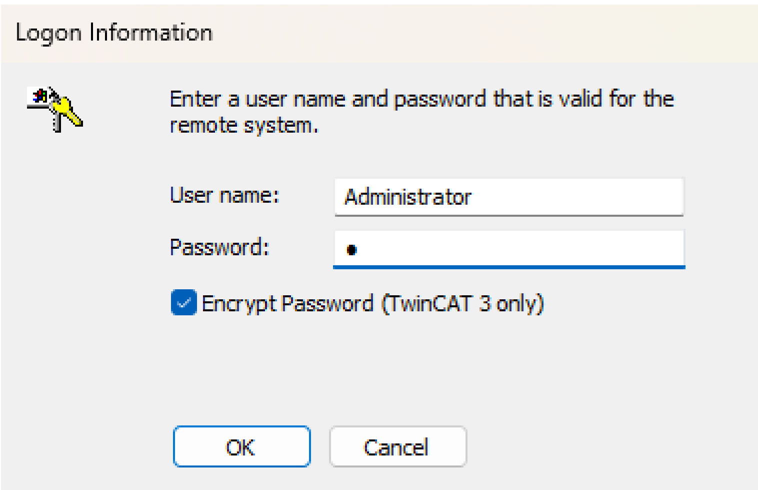
5. Select your `PLC` from the list (it should have the IP address you configured above: `192.168.2.100`). Under **Address Info** select **IP Address**, then click **Add Route**:



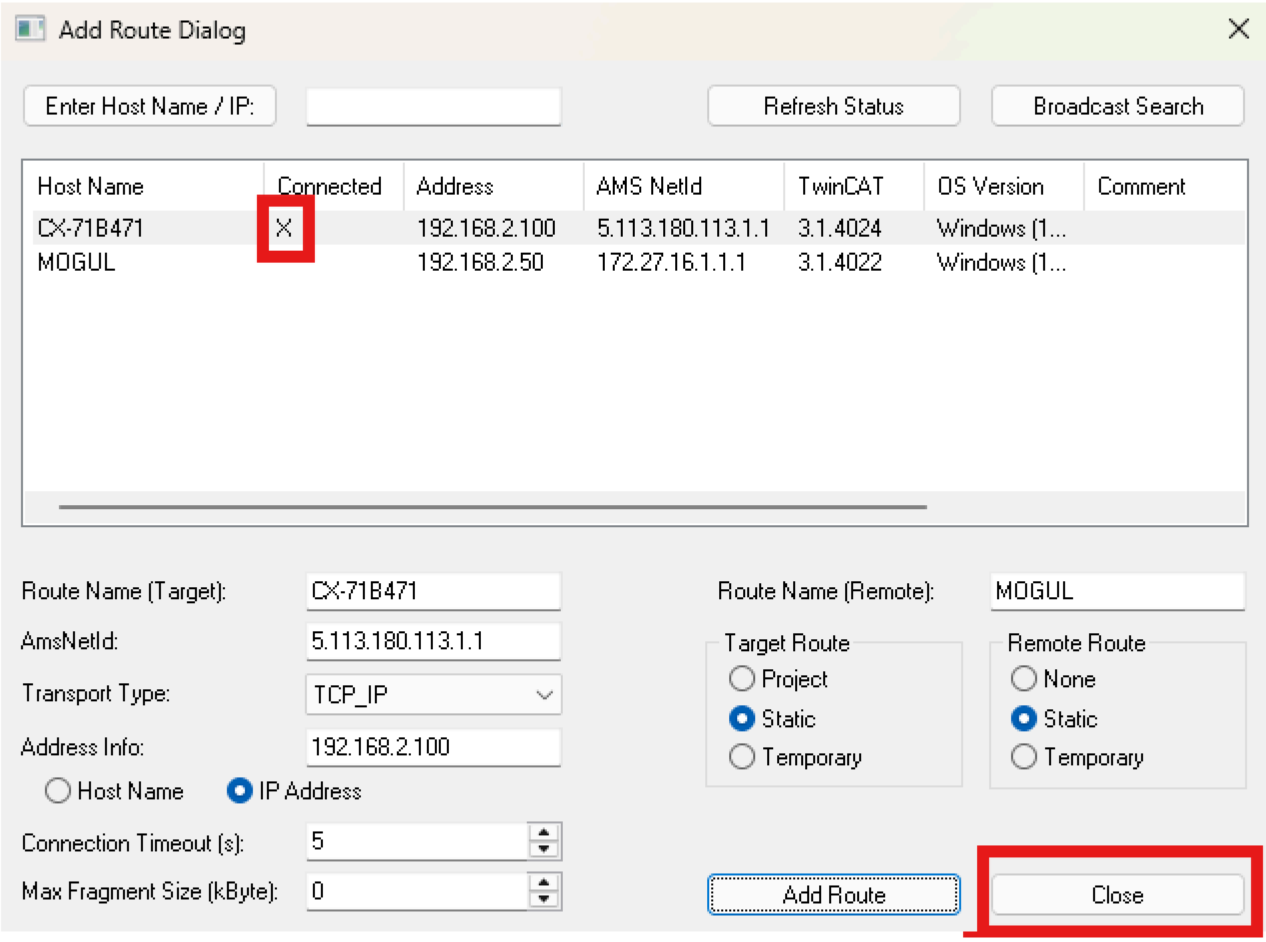
6. If you want to use `Secure ADS`, check `Secure ADS` and copy the `Check Fingerprint` value to the `Compare with` box. In **Remote User Credentials**, enter the username and password for your PLC, then click **OK**. If you haven't changed it, the default Beckhoff credentials are:
    - Username: `Administrator`
    - Password: `1`



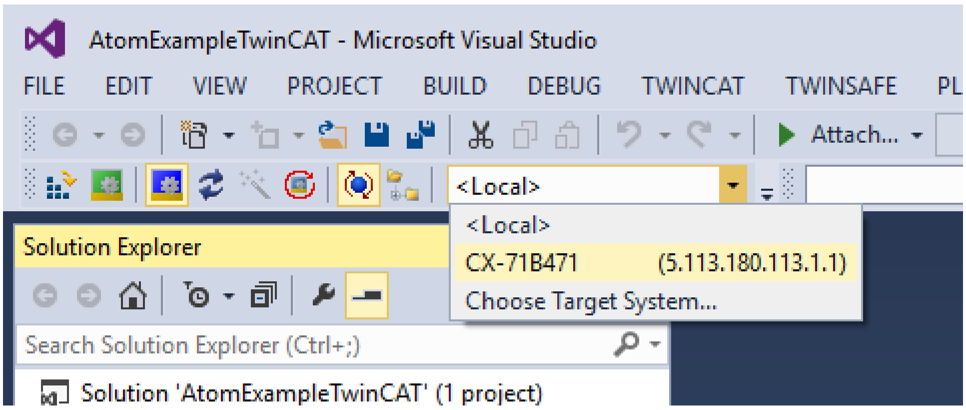
7. If a password prompt appears, enter the same username & password for your PLC as in step 6. then click **OK**:



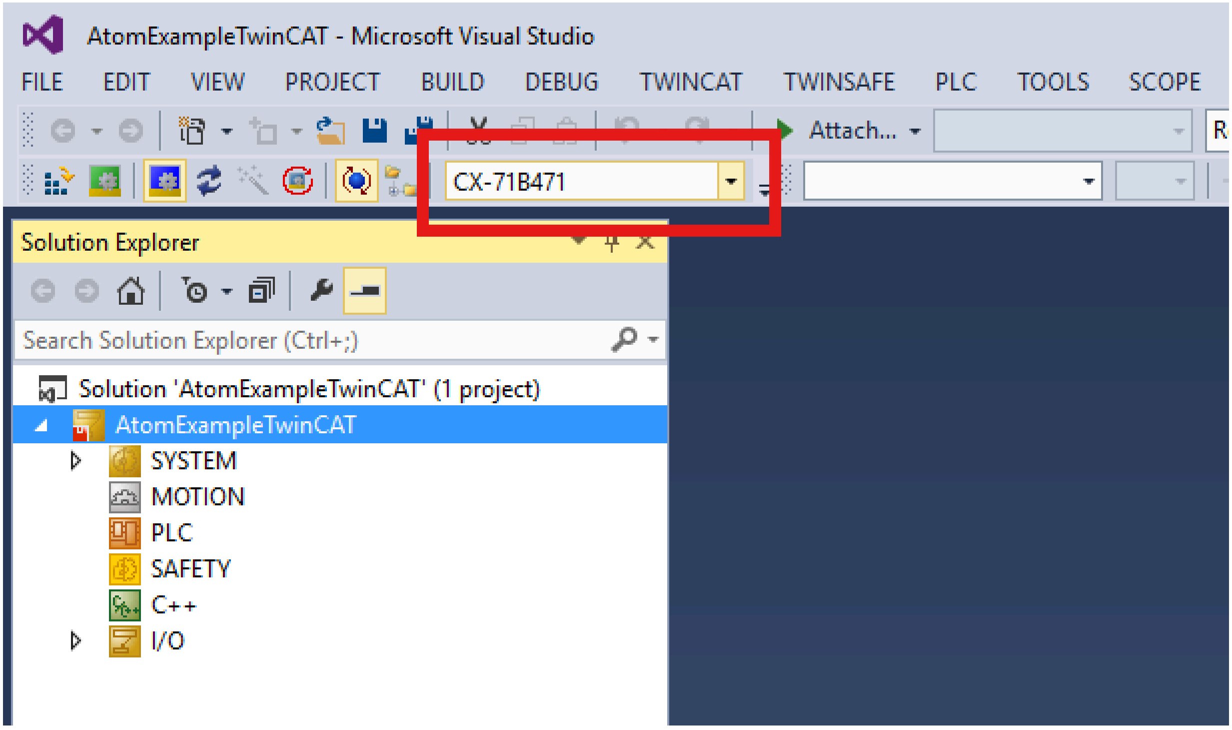
8. If everything worked correctly, you should see a check or X under the `Connected` column next to your PLC. If it does, hit **Close**:



9. Select the target dropdown and select your PLC (in our case, `CX-71B471`):

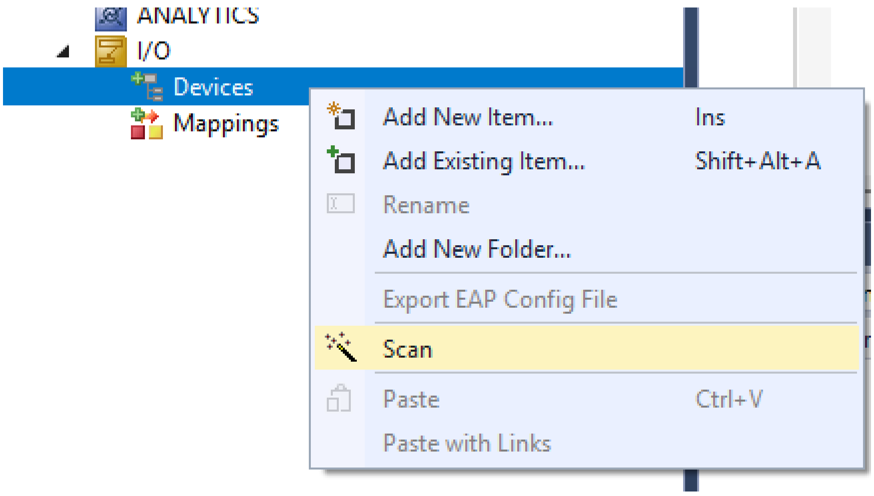


10. If everything worked correctly, you should see `CX-71B471` become the value in the dropdown with *no* `(ERROR)` suffix:

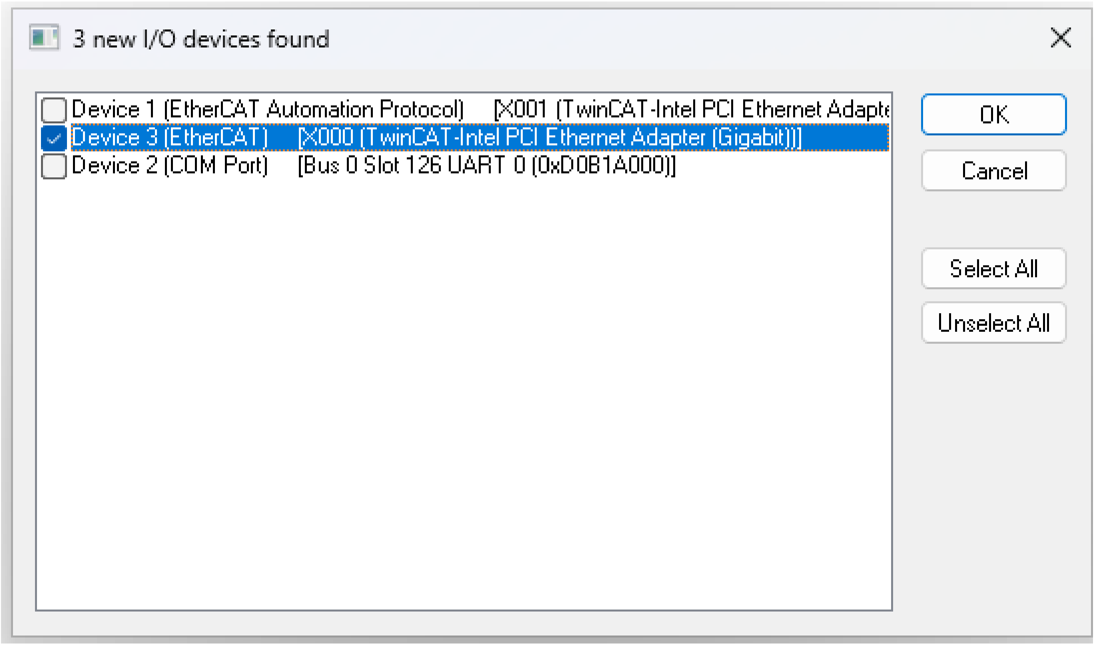


Adding and configuring Atom

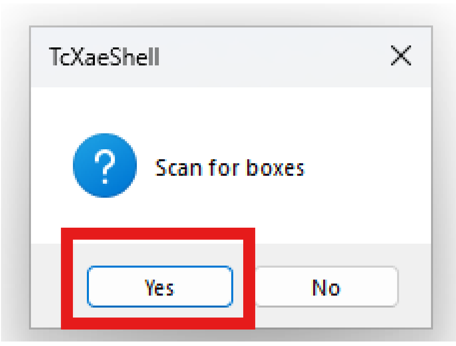
1. Right-click **I/O** > **Devices** and select **Scan**:



2. Select the entry that reads `Device 3 (EtherCAT)` and click **OK** (this might be slightly different for you):

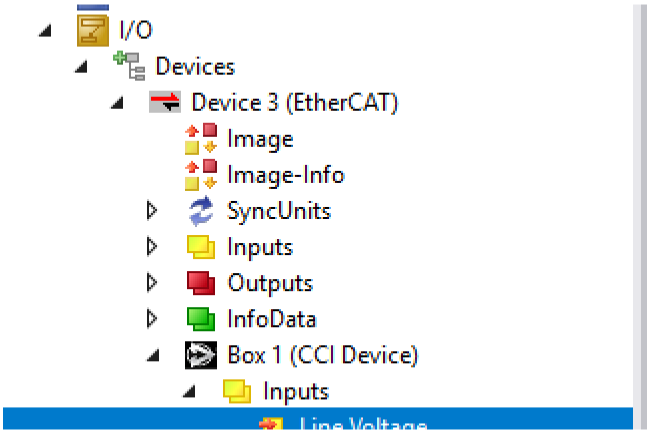


3. When prompted to **Scan for boxes**, select **Yes**:



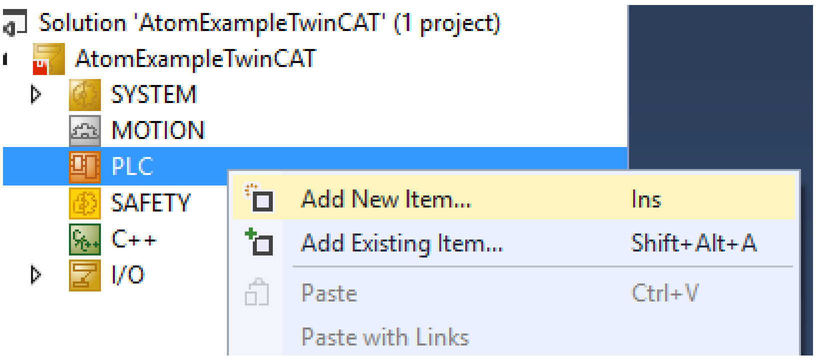
4. If everything worked correctly, you should see `Device 3 (EtherCAT)` (your PLC) and `Box 1 (CCI Device)` (ATOM) in the **I/O Devices** tree. If you do not, see troubleshooting.

