Controlling Synapticon Devices Using Rockwell Studio 5000 Logix Designer - a Quick Start Guide

1. Notices

The information contained in this document is subject to change without notice. Synapticon GmbH provides this material "as is" and makes no warranty of any kind, expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Synapticon GmbH shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material, whether based on warranty, contract, or other legal theory.

1.1. Copyrights and Trademarks

Synapticon GmbH
Steinbeisstr. 1
71101 Schönaich
Germany, Europe
www.synapticon.com

2. Table of Contents

3. Disclaimer

This document provides an example of how to turn a Brushless DC (BLDC) motor by utilizing Rockwell Studio 5000 Logix Designer (on a Windows platform) and Synapticon's SOMANET Integro. Because it is an example, content may not match with the latest version of Rockwell Studio 5000 Logix Designer or with the SOMANET firmware.

IMPORTANT
The included code is an example; it cannot be used as a reference. In case of doubt, refer to the official Rockwell Studio 5000 Logix Designer and/or the SOMANET Integro documentation.

Synapticon GmbH refuses any responsibility for any problem or damage incurred from the use of the example configuration and code.

3.1. Reporting Errors

It is in our best interest to provide detailed, user-oriented and error-free documentation. If something is missing, incomprehensible or incorrect, go to Suggest Documentation Improvement and complete the form displayed there. In the "Link to documentation page" field, enter **Controlling Synapticon Devices Using Rockwell Studio 5000 - a Quick Start Guide v1.4**.

3.2. Versioning

This documentation will be constantly updated and improved. The version in the footer reflects the current documentation release.

4. Hardware and Software Overview

4.1. Hardware Overview

A diagram of the wiring used in this example is provided in *Figure 4.1* below. The provided code assumes that the laptop on which the setup is used has only one Ethernet port and that only one device is connected at the time.

See Connection Guide for additional information.

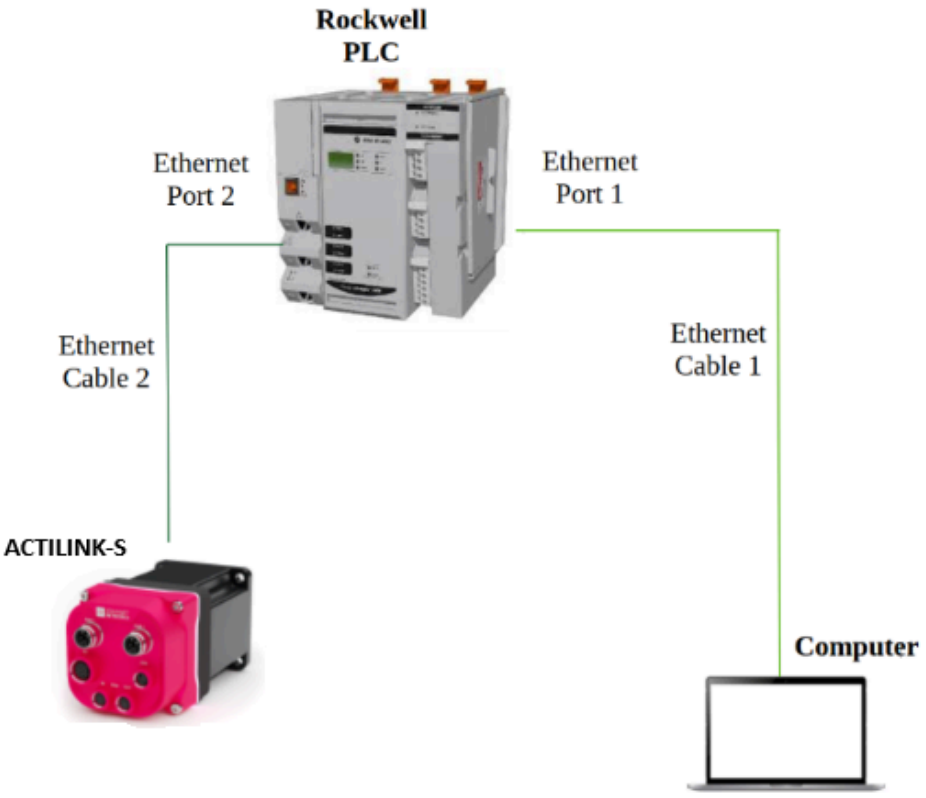


Figure 4.1 - Connection diagram

NOTE
While the L3x0ER and L3x0ERM controllers have been used as examples in this documentation, any PLC can be used with the drive.

4.2. Rockwell Studio 5000 Logix Designer Software

This documentation assumes that the user has downloaded the Rockwell Studio 5000 Logix Designer software and is familiar with its usage.

> **NOTE**
> This documentation is compatible with Rockwell Studio 5000 Logix Designer Software version 37.00.00 or higher.

5. Rockwell Studio 5000 Library for Synapticon ACTILINK-S

This section provides a comprehensive overview of the Rockwell Studio 5000 library designed for controlling Synapticon ACTILINK-S drives.

The library facilitates the implementation of the CiA 402 drive profile by providing the following:

- a set of User-Defined Data Types (UDTs) for structured data handling, and
- Add-On Instructions (AOIs) for drive control.

5.1. Library Content

5.1.1. User-Defined Data Types (UDTs)

The following UDTs provide a structured and standardized way to handle data within the library, serving as the foundational data structures for the AOIs.

UDT	Description
`ACTILINK_CIA402states`	Defines the integer values for the different CiA 402 internal state machine states.
`ACTILINK_CIA402statesStatusword`	Defines the masked integer values of the CiA 402 Status Word for state detection.
`ACTILINK_CIA402statesTransitionsCommands`	Defines the integer values for the Control Word to command specific state transitions.
`ACTILINK_inputsOffsets`	Defines the byte offsets within the drive's input I/O data structure. These offsets are used to correctly parse the data received from the drive.
`ACTILINK_machineInternalstates`	Defines the numerical values for the internal state machine logic of the `ACTILINK_ServoState AOI`. These values are used to manage the sequence of drive transitions.
`ACTILINK_modesOfOperation`	Defines the integer values that define the various operating modes of the drive, as per the CiA 402 standard.
`ACTILINK_outputsOffsets`	Defines the byte offsets within the drive's output I/O data structure. These offsets are used to correctly write command data to the drive.
`ACTILINK_ServoValues`	This UDT collects all servo values under one namespace. Also the status and control word is located here.
`ACTILINK_SubOutputs`	Defines commonly used types for modes and value flags.

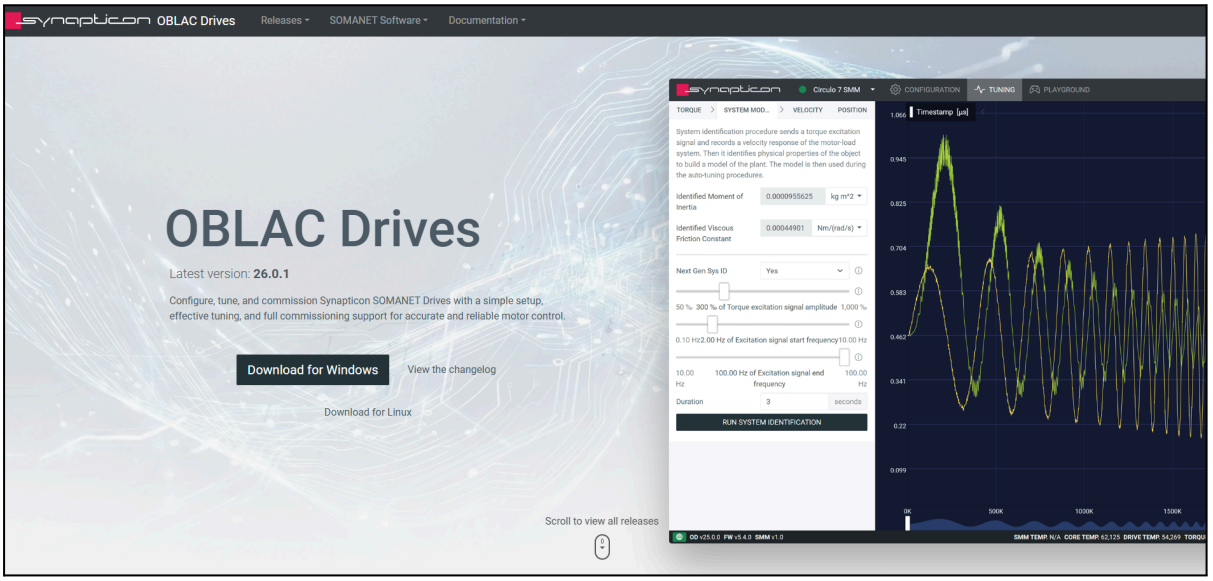
5.1.2. Add-On Instructions (AOIs)

In this library, several AOI are defined to work with the servo drive.

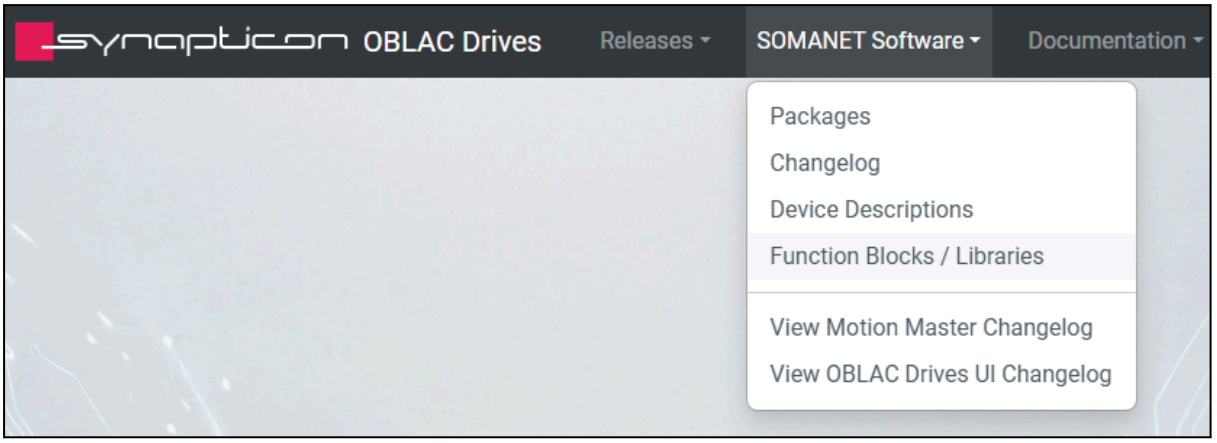
To download the library, follow the procedure below.

> NOTE:
> The *Rockwell_AOI_Demo-v1.0.zip* file is only available for firmware equal to, or greater than, v5.6.5.

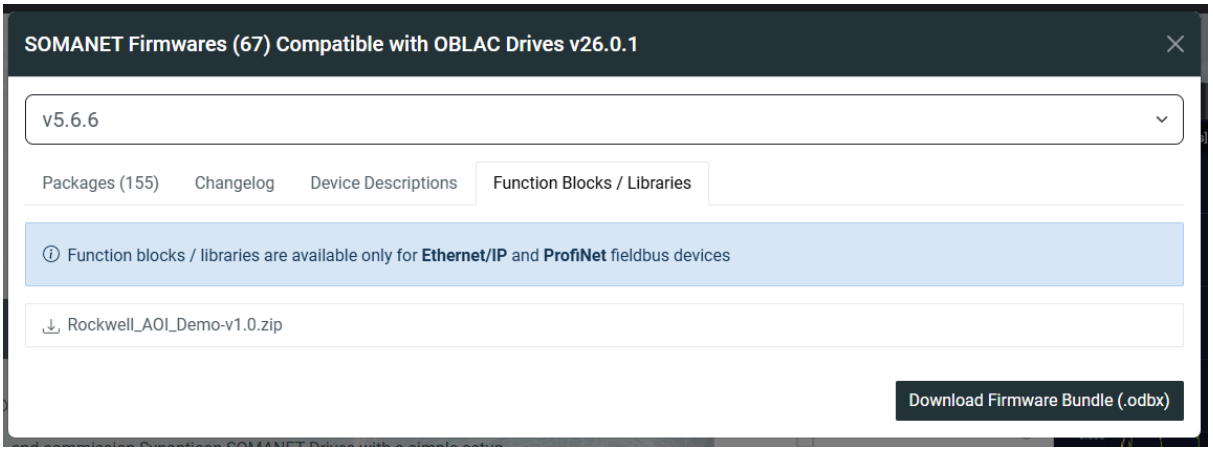
1. Go to OBLAC Drives. The following screen appears.



2. From the main OBLAC Drives screen, go to the Navigation bar at the top. From the *SOMANET Software* pulldown menu, select **Function Blocks / Libraries**.



The following screen appears:

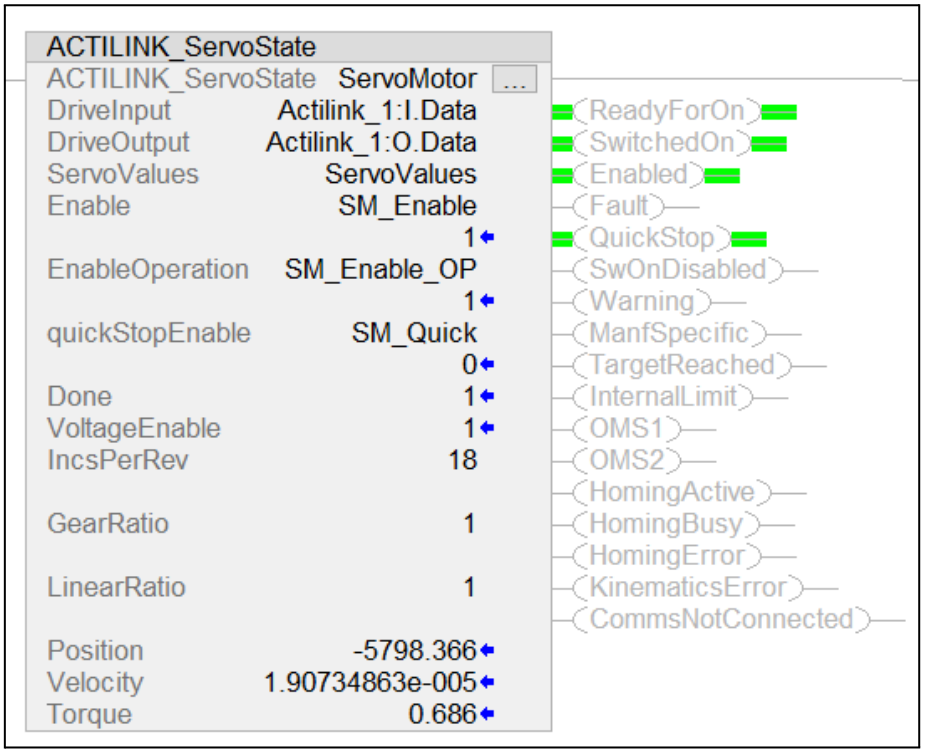


3. Use the drop-down menu at the top to select the firmware version desired then click **Rockwell_AOI_Demo-v1.0.zip**. The package is downloaded.

The AOI ACTILINK_ServoState is necessary to initialize the drive and take it to a working state. For Profile Velocity and Profile Position it is also recommended to run ACTILINK_SetProfileValues to initialize the profile velocity, acceleration and deceleration. After the initialization, ACTILINK-S can be used in profile position, profile velocity, or profile torque with the respective AOI. Modes can be changed during operation without the need to reinitialize the drive. See below for a detailed description of the AOI.

5.1.2.1. ACTILINK_ServoState

This is the main AOI to control an ACTILINK-S drive. In this AOI, the data I/O is handled and the drive state machine. Users with experience with DS402 will find themselves familiar with the states. Additionally, the AOI shows the actual velocity, position, and torque of the device.



Inputs:

- Enable: BOOL - Enable the AOI
- EnableOperation: BOOL - trigger operation enable bit
- quickStopEnable: BOOL - set quick stop enable bit. This will result in the state transition (11) to QuickStopActive. When the drive comes to halt, the state machine automatically advances to SwitchOnDisabled (transition 12). See also the state diagram below.
- VoltageEnable: BOOL - Enable voltage of the power stage.
- EnableHoming:BOOL - Enable homing mode.
- StartHoming:BOOL - Start the homing procedure.
- FaultReset:BOOL - Send fault reset to the drive.
- GearRatio:REAL - Setting of the gear box transmission ratio. See Usage of Gear and Linear Ratio below.
- LinearRatio:REAL - Same as gear ratio but for linear transmission. See Usage of Gear and Linear Ratio below.
- IncsPerRev - Setting of encoder increments per motor shaft revolution.