> **INFO**
>
> If you are using the TwinCAT soft PLC simulator, you should only see `Box 1 (CCI Device)` in the **I/O Devices** tree.

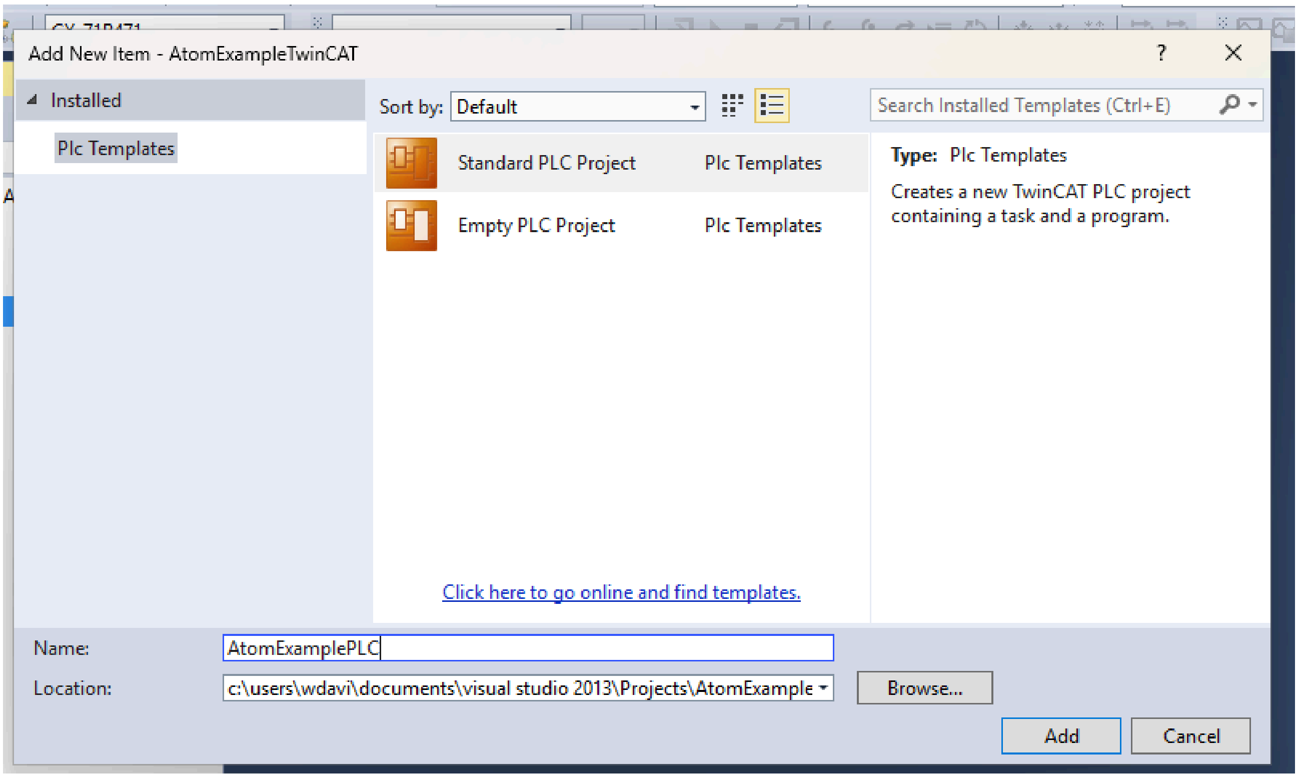


Configuring your TwinCAT 3 project

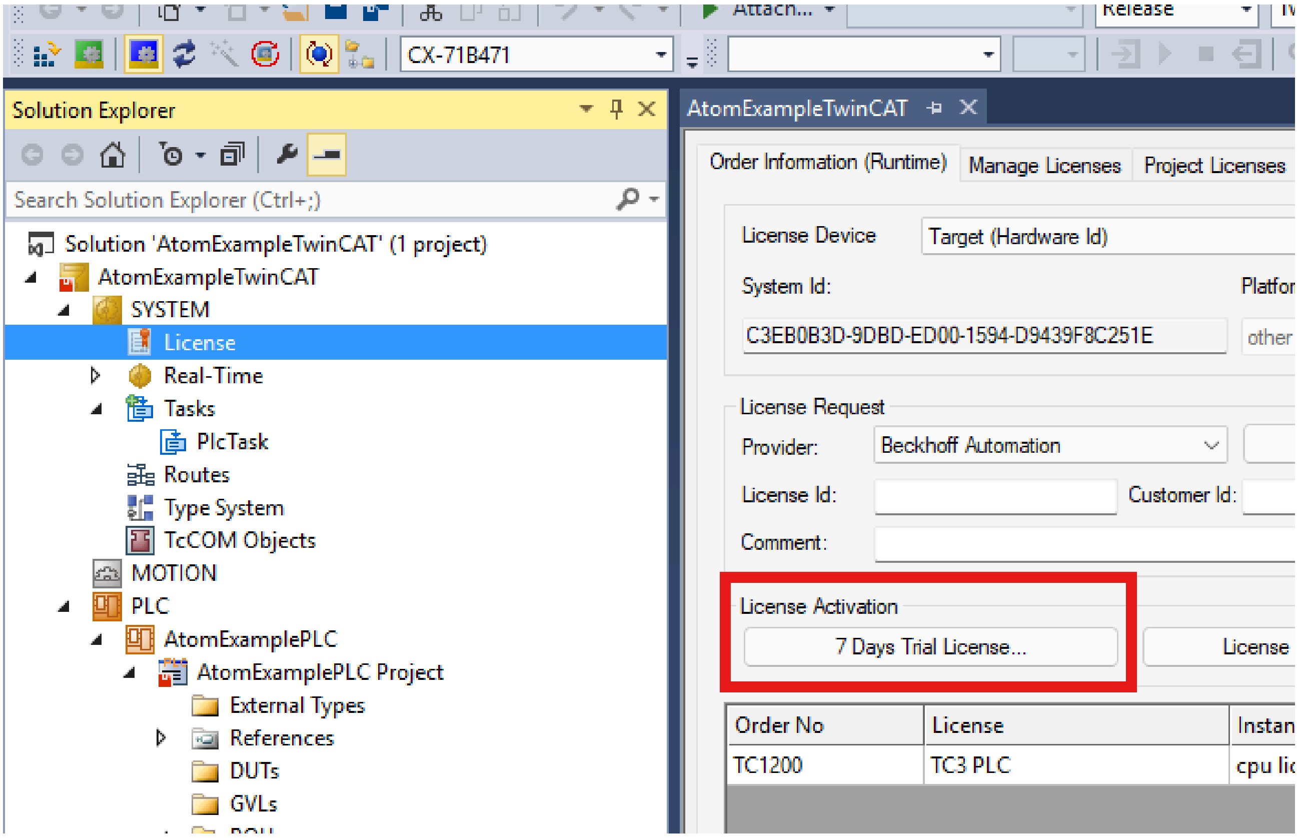
1. Right-click **PLC** and select **Add New Item...**:

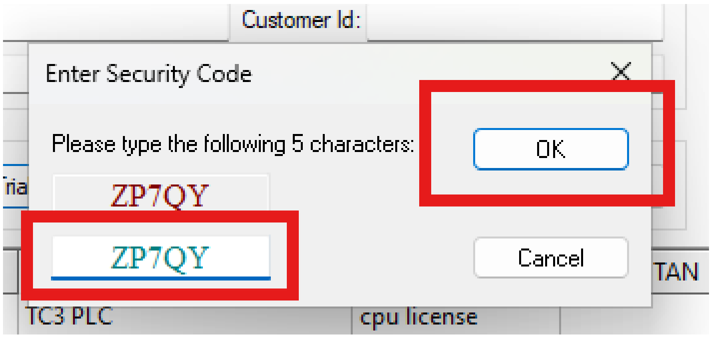


2. Select **Standard PLC Project**, name it `AtomExamplePLC`. Click **Add**:

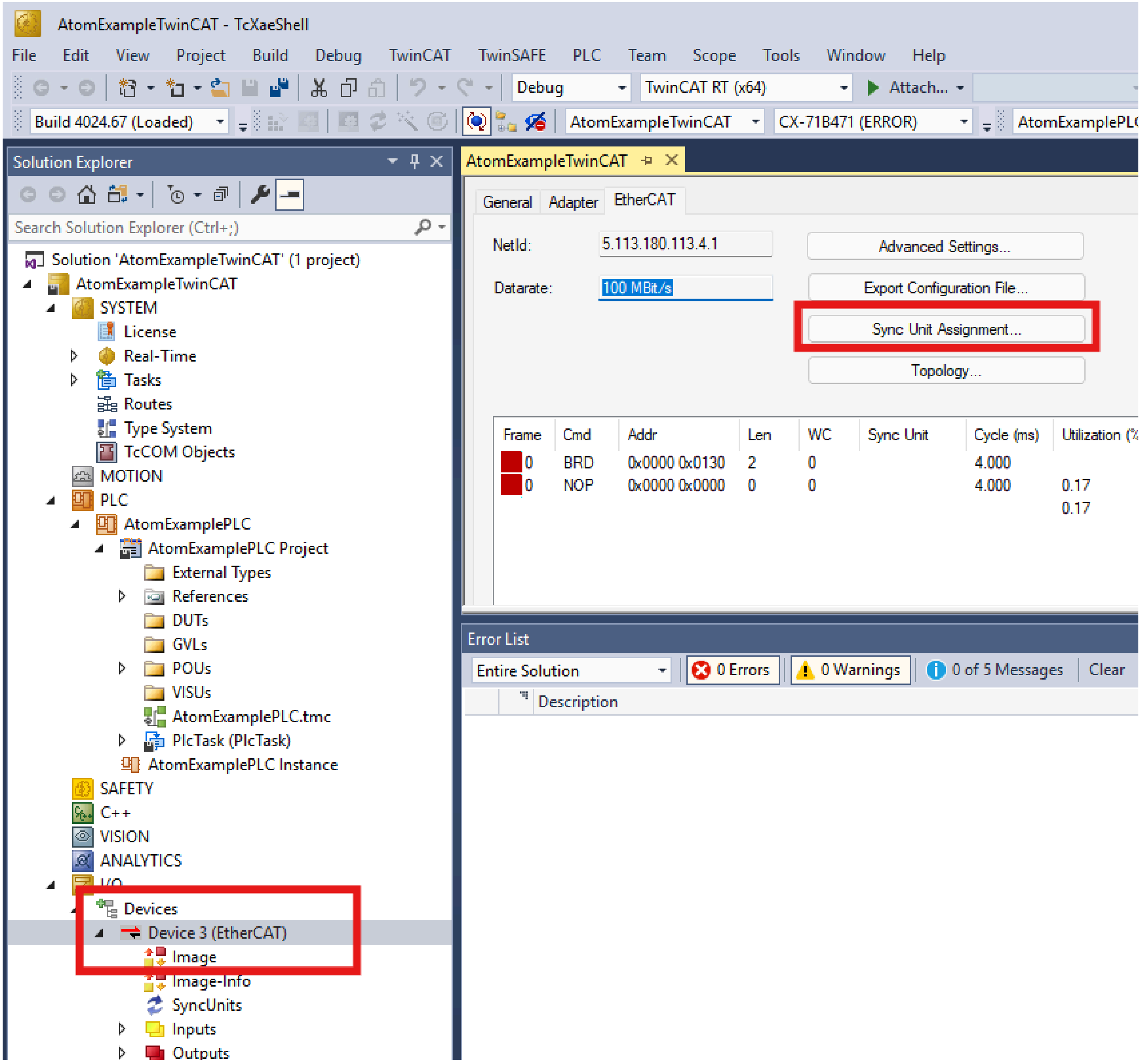


3. Select **License** and click **7 Days Trial License...*. Then complete the CAPTCHA:

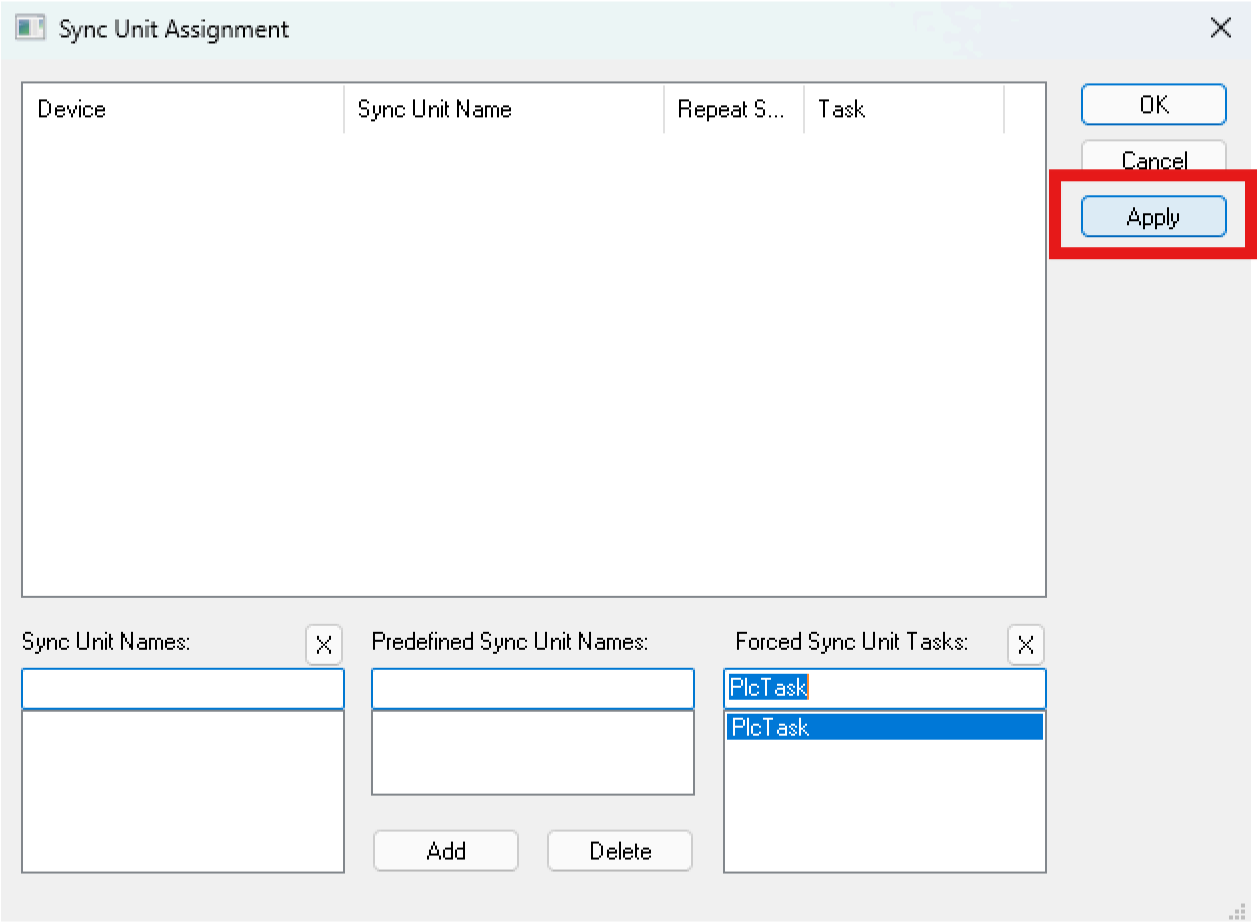




4. Select **Device 3 (EtherCAT)** and click **Sync Unit Assignment**:

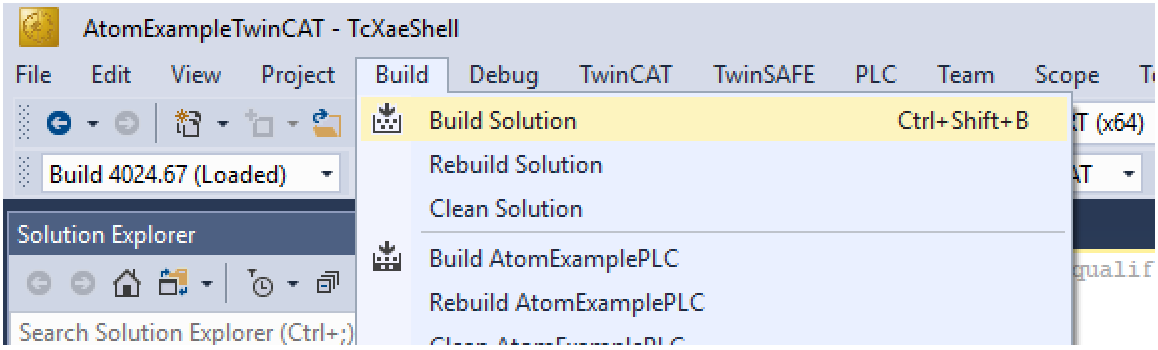


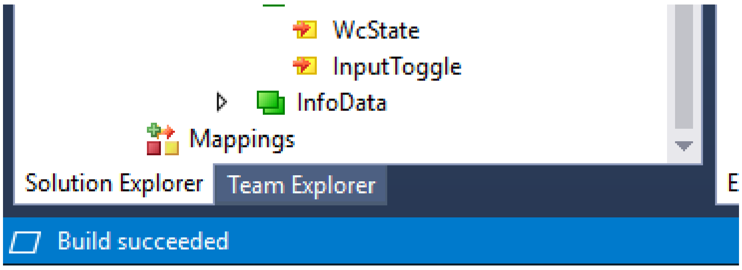
5. Select **PlcTask** and hit **Apply**:



> **NOTE**
>
> If you get an error message like "needs sync master (at least one variable linked to a task variable)", redo this step.

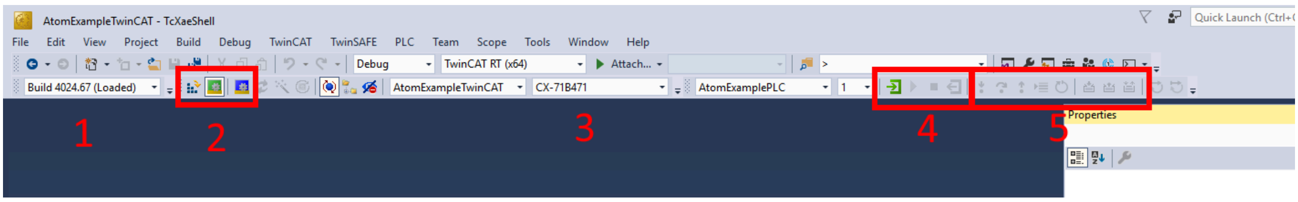
6. Select **Build** > **Build Solution**. If you configured everything correctly, you should see a message like `Build succeeded` in the lower left and no error messages should pop up:





Crash course in TwinCAT 3

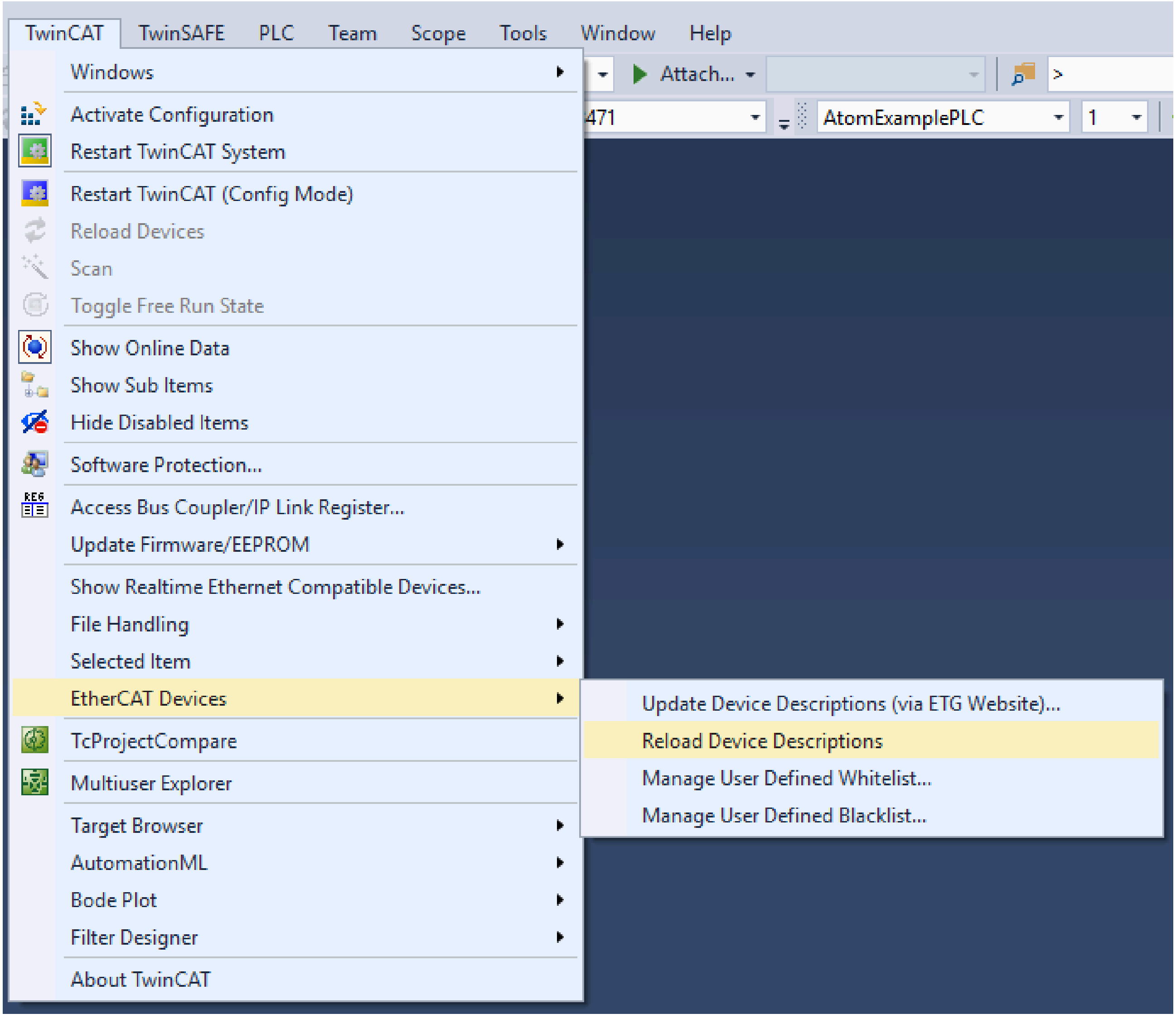
Most actions can be performed from the **TwinCAT** menu in the top bar. Here are some of the most common actions:



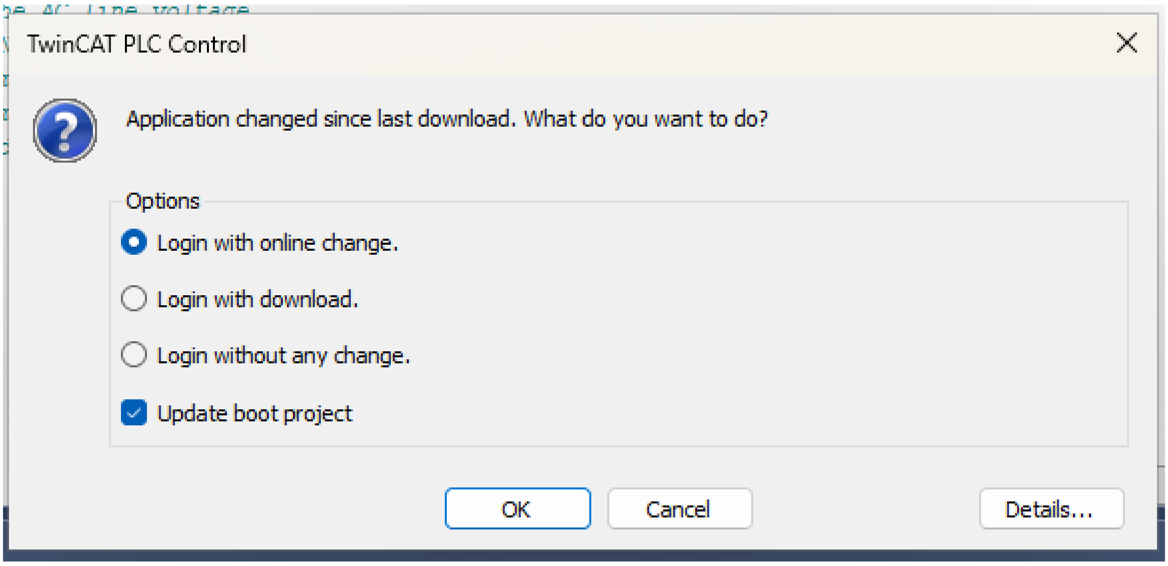
- `1` - The TwinCAT build version to use. This should match the version of TwinCAT on your PLC.
- `2` - From left to right:
  - `Activate the configuration`: Loads the PLC project onto the PLC as the boot project (the project the PLC will auto-start when it boots)
  - `Run`: Puts the TwinCAT system into run mode, allowing the PLC to execute the project

- `Config`: Puts the TwinCAT system into configuration mode, allowing you to modify the PLC project.
  - You may occasionally get a prompt reading `Activate free run?` - free run is a special PLC mode that allows you to edit EtherCAT variables on your I/O devices manually without having to put the PLC into run mode.
  - When adding new devices or installing new ESI files, you may need to click activate config mode even if config mode is already active. Clicking **Config** mode will restart the PLC in config mode.
- `3` - This is the PLC you want to program. `<Local>` is the built in soft PLC for testing & development.
- `4`: From left to right:
  - `Login` - Log in to download changes to the PLC and debug in real-time.
  - `Start` - Start PLC program execution.
  - `Stop` - Pause PLC program execution.
  - `Logout` - Log out of the PLC to make changes/modify the PLC program.
- `5`: Debugging tools. You can set breakpoints, step through code, force variables, and more.

If you install a new ESI file in `C:\TwinCAT\3.1\Config\Io\EtherCAT`, you may need to **Reload Device Descriptions** and restart the PLC in config mode:



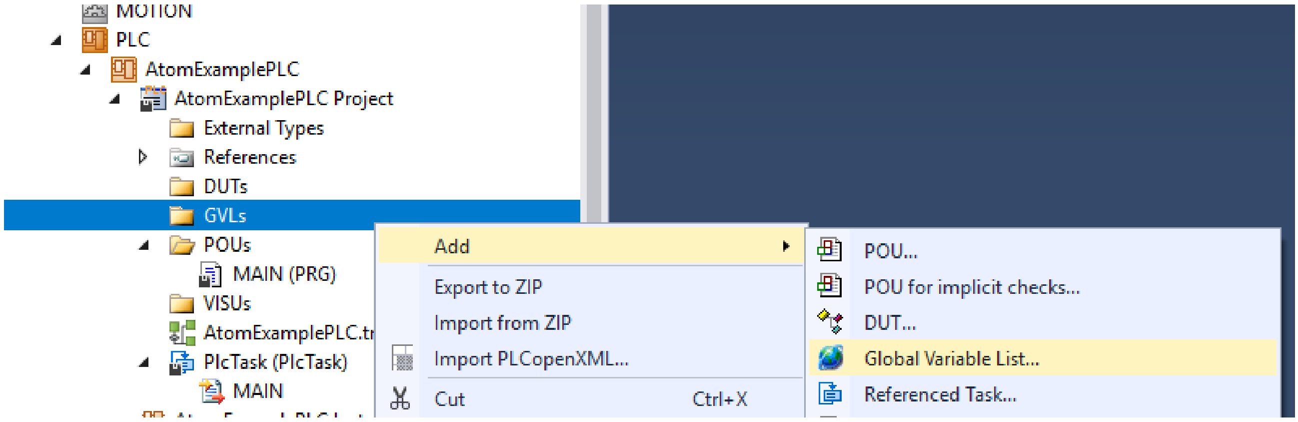
When you login to the PLC, you may be prompted with this dialog:

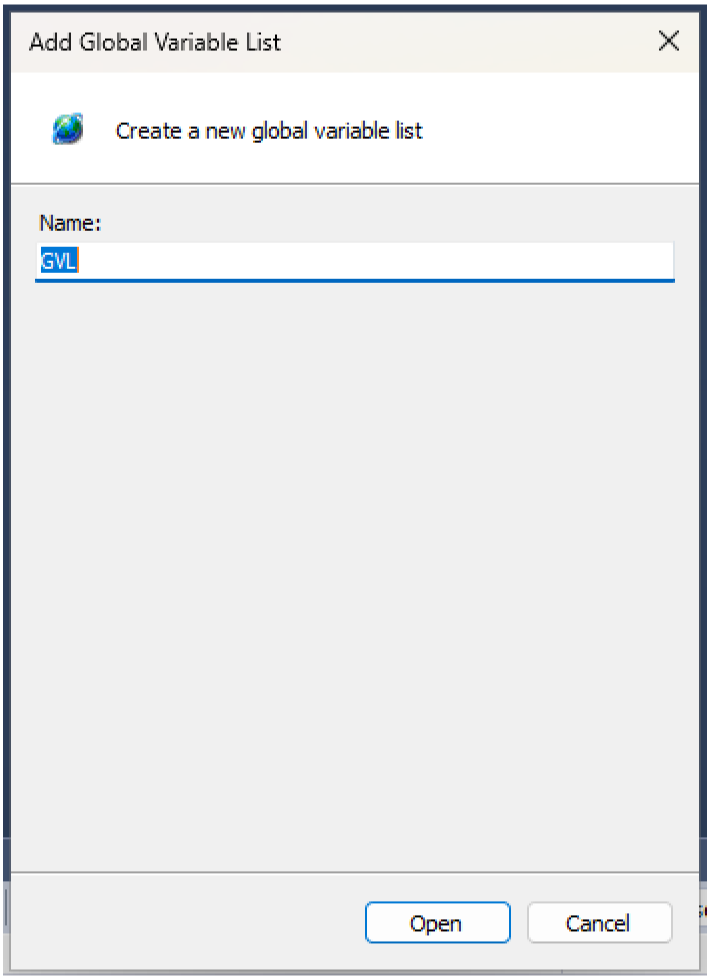


- `Login with online change` - Changes you made to the project with be pushed in real-time to the PLC without restarting the current program. Essentially, this pushes the "delta-updates".
- `Login with download` - The entire project is re-downloaded to the PLC, overwriting & restarting whatever program was running.
- `Login without any change` - Do not download any changes you made to the PLC.
- `Update boot project` - Update the default boot project to the one you are downloading.