Outputs:

- ReadyForOn:BOOL - The drive is in state SwitchOn disabled.
- SwitchedOn:BOOL - The drive is in the state SwitchOn
- Enabled:BOOL - The drive is in the state operation enabled.
- Fault:BOOL - The drive is in the fault state. Use FaultReset to release the error if possible. If the error persists,remove the cause of the error and power cycle the drive.
- QuickStop:BOOL - The drive is in QuickStop state.
- SwOnDisabled:BOOL - The drive is in the state SwitchOnDisabled.
- Warning:BOOL - The warning bit of the drive is set.
- ManfSpecific:BOOL - Manufacturer specific status signal, currently unused.
- TargetReached:BOOL - In operation modes that support this flag, indicates that the requested action is achieved. For example, the target velocity is reached (in profile velocity) or the requested position is reached in profile position mode.
- internalLimit:BOOL - The internal software limits are reached.
- OMS1:BOOL - (Operation Mode Specific) The value of this field depends on the operation mode used. Refer to the operation mode description below if, and how, a specific operation mode uses this bit.
- OMS2:BOOL - (see OMS1)
- HomingActive:BOOL - Homing is achieved and active.
- HomingBusy:BOOL - Homing procedure is currently active.
- HomingError:BOOL - An error occurred during the homing operation.
- KinematicsError:BOOL - Invalid setting of gear ratio or linear ratio.

- CommsNotConnected:BOOL - Active when there is no communication to the ACTILINK
- Position:REAL - Display the actual position in user units.
- Velocity:REAL - Display actual velocity in units per second.
- Torque:REAL - Display actual torque in percent of rated torque, see also drive parameter Rated Torque (CIP: 112.1.7552).

In/Out:

- DriveInput:SINT[47] - Raw values to transmit to the drive.
- DriveOutput:SINT[35] - Raw values received from the drive.
- ServoValues:ACTILINK_ServoValues - Parsed parameter values.

5.1.2.1.1. Usage of Gear and Linear Ratio

The gear ratio represents the number of revolutions a motor shaft must make for the output shaft of the gear box to turn a single revolution.

The linear ratio is the number of units the motion will move for a single revolution of the shaft.

For rotary motion without a gear box, both values must be set to 1.
This will result in one unit per revolution.

For rotary motion in degrees, set the gear ratio to 1 and the linear ratio to 360.

5.1.2.1.2. CiA402 State Machine

To control the drive, the CiA402 state machine is used.

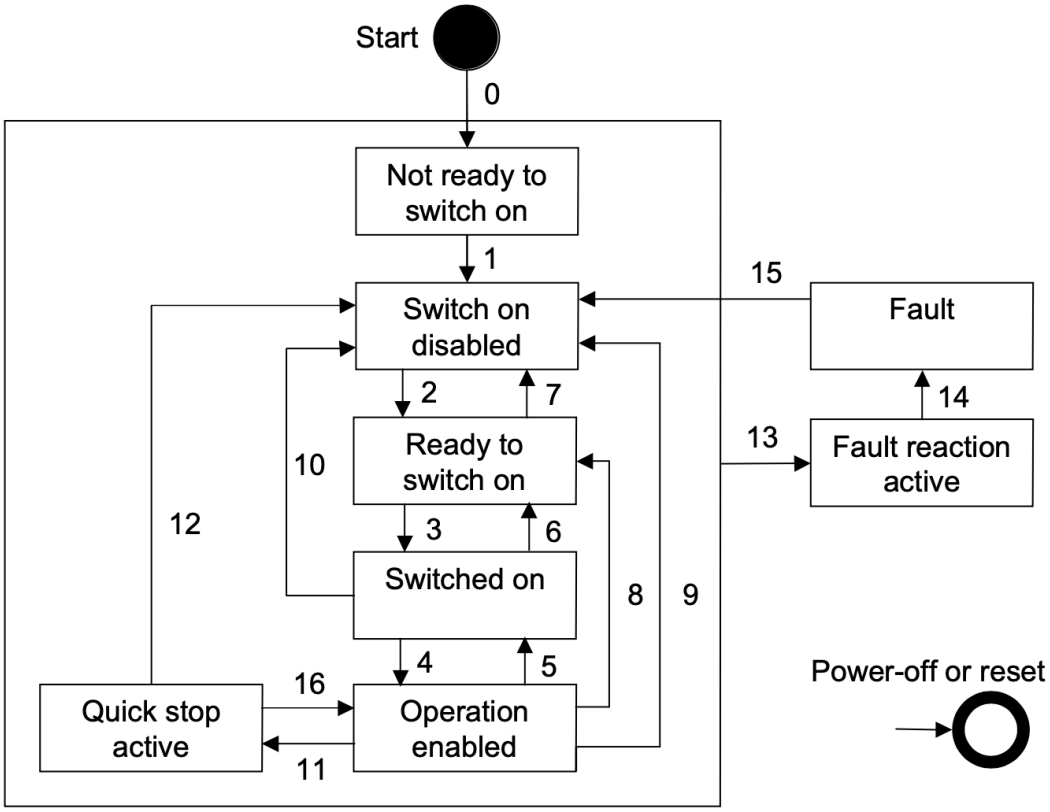


Figure 5.1 - CiA402 state machine
See IEC 61800 for additional information

The transition 0, 1, 12 (if quick stop method is 2), 14 are automatic transitions, no user interaction is required. Transition 15 is only possible with the master application setting the Fault Reset bit (sometimes automatically in the main task). All other transitions are commanded by the control word bits.

5.1.2.1.3. Homing

For homing, no extra AOI is necessary. To set a new home position, follow these steps:

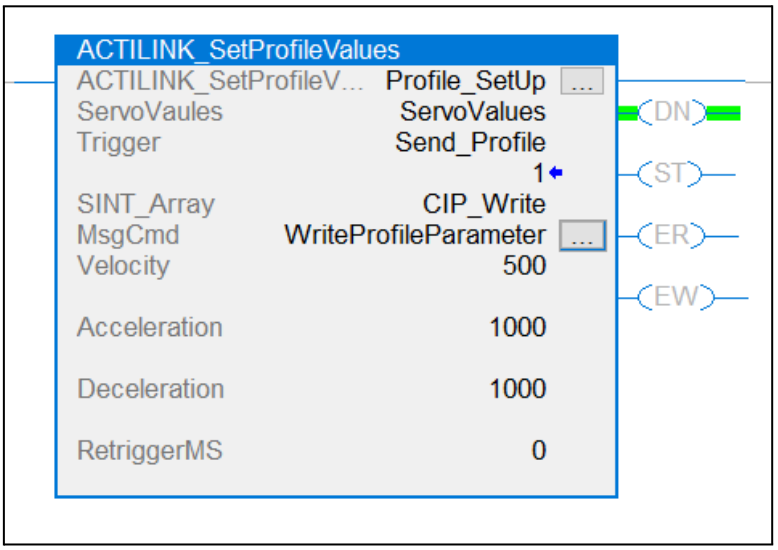
1. Move the drive to the position where it should be homed.
2. Activate the homing by setting a rising flank 0->1 on input EnableHoming.
3. Start the homing by setting true on input StartHoming.

If the output HomingActive is TRUE, the homing has completed successfully.
If HomingError is TRUE, try the procedure again.

After a successful homing, the current position now displays a value of 0.

5.1.2.2. ACTILINK_SetProfileValues

It is necessary for profile position and profile velocity to have acceleration and deceleration set for the drive to be able to reach the aspired velocity. For profile position mode, the profile velocity is also necessary as the velocity to reach the respective setpoints. The output signals are directly linked to the output signals of the MESSAGE object responsible for transmitting the parameter.



Inputs:

- Trigger:BOOL - to send the pre-defined profile parameter, such as profile velocity, profile acceleration, and profile deceleration to the drive.
- RetriggerMS:DINT - Retrigger after count milliseconds.
- Velocity:REAL - Profile velocity in user units.
- Acceleration:REAL - Profile acceleration in user units.
- Deceleration:REAL - Profile deceleration in user units.

Outputs:

- DN:BOOL - Download was completed and successful. Mirrored from MsgCmd.DN
- ER:BOOL - An error occurred while downloading the values. Mirrored from MsgCmd.ER
- EW:BOOL - Mirrored from MsgCmd.EW.
- ST:BOOL - Mirrored from MsgCmd.ST

InOut

- MsgCmd:MESSAGE - The message object to communicate with the device.
- ServoValues:ACTILINK_ServoValues - Exchange of serve values with other AOI.
- SINT_Array:SINT[2] - The array of SINTs used in the MsgCmd.

5.1.2.2.1. The Message Command Object

A message object is necessary to perform the write of the profile parameter velocity, acceleration, and deceleration. This message object is used for the explicit message, and uses the Service Type: *Set Attribute Single*. Use the **Configuration** menu to configure the message object; use the **Communication** tab to set the path to the ACTILINK drive used.

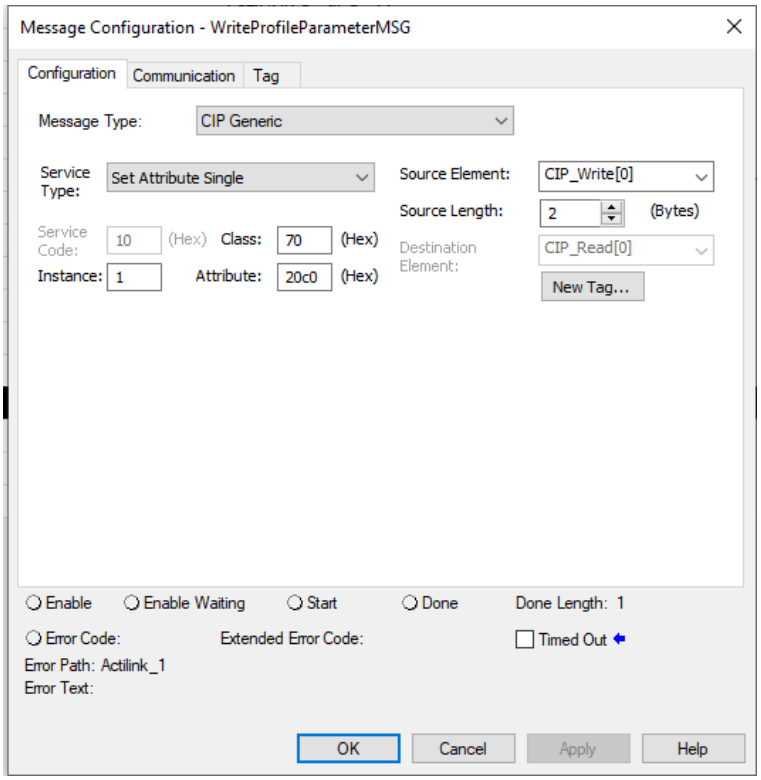


Figure 5.2 - Configuration of the message object used in ACTILINK_SetProfileValues

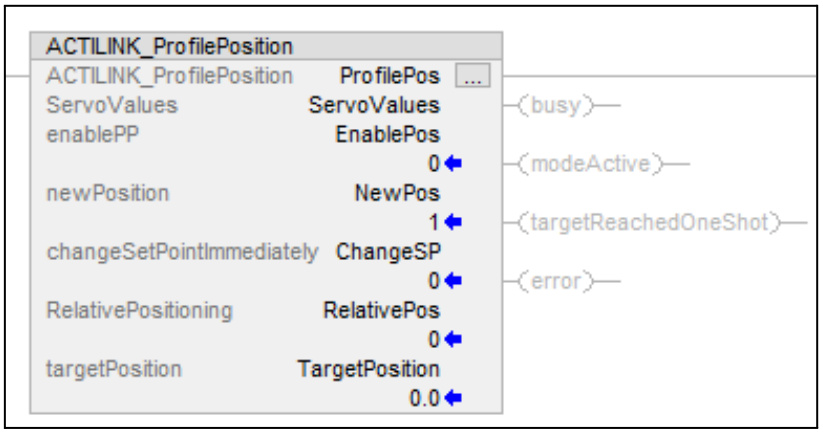
The values for the CIP address are not important because they are overwritten by the AOI at runtime to match the actual parameter address. The local tag CIP_Write[] holds the value, which will be written by the next trigger of the write operation. All values are in user units.

NOTE

For additional information on how to configure a message object, refer to Explicit Messaging with the MSG Instruction.

5.1.2.3. ACTILINK_ProfilePosition

This block controls the drive in Profile Position Mode, commanding the drive to move to a specific position. It handles the I/O mapping and control word bits for motion commands.



Usage:

- To use this AOI the ACTILINK_SetProfileValues needs to set all profile parameters.
- Calling this AOI continuously after the drive is in the `Operation Enabled` state.
- A rising edge on the `newPosition` input will trigger a new position move.
- A rising edge on `changeSetPointImmediately` will initiate an immediate update to the target position without waiting for the previous one to complete.

Inputs:

- `enablePP`: `BOOL` - A rising edge on `enablePV` will activate Profile Position Mode.
- `newPosition`: `BOOL` - A rising edge on this input triggers a new position command.
- `changeSetPointImmediately`: `BOOL` - A rising edge on this input will trigger an immediate change in the target position.
- `targetPosition`: `REAL`- The target position in user defined units, see also ACTILINK_ServoState.

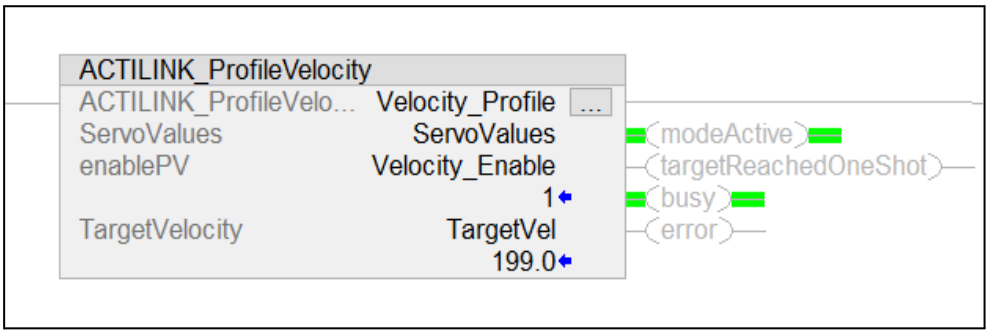
Outputs:

- `error`: `BOOL` - `1` if a drive fault is detected.
- `modeActive`: `BOOL` - `1` when the drive is in Profile Position Mode.
- `busy`: `BOOL` - `1` while a position command is in progress.
- `targetReachedOneShot:BOOL` - A one-shot pulse upon reaching the target position.

InOuts:

- ServoValues:ACTILINK_ServoValues - Sharing of parsed drive parameters.

5.1.2.4. ACTILINK_ProfileVelocity



This block controls the drive in Profile Velocity Mode.

Usage:

- To use this AOI the ACTILINK_SetProfileValues needs to set profile acceleration and profile deceleration.
- Calling this AOI continuously after the drive is in the `Operation Enabled` state.
- A rising edge on `enablePV` will activate this mode. The mode remains active as long as this input is `TRUE`.
- The drive will move at the `targetVelocity` specified.

Inputs:

- `enablePV`: `BOOL` - A rising edge on this input will activate the mode.
- `TargetVelocity`: `REAL`- The desired velocity in RPM.

Outputs:

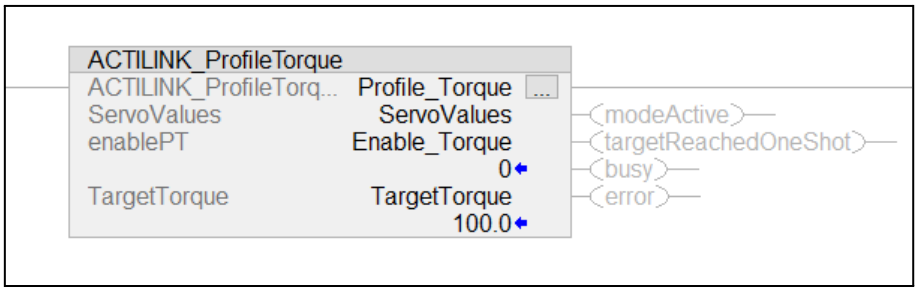
- `error`: `BOOL` - `TRUE` if a drive fault is detected.
- `modeActive`: `BOOL` - `TRUE` when the drive is in Profile Velocity Mode.
- `busy`: `BOOL` - `TRUE` while the Actuator is reaching the target velocity.
- `targetReachedOneShot`: `BOOL` - A one-shot pulse upon reaching the commanded velocity.

InOuts:

- ServoValues:ACTILINK_ServoValues - Sharing of parsed drive parameters.

5.1.2.5. ACTILINK_ProfileTorque

This block operates the drive in Profile Torque Mode, applying a commanded motor torque.