A basic example program

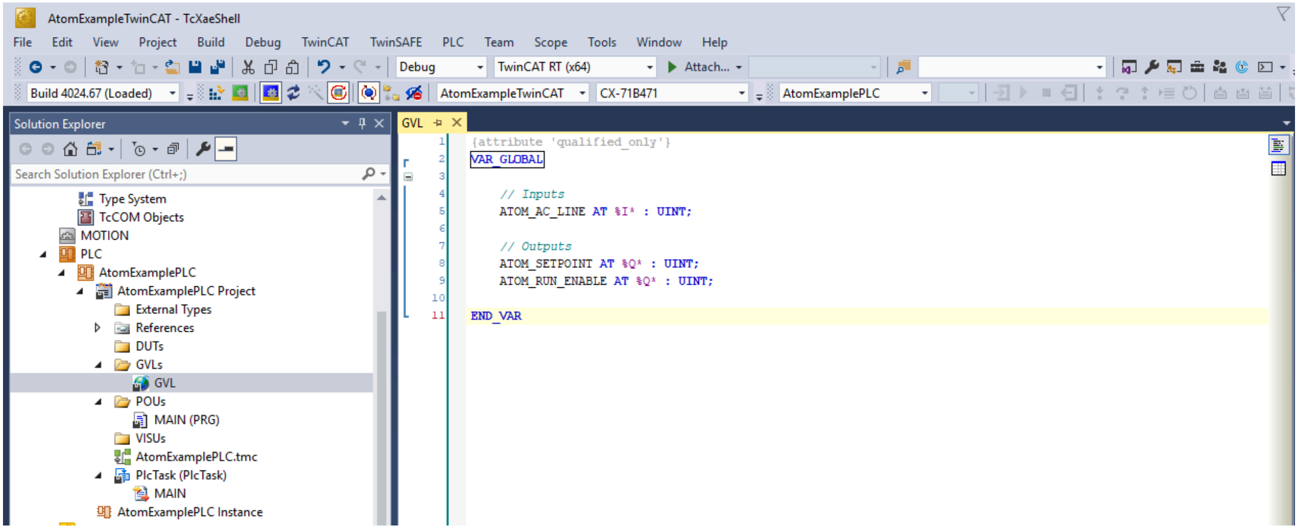
1. Right-click **GVLs** and select **Add** > **Global Variable List**. Name it `GVL` and click **Add**:





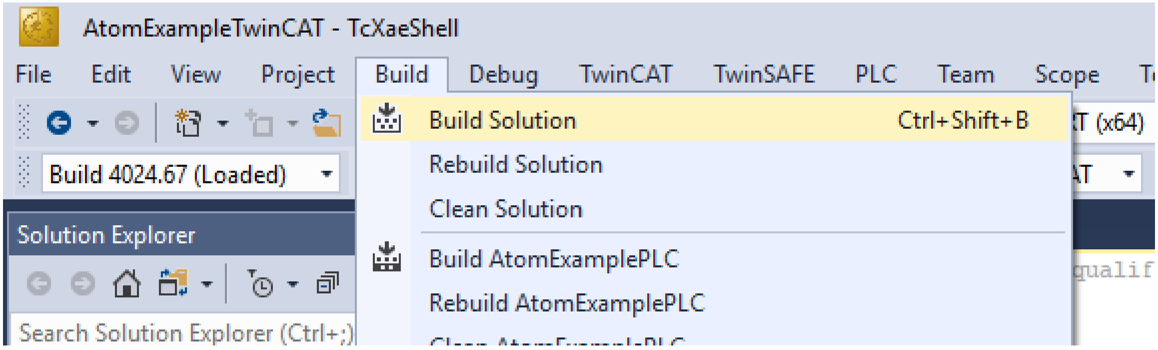
2. Create three variables:

```
  // Inputs
  ATOM_AC_LINE AT %I* : UINT; // AC Line Voltage in Volts as reported by ATOM

  // Outputs
  ATOM_SETPOINT AT %Q* : UINT; // The setpoint/output command to ATOM
  ATOM_RUN_ENABLE AT %Q* : UINT; // Put ATOM in RUN/STOP
```

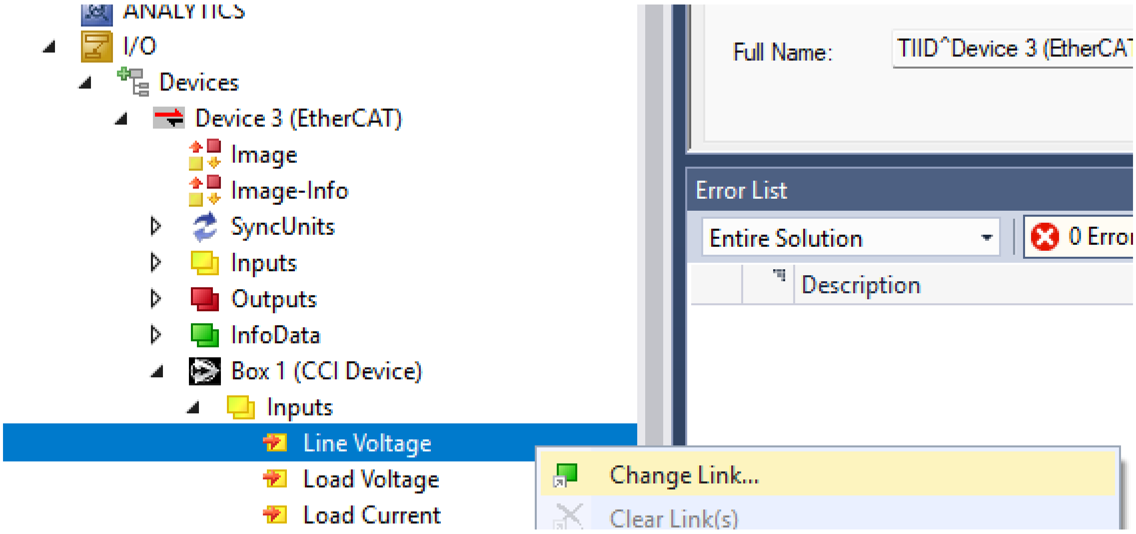


> **INFO**
>
> Global variables take the format: `<NAME> AT %<I/Q>* : <DATA-TYPE>;`. Use `%I*` for inputs and `%Q*` for outputs. All ATOM EtherCAT variables are `UINT` (unsigned integers).

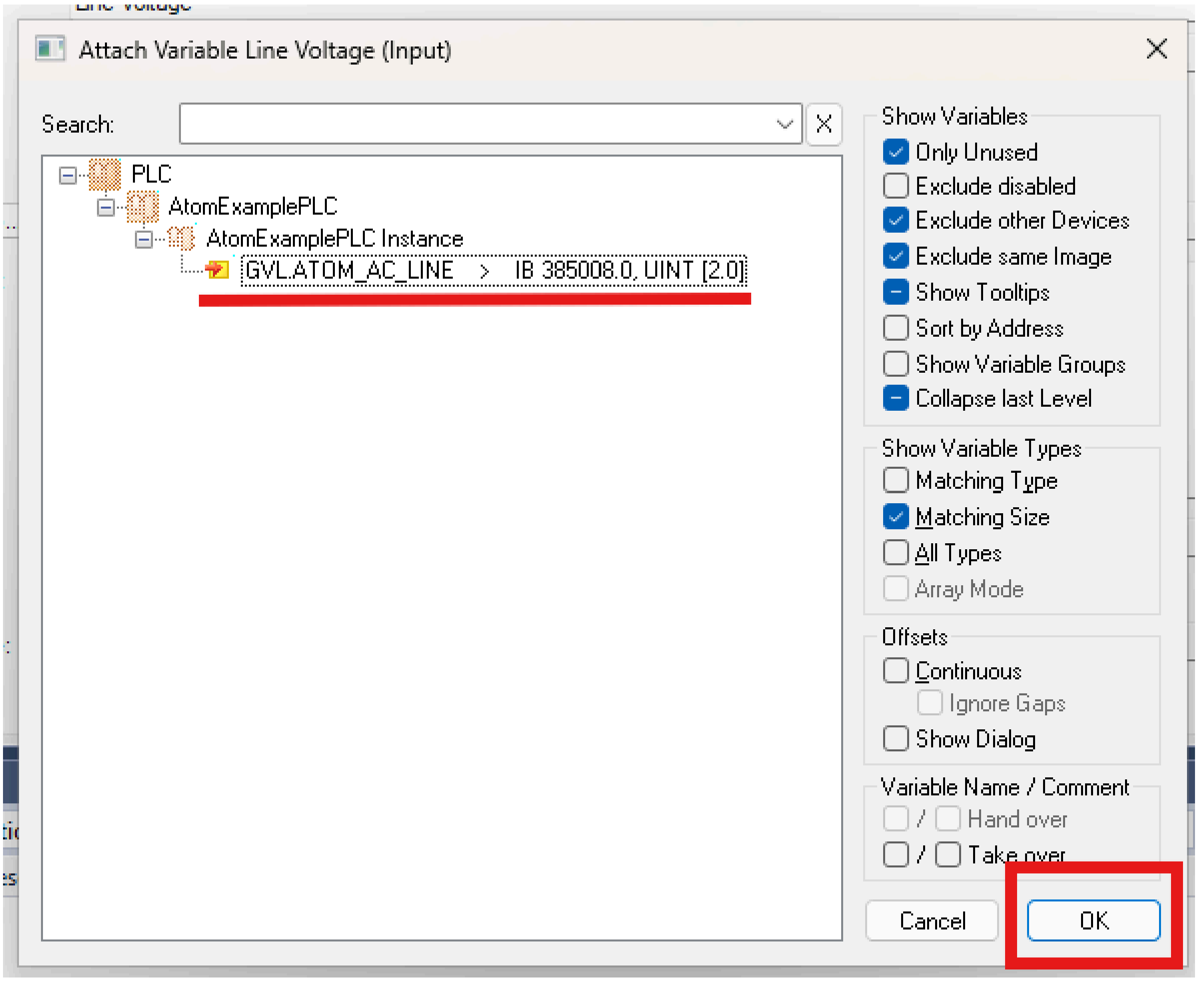
3. You must build the project to register the global variables. Select **Build** > **Build Solution**. If you configured everything correctly, you should see a message like `Build succeeded` in the lower left and no error messages should pop up:



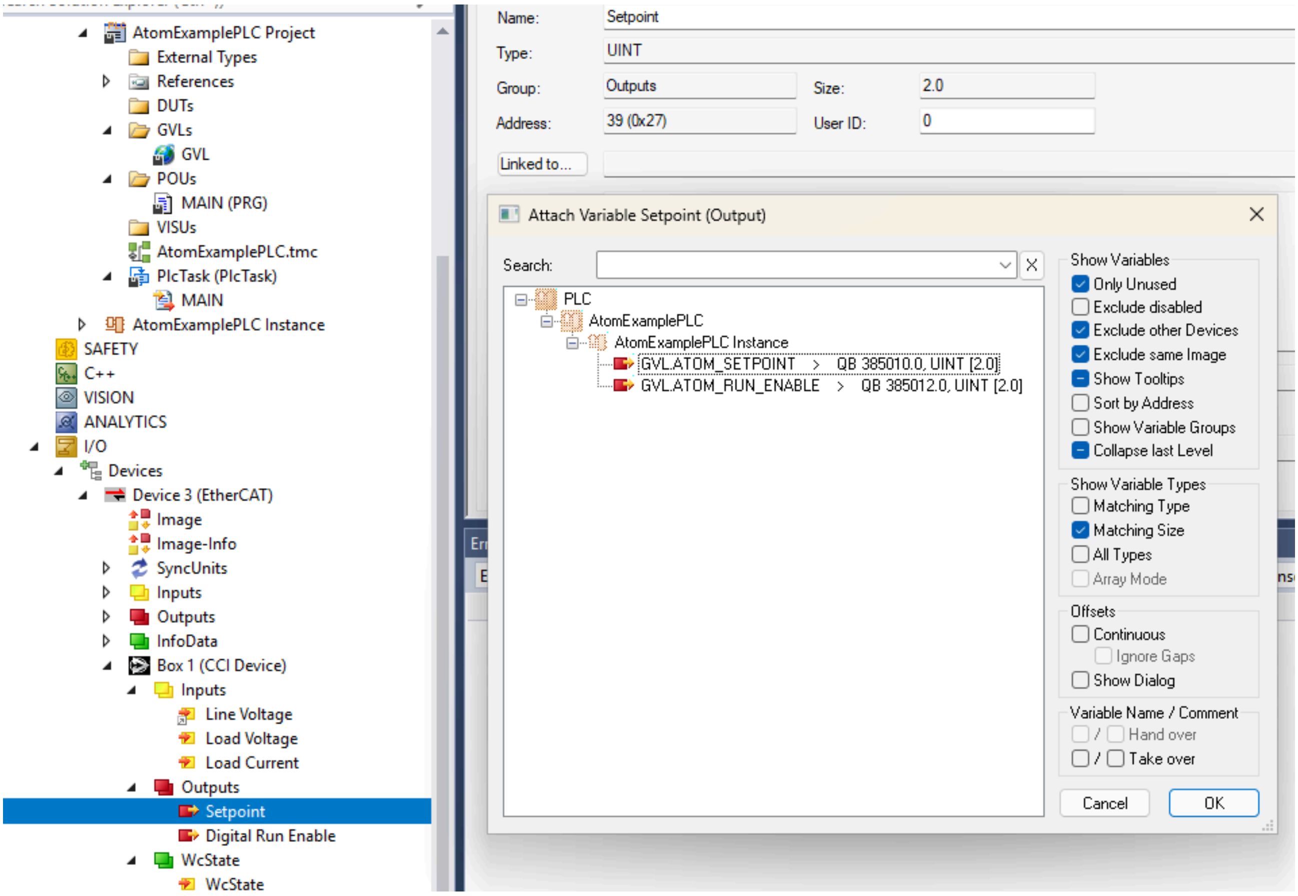
4. Right-click `Box 1 (CCI Device)` > `Inputs` > `Line Voltage` and select **Change Link...*:



5. Select the corresponding global variable `GVL.ATOM_AC_LINE` and click **OK**:

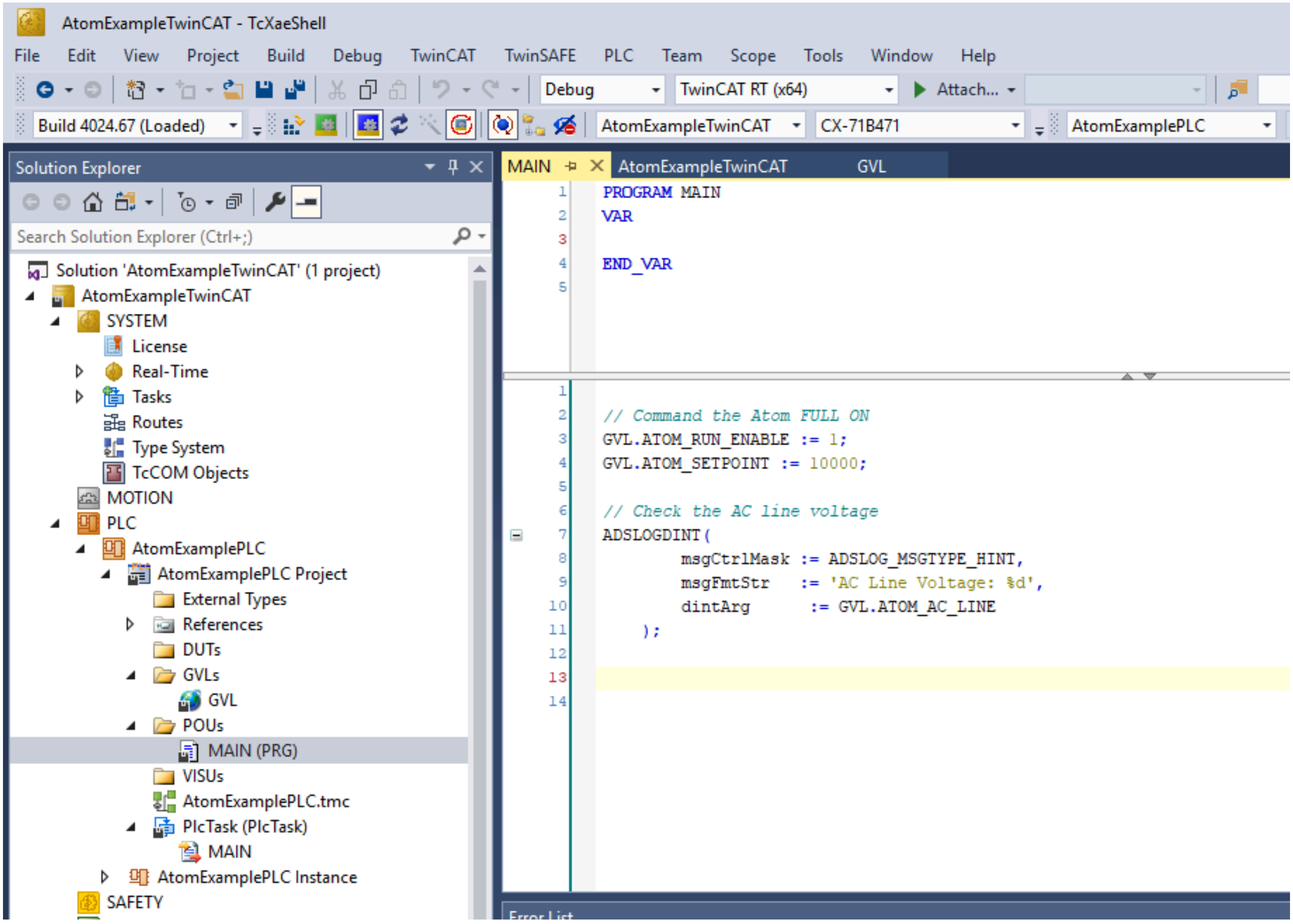


6. Repeat the process for `Outputs` > `Setpoint` and `Outputs` > `Digital Run Enable`:

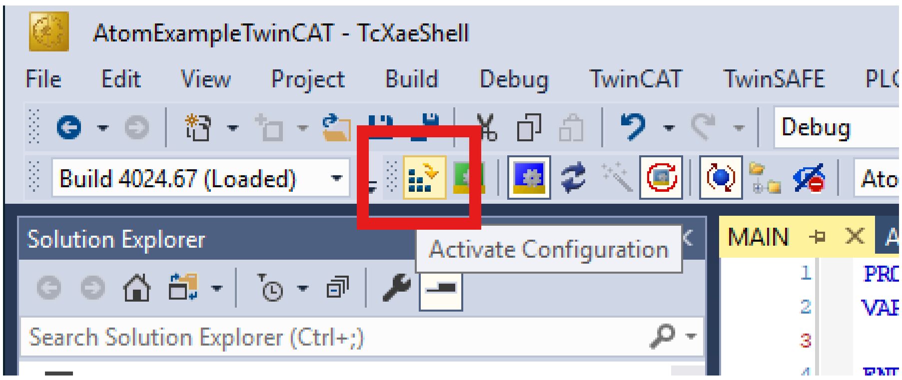


7. Select the main program `MAIN (PRG)` and add the following code:

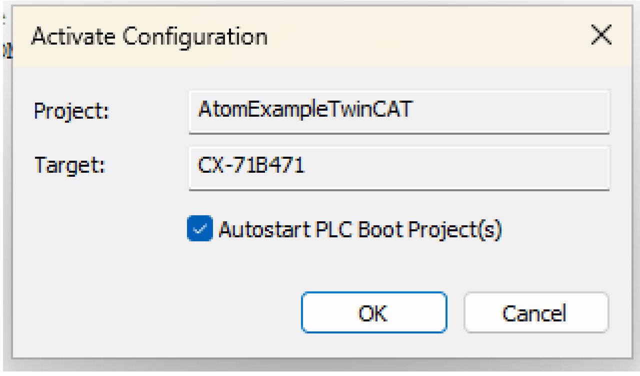
```
// Command the Atom FULL ON
GVL.ATOM_RUN_ENABLE := 1;
GVL.ATOM_SETPOINT := 10000;

// Print the AC line voltage:

ADSLOGDINT(
  msgCtrlMask := ADSLOG_MSGTYPE_HINT,
  msgFmtStr := 'AC Line Voltage: %d V',
  msgArgs := GVL.ATOM_AC_LINE
);
```



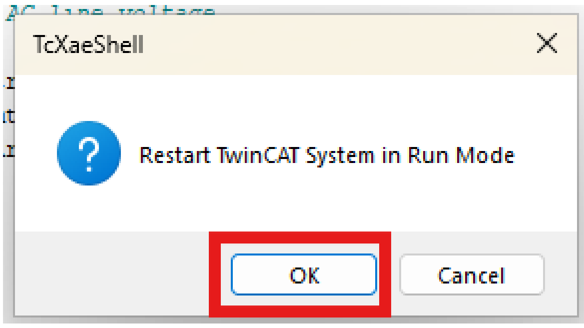
8. Activate the configuration:



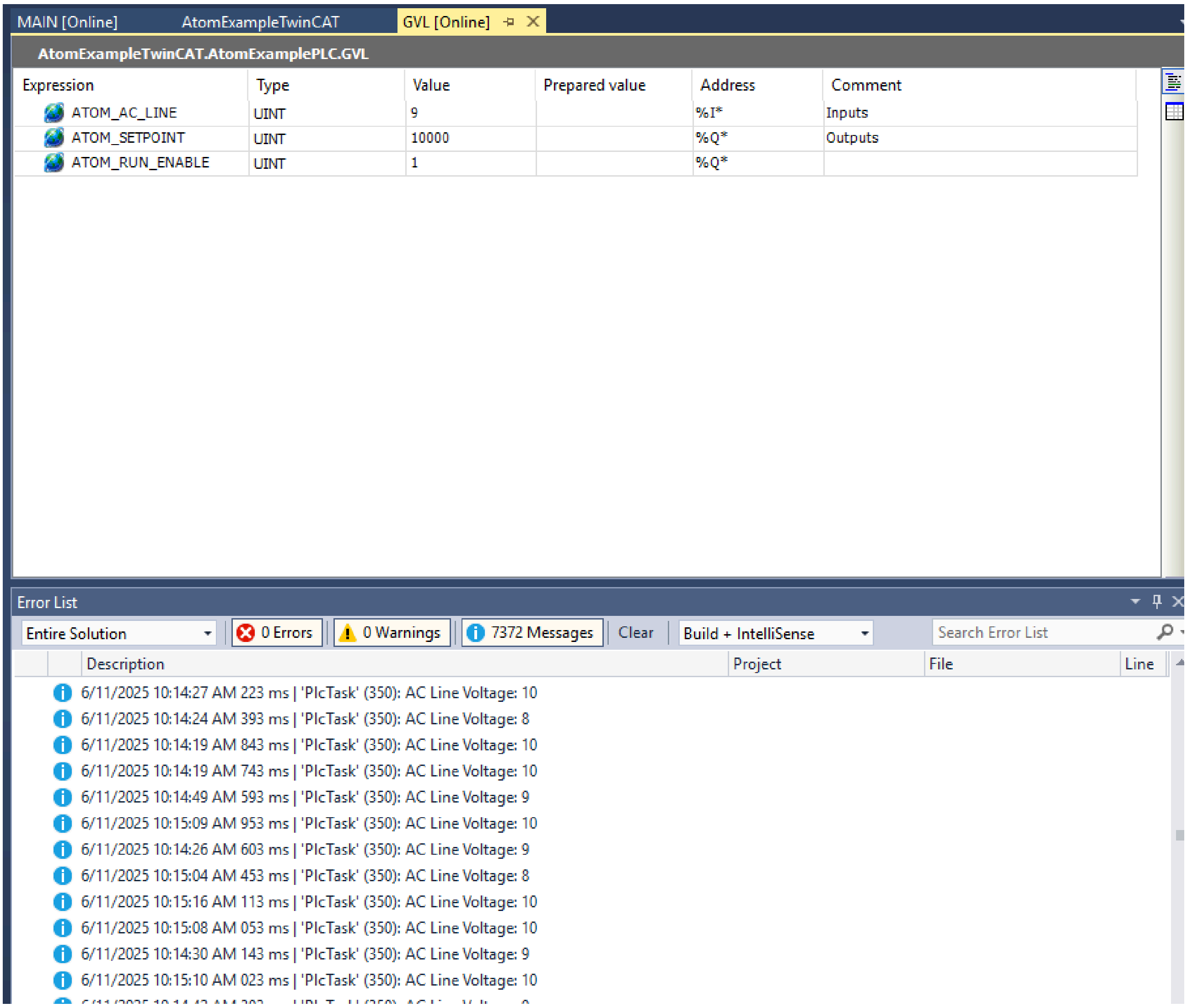
9. Check **Autostart PLC Boot Project(s)** and click **OK**:



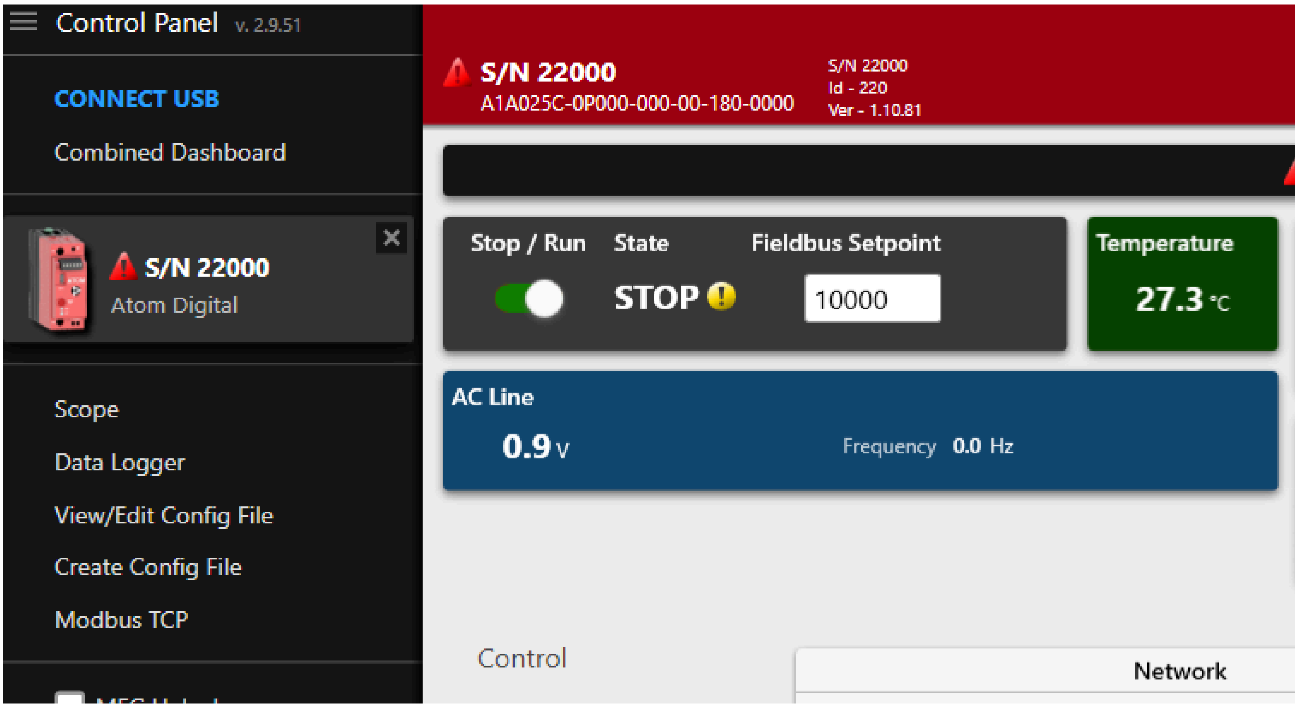
10. When prompted to **Restart TwinCAT System in Run Mode**, click **Yes**:



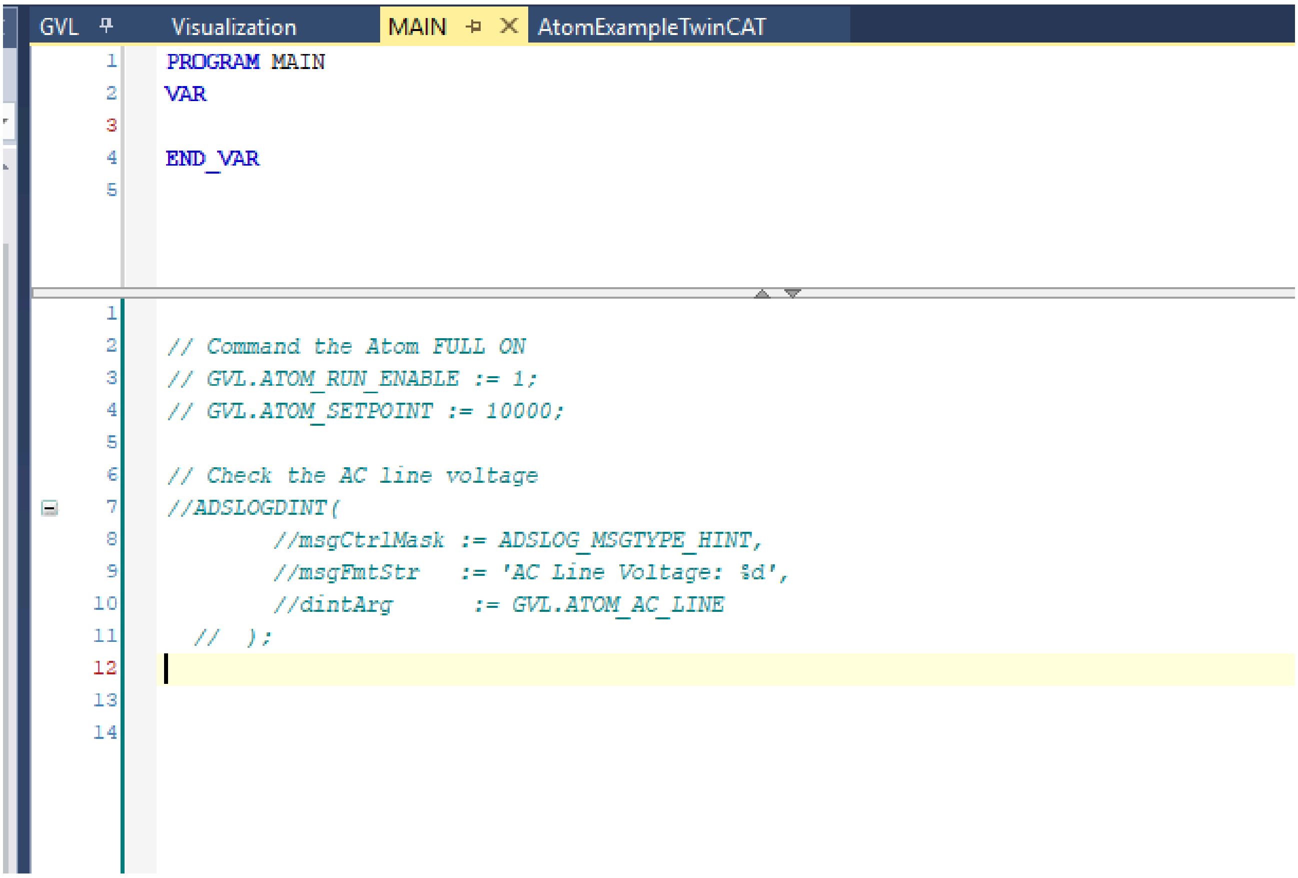
11. Login to the PLC by clicking the **Login** button in the top bar. You should see the AC Line Voltage updated in the debug window and printed to the console:



Additionally, if you connect to ATOM with Control Panel over USB, you can see the ATOM went into RUN mode and the output setpoint was set to `10000` (which is `100.0%` of the output):



Once you've verified that ATOM is working, you can comment out the code in `MAIN (PRG)`:



Next, we'll create a simple user interface to control ATOM. You can follow along with either the Structured Text or Ladder Logic examples below.

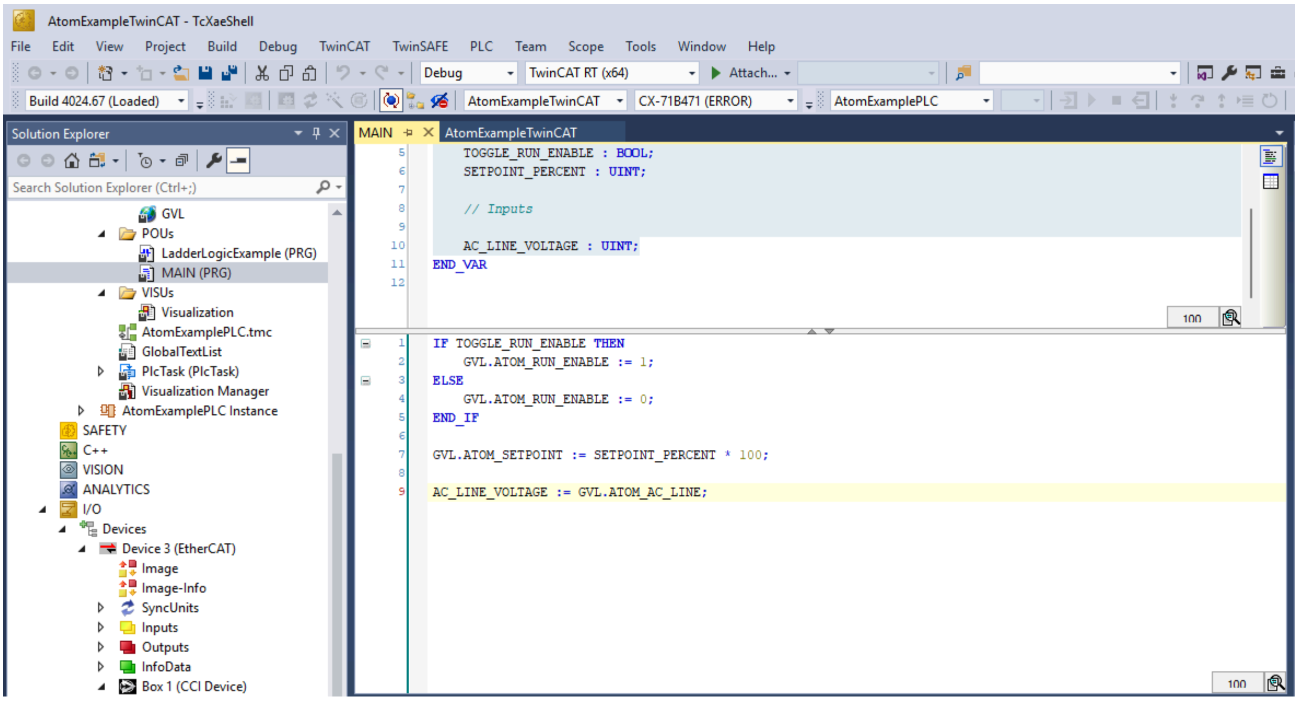
Structured Text

1. Define the variables:

```
// Outputs

TOGGLE_RUN_ENABLE : BOOL;
SETPOINT_PERCENT : UINT;

// Inputs

AC_LINE_VOLTAGE : UINT;
```