Usage:

- Calling this AOI continuously after the drive is in the `Operation Enabled` state.
- A rising edge on `enablePT` will activate this mode. The mode remains active as long as this input is `TRUE`.
- The `targetTorque` is applied when `NullTorque` is `0`.

Inputs:

- `enablePT`: `BOOL` - A rising edge on this input will activate the mode.
- `targetTorque`: `REAL`- The desired motor torque value. The torque is provided as a percentage of rated torque.

Outputs:

- `busy`: `BOOL` - `1` while the actuator is reaching the target torque.
- `error`: `BOOL` - `1` if a drive fault is detected.
- `modeActive`: `BOOL` - `1` when the drive is in Profile Torque Mode.
- `targetReachedOneShot`: - A one-shot pulse upon reaching the target torque.

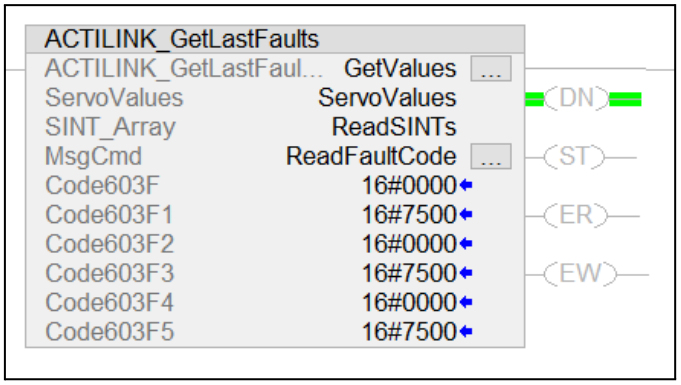
InOuts:

- ServoValues:ACTILINK_ServoValues - Sharing of parsed drive parameters.

> **NOTE**
> Target Reached Bit and Window Parameters
>
> For the `targetReachedOneShot` output on the profile mode AOIs (`FB_ProfilePosition`, `FB_ProfileVelocity`, `FB_ProfileTorque`) to function correctly, the drive's window parameters must be configured. These parameters define the tolerance and time delay for determining if a target has been successfully reached.
>
> The following parameters must be configured via the OBLAC configuration tool or by using explicit messaging:
> - Velocity Window: CIP: 112.1.6976 CiA: `0x606D:0`
> - Velocity Window Time: CIP: 112.1.7040 CiA: `0x606E:0`
> - Torque Window: CIP: 104.1.1281 CiA: `0x2014:1`
> - Torque Window Time: CIP: 104.1.1282 CiA: `0x2014:2`
> - Position Window: CIP: 112.1.6592 CiA: `0x6067:0`
> - Position Window Time: CIP: 112.1.6556 CiA: `0x6068:0`

5.1.2.6. ACTILINK_GetValues

With this AOI, the last five faults of the drive are shown. It is triggered whenever a fault is detected, or a fault is cleared and the fault register CIP: 112.1.4032 (CiA: `0x603f:0`) is read.



Usage:

- Add this AOI on a new rung.
- Link to ACTILINK ServoValues and provide the SINT_Array for the output data and the message ReadFaultCode.
- To properly configure the message command, see Explicit Messaging with the MSG Instruction.

Inputs:

- EnableIn:BOOL - Enable this AOI.
- GetCustom:BOOL - Activate reading of the fault code.
- CustomClass:DINT - Configured class for the fault code parameter (default: 0x70).
- CustomInance:INT - Configured instance for the fault code parameter (default: 0x70).
- CustomAttribute:DINT - Configured attribute for the fault code parameter (default: 0x70).

Outputs:

- DN:BOOL - Download was completed and successful. Mirrored from MsgCmd.DN.
- ER:BOOL - An error occurred while downloading the values. Mirrored from MsgCmd.ER.
- EW:BOOL - Mirrored from MsgCmd.EW.
- ST:BOOL - Mirrored from MsgCmd.ST.
- Step:DINT - Internal process step.
- EnableOut:BOOL - Enable output, default on.
- Code603F:INT - Last read error code.
- Code603F1:INT - Previous error code.
- Code603F2:INT - Predecessor to Code603F1
- Code603F3:INT - Predecessor to Code603F2.
- Code603F4:INT - Predecessor to Code603F3.
- Code603F5:INT - Predecessor to Code603F4.
- Active:BOOL - Output active.

InOuts:

- MsgCmd:MESSAGE - Linked message object for the explicit message transfer.
- ServoValues:ACTILINK_ServoValues - Reference to ACTILINK_ServoValues for fault signaling.
- SINT_Array:SINT[8] - Temporary variable to store the result of the message command.

5.2. Adding the Library

This section provides instructions on how to add the Synapticon library to a Rockwell Studio 5000 Logix Designer project.

5.2.1. Import Demo Application

As part of the distribution package, a demo program in ladder logic is provided. The demo is provided as an L5X export of the main Routine.

1. To open the project, start Rockwell Studio 5000 and open a new project.
   - To set up a new project, refer to Establishing a New Project.
   - To add an ACTILINK-S drive, see Adding I/O Devices.

2. When the project is setup, right click on the **MainProgram** and select **Add -> Import Routine**.

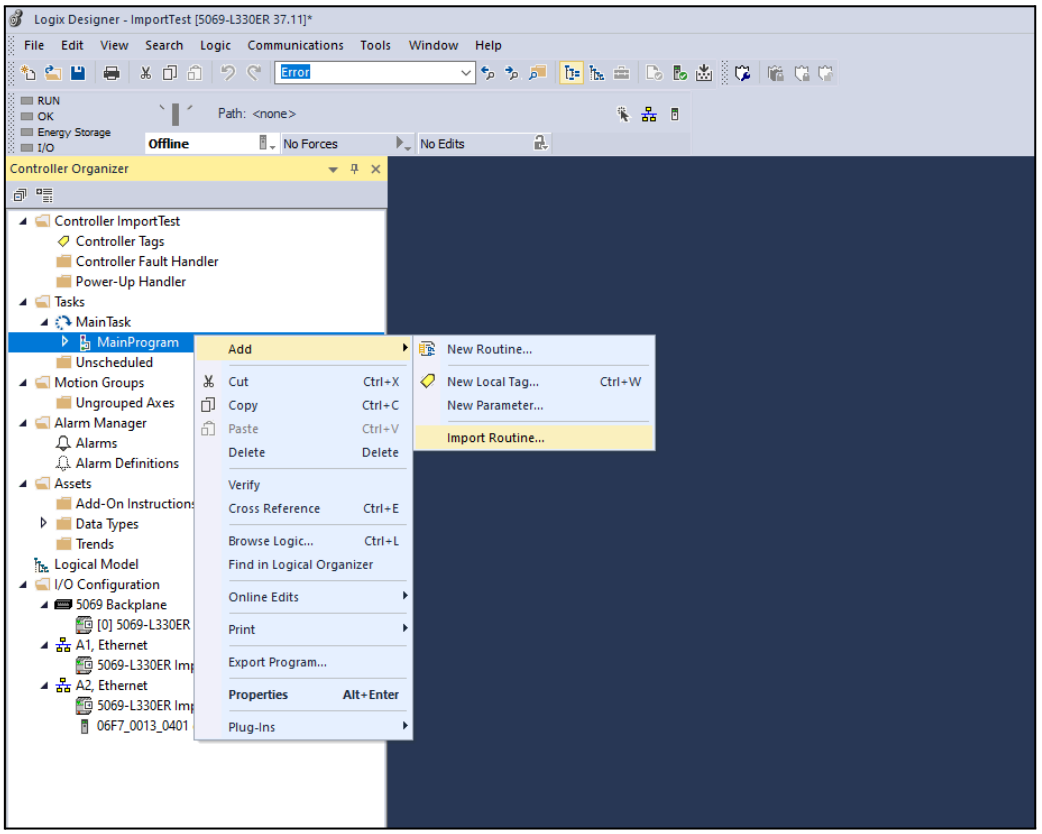


Figure 5.3 - Right click on MainProgram and import a routine

NOTE:
The import may take some time to occur until Studio 5000 has set up everything properly.

3. In the dialog box, select the L5X routine associated with this documentation.

4. Depending on the name selected for the ACTILINK-S drive, the linked names must also be updated.
   - To change the input and output links, refer to *Figure 5.4*.
   - For information on how to fix the drive alias, see *Figure 5.5*.