Add the following code:

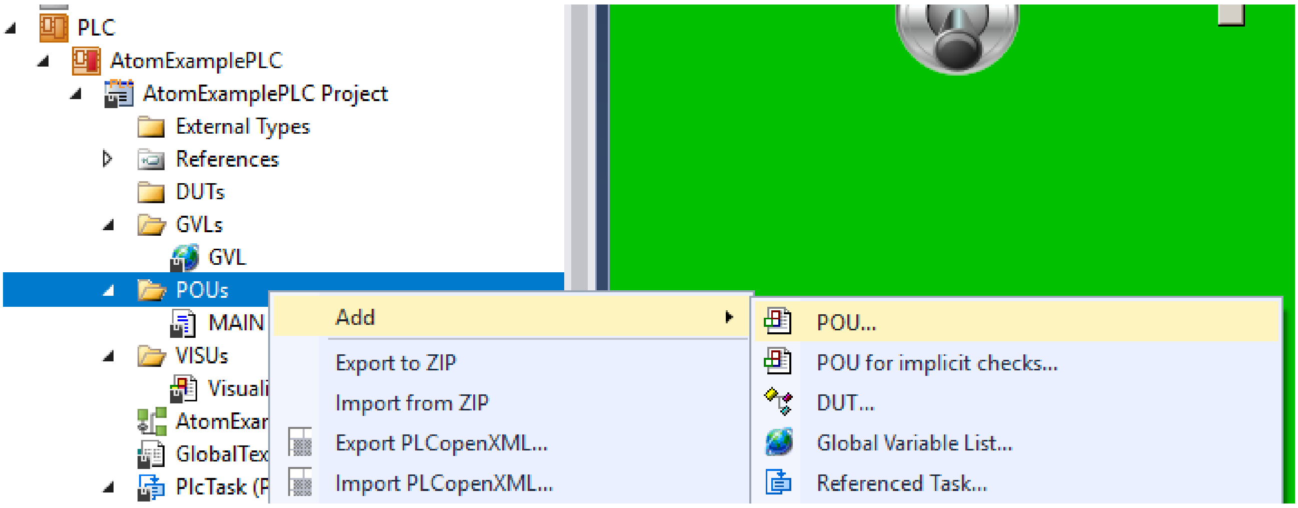
```
IF TOGGLE_RUN_ENABLE THEN
        GVL.ATOM_RUN_ENABLE := 1;
ELSE
        GVL.ATOM_RUN_ENABLE := 0;
END_IF

GVL.ATOM_SETPOINT := SETPOINT_PERCENT * 100;

AC_LINE_VOLTAGE := GVL.ATOM_AC_LINE;
```



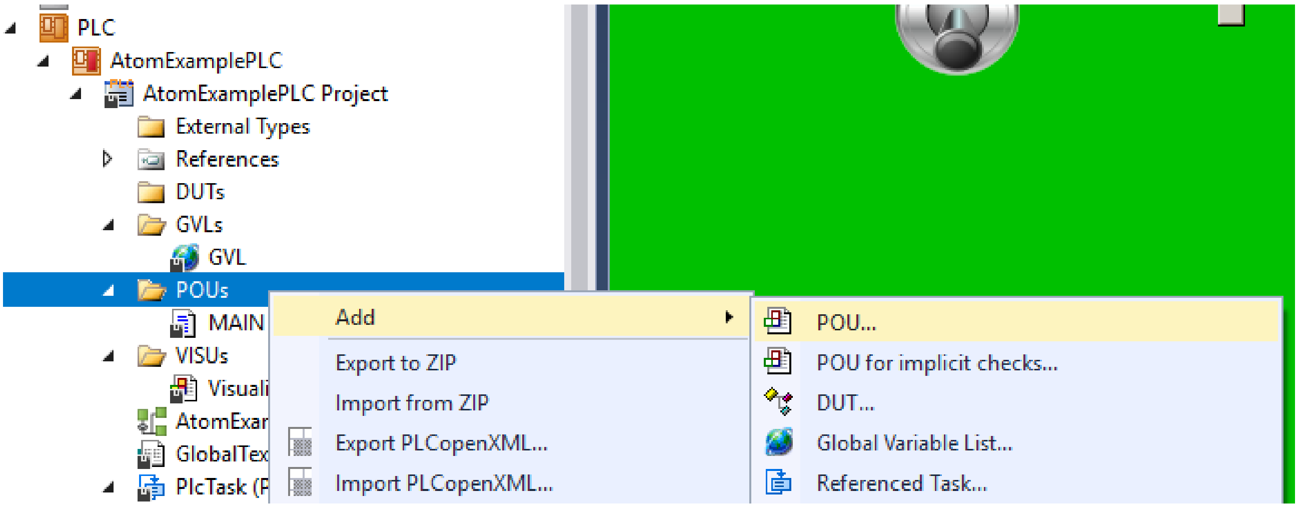
Next, head to the interface section to create a simple user interface to control ATOM.

Ladder logic

Right click **POUs** and select **Add > POU...**



Set the name to `LadderLogicExample`, set type to `Program` and select **Ladder Logic Diagram (LD)** as the Implementation language:

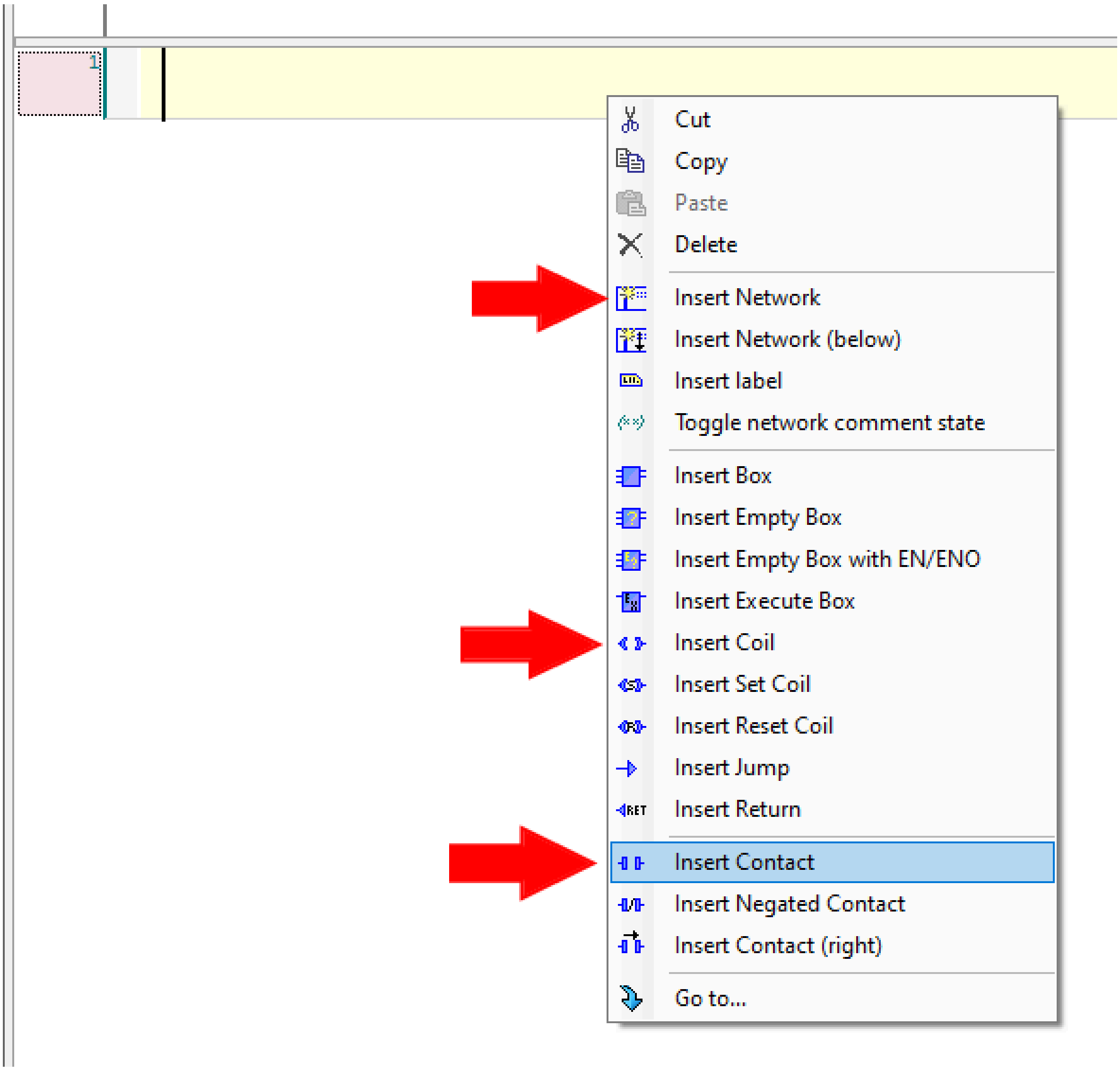


Copy the following code into the top panel of the **LadderLogicExample** editor:

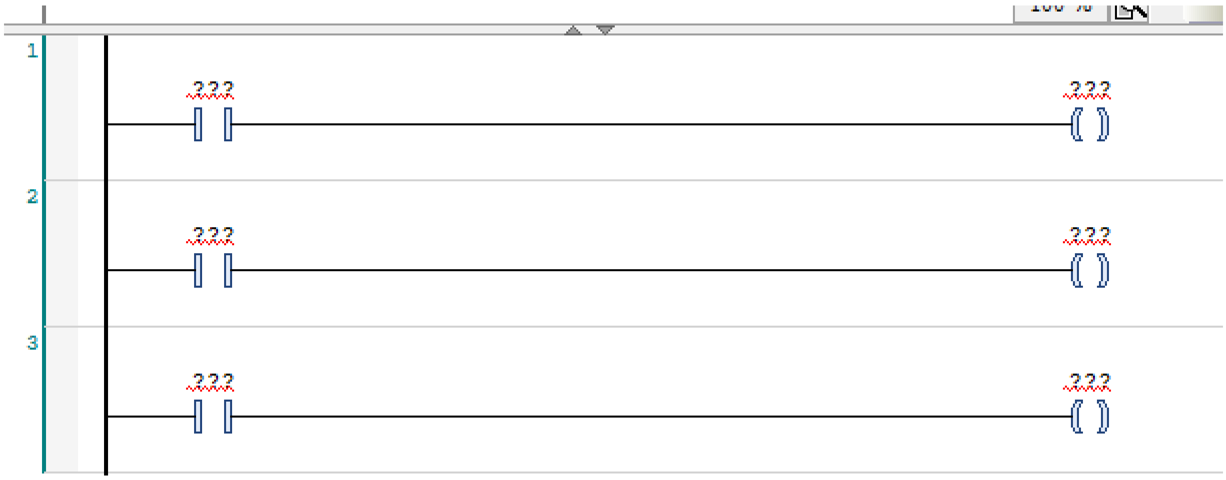
```
// ...
VAR_OUTPUT
        TOGGLE_RUN_ENABLE: BOOL;
  SETPOINT_PERCENT : UINT;
END_VAR

VAR_INPUT
        AC_LINE_VOLTAGE : UINT;
END_VAR
```

In the bottom panel of the editor, we'll create a simple ladder logic program using the variables we just added above.

1. Create **3** networks total by right-clicking and selecting **Insert Network**
2. For each network, right click and insert **one** contact and **one** coil



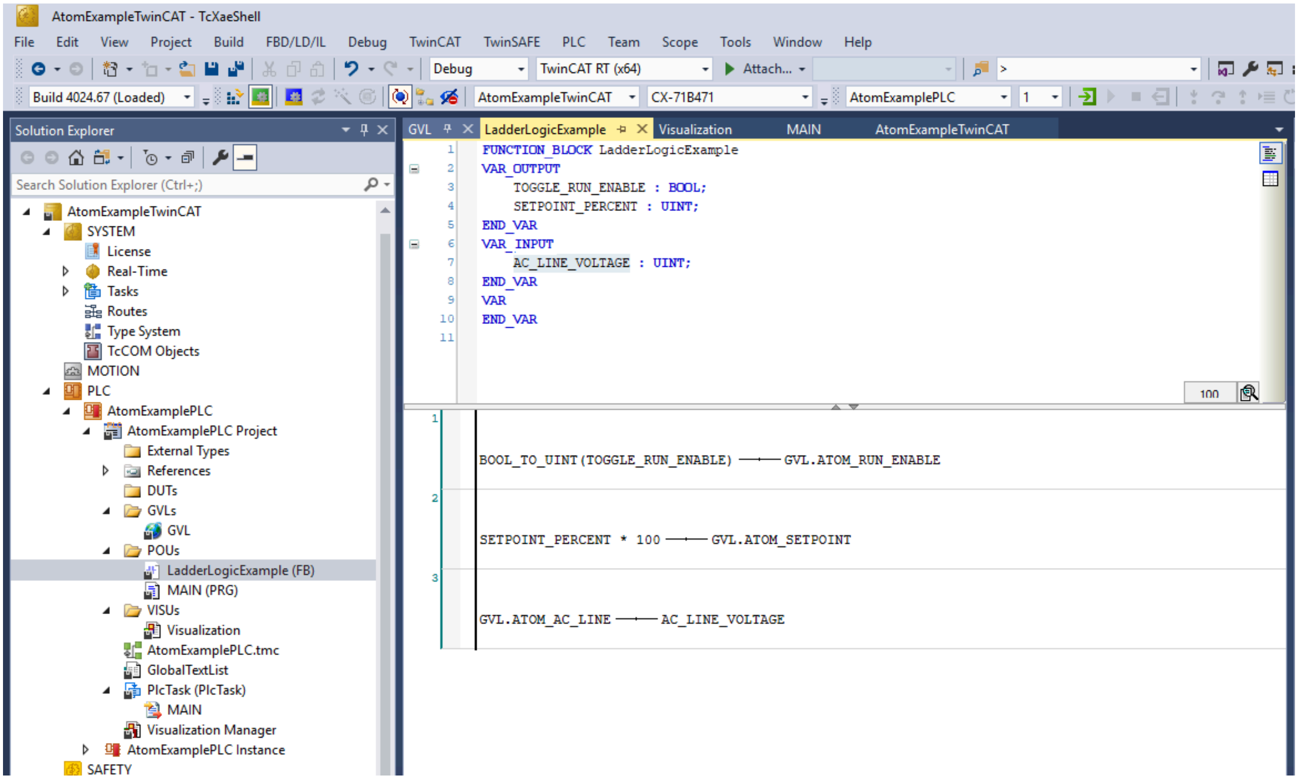
After you're finished, your ladder logic program should look like:



For each rung, replace the `???` with the corresponding variables:

1. **Rung #1** - `BOOL_TO_UINT(TOGGLE_RUN_ENABLE)` and `GVL.ATOM_RUN_ENABLE`
2. **Rung #2** - `SETPOINT_PERCENT * 100` and `GVL.ATOM_SETPOINT`
3. **Rung #3** - `GVL.ATOM_AC_LINE` and `AC_LINE_VOLTAGE`

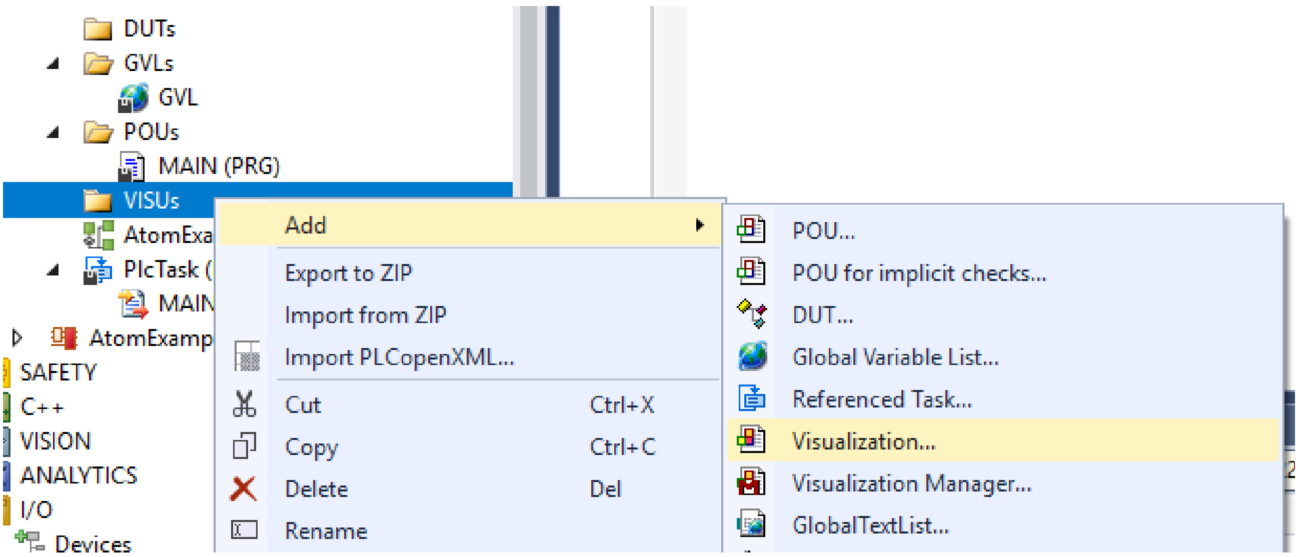
After you're finished, your ladder logic program should look like:



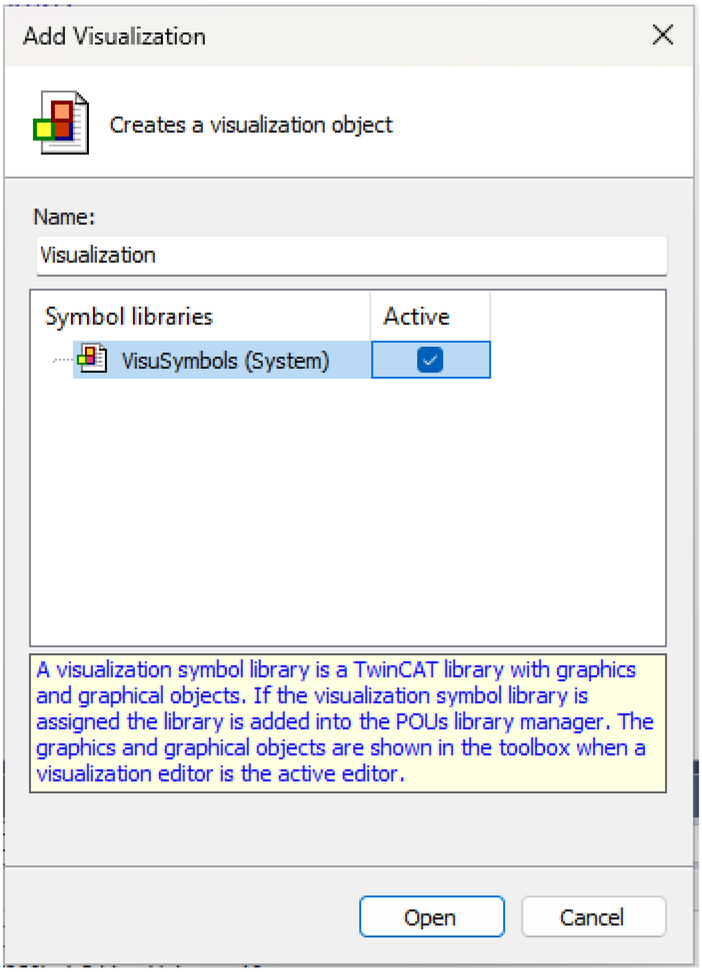
Next, head to the interface section to create a simple user interface to control ATOM.

Creating a user interface

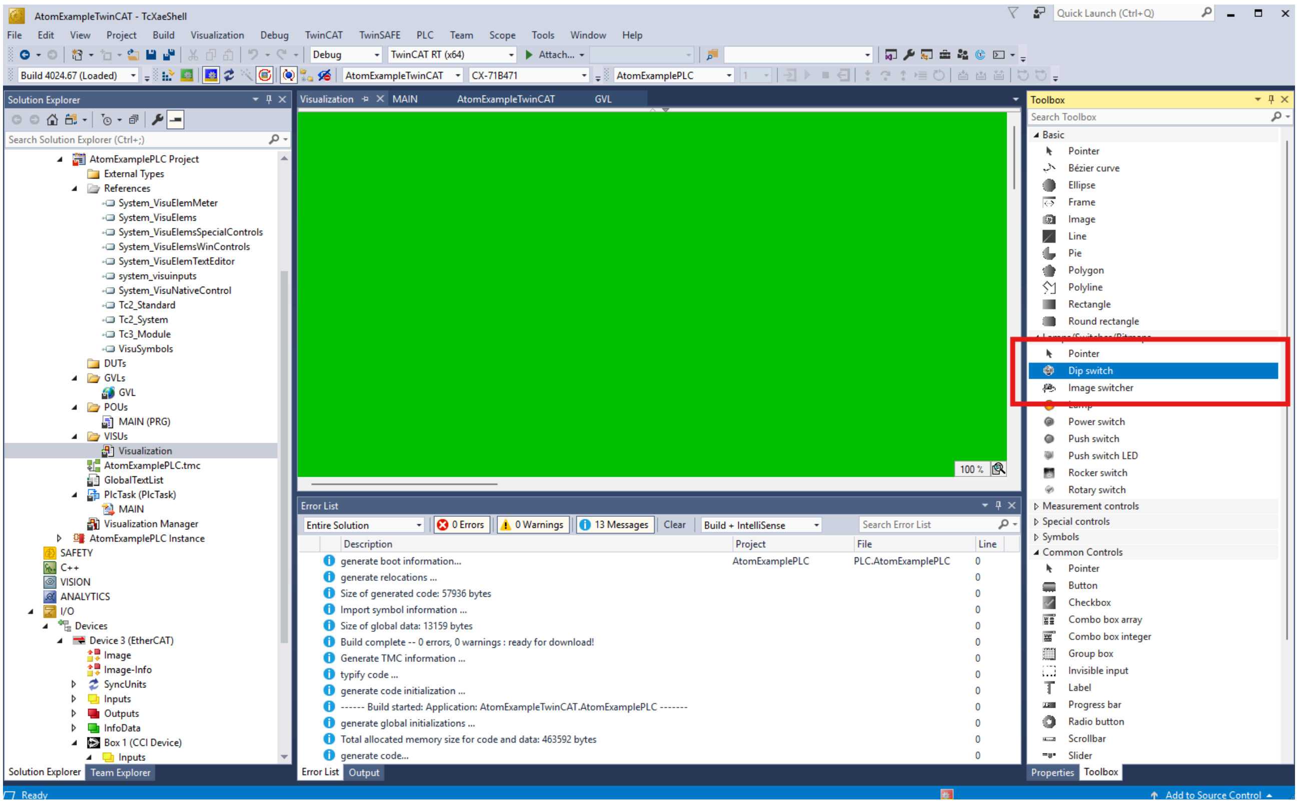
1. Right click **VISUs** and select **Add > Visualization...**:



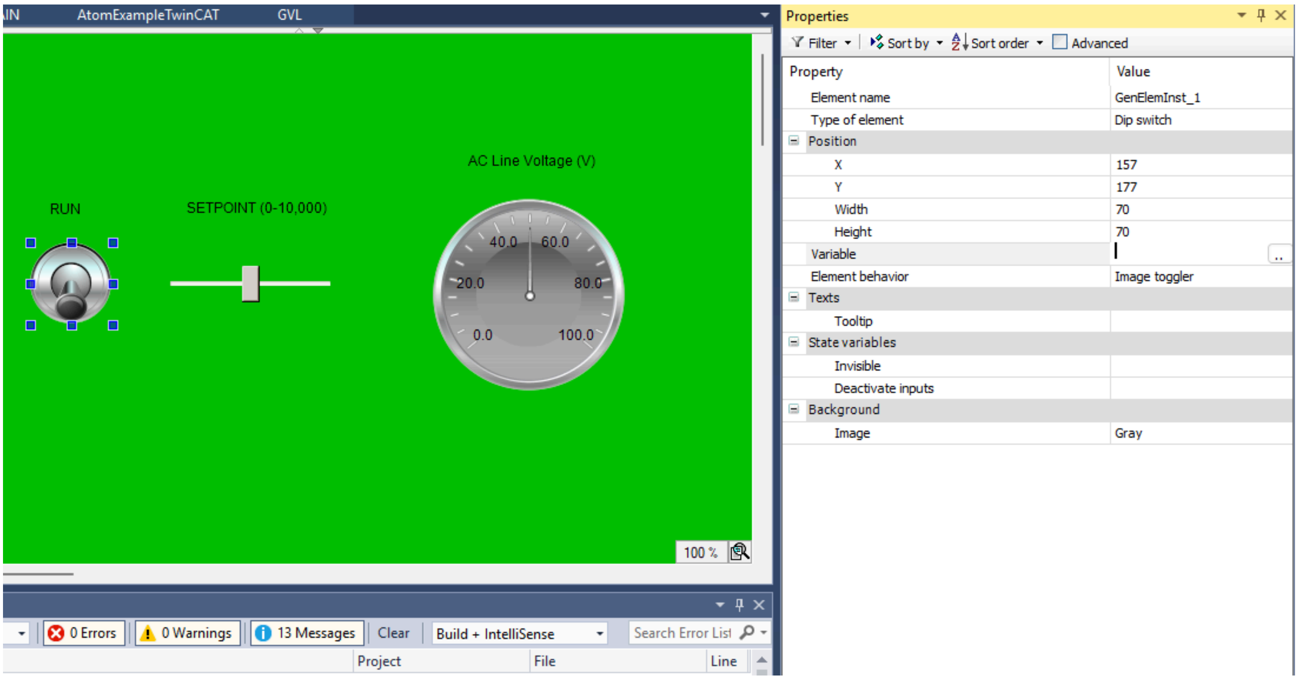
2. Name it `Visualization` and check **Active** on `VisuSymbols (System)`, then hit `Open`:



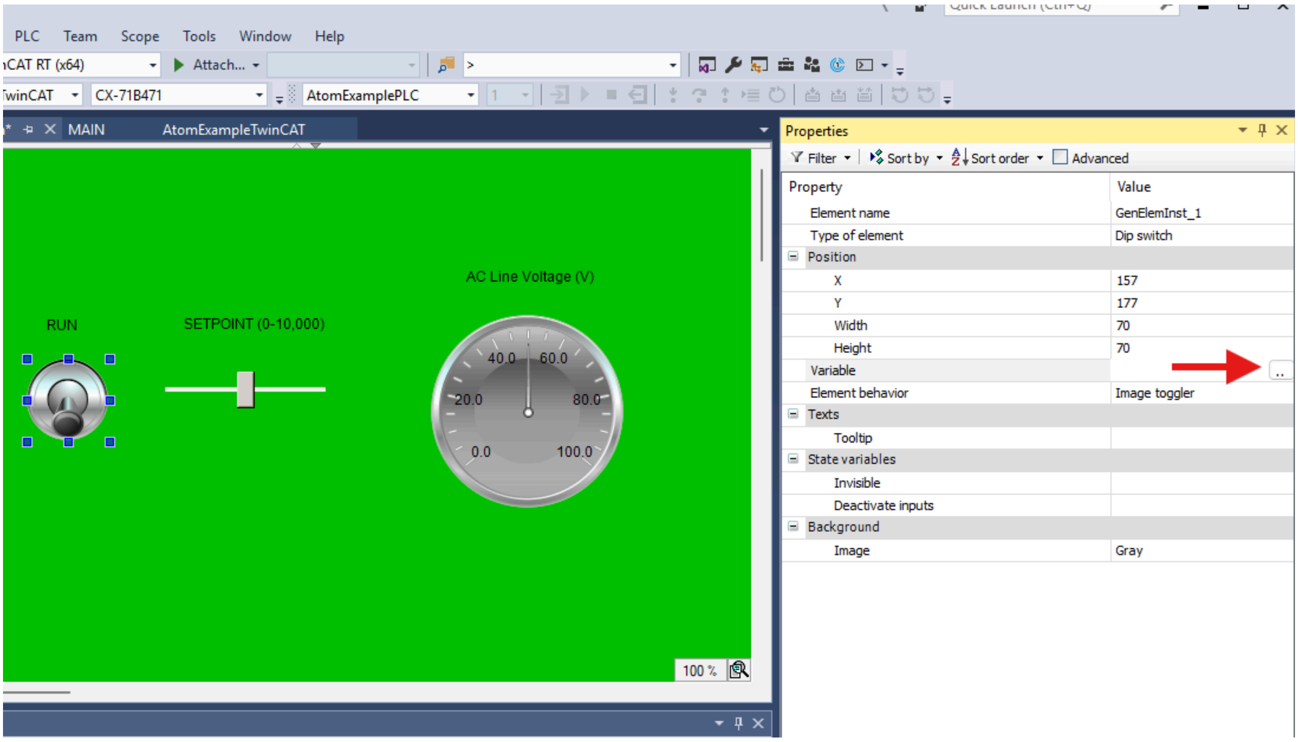
3. Select components in to the toolbox on the right and drag them onto the canvas:

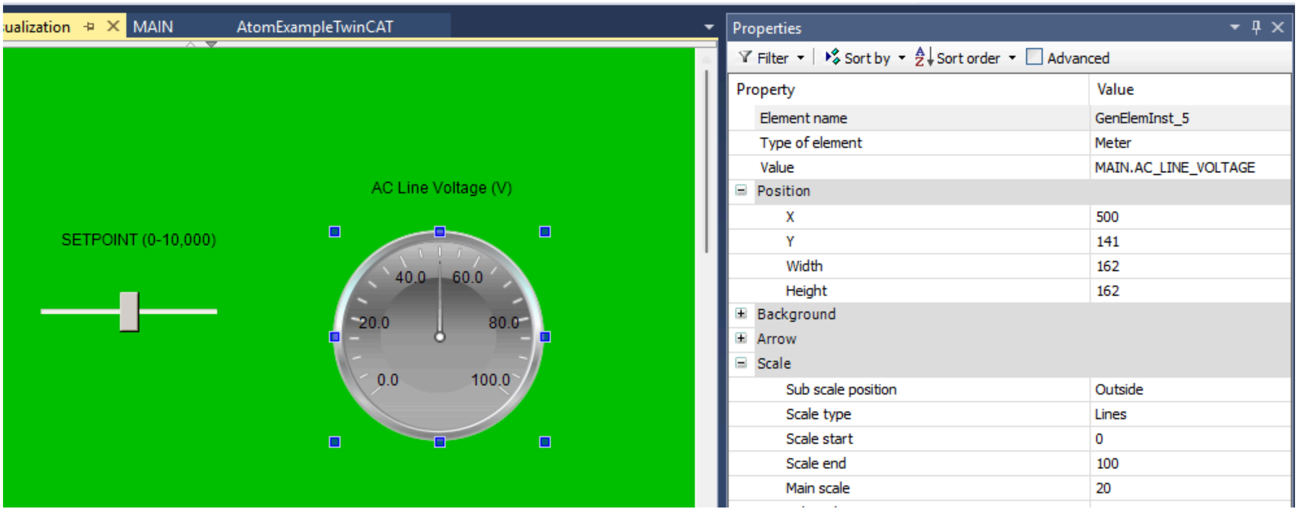


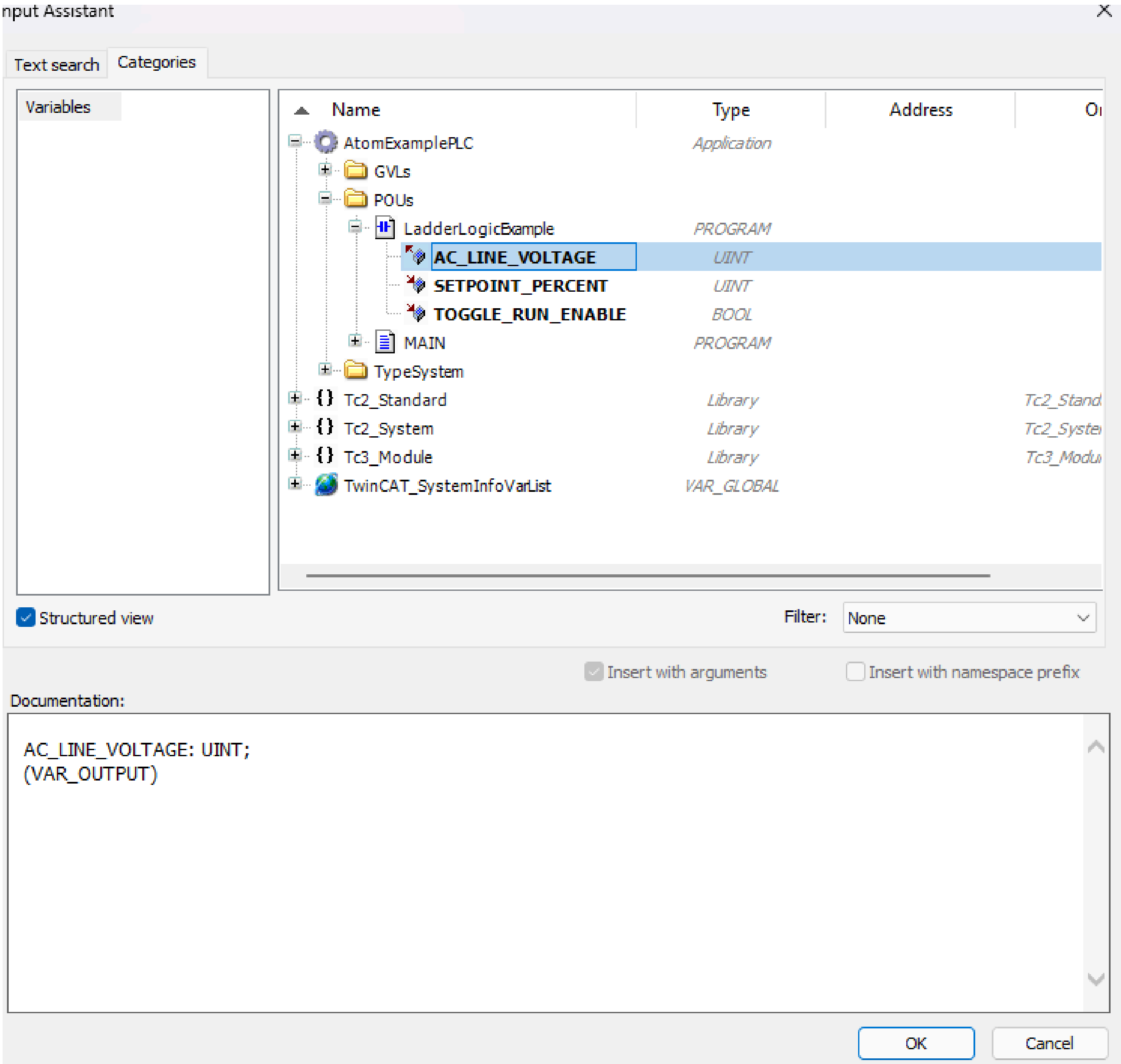
4. In this example, we will add:
   - A **Dip switch** to toggle the `Run Enable` state
   - A **Slider** to set the output setpoint percentage
   - A **Meter** to display the AC line voltage