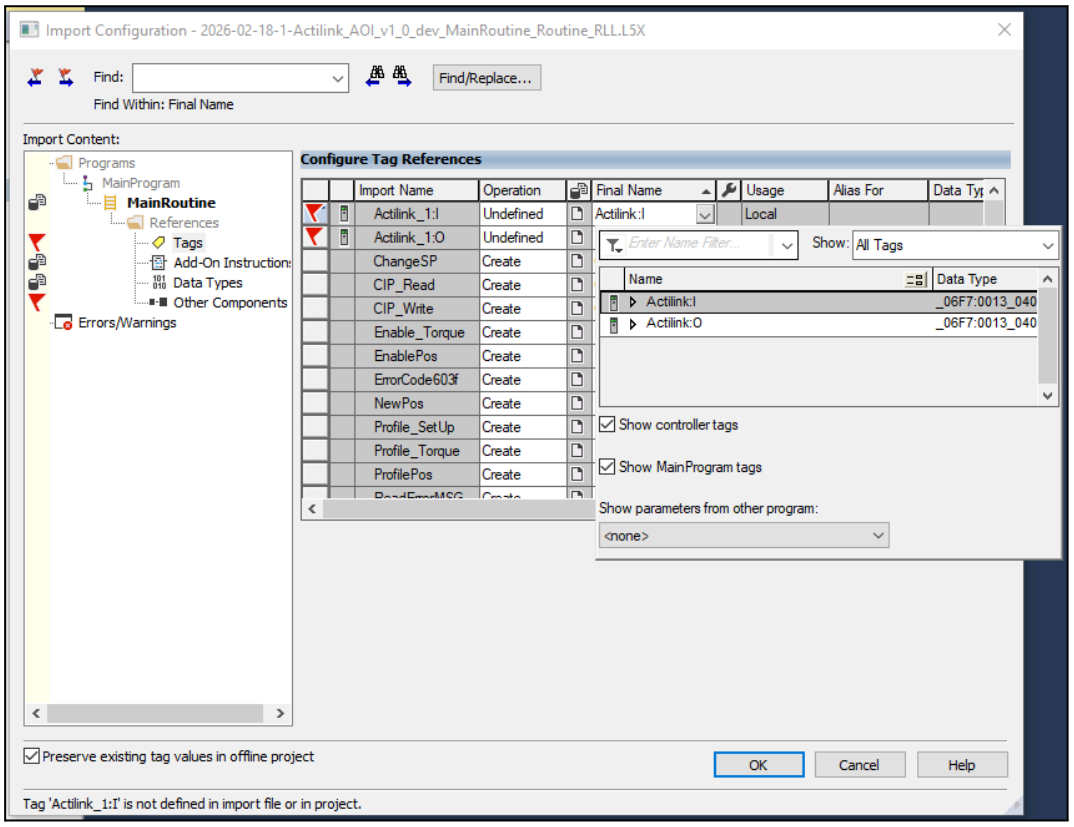


Figure 5.4 - Fix link tags for input and output

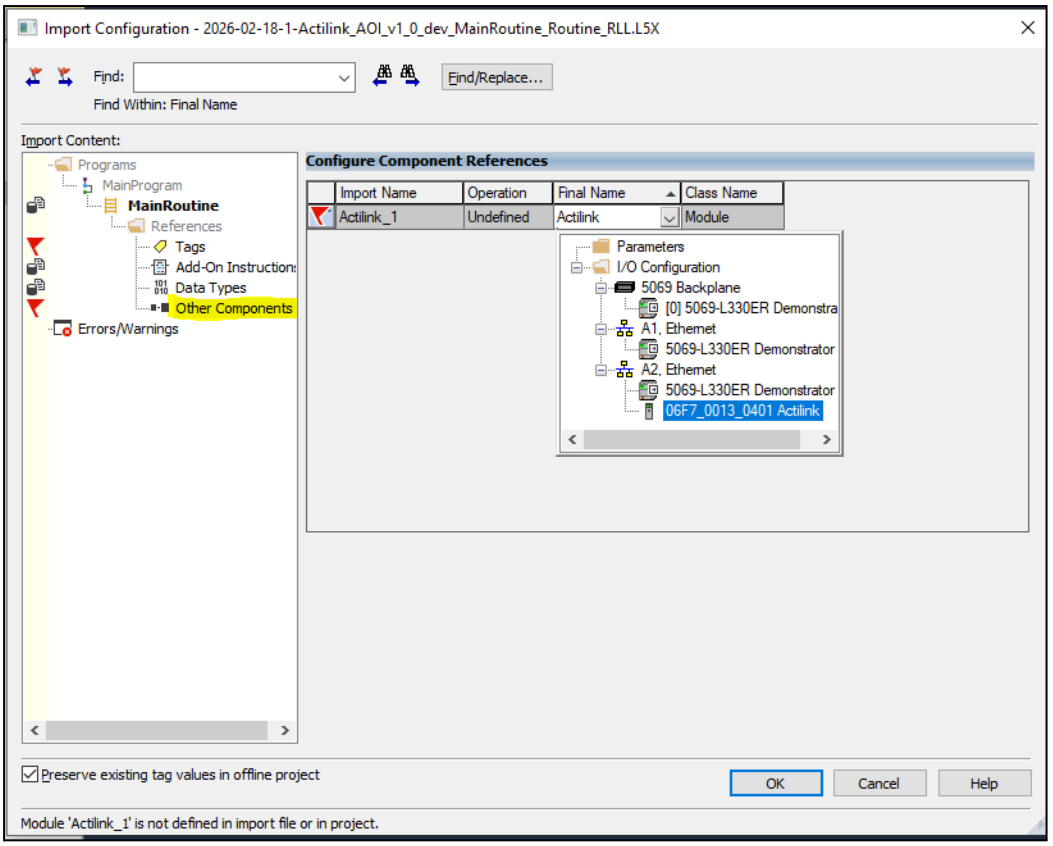


Figure 5.5 - Fix the link to the added ACTILINK module

5. Once the project is ready, select the **MainRoutine** task with:

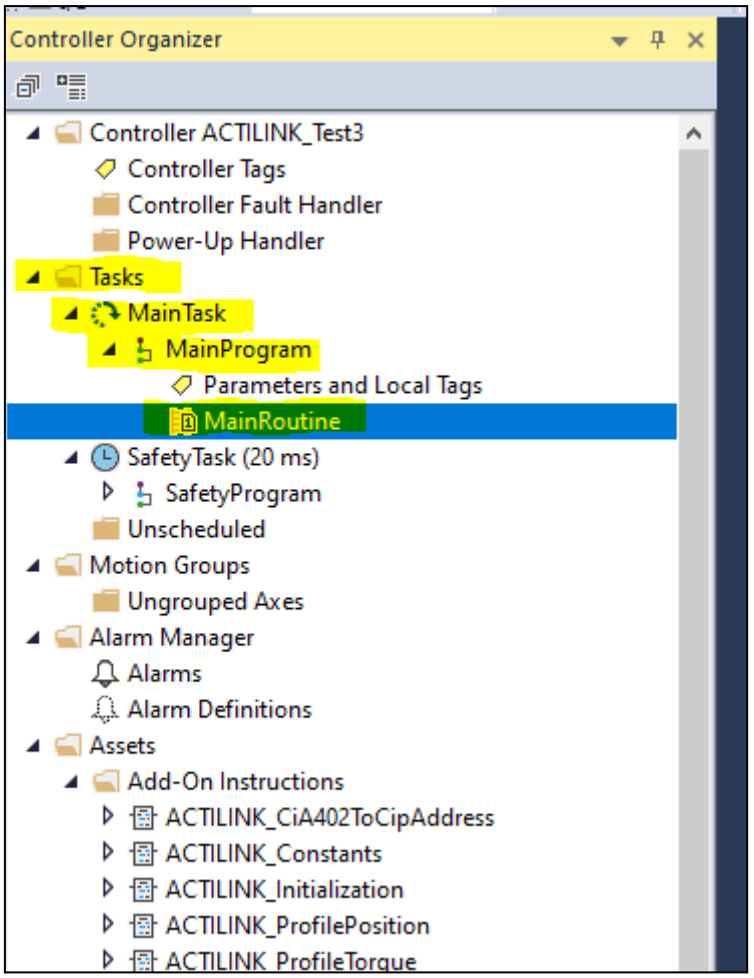


Figure 5.6 - Navigate the Controller Organizer to the MainRoutine

6. With a double click, the current main task of the project opens.

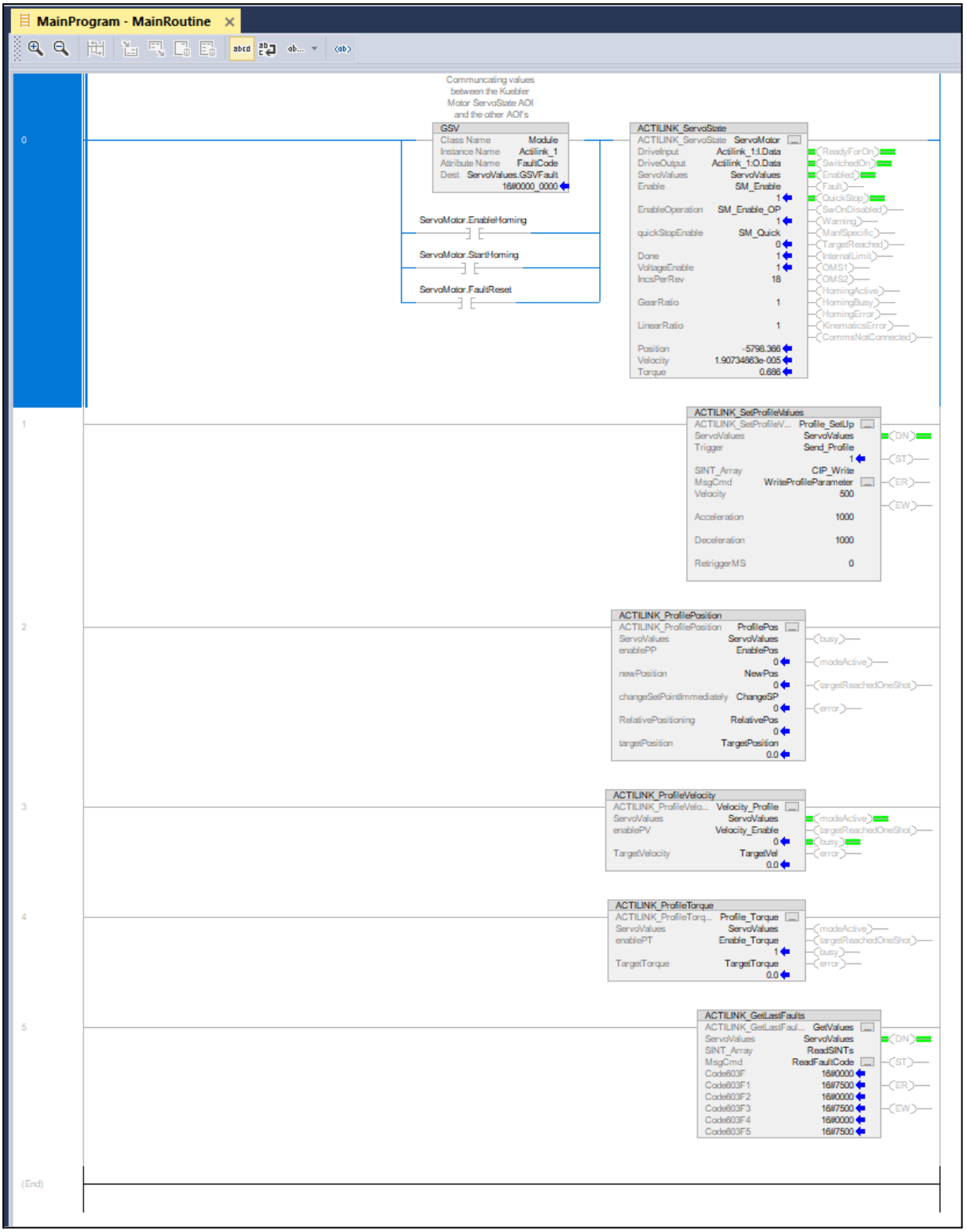


Figure 5.7 - Display of the main task in ladder logic

In the above overview, all the main AOI are visible. *Figure 5.7* provides an example of how to wire homing and fault reset signals as input to ACTILINK_ServoState.

This application displays all available AOI to implement a custom application.

Mandatory AOI are ACTILINK_ServoState as the main entry point and ACTILINK_SetProfileValues if no default profile parameters are set or, if the profile parameter must be set by the application.

5.3. Explicit Messaging with the MSG Instruction

NOTE
In addition to the standard cyclic I/O data, the Synapticon ACTILINK-S drive contains a wide range of parameters that are configured and accessed using explicit messaging. These parameters are not part of the standard I/O mapping defined by the drive's EDS file. A key example is the configuration of window parameters, which is necessary for the `targetReachedOneShot` output on the profile mode AOI to function correctly.

This section explains how to use the **Message (MSG)** instruction in Rockwell Studio 5000 Logix Designer for explicit messaging with the drive.

The MSG instruction is an asynchronous command that reads or writes blocks of data to another module on a network. The single operand for this function is a tag of the **Message** type.

5.3.1. Configuring the MSG Instruction

First, a tag of the **MSG** data type must be created in your Controller Tags.

1. Place themMSG ladder element to a new rung.

Figure 5.8 - Adding the MSG Ladder Element

2. Define the Message Type:
   - On the **Configuration** tab, select **CIP Generic** as the Message Type. This is the standard protocol for explicit messaging to the drive.

3. Set the Service and CiA 402 Parameters:
   - Select the **Service Type**:
     For a read operation, select **Get Attribute Single**.
     For a write operation, select **Set Attribute Single**.
   - See the CIP Parameters List for all available parameters.
   - For example, to configure the Position Window (*Figure 5.9*), the value from tag velocityWindowTag will be written, and the **Class**, **Instance**, and **Attribute** values must be specified:
     - **Class:** 70
     - **Instance:** 1
     - **Attribute:** 1B40
   - Another example is reading the error parameter (*Figure 5.10*) into the tag ErrorCode603f; the following parameter for the CIP address must be set as follows:
     - **Class**: 70
     - **Instance**: 1
     - **Attribute**: FC0
     - The target tag to store the result to - here, ErrorCode603f - with 16bit size data type UINT.

4. Link the Source and Destination Tags:
    - On the **Source/Destination** tab, specify the **Source Element** or **Destination Element**. This is the PLC tag that will either provide the data to be written to the drive, or store the data read from the drive. The tag's data type must match the parameter to be accessed (e.g., `UINT` for the `velocity window` parameter).

The figure below illustrates how to configure the MSG instruction with the CIP Generic message type, setting the service and CiP parameters and linking the source/destination tags.

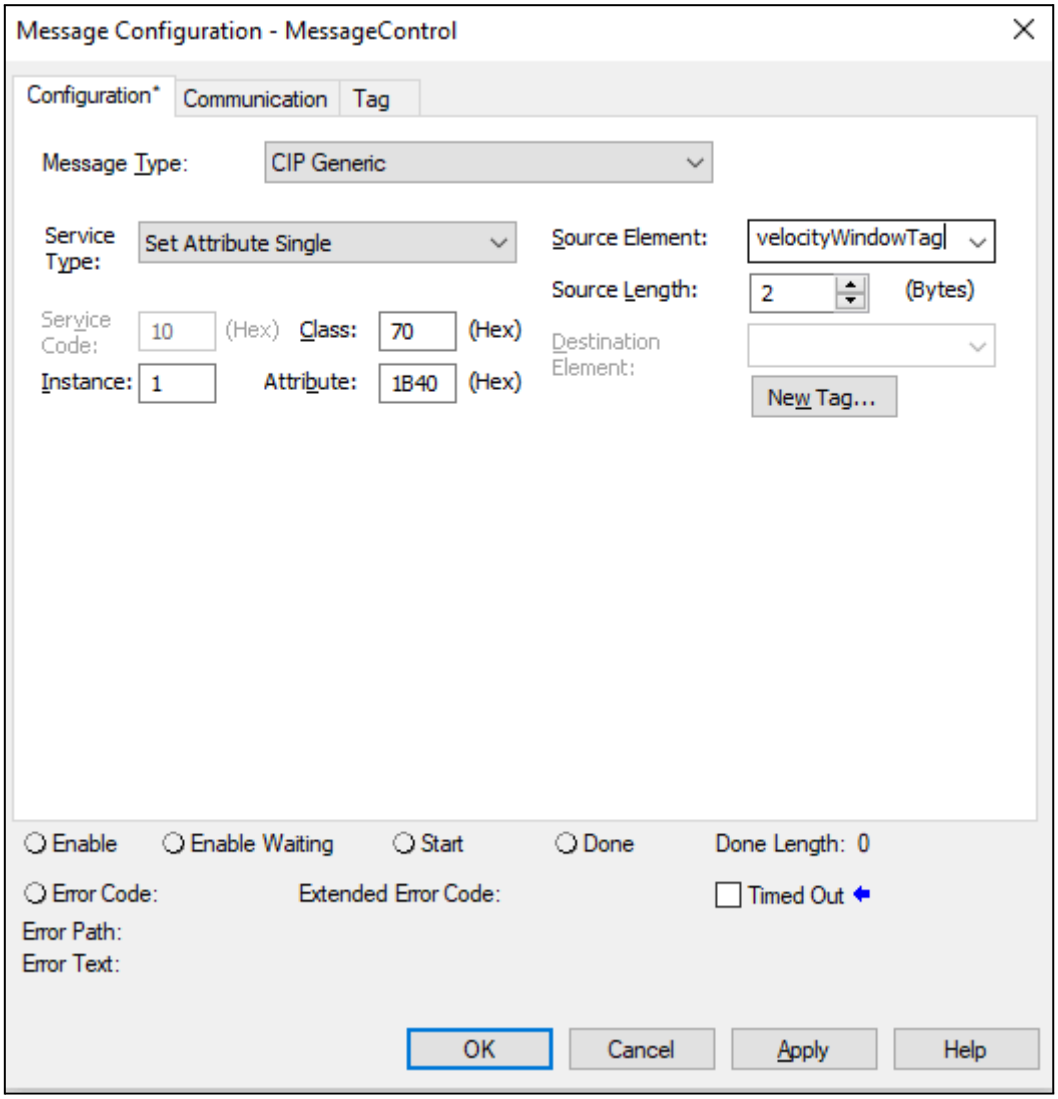


Figure 5.9 - Configuring the MSG instruction

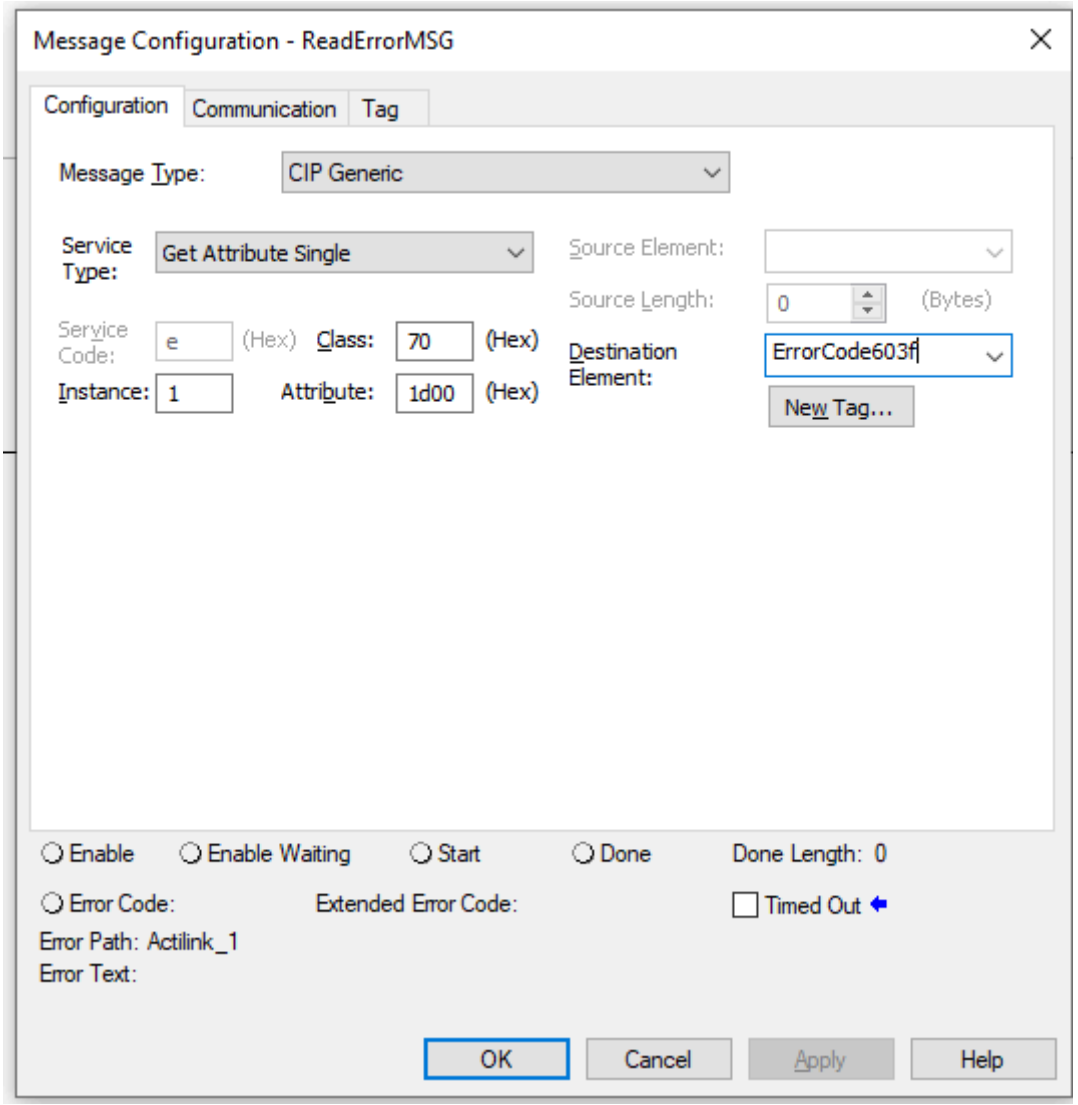


Figure 5.10 - Example configuration to read the Error Code with CIP address 70.1.1D00 (CiA402 0x603F:0)

5. Specify the Communication Path:
    - On the **Communication** tab, enter the path to the ACTILINK-S drive. This path defines the route from the controller to the specific drive on the network.

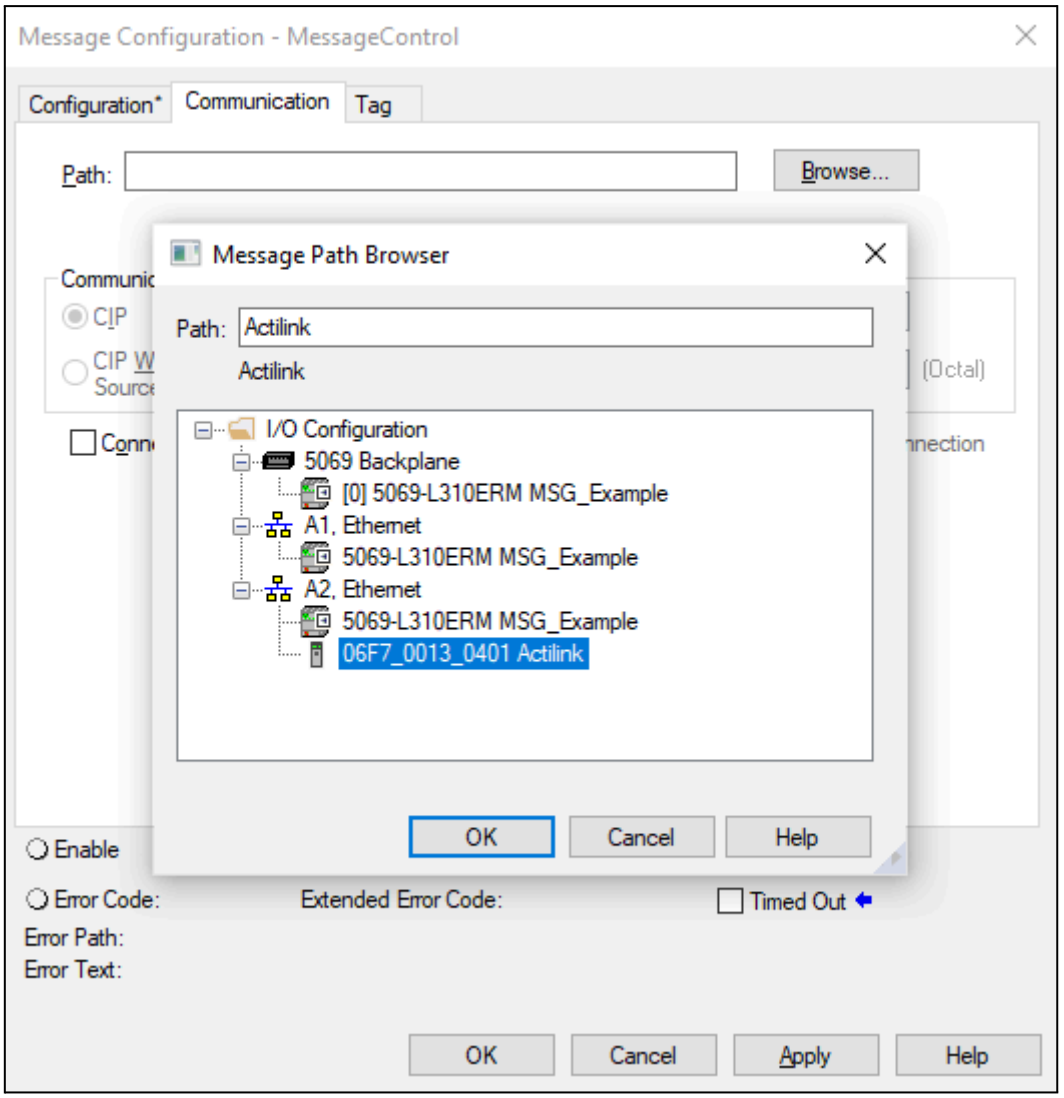


Figure 5.11 - Specifying the communication path to the ACTILINK-S drive

6. Click **Apply** to finish the configuration.

6. Additional Information

6.1. Using Rockwell Studio 5000 Logix Designer

Rockwell Studio 5000 Logix Designer is used to program Rockwell PLCs. This section explains the process of establishing a new project and configuring the network to integrate the ACTILINK-S drive.

6.1.1. Establishing a New Project

Before configuring your drive, create a new project within Rockwell Studio 5000 Logix Designer.

To do this:

1. Open Rockwell Studio 5000 Logix Designer.
2. From the welcome screen, select **New Project**.

Figure 6.1 - Rockwell Studio 5000 Logix Designer welcome screen

3. Select your specific PLC controller from the list, which depends on your hardware.

4. Name the project and select a location where it will be saved.

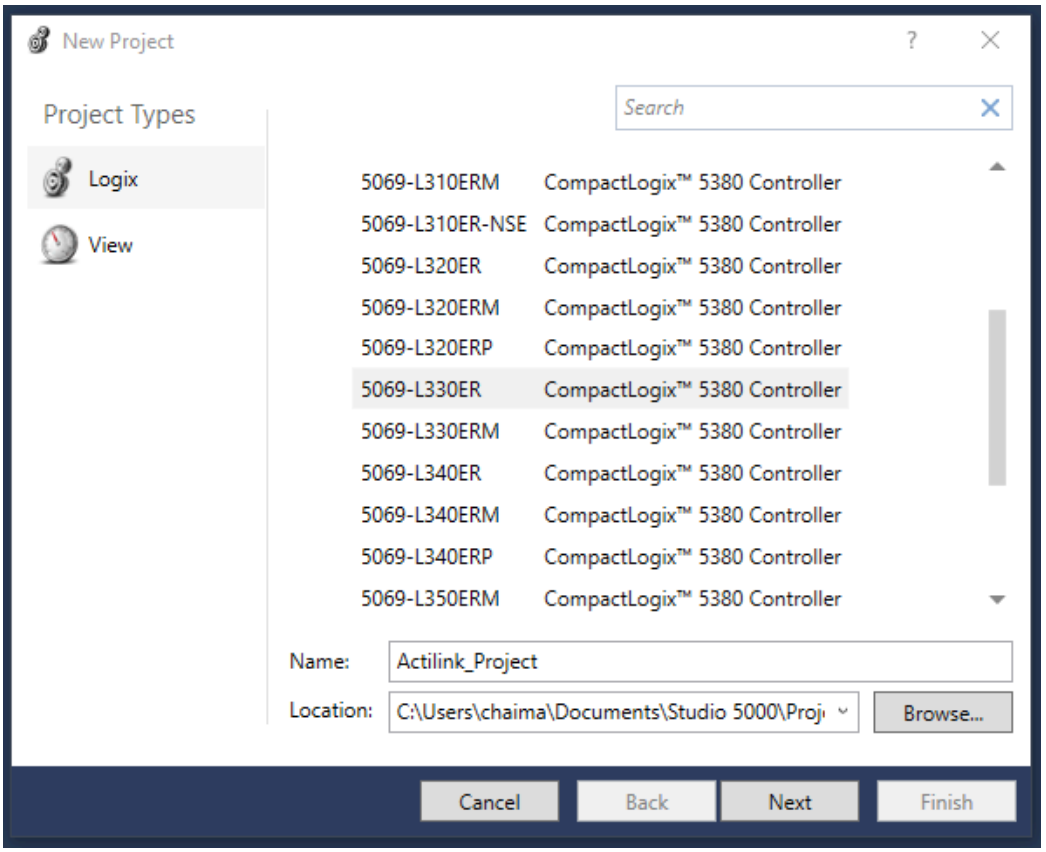


Figure 6.2 - Selecting the specific PLC controller and defining the project name and save location

5. Click **Next** and then **Finish** to create the project. The software will generate the project structure.

6.1.2. Establishing the Network Configuration

Proper network configuration is critical for successful communication between your PC, the PLC, and the ACTILINK-S drive. This involves two main parts: configuring your local computer's network adapter and establishing the network within the Rockwell Studio 5000 Logix Designer software.

The PLC is connected to the computer via an Ethernet port.

IMPORTANT
The PLC's Ethernet ports have a static network and IP address, which are set by the manufacturer. If your PLC has a screen display, it will show the static configuration for all of its ports. If it does not, consult the PLC's documentation to find this information.

In this demo, the CompactLogix 5380 5069-L310ERM PLC is used, which has two Ethernet ports: A1 and A2. The PLC's Ethernet ports have a static network and IP address that are set by the manufacturer. If your PLC has a screen display, it will show the static configuration for all its ports. If it does not, consult the PLC's documentation to find this information.

In this example, port A1 is used, which has the static IP address `192.168.0.10` and is part of the network subnet `192.168.0.0/24`. This port will be used to connect the PLC to the computer.

After obtaining the IP address and the subnet of the port used to connect to the computer, proceed with the following steps.

6.1.2.1. Local Computer Configuration

To ensure that your computer can communicate with the PLC, set a static IP address on your computer's Ethernet adapter, which is on the same subnet as the PLC's port.

To do this:

1. Go to the Control **Panel**.
2. Select **Network and Internet**.
3. Select **Network and Sharing Center**.
4. From the left-hand menu, select **Change adapter settings**.
5. Right-click on the Ethernet adapter to be used for the PLC connection and select **Properties**.
6. In the list of items, click **Internet Protocol Version 4 (TCP/IPv4)** and then click the **Properties** button.
7. Select the **Use the following IP address** option.
8. Set an IP address for the adapter on the same subnet as the PLC's Ethernet port. For this example, since the PLC is `192.168.0.10`, the computer's IP address is set to `192.168.0.3`.
9. Set the **Subnet mask** to `255.255.255.0` to match the PLC's network.

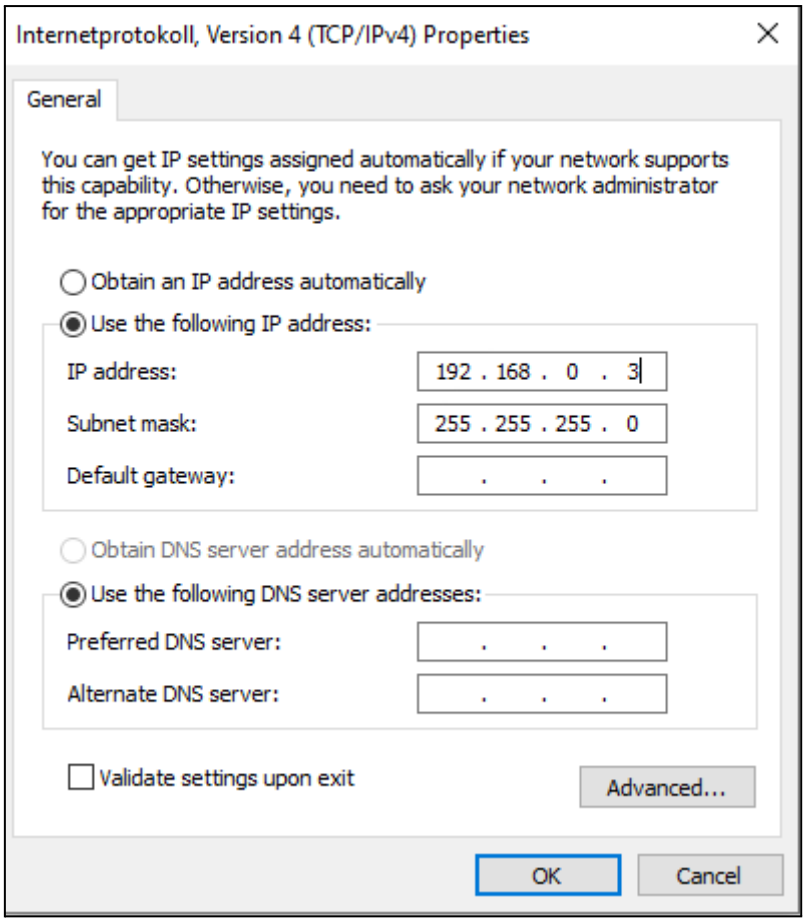


Figure 6.3 - Configuring the IP address and subnet mask for the Ethernet adapter

10. Click **OK** on both windows to save the changes.
11. Finally, connect the PLC to your computer using an Ethernet cable.

6.1.2.2. In-Software Configuration

1. In Rockwell Studio 5000 Logix Designer, go to **Communications** -> **Who Active**.
2. Go to Ethernet, Ethernet
3. Select your PLC (`192.168.0.10`) from the list and confirm the communication path. This will establish a connection between your software and the PLC.

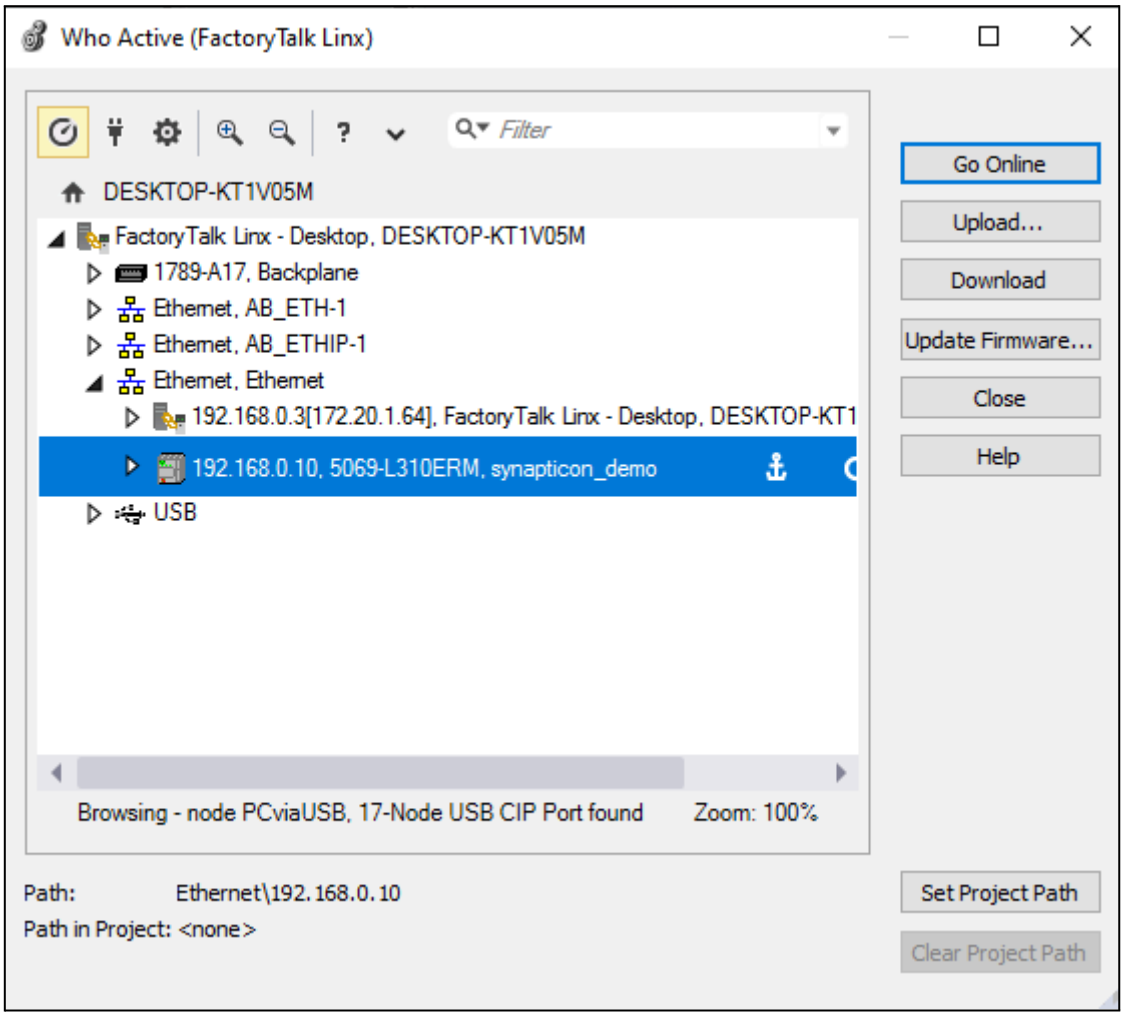


Figure 6.4 - Establishing the communication path and connecting to the PLC in Rockwell Studio 5000 Logix Designer

4. Once the path is confirmed, click **Go Online** to verify that the connection is established between the PLC and the computer. If the connection is successful, the indicators in the software display **Controller OK**.

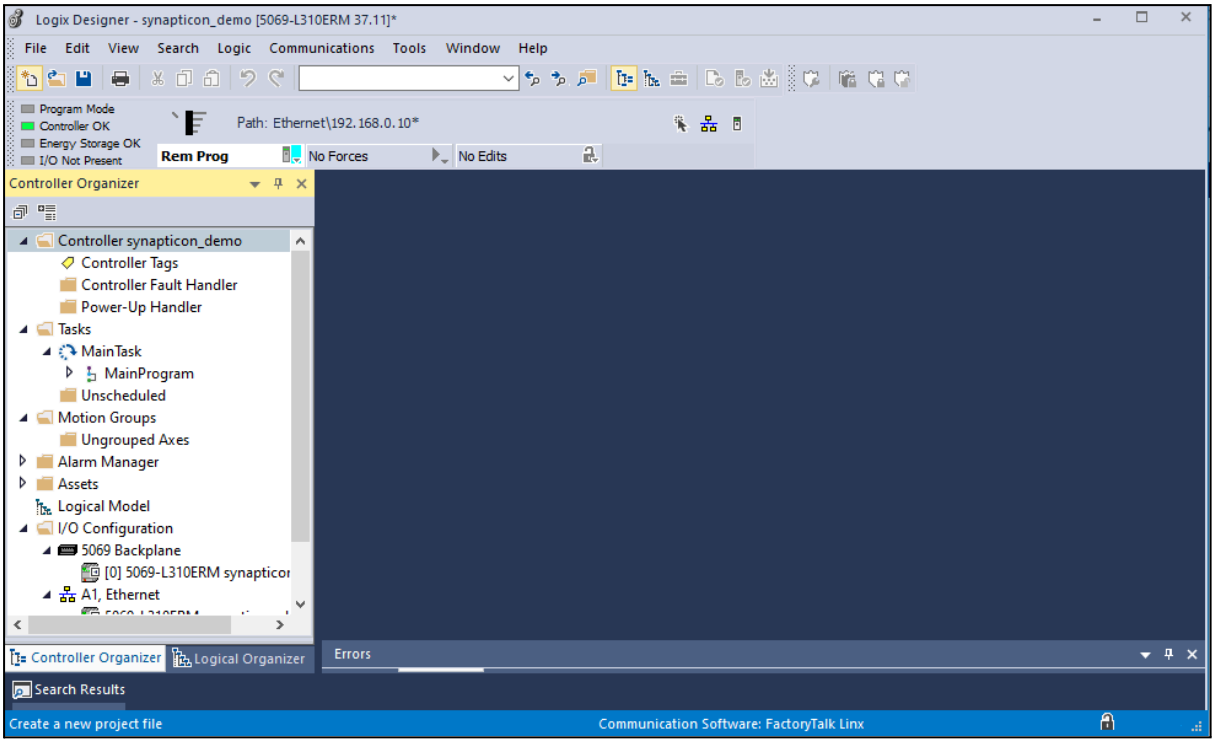


Figure 6.5 - Verifying the connection to the PLC and confirming the "Controller OK" status

6.1.2.3. Customizing PLC Ethernet Port Settings (Optional)

While the PLC's Ethernet ports have a default static IP, these settings can also be customized to fit a specific network architecture. This can be useful for creating a dedicated, isolated network between the PLC and the ACTILINK-S drive.

To customize the network settings for the port connecting to the drive (in this case, port A2), follow these steps:

1. In the project tree on the left, under the **I/O Configuration** section, locate the Ethernet port to be used for the drive connection (port A2).
2. Right-click on the port and select **Properties**.
3. Go to the **Internet Protocol** tab.
4. Select **Manually Configure IP Settings**.
5. Set the new IP address and network mask to be used for this port.
6. Click **OK** to apply the changes.

IMPORTANT
If the IP address and subnet mask of the PLC's Ethernet port are changed, **the ACTILINK-S drive's IP address and subnet mask must be configured to match the new settings**.The drive and the PLC port must be on the same network to communicate successfully.

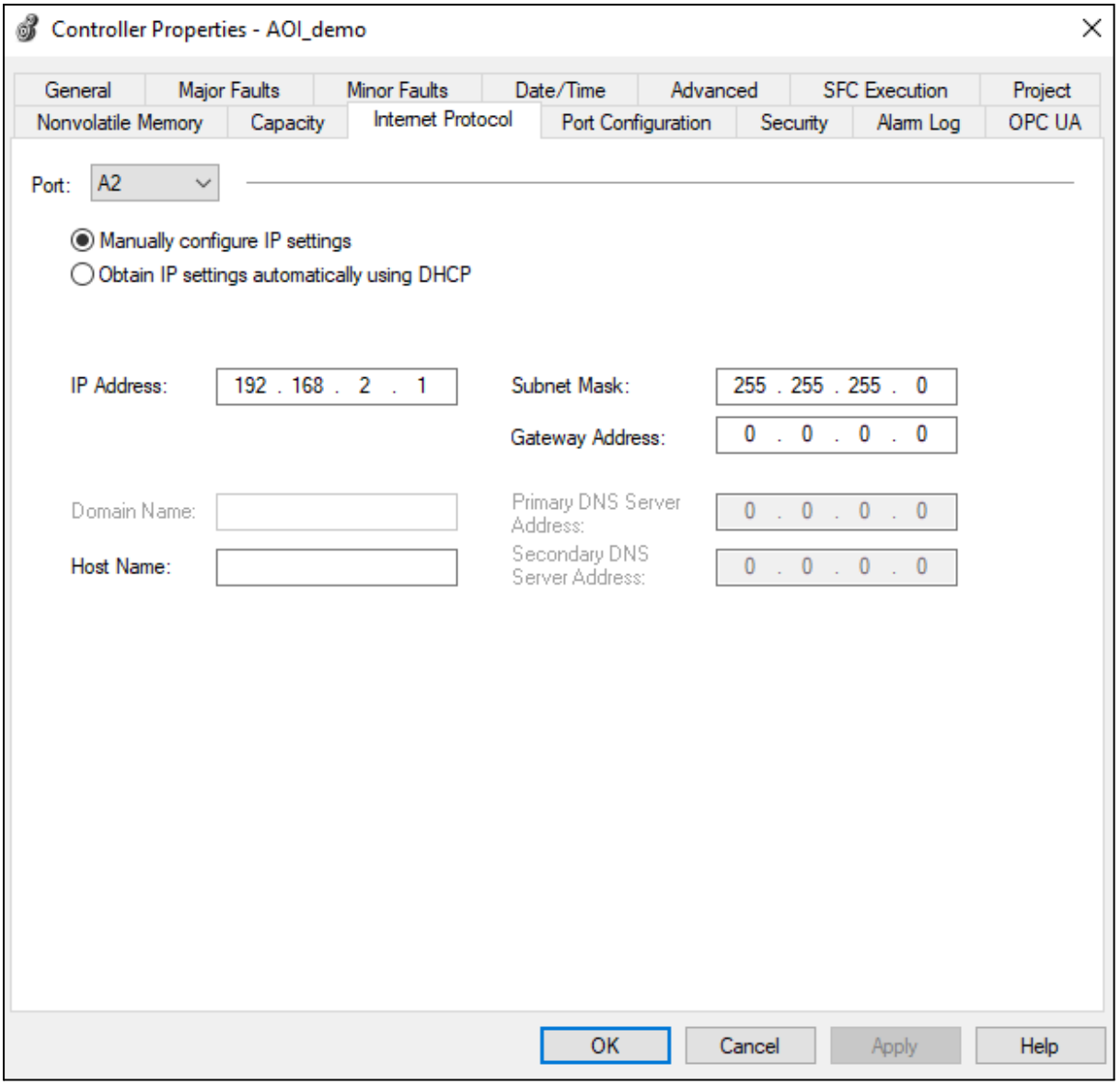


Figure 6.6 - Customizing the network settings for the Ethernet port connecting to the drive

6.1.3. Adding I/O Devices

Once the project is created and the network is configured, the ACTILINK-S drive can be added to the I/O tree. This process requires the drive's EDS file and proper physical and logical connections.

6.1.3.1. Drive Configuration: Configuring the IP Address

Before connecting the drive in Rockwell Studio 5000 Logix Designer, the drive's IP address and subnet mask must be configured to match the network configurations of the port on the PLC that will be used to connect to the ACTILINK-S. Here, the Ethernet port A2 on the CompactLogix 5380 is used. This port has the network subnet `192.168.1.0/24` with the IP address `192.168.1.10`.)

This IP address assignment process can be accomplished using a two-step method, with the initial IP assignment facilitated by a DHCP server.

The TCP/IP network configurations are stored in object `0x2E00` of the drive. By default, the IP address (`0x2E00:1`) and subnet mask (`0x2E00:2`) are set to 0.0.0.0. An IP address of 0 indicates that the drive will request an IP from a DHCP server.

6.1.3.1.1. Step 1: Initial IP Assignment via DHCP

To temporarily assign an IP address to the drive, a DHCP server tool is required. The Rockwell DHCP Tool can be used for this purpose.

1. Download and install the Rockwell DHCP Tool here.
2. Connect the ACTILINK-S drive directly to your computer using an Ethernet cable.
3. Open the Rockwell DHCP Tool.

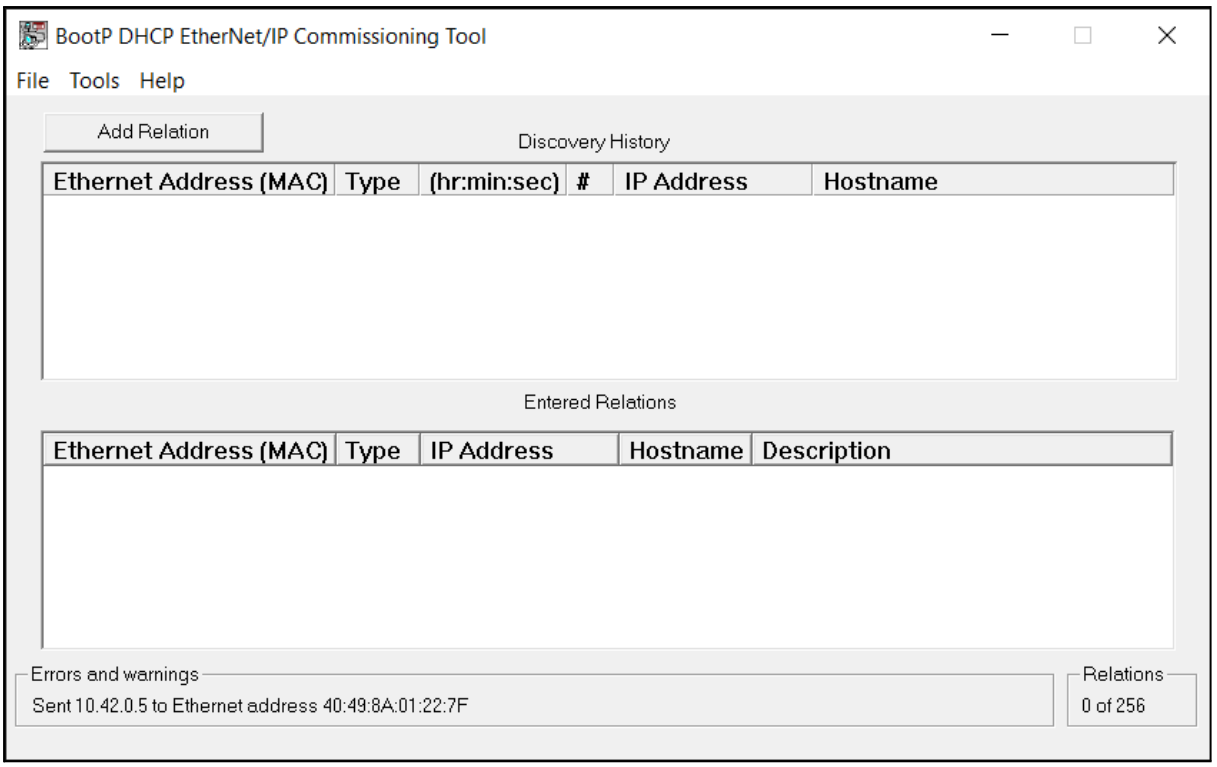


Figure 6.7 - The Rockwell DHCP Tool interface after opening the application

4. Select the Ethernet adapter on your computer that is connected to the drive.

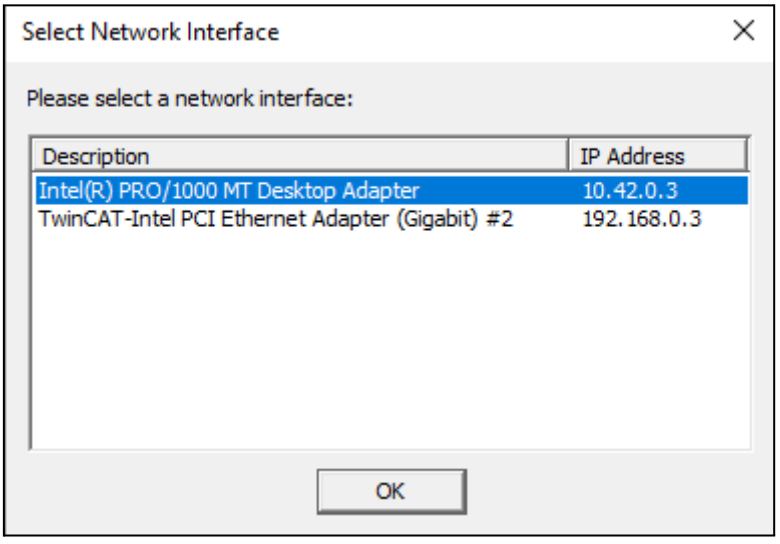


Figure 6.8 - Selecting the Ethernet adapter connected to the drive

5. The drive will begin sending DHCP requests, which will be visible in the tool.

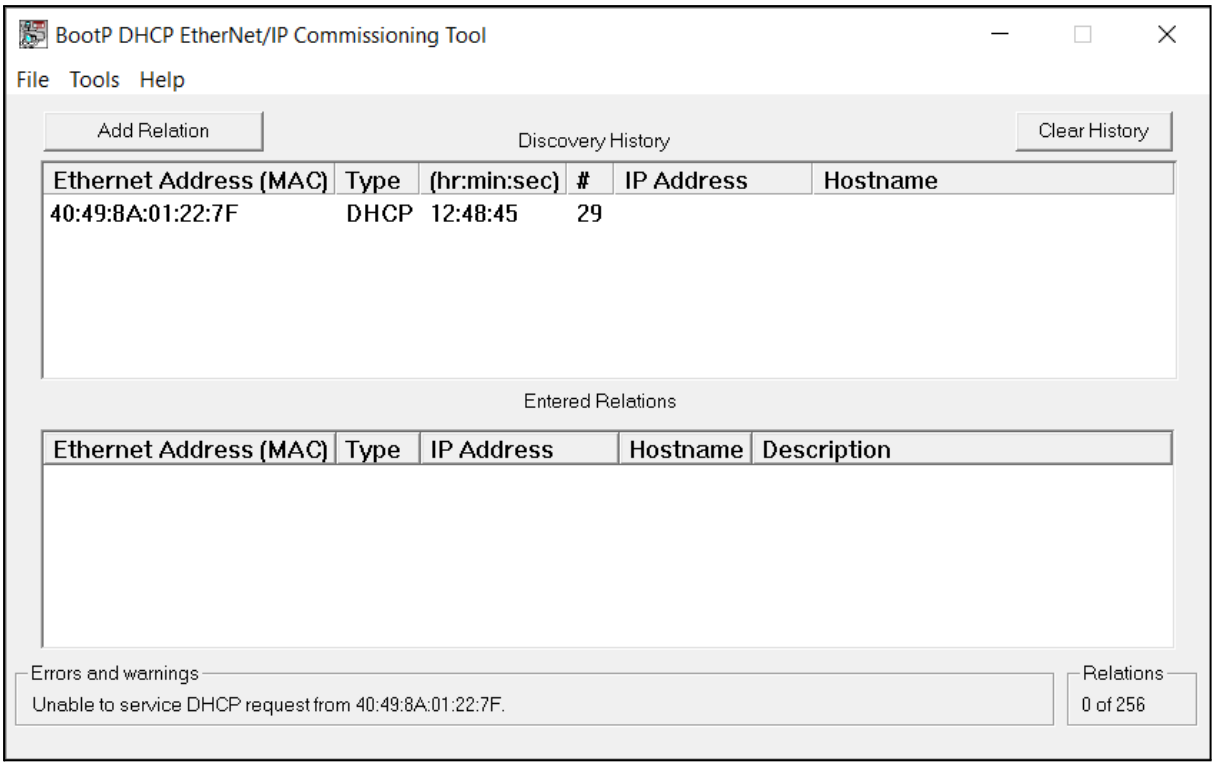


Figure 6.9 - The Rockwell DHCP Tool displaying the DHCP requests sent by the drive

6. Click on the drive's request entry and manually assign an IP address that is on the same subnet as your computer's Ethernet adapter. For example, if the adapter is `10.42.0.3`, then `10.42.0.70` can be assigned to the drive.

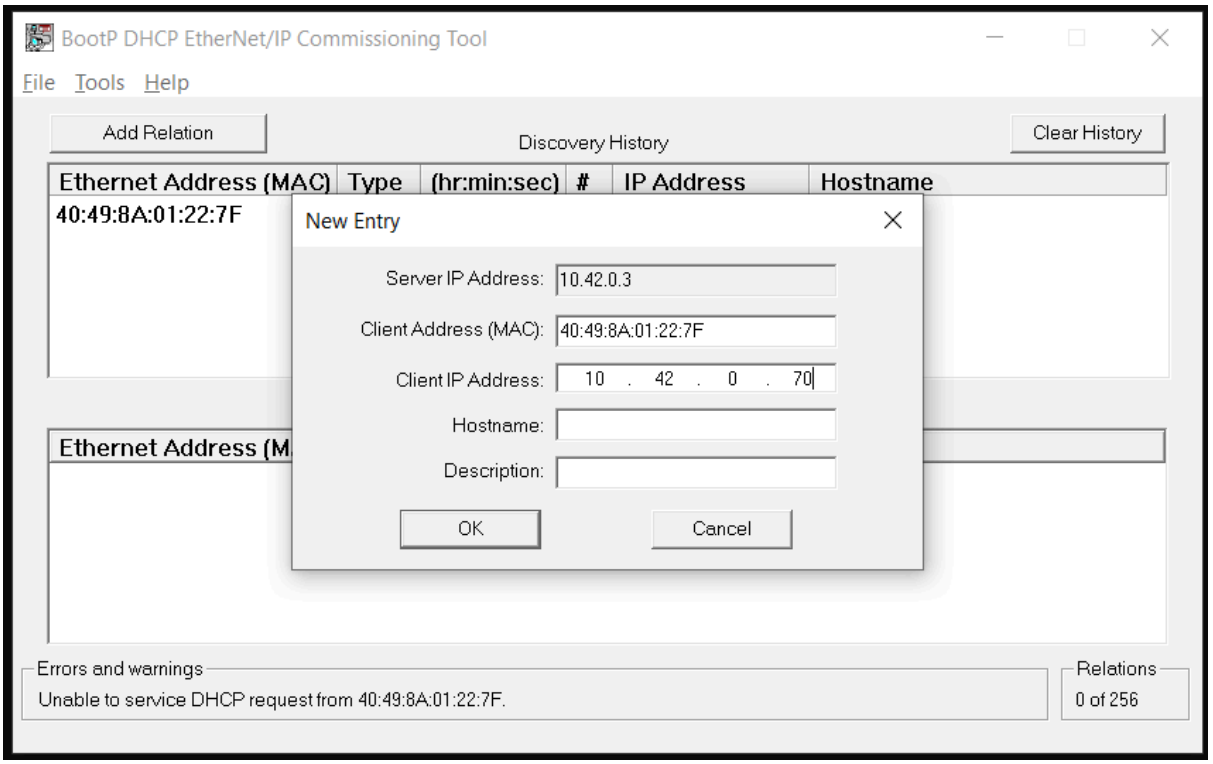


Figure 6.10 - Assigning an IP address to the drive within the Rockwell DHCP Tool

7. Next, the tool sends this IP address to the drive, allowing you to connect to it.

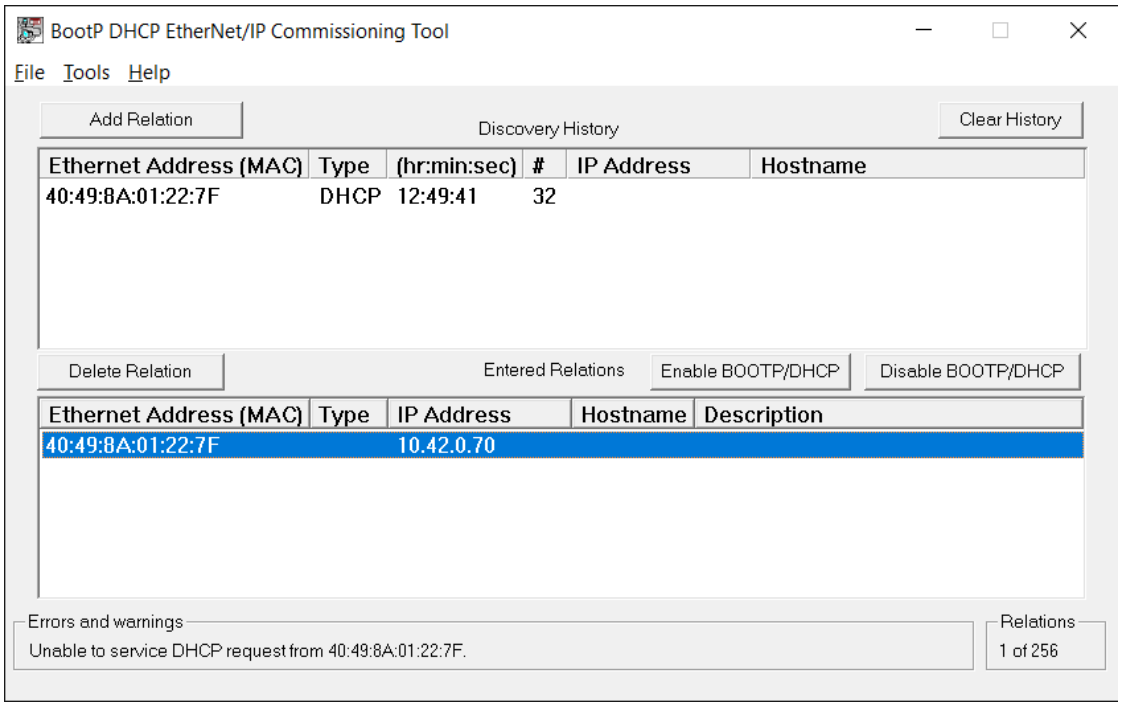


Figure 6.11 - The tool sending the new IP address to the drive

8. The successful assignment of the IP address can be verified by pinging the drive from a command prompt using the assigned IP address.

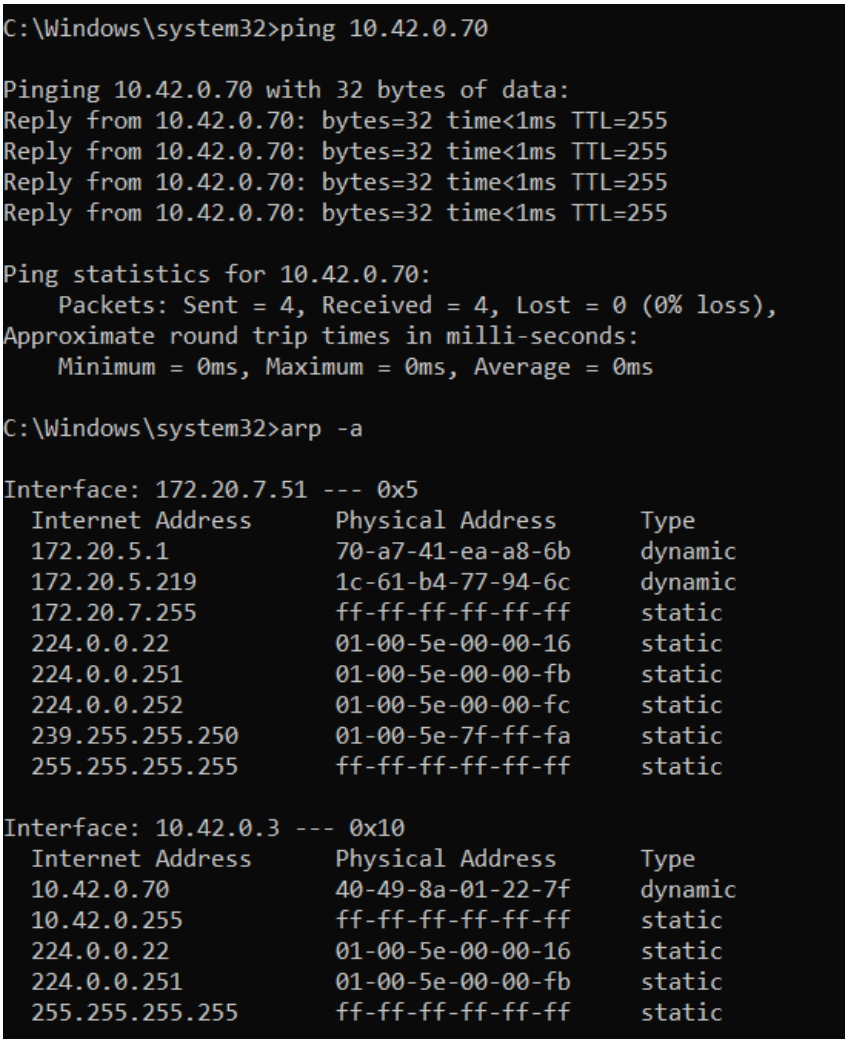


Figure 6.12 - Verifying the successful IP address assignment by pinging the drive from the command prompt

NOTE

The DHCP server tool must be active and running on your computer during this configuration process to successfully assign a temporary IP address to the drive.

6.1.3.1.2. Step 2: Setting the Static IP Address

Once the drive has an IP address, the Synapticon configuration tool OBLAC Drives App can be used to set a permanent, static IP address. This ensures consistent communication with the PLC.

1. Open the OBLAC Drives App (version 25.0.0 or later).
2. Navigate to the **Ethernet** section and select **Add New Device**.
3. Enter the temporary IP address that was assigned in the previous step.

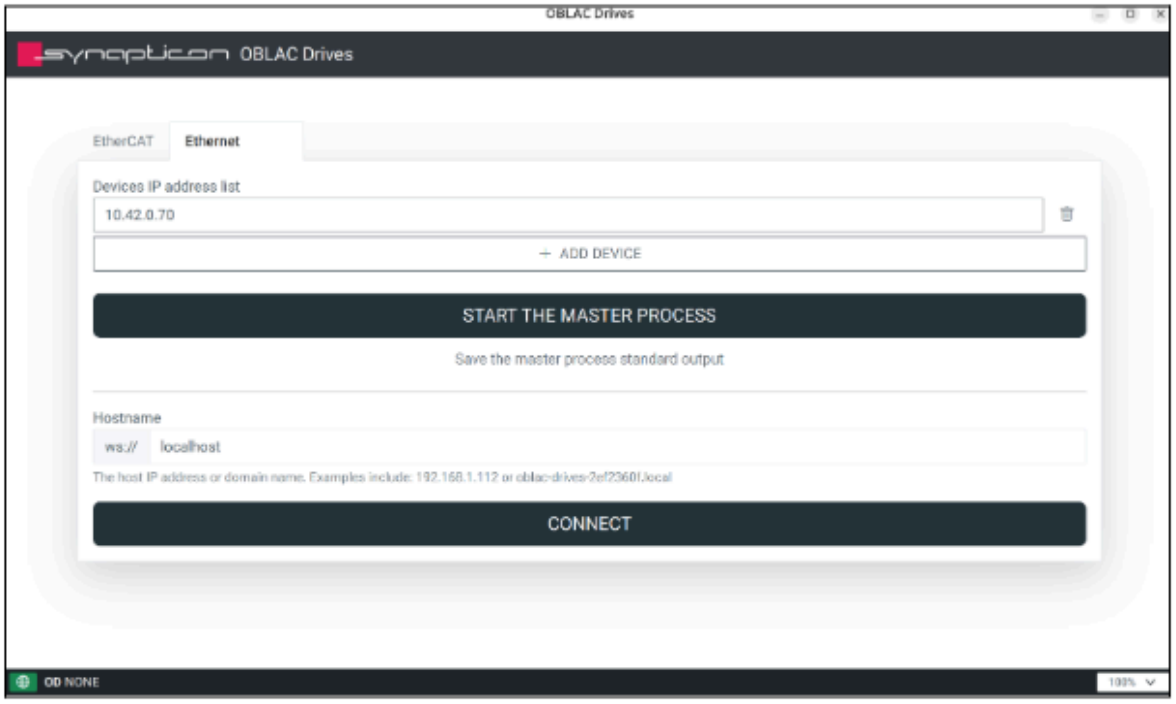


Figure 6.13 - Adding a new device in the OBLAC Drives App using the assigned temporary IP address

4. If the connection is successful, the drive's interface loads.

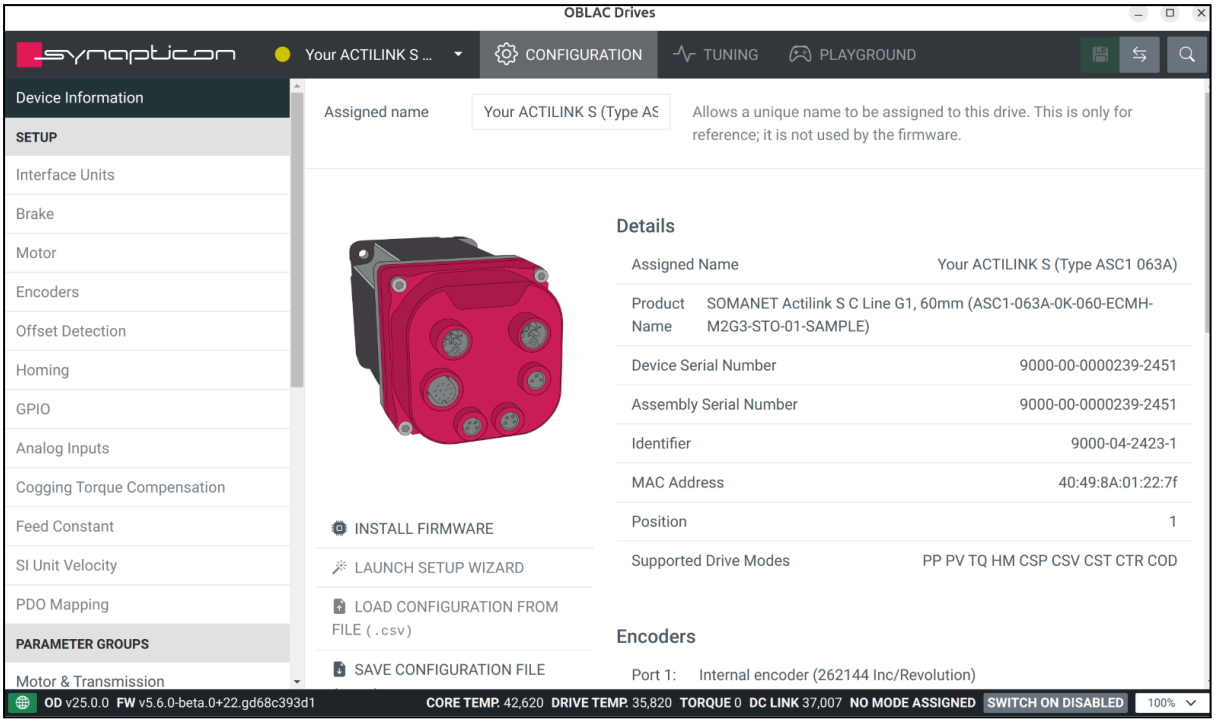


Figure 6.14 - The drive's configuration loaded in the OBLAC Drives App after a successful connection

5. Navigate to object `0x2E00` in the drive's object dictionary.
6. Write the desired static IP address to sub-index `0x2E00:1`. For example, to match a PLC port with a subnet of `192.168.1.0/24`, assign the IP address `192.168.1.11`.
7. Write the corresponding subnet mask to sub-index `0x2E00:2`. For the `192.168.1.0/24` subnet, the mask is `255.255.255.0`.

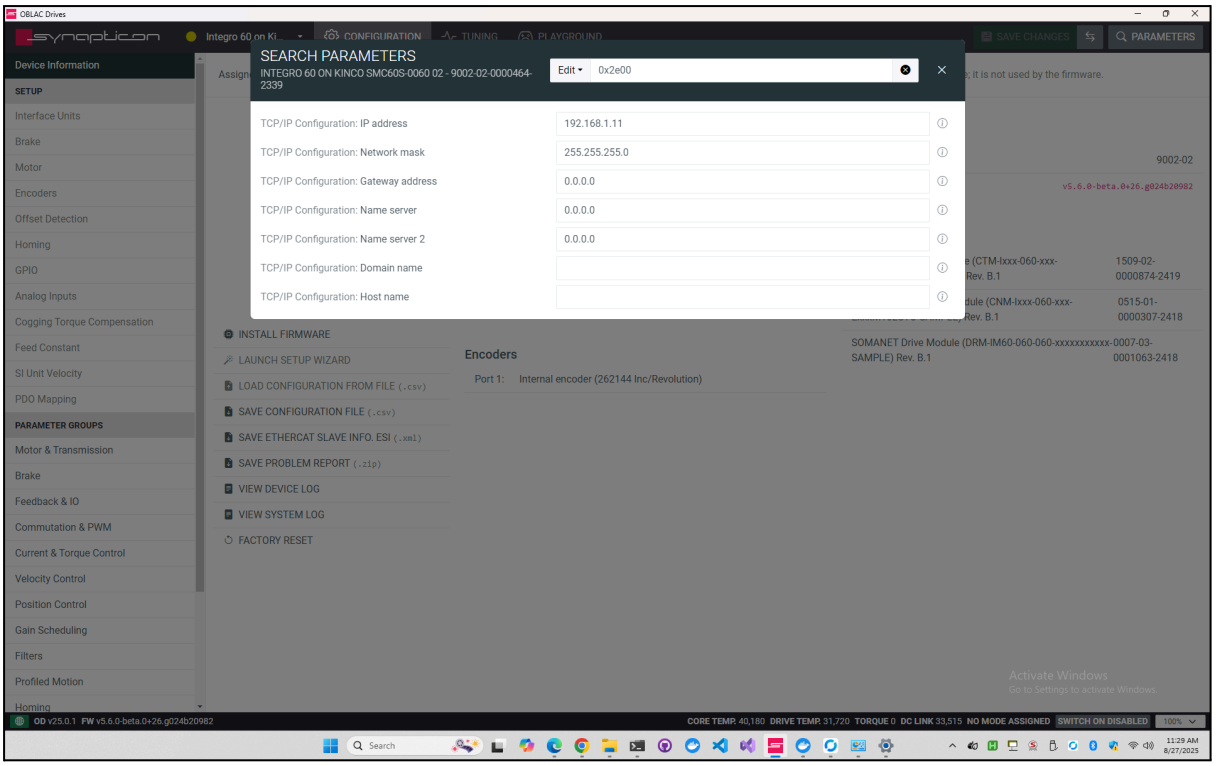


Figure 6.15 - Configuring the static IP address and subnet mask in the drive's object dictionary

8. Save these settings to the drive's memory.
9. On the next power cycle, the drive will use the newly configured static IP address, allowing it to be integrated into the PLC network.

6.1.3.2. Software Side

The following information explains how to integrate the ACTILINK-S drive into your Rockwell Studio 5000 Logix Designer project.

6.1.3.2.1. Download and Install the EDS File

The EDS (Electronic Data Sheet) file is essential for the PLC to understand the drive's communication and data structure.

The EDS file is:

- distributed with the package,
- downloaded from <bundles.synapticon.com>, or
- read directly with the Synapticon configuration tool.

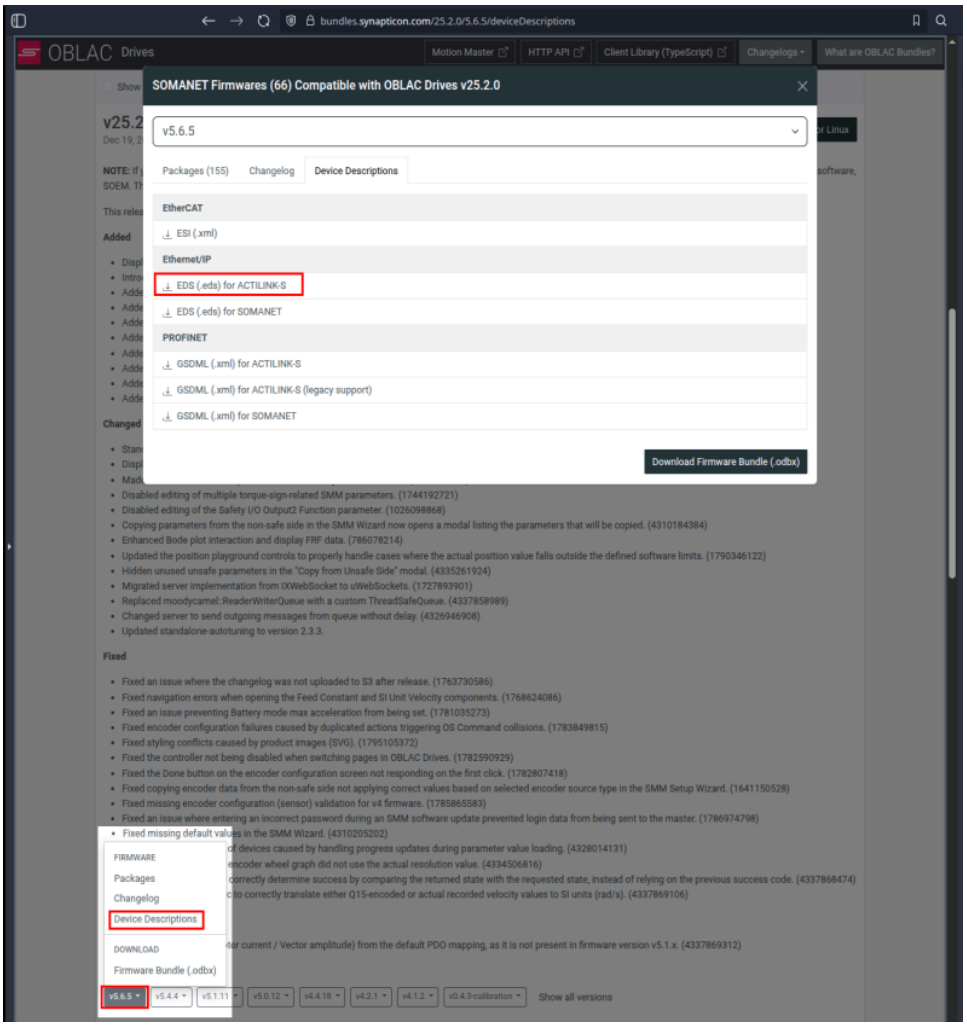


Figure 6.16 - Downloading the device description (EDS)

1. Download the EDS file for the ACTILINK-S from the provided location.
2. In Rockwell Studio 5000 Logix Designer, go to **Tools** -> **Device description installation tool**.

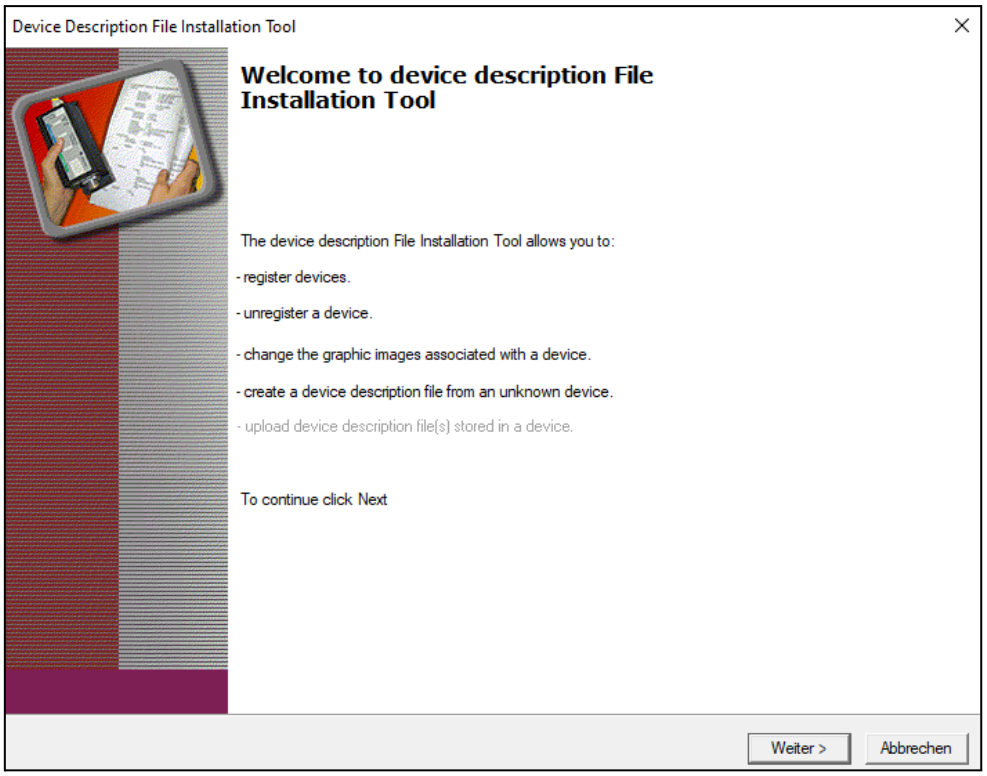


Figure 6.17 - The Device Description Installation Tool in Rockwell Studio 5000 Logix Designer

3. Click on register a device description file

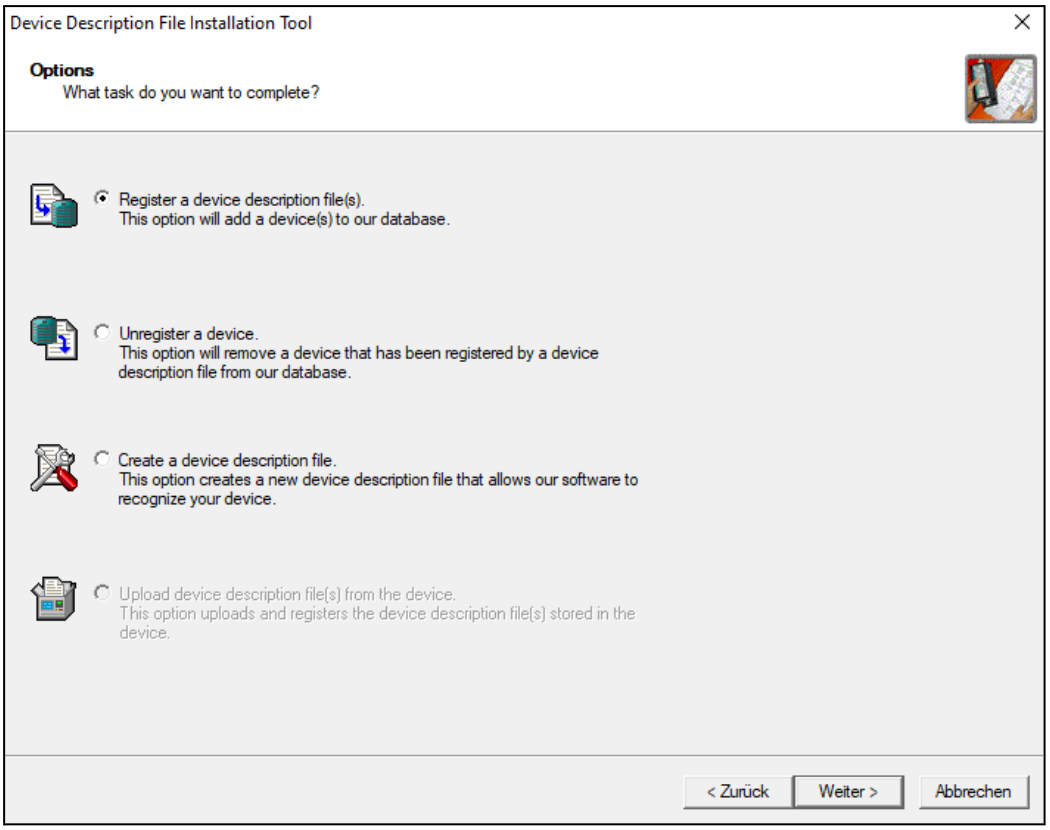


Figure 6.18 - Registering a new device description file

4. Follow the wizard to install the EDS file. This will make the ACTILINK-S device available in the device list.

6.1.3.2.2. Connect the Device

To add the new device:

1. Go to the project tree on the left side of the screen. Under the **I/O Configuration** section, locate the Ethernet port that you are using for the drive connection (in this case, A2) and right-click on it.
2. Select **New Module**.
3. In the device list, the ACTILINK-S drive is displayed. Select it and click **Create**.

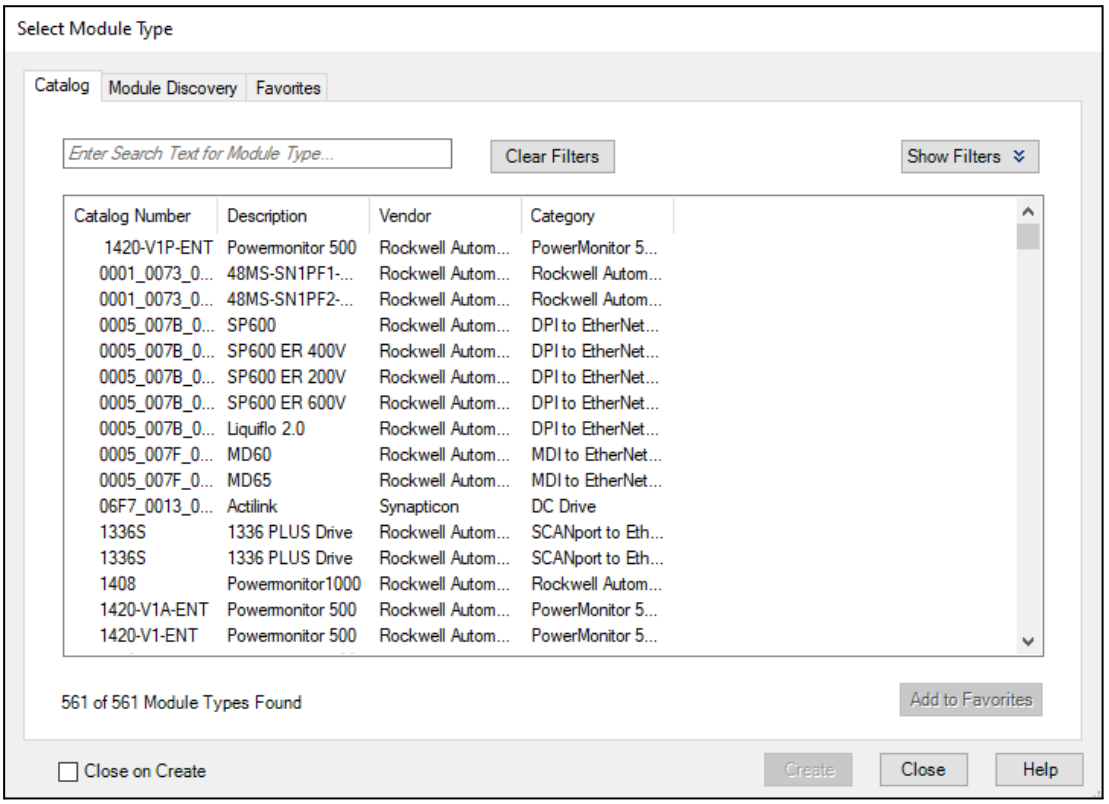


Figure 6.19 - Adding the ACTILINK-S drive as a new module to the I/O Configuration

6.1.3.2.3. Configure the Drive

1. A new window will appear for the module properties.
2. Give the device a descriptive name.
3. Enter the IP address that was previously configured on the ACTILINK-S drive (e.g., 192.168.1.11).

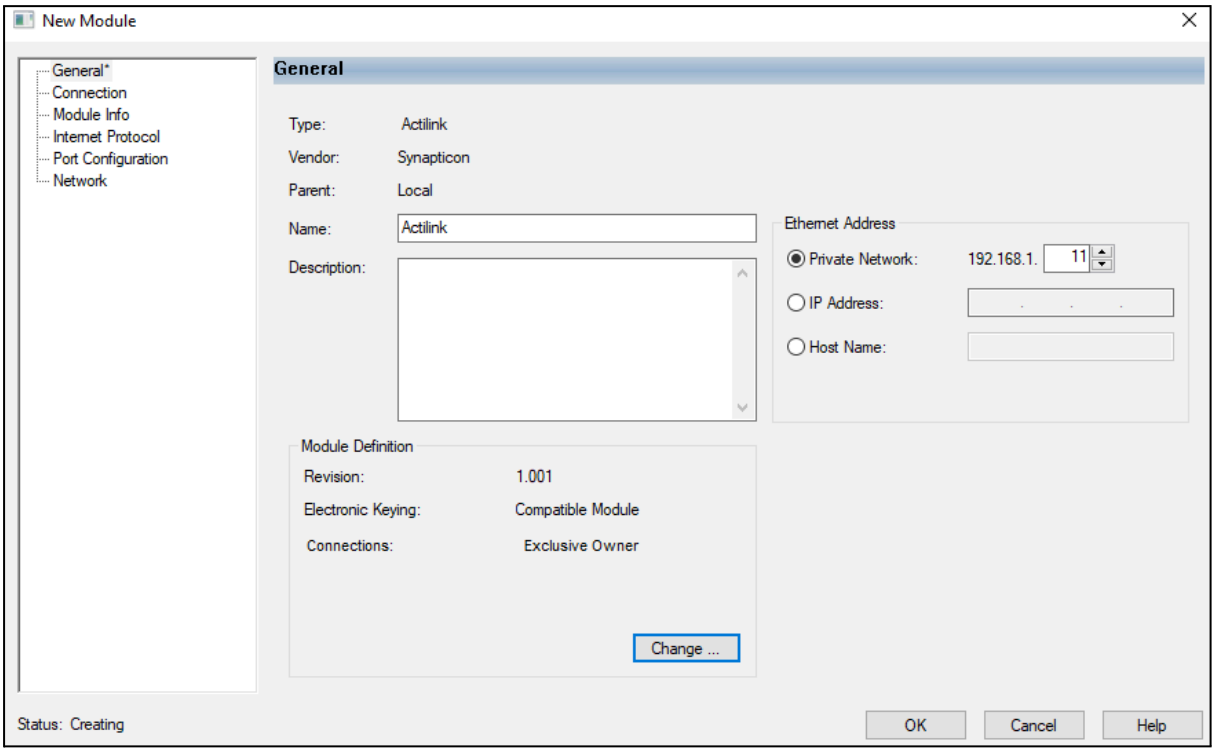


Figure 6.20 - Configuring the module properties and IP address for the ACTILINK-S drive

4. Click **OK** to add the device to your project.

6.1.3.3. Verify the Connection

Once the project is fully configured, download it to the PLC.

To do this:

1. Go to **Communications** -> **Download**.
2. After the download is complete, the PLC will attempt to establish a connection with the ACTILINK-S drive.
3. To verify the connection, look at the I/O tree. The icon for the ACTILINK-S drive will turn green, and the status will show "OK" if the connection is successful.

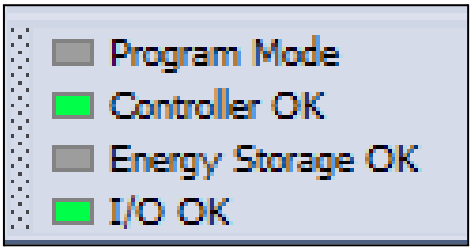


Figure 6.21 - Downloading the project to the PLC and verifying the successful connection to the ACTILINK-S drive

4. If the connection fails, the icon will have a red "X" or a fault indicator. Check the logs in Rockwell Studio 5000 Logix Designer for specific error messages to troubleshoot the issue. Common causes include an incorrect IP address, a network cable issue, or a missing/incorrect EDS file.