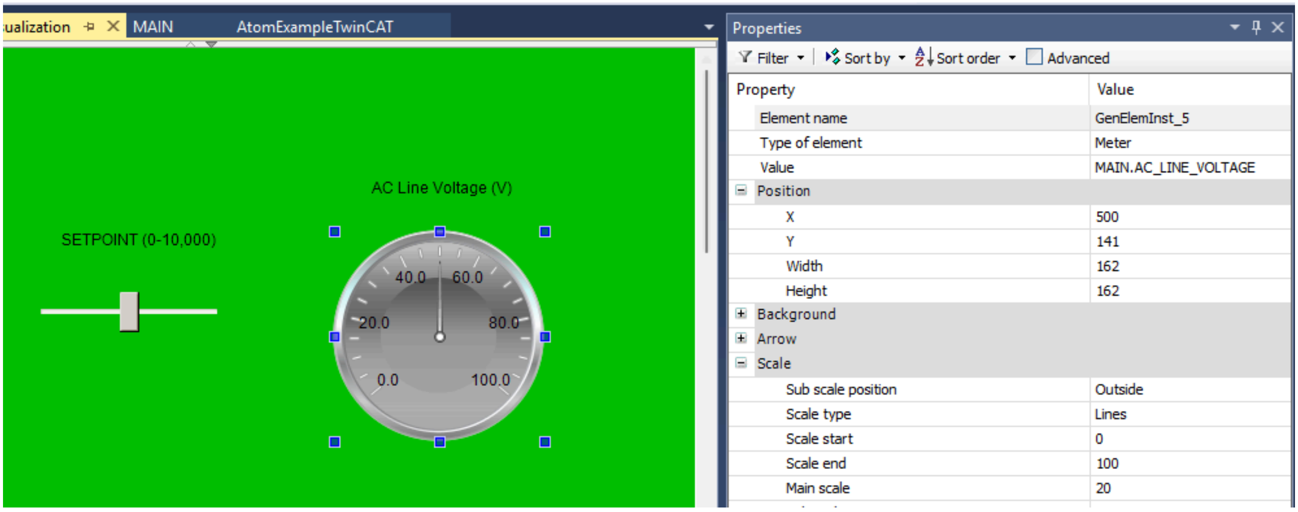
5. Connect the components to the variables we defined earlier:
    - For the **Dip switch**, set the `Variable` property to `MAIN.TOGGLE_RUN_ENABLE`, if using structured text, or `LadderLogicExample.TOGGLE_RUN_ENABLE` if using ladder logic.
    - For the **Slider**, set the `Variable` property to `MAIN.SETPOINT_PERCENT`, if using structured text, or `LadderLogicExample.SETPOINT_PERCENT` if using ladder logic.
    - For the **Meter**, set the `Value` property to `MAIN.AC_LINE_VOLTAGE` if using structured text, or `LadderLogicExample.AC_LINE_VOLTAGE` if using ladder logic.

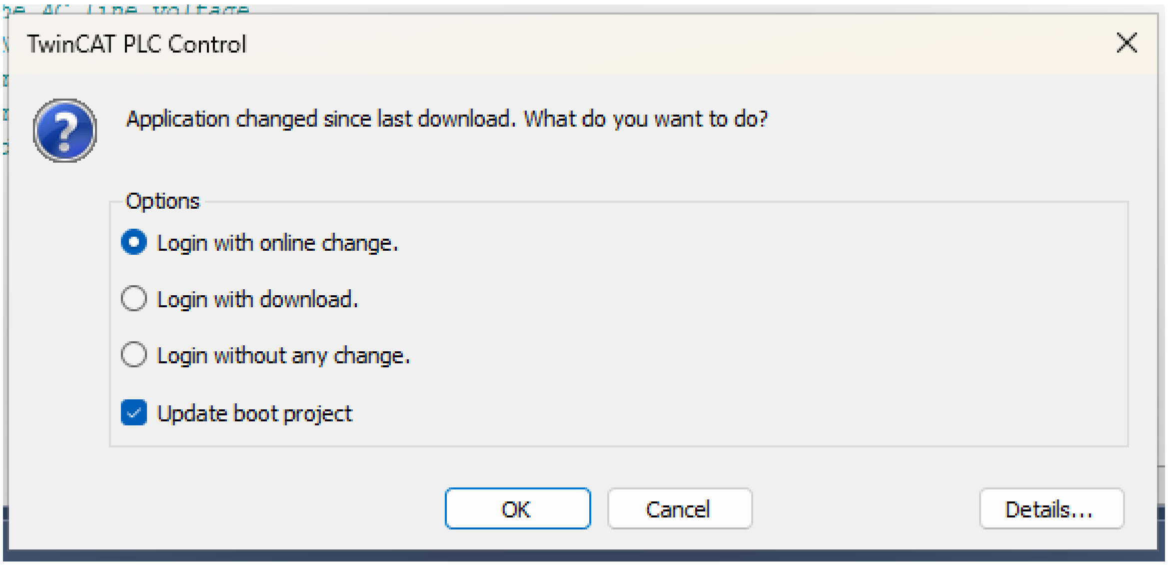

PLC Team Scope Tools Window Help
Attach...
CX-71B471
AtomExamplePLC
MAIN
AtomExampleTwinCAT
AC Line Voltage (V)
RUN
SETPOINT (0-10,000)
40.0
60.0
20.0
80.0
0.0
100.0
100 %
Properties
Filter
Sort by
Sort order
Advanced
Property
Value
Element name
GenElemInst_1
Type of element
Dip switch
Position
X
157
Y
177
Width
70
Height
70
Variable
Element behavior
Image toggler
Texts
Tooltip
State variables
Invisible
Deactivate inputs
Background
Image
Gray


MAIN
AtomExampleTwinCAT
AC Line Voltage (V)
SETPOINT (0-10,000)
40.0
60.0
20.0
80.0
0.0
100.0
Properties
Filter
Sort by
Sort order
Advanced
Property
Value
Element name
GenElemInst_5
Type of element
Meter
Value
MAIN.AC_LINE_VOLTAGE
Position
X
500
Y
141
Width
162
Height
162
Background
Arrow
Scale
Sub scale position
Outside
Scale type
Lines
Scale start
0
Scale end
100
Main scale
20

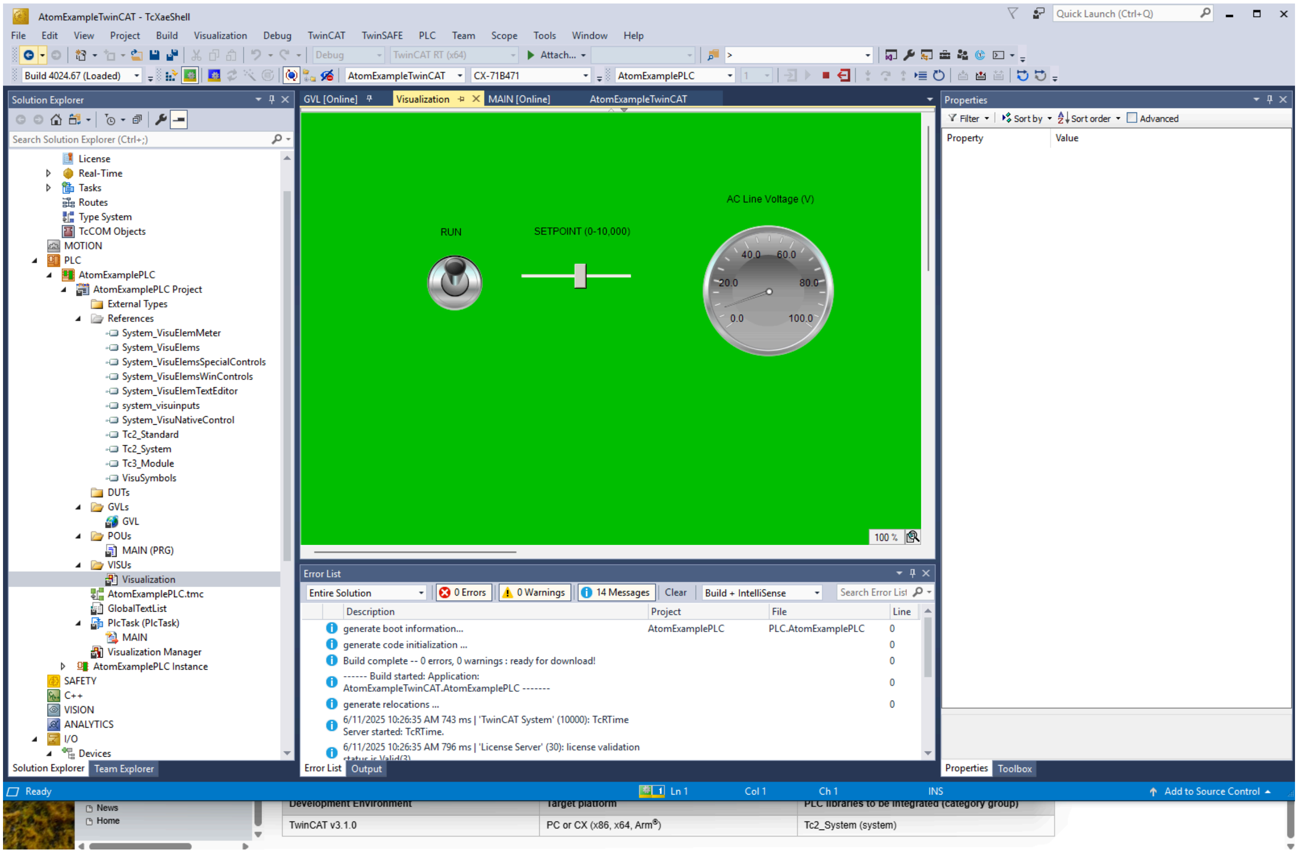


6. Build the project and hit **Login**. Select **Login with online change** and click **OK**:





7. If everything worked correctly, you should see the ATOM's AC line voltage in the meter, and you can toggle the run enable state and set the output setpoint percentage using the dip switch and slider, respectively:



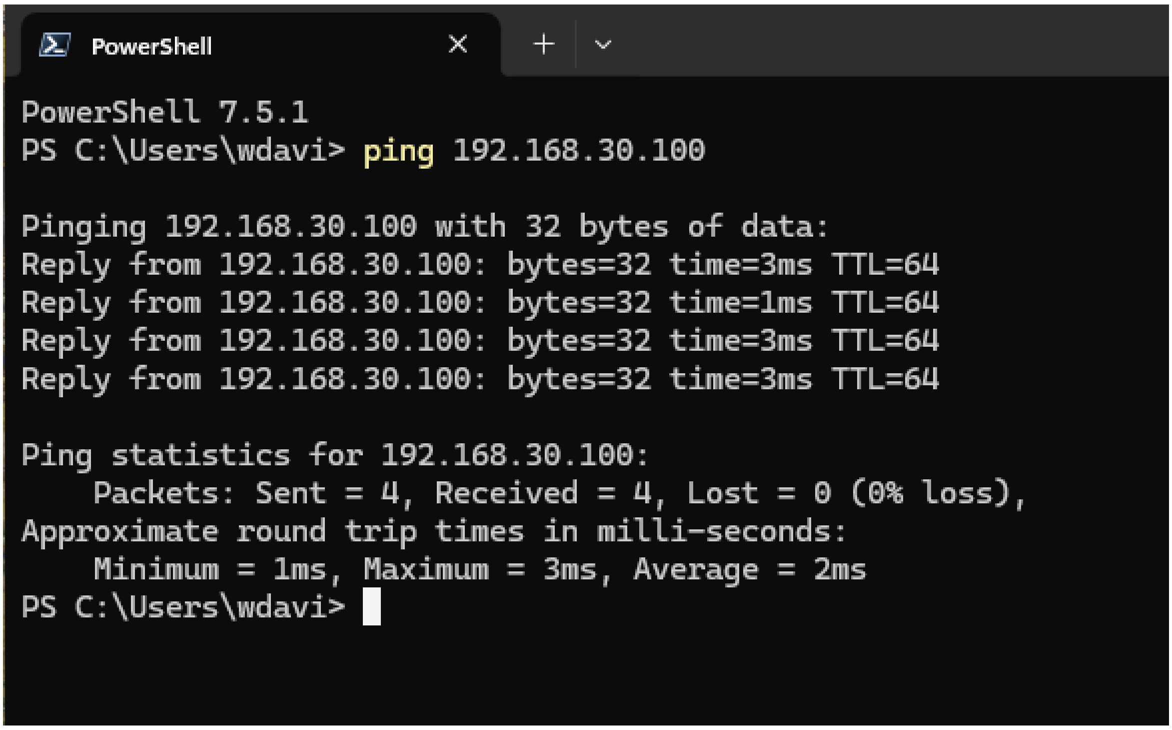
Troubleshooting

- My ATOM does not appear in the TwinCAT I/O Devices tree.
  - Ensure that ATOM is powered on and connected to the same network as your PC.
  - Check that the `Atom.xml` file is in the correct directory: `C:\TwinCAT\3.1\Config\Io\EtherCAT`.
  - Ensure that you have installed the ESI file correctly and reloaded the device descriptions in TwinCAT.
  - Check the network cable connection between your PC, ATOM, and PLC.
- I cannot connect to my PLC.
  - Ensure that the PLC is powered on and connected to the same network as your PC.

- Check that you have configured the correct IP address and subnet mask for both the PLC and your PC.
- Ensure that you have the correct username and password for your PLC.
- My variables are missing or an ESI file is working:
  - Select Reload Device Descriptions from the TwinCAT menu bar.
  - Click Config to restart the PLC in config mode.
  - Select the *Activate the configuration* button in the TwinCAT menu bar.

Can't connect to PLC or ATOM

Use the `ping` utility on Windows to check if your PC can reach the PLC/ATOM:



If:

- Ping is successful - you have a configuration problem with your PC
- Ping is unsuccessful - you have a hardware configuration, PLC configuration, or ATOM configuration problem.

Advanced

ATOM supports `FOE` (File Over EtherCAT) for firmware updates. If you receive a firmware update file from Control Concepts, you can update ATOM's firmware using TwinCAT 3: