



Smart Camera Industrial Protocol

Operation Manual

Legal Information

© 2021 Hangzhou Hikrobot Co., Ltd. All rights reserved.

About this Manual

The Manual includes instructions for using and managing the Product. Pictures, charts, images and all other information hereinafter are for description and explanation only. The information contained in the Manual is subject to change, without notice, due to firmware updates or other reasons. Please find the latest version of this Manual at the Hikrobot website (<https://en.hikrobotics.com/>). Please use this Manual with the guidance and assistance of professionals trained in supporting the Product.

Trademarks

HIKROBOT and other Hikrobot's trademarks and logos are the properties of Hikrobot in various jurisdictions. Other trademarks and logos mentioned are the properties of their respective owners.

Disclaimer

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THIS MANUAL AND THE PRODUCT DESCRIBED, WITH ITS HARDWARE, SOFTWARE AND FIRMWARE, ARE PROVIDED "AS IS" AND "WITH ALL FAULTS AND ERRORS". HIKROBOT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, MERCHANTABILITY, SATISFACTORY QUALITY, OR FITNESS FOR A PARTICULAR PURPOSE. THE USE OF THE PRODUCT BY YOU IS AT YOUR OWN RISK. IN NO EVENT WILL HIKROBOT BE LIABLE TO YOU FOR ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL, OR INDIRECT DAMAGES, INCLUDING, AMONG OTHERS, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, OR LOSS OF DATA, CORRUPTION OF SYSTEMS, OR LOSS OF DOCUMENTATION, WHETHER BASED ON BREACH OF CONTRACT, TORT (INCLUDING NEGLIGENCE), PRODUCT LIABILITY, OR OTHERWISE, IN CONNECTION WITH THE USE OF THE PRODUCT, EVEN IF HIKROBOT HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR LOSS.

YOU ACKNOWLEDGE THAT THE NATURE OF INTERNET PROVIDES FOR INHERENT SECURITY RISKS, AND HIKROBOT SHALL NOT TAKE ANY RESPONSIBILITIES FOR ABNORMAL OPERATION, PRIVACY LEAKAGE OR OTHER DAMAGES RESULTING FROM CYBER-ATTACK, HACKER ATTACK, VIRUS INFECTION, OR OTHER INTERNET SECURITY RISKS; HOWEVER, HIKROBOT WILL PROVIDE TIMELY TECHNICAL SUPPORT IF REQUIRED. YOU AGREE TO USE THIS PRODUCT IN COMPLIANCE WITH ALL APPLICABLE LAWS, AND YOU ARE SOLELY RESPONSIBLE FOR ENSURING THAT YOUR USE CONFORMS TO THE APPLICABLE LAW. ESPECIALLY, YOU ARE RESPONSIBLE, FOR USING THIS PRODUCT IN A MANNER THAT DOES NOT INFRINGE ON THE RIGHTS OF THIRD PARTIES, INCLUDING WITHOUT LIMITATION, RIGHTS OF PUBLICITY, INTELLECTUAL PROPERTY RIGHTS, OR DATA PROTECTION AND OTHER PRIVACY RIGHTS. YOU SHALL NOT USE THIS PRODUCT FOR ANY PROHIBITED END-USES, INCLUDING THE DEVELOPMENT OR PRODUCTION OF WEAPONS OF MASS DESTRUCTION, THE DEVELOPMENT OR PRODUCTION OF CHEMICAL OR BIOLOGICAL WEAPONS, ANY ACTIVITIES IN THE CONTEXT RELATED TO ANY NUCLEAR

EXPLOSIVE OR UNSAFE NUCLEAR FUEL-CYCLE, OR IN SUPPORT OF HUMAN RIGHTS ABUSES.

THE PERFORMANCE DATA IN THIS PUBLICATION IS BASED ON HIKROBOT'S INTERNAL RESEARCH/EVALUATION. ACTUAL DATA MAY VARY DEPENDING ON SPECIFIC CONFIGURATIONS AND OPERATING CONDITIONS AND HIKROBOT SHALL NOT BEAR THE CONSEQUENCES ARISING THEREFROM.

IN THE EVENT OF ANY CONFLICTS BETWEEN THIS MANUAL AND THE APPLICABLE LAW, THE LATTER PREVAILS.

Symbol Conventions

The symbols that may be found in this document are defined as follows.

Symbol	Description
 Danger	Indicates a hazardous situation which, if not avoided, will or could result in death or serious injury.
 Caution	Indicates a potentially hazardous situation which, if not avoided, could result in equipment damage, data loss, performance degradation, or unexpected results.
 Note	Provides additional information to emphasize or supplement important points of the main text.

Available Model

This manual is applicable to the SC2000Pro Series, SC2000E Series, SC3000 Series, and SC7000Pro Series smart camera.

Contact Information

Hangzhou Hikrobot Co., Ltd.

E-mail: tech_support@hikrobotics.com

Website: <https://en.hikrobotics.com/>

Contents

Chapter 1 EtherNet/IP	1
1.1 Introduction.....	1
1.2 Hardware Wiring.....	1
1.3 Smart Camera Settings.....	1
1.4 PLC Settings.....	4
1.4.1 Create Sysmac Studio Project.....	4
1.4.2 Network Configurator Settings.....	9
1.5 Input and Output Mapping.....	13
1.5.1 Input Mapping.....	13
1.5.2 Output Mapping.....	14
1.6 Trigger Test.....	16
1.7 Project Switching Test.....	18
Chapter 2 Profinet	20
2.1 Introduction.....	20
2.2 Hardware Wiring.....	20
2.3 Smart Camera Settings.....	20
2.4 PLC Settings.....	23
2.5 Device Module Definition.....	26
2.6 Trigger Test.....	30
2.7 Project Switching Test.....	34
Chapter 3 ModBus	37
3.1 Introduction.....	37
3.2 Hardware Wiring.....	37
3.3 Smart Camera Settings.....	37
3.4 ModBus TCP Test Tool.....	40
3.5 PLC Settings.....	42
3.6 Device Register Definition.....	45
3.7 Trigger Test.....	48
3.8 Project Switching Test.....	53

Chapter 1 EtherNet/IP

1.1 Introduction

EtherNet/IP is a network suitable for industrial environment and time-critical applications. EtherNet/IP uses standard Ethernet, TCP/IP technology and CIP's open application layer protocol, and it can share the network with normal Ethernet communication.

Note

This chapter takes Omron NX series PLC as an example to explain how to communicate with the smart camera via the EtherNet/IP communication protocol. For other devices, refer to the user manual provided by the manufacturer and make related settings by reading contents of this chapter.

1.2 Hardware Wiring

The wiring of Omron NX series PLC and the smart camera is shown below.

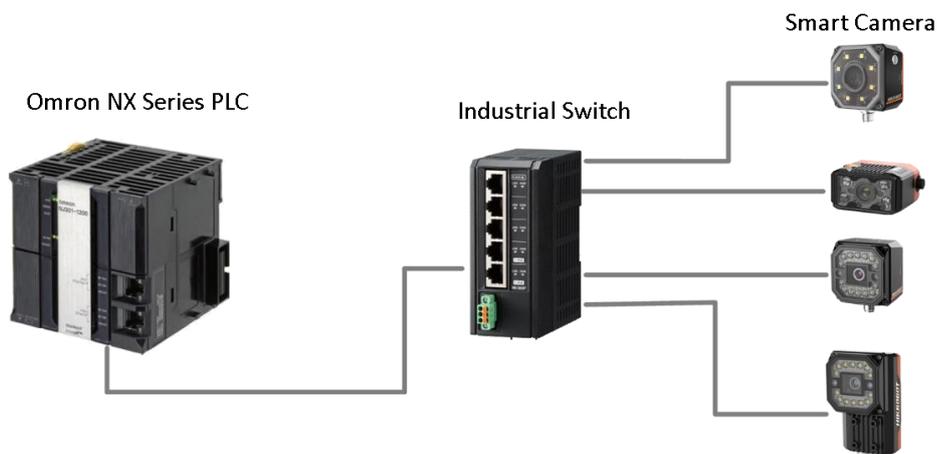


Figure 1-1 Hardware Wiring

1.3 Smart Camera Settings

Before using the smart camera's EtherNet/IP function to communicate with PLC devices, you need to set the smart camera first via the SCMVS client software.

Before you start:

- Make sure that the PC has installed the SCMVS client software.

- Check the device's firmware version.

Note

Regarding SC2000Pro series smart camera, you need to operate it via the web, and operations are similar to those via the SCMVS client software.

Steps

1. Log in the device via the SCMVS client software.
 2. Click **Communication** on the menu bar.
 3. Click  to add Ethernet/IP communication.
 4. Switch on  to enable Ethernet/IP communication.
 5. Set **Input Size** and **Output Size** according to actual demands.
-

Note

The default value is 200.

6. (Optional) Enable **Result Byte Swap** according to actual demands.

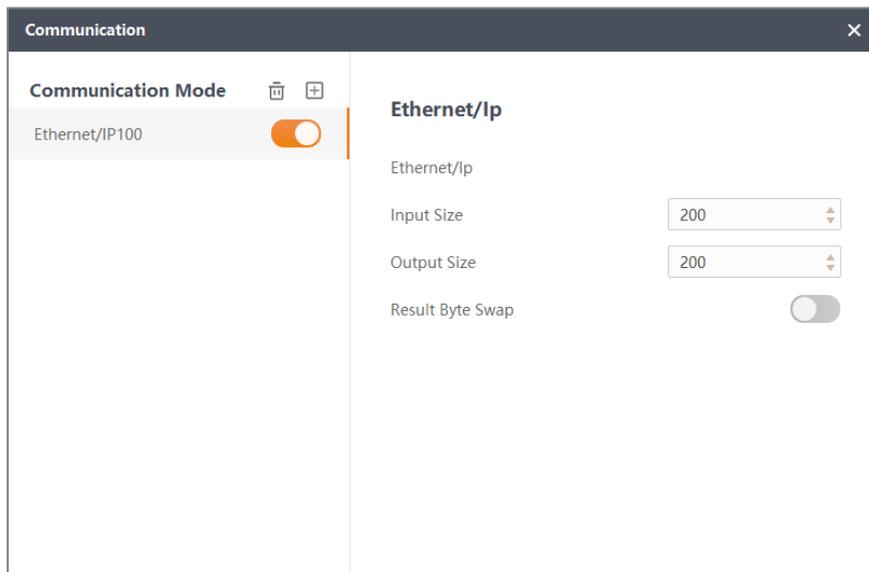


Figure 1-2 Set Communication Parameters

7. Create a project and set parameters for cameras, base images, and vision tools according to actual demands.

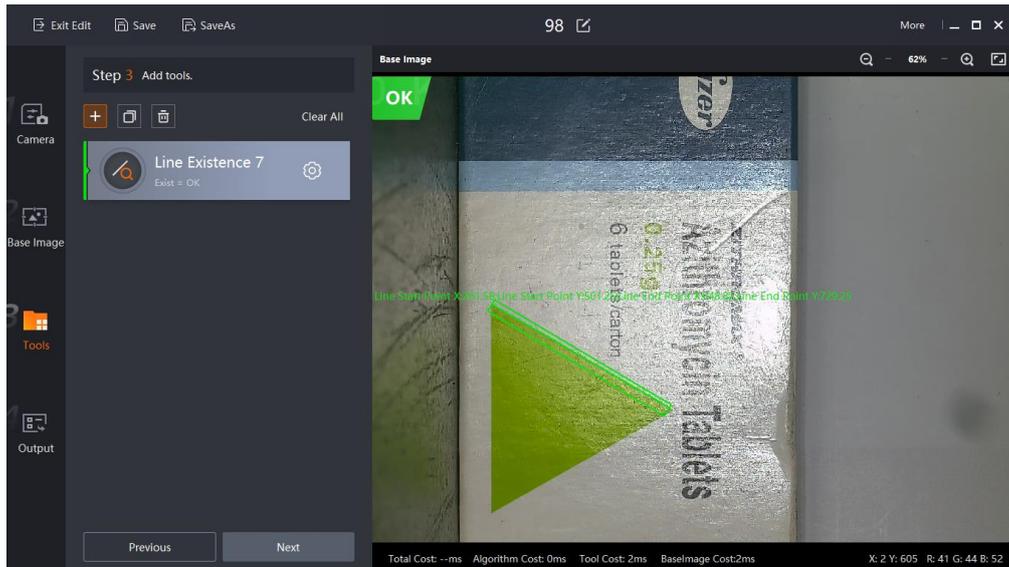


Figure 1-3 Line Existence

Note

Here we take the tool of line existence as an example.

8. Go to **Output** → **Tool Results** → **Edit** to set the outputted content.

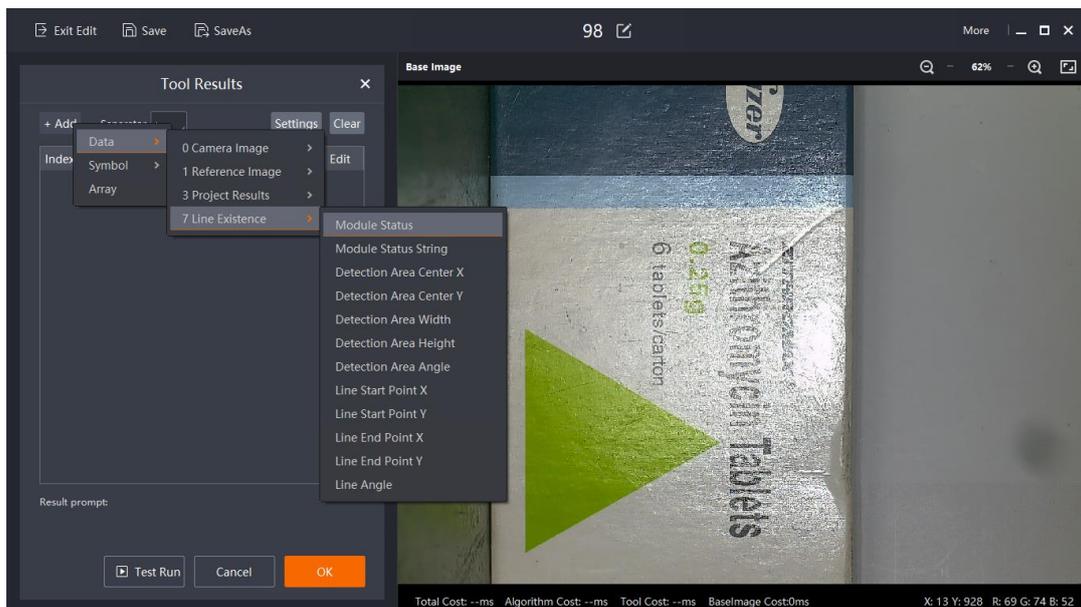


Figure 1-4 Set Tool Results

Note

Here we take the module status and module status string of line existence as an example to introduce tool results.

9. Save the project.
10. If you want to switch projects via communications, go to project management, select **Trigger Communication** as **Mode**, enter **Communication String**, and **Switch Return String** according to actual demands.

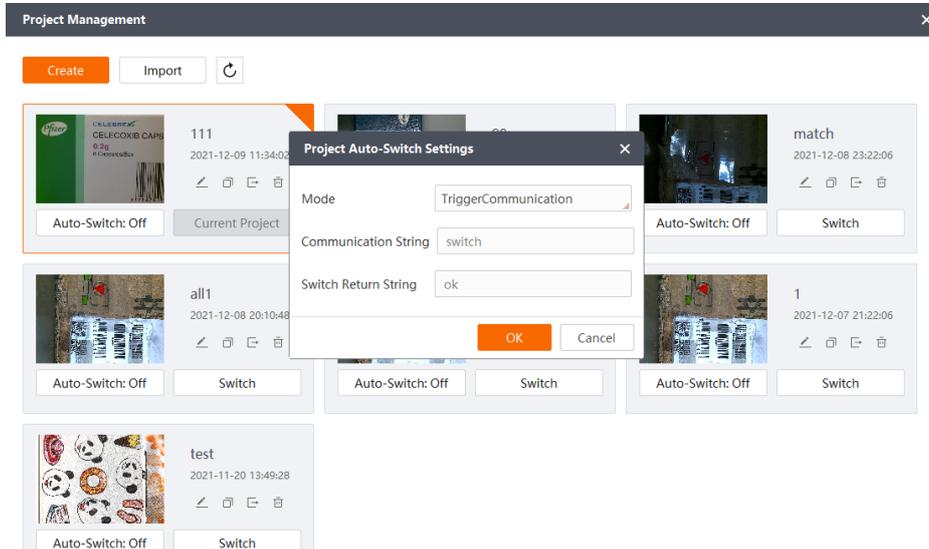


Figure 1-5 Project Switch Settings

1.4 PLC Settings

This section takes Omron NX series PLC as an example to explain how to set PLC.

1.4.1 Create Sysmac Studio Project

You need to create Sysmac Studio project first when using Omron NX series PLC.

Steps

1. Run Sysmac Studio, click **New Project**, enter parameters, and click **Create**.

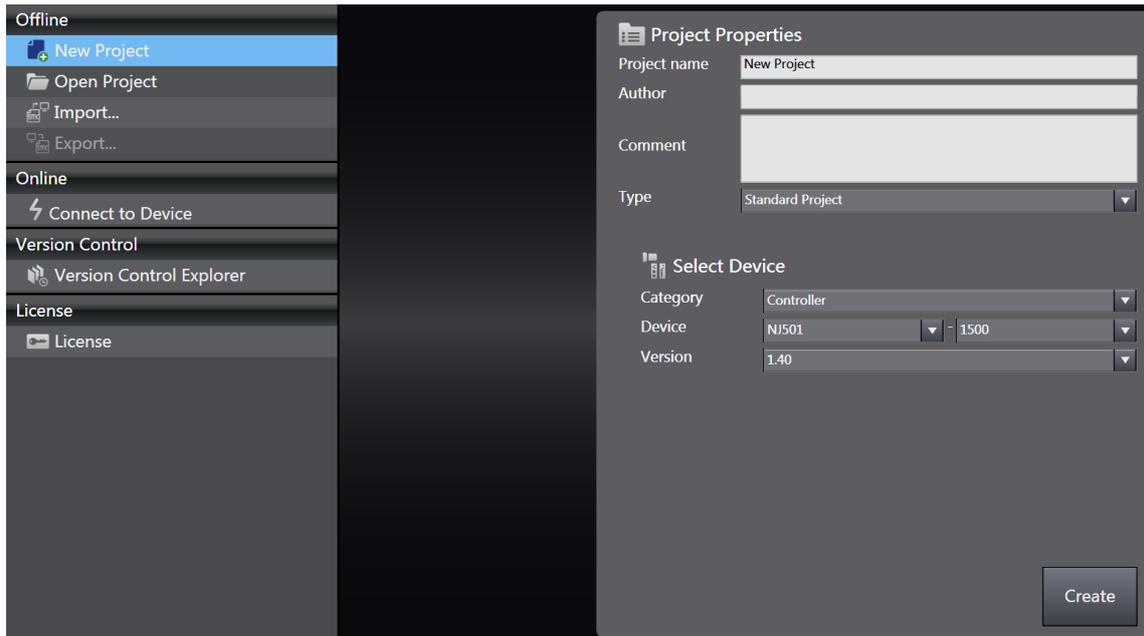


Figure 1-6 Create New Project

2. Create a new **Global Variables** whose size should be the same with that of the smart camera configured in the SCMVS.

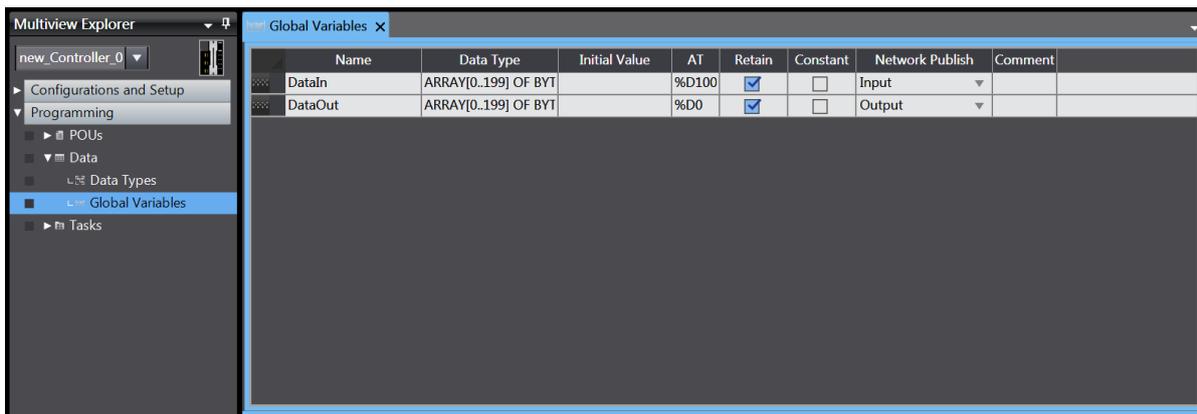


Figure 1-7 Create Global Variables

Note

When connecting multiple smart cameras, you should create several global variables with different names.

3. Go to **Controller Setup** → **Memory Settings**, check **Enable** of DM, and allocate the global variables created in step 2 to DM memory.

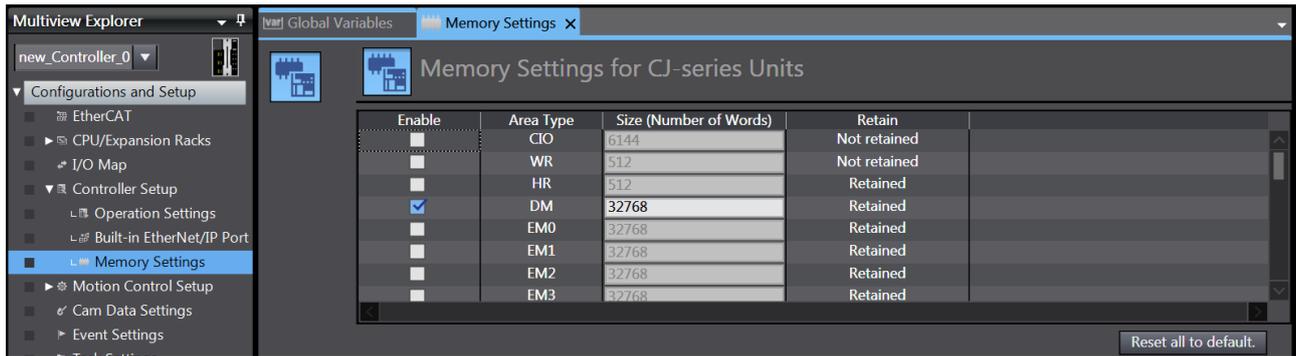


Figure 1-8 Enable DM Area Type

4. Go to **Controller Setup** → **Built-in EtherNet/IP Port Settings**, and set the IP address of PLC port 1.

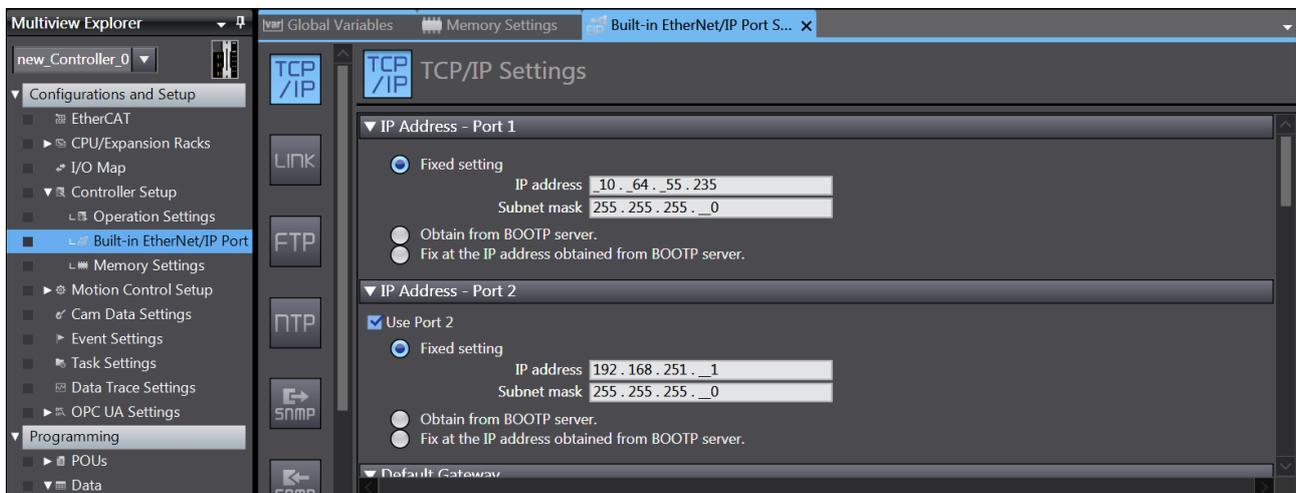


Figure 1-9 Set IP Address

Note

Make sure that the PLC and the smart camera's IP address are in the same network segment.

5. Click **Tools** in the menu, and select **EtherNet/IP Connection Settings**.

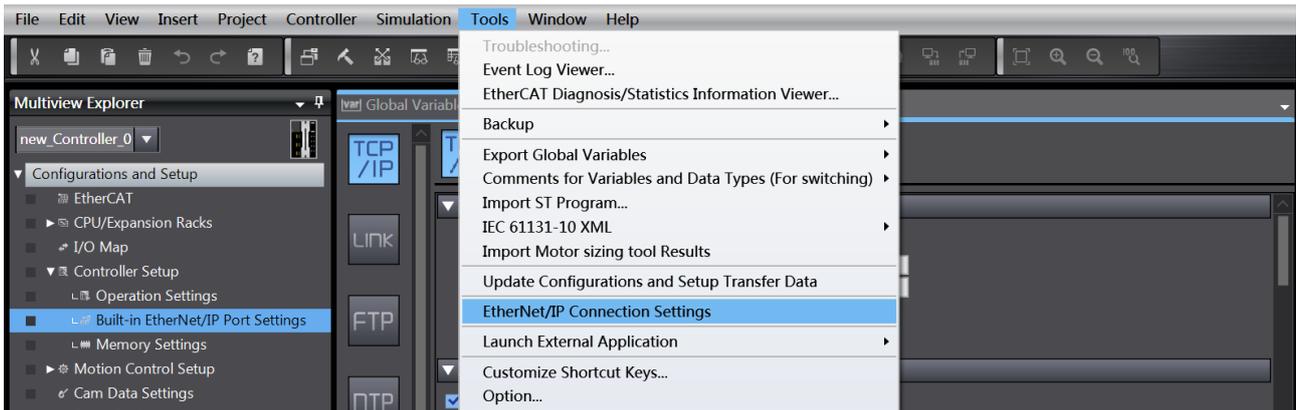


Figure 1-10 Open EtherNet/IP Connection Settings

6. Double click port 1 in **EtherNet/IP Device List**, and click **Registration All** in tag sets to register global variables.

Node Address	Device	Description
10.64.55.235	Built-in EtherNet/IP Port Settings - Port 1	NX102-9000
192.168.251.1	Built-in EtherNet/IP Port Settings - Port 2	NX102-9000

Figure 1-11 EtherNet/IP Device List

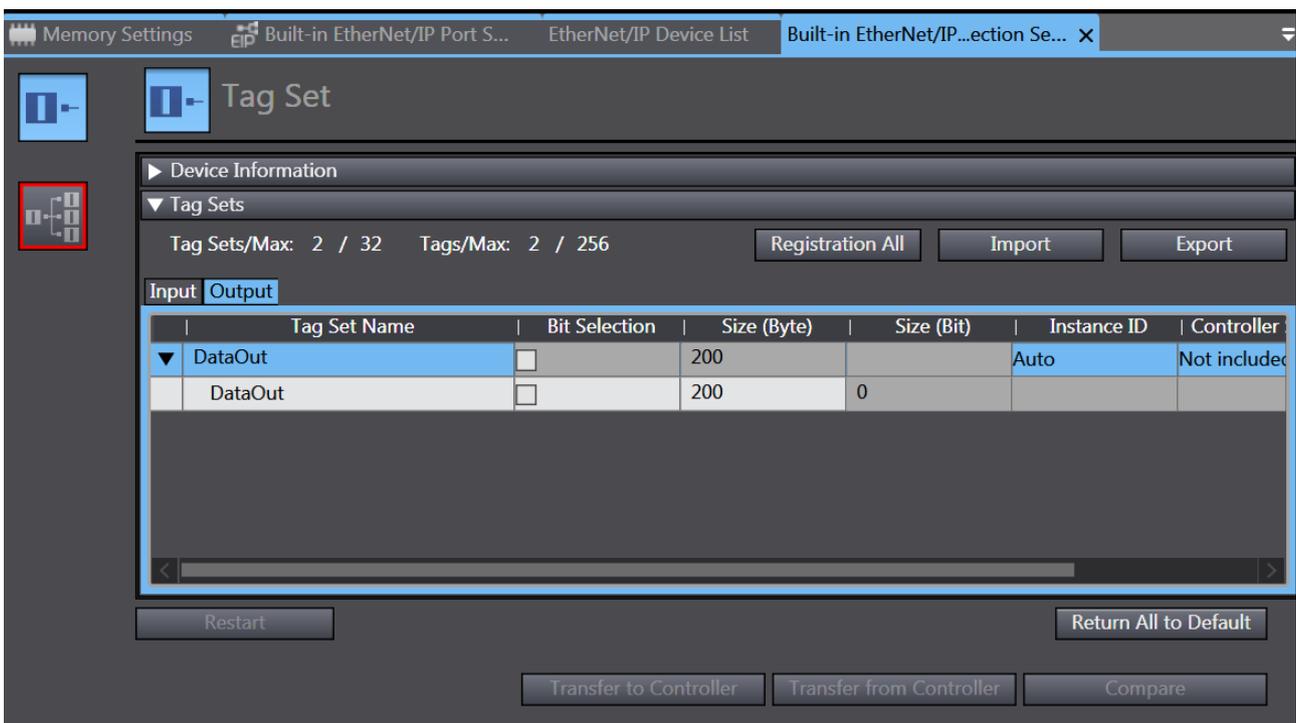


Figure 1-12 Register Global Variables

7. Right click in the **Target Device** area, click **Display EDS Library**, select the corresponding EDS file in the EDS library, and click **Install**.

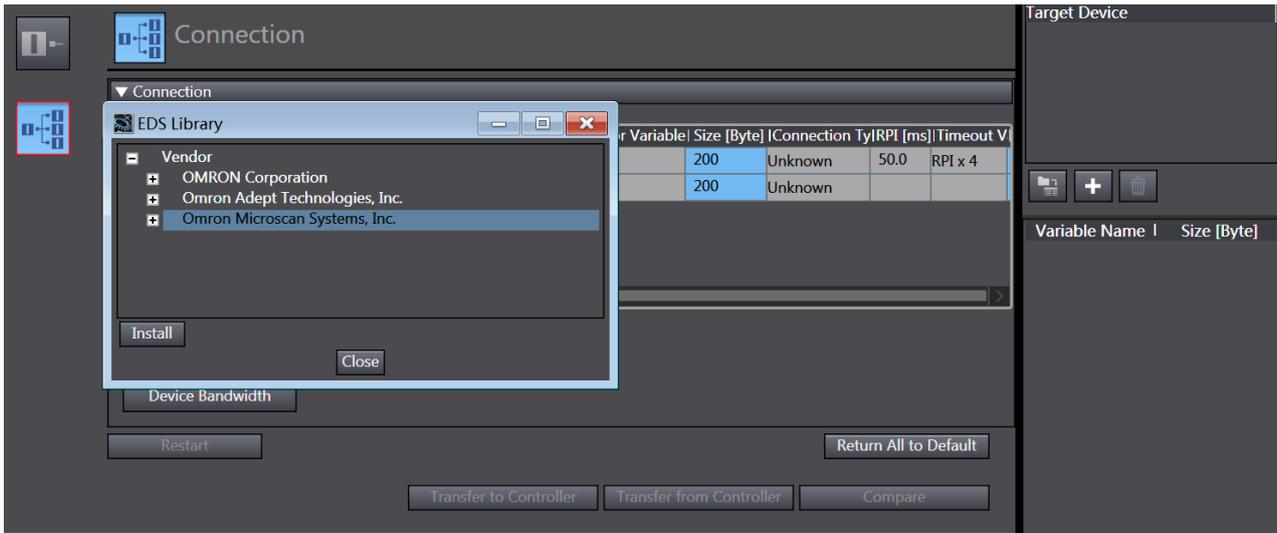


Figure 1-13 Install EDS File

8. Click **+** in the **Target Device** area to add devices, enter **Node address**, **Model name**, and **Revision** according to actual demands, and click **Add**.
9. Click **+** in **Built-in EtherNet/IP Port Settings**, add created target devices, select **Consume Data From/Produce Data To** as **Connection I/O Type**, and set size and originator variable that are global variable created before.

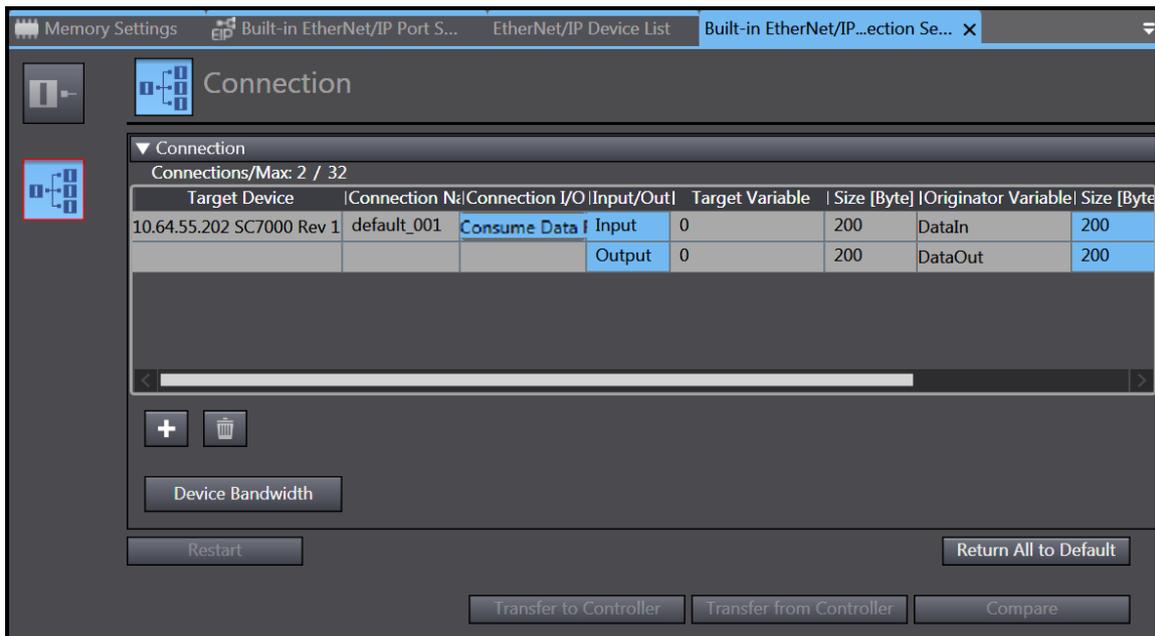


Figure 1-14 Create Target Device

10. Go to **Tools** → **Export Global Variables** → **Network Configurator...**, and export the global variables.

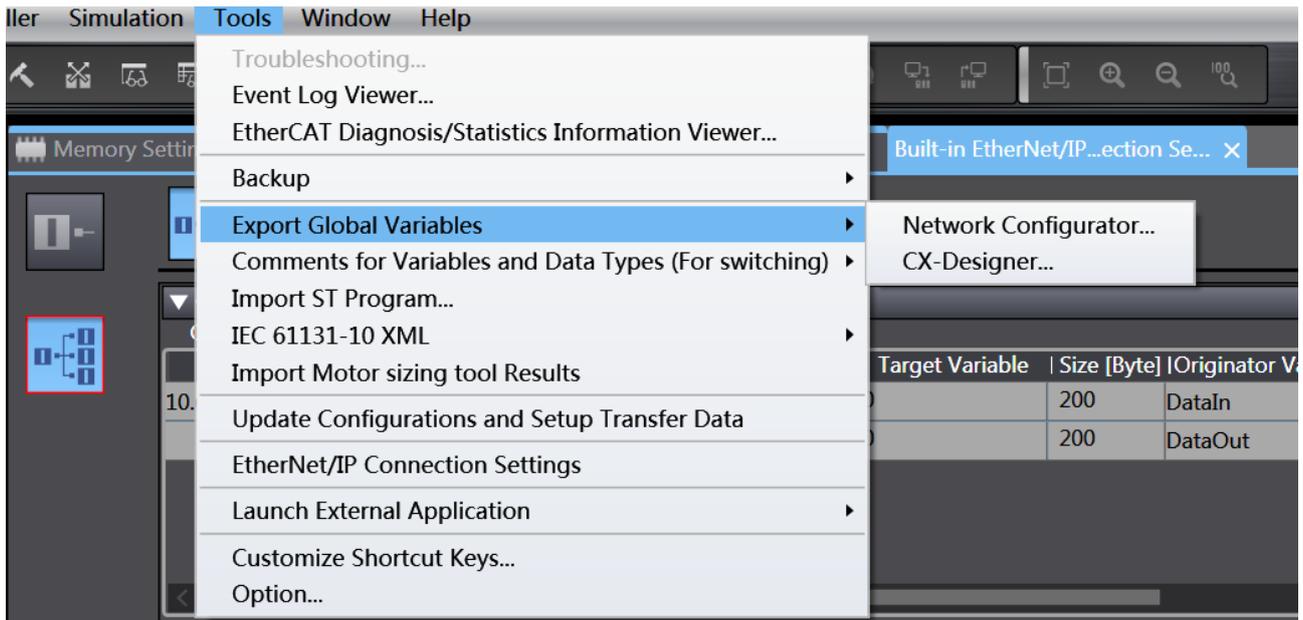


Figure 1-15 Export Global Variable File

11. Save and compile the project, and synchronize it to the PLC controller.

1.4.2 Network Configurator Settings

After the project is created, you need to set it via the network configurator.

Steps

1. Run Network Configurator, click **EDS File** and select **Install...** to import the device's EDS file.

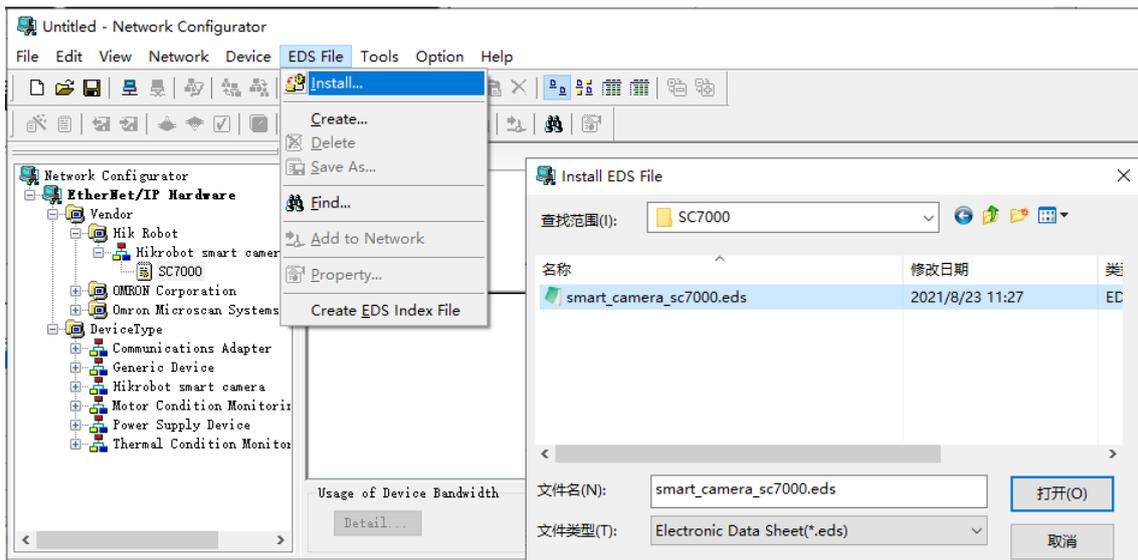


Figure 1-16 Import EDS File

2. Drag in the corresponding PLC controller and the smart camera.

3. Right click the smart camera to modify its IP address.

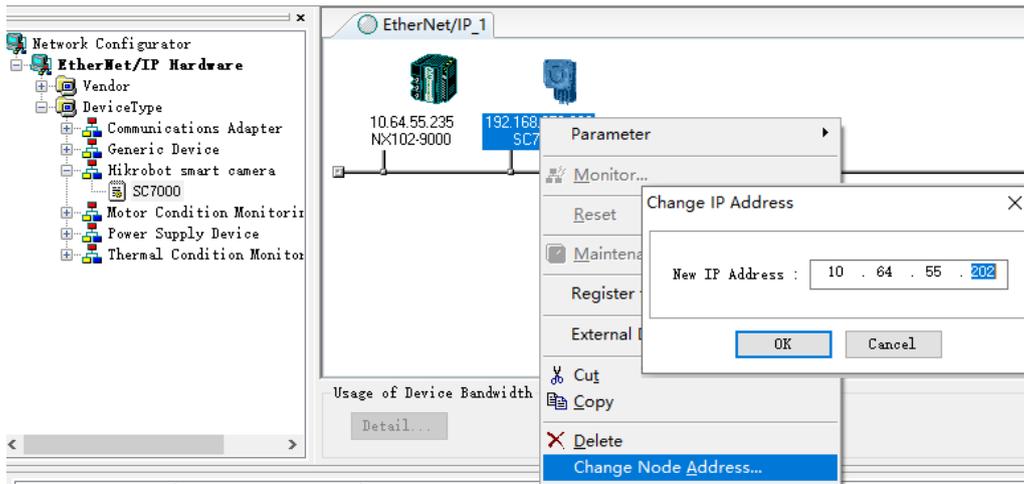


Figure 1-17 Modify IP Address

4. Double click the PLC, go to **Tag Sets** → **To/From File** → **Import from File**, and import the global variables created in the Sysmac Studio.

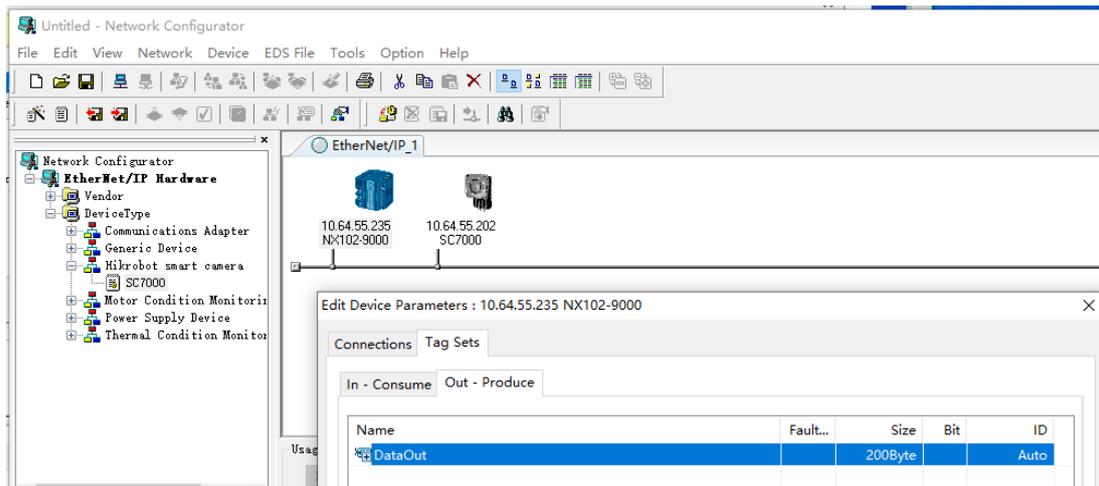


Figure 1-18 Import Global Variables

5. Establish the connection between the smart camera's input/output and the global variables of PLC controller. **Data In** connects to **Input_13**, and **Data Out** connects to **Output_22**, and set **Packet Interval** as 50 ms.

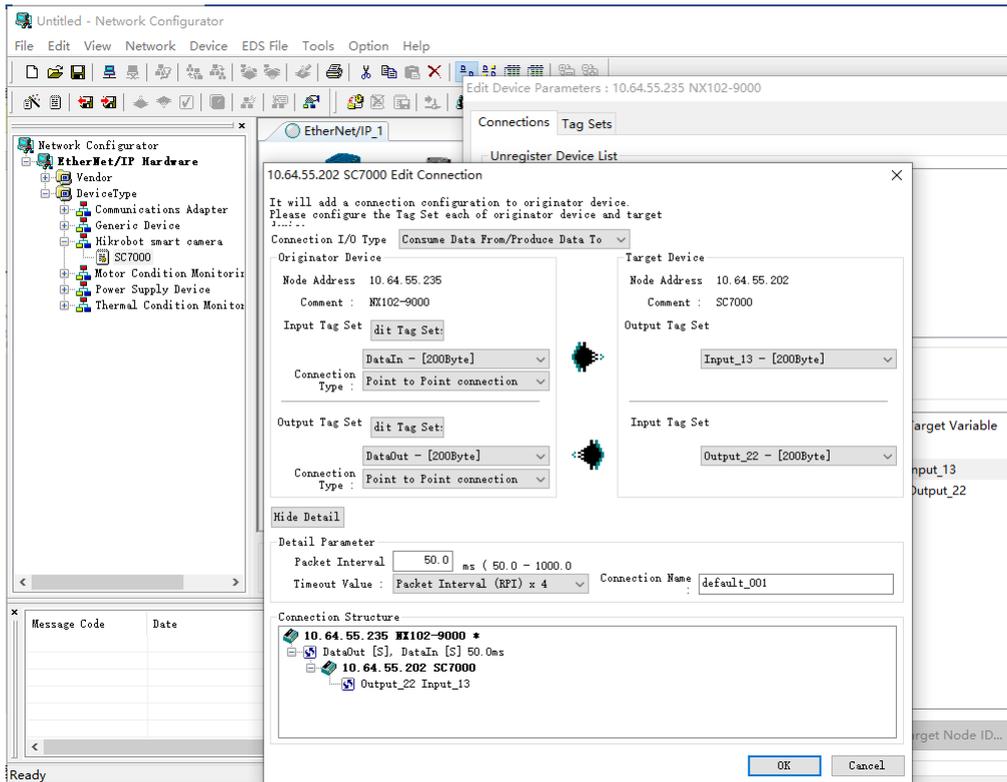


Figure 1-19 Establish Connection

6. Select interface for PLC controller and download configuration to the PLC.

Note

After the connection between the PLC and the smart camera is finished, the PLC's red NET ERR indicator will unlit.

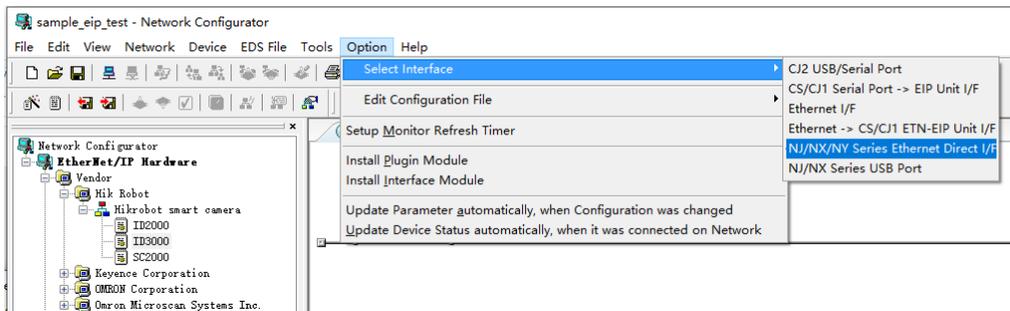


Figure 1-20 Select PLC's Communication Interface Type

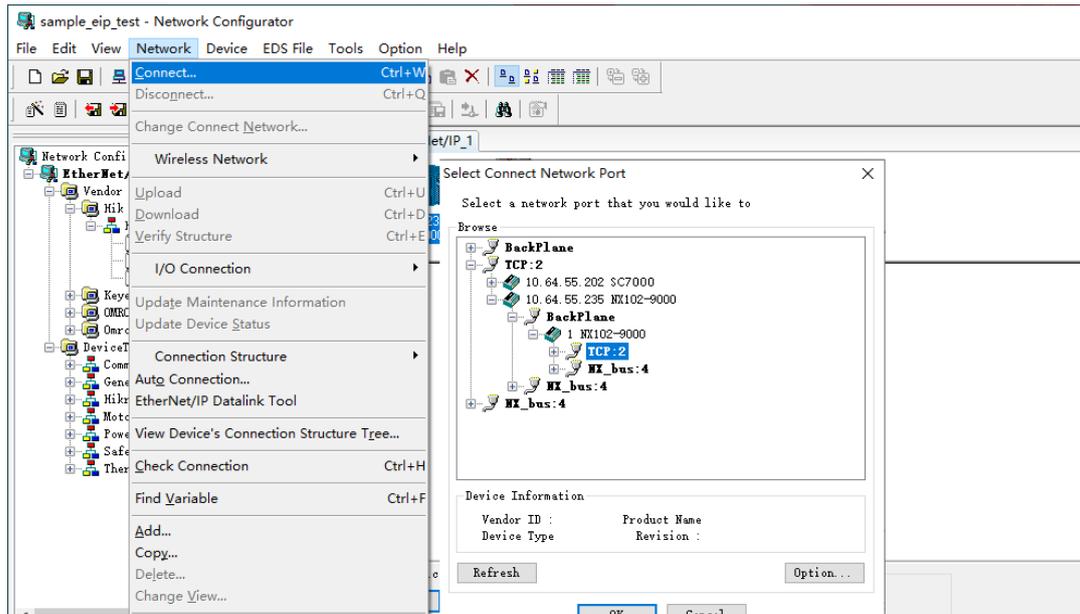


Figure 1-21 Connect

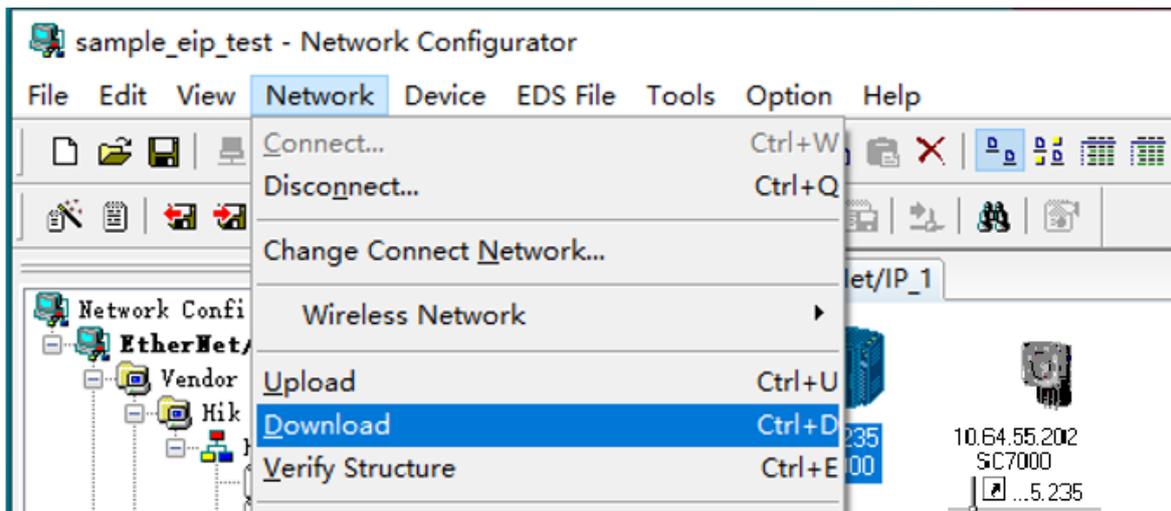


Figure 1-22 Download

1.5 Input and Output Mapping

1.5.1 Input Mapping

The definition of input mapping (smart camera > PLC controller) is shown below.

Table 1-1 Input Mapping Table

Byte	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
0-1	useful data length							
2					Decoding	Acquiring	Trigger Ack	Trigger Ready
3	General Fault						Results Timeout	Results Available
4-17	Reserved							
18-19	Result Length							
20	Result Data 0							
...	...							
199	Result Data 179							
...	Result Data ...							

Table 1-2 Input Mapping Table Description

Name	Description
Trigger Ready	The device is ready to receive new trigger signal. When the Trigger Enable is set and the device is ready to receive next trigger signal, the Trigger Ready is set.
Trigger Ack	The device has already received the trigger signal.
Acquiring	The device is acquiring images.
Decoding	The device is recognizing decodes on images.
Reserved	Reserved.
Results Available	The device outputs new results. When the PLC is set to Result Ack, the Results Available will be cleared.

Results Timeout	Results are not gotten when the time is out, and internal timeout is 6 sec. When the PLC is set to Result Ack, the Results Timeout will be cleared.
General Fault	The device's internal fault, and you can clear this signal via Clear Error.
Result Length	The invalid data length in the result area.
Result Data	<p>The reading result of the device, and the reading result is in ASCII format.</p> <ul style="list-style-type: none"> • When the result data length is less than the configured result module, the remaining data is filled with 0. • When the result data length is greater than the configured result module, the extra data is cut off.

1.5.2 Output Mapping

The definition of output mapping (PLC controller > smart camera) is shown below.

Table 1-3 Output Mapping Table

Byte	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
0						Results Ack	Trigger	Trigger Enable
1	Clear Error							Execute Command
2-17	Reserved							
18-19	User Data Length							
20	User Data 0							
...	...							
199	User Data 179							
...	User Data ...							

Table 1-4 Output Mapping Table Description

Name	Description
Trigger Enable	The PLC controls the device's trigger enable function via this bit.
Trigger	When following conditions are met, the PLC sets this bit to trigger the device to acquire an image and run the algorithm once. <ul style="list-style-type: none">• Trigger Enable is set.• The device is not currently acquiring images and running algorithms.• Trigger Ready is set.
Results Ack	After the PLC reads the trigger result, it sets this bit to notify the device, and the device clears Results Available and Results Timeout after confirmation.
Execute Command	Execute the command specified in the User Data area once on the rising edge.
Clear Error	Clear error status.
Reserved	Reserved.
User Data Length	The User Data area contains the length of valid data.
User Data	Data or commands sent to the device.

1.6 Trigger Test

Communication Sequence Diagram

The communication sequence diagram of PLC and the device is shown below.

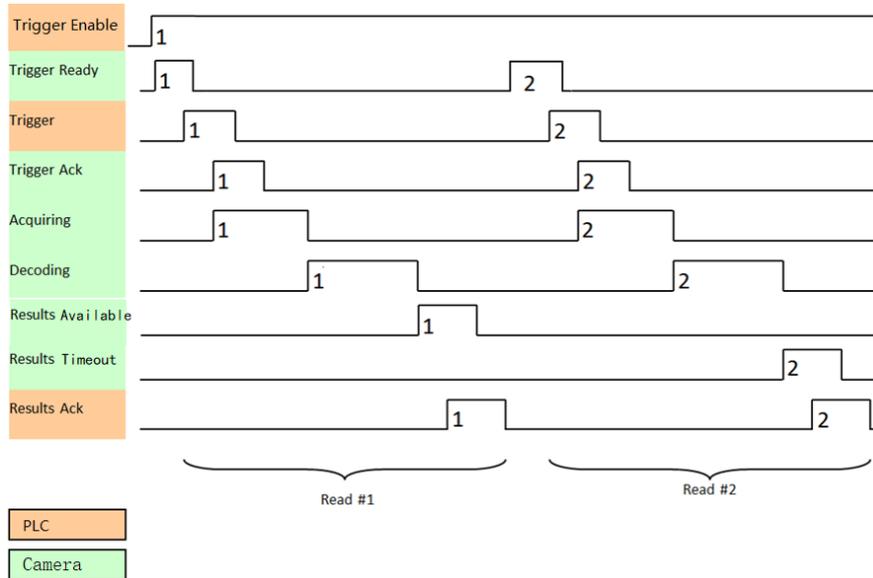


Figure 1-23 Communication Sequence Diagram

The logic of the above sequence diagram is shown below:

1. PLC sets Trigger Enable signal DataOut[0].0. After the device is ready, set Trigger Ready signal DataIn[2].0.
2. After detecting the device's Trigger Ready signal, the PLC sends Trigger signal DataOut[0].1 and controls the device to operate once.
3. The device starts to acquire images and runs the algorithm after receiving Trigger signal.
 - If the result of the algorithm tool is outputted correctly, set the Results Available signal DataIn[3].0 and put contents of the configuration result into the address starting from DataIn [20].
 - If the result output times out, set the Results Timeout signal DataIn[3].1 and clear the start address of DataIn [20].

Note

For tools like character recognition, the result will be outputted after the character is recognized. If the character is not recognized, the result will not be outputted, and the result output will time out at this time. If you need to return results quickly, you can use the exception output function.

4. After detecting Results Available, the PLC starts to read results from DataIn [20].
5. After reading results is finished, set Results Ack signal DataOut[0].2, and notify the device.

6. After the trigger flow is finished, PLC sends Trigger signal again and next round of trigger flow begins.

Create Variables

Create variables in accordance with the device's input/output mapping area.

Ladder Diagram

The PLC triggers the device to run the project, and related ladder diagrams are shown below.

- Enable Trigger

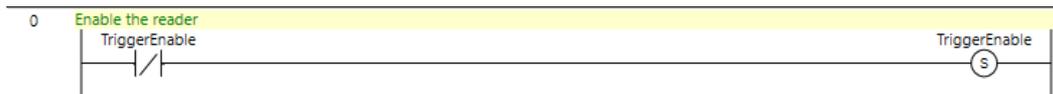


Figure 1-24 Enable Trigger

- Send Trigger Signal

The trigger signal can be generated by adding an edge signal before Trigger Ready according to actual demands. For example, when the PLC detects the photoelectric input signal, it sends a trigger signal once.

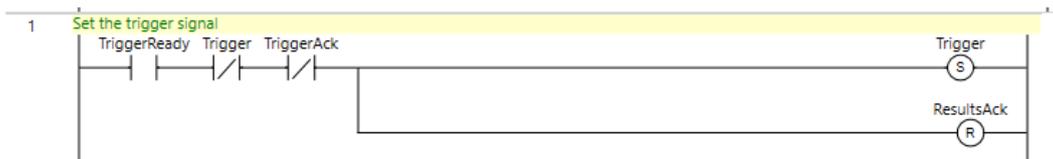


Figure 1-25 Send Trigger Signal

- Clear Trigger Signal

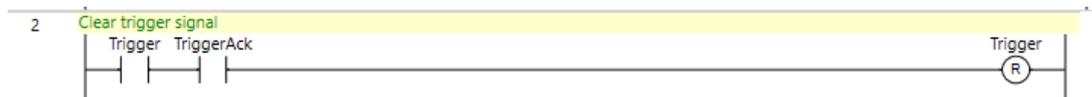


Figure 1-26 Clear Trigger Signal

- Get Device Results

When the device feedbacks the Results Available signal, it means that result data has been updated, and the PLC can read result data to the user's storage area. After the reading is complete, set the Results Ack signal to acknowledge that the device has completed reading

result data.

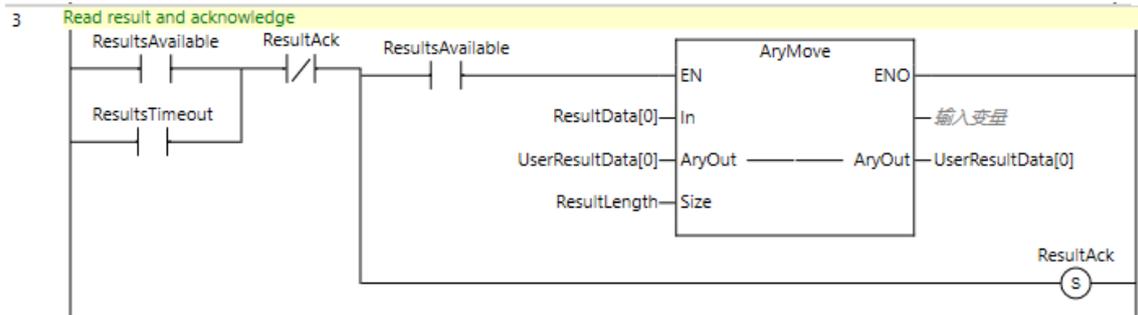


Figure 1-27 Get Device Results

View Results

Open the variable monitoring window of the Sysmac Studio, and add the created DataIn input variable and DataOut output variable. The online value of the variable is the internal input/output value of the device.

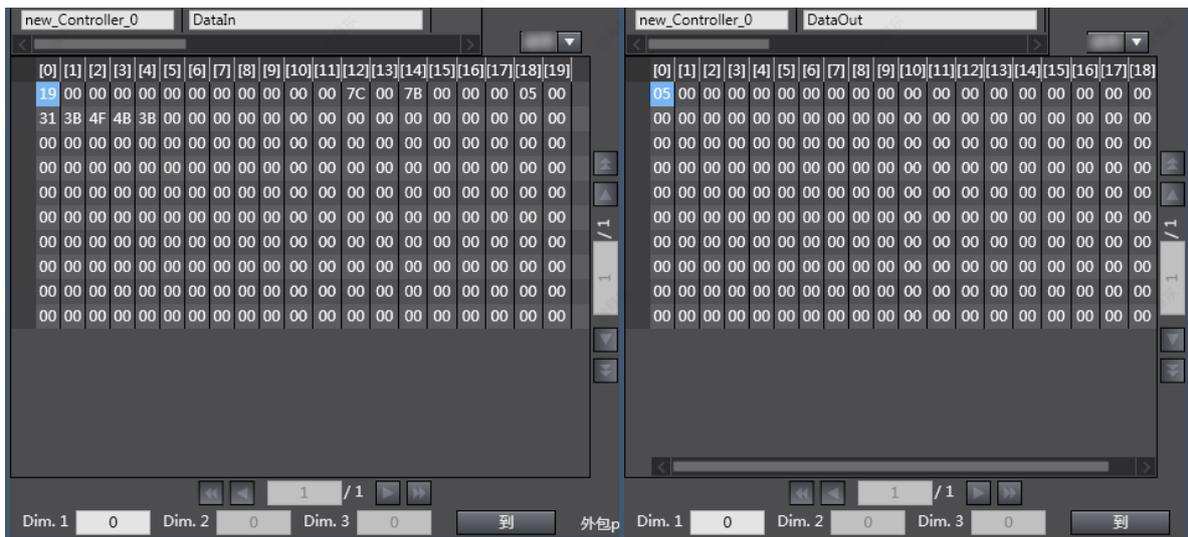


Figure 1-28 View Results

1.7 Project Switching Test

The related ladder diagram of project switching test is shown below.

- Send Switching Command

Write command character to User Data area, and write command character length to User

Data Length.

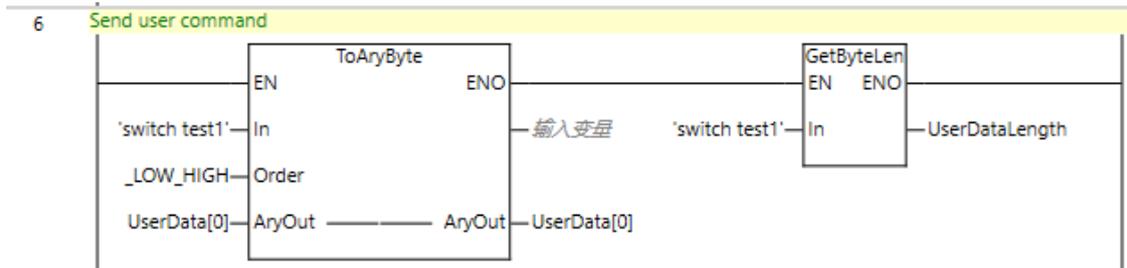


Figure 1-29 Send Switching Command

- Execute Switching Command

Write a rising edge to the Executive Command bit in the control area to execute a switching command for the project.

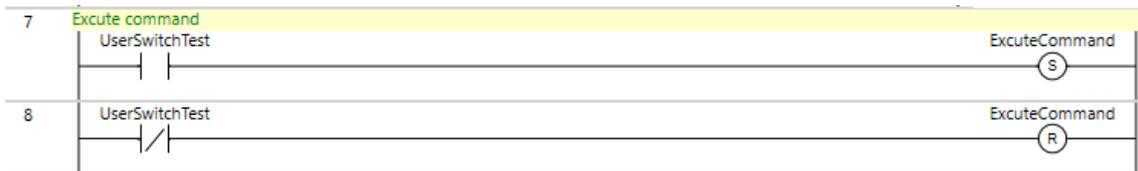


Figure 1-30 Execute Switching Command

Chapter 2 Profinet

2.1 Introduction

Profinet is an open industrial EtherNet standard launched by PI for automation. Profinet uses TCP/IP and IT standards. It is a real-time EtherNet that not only supports standard TCP/IP, but also provides time precision that standard EtherNet cannot achieve.

Note

This chapter takes Siemens S7 series PLC as an example to explain how to communicate with smart cameras via Profinet communication protocol. For other devices, refer to the user manual provided by the manufacturer and make related settings by reading contents of this chapter.

2.2 Hardware Wiring

The wiring of Siemens S7 series PLC and the smart camera is shown below.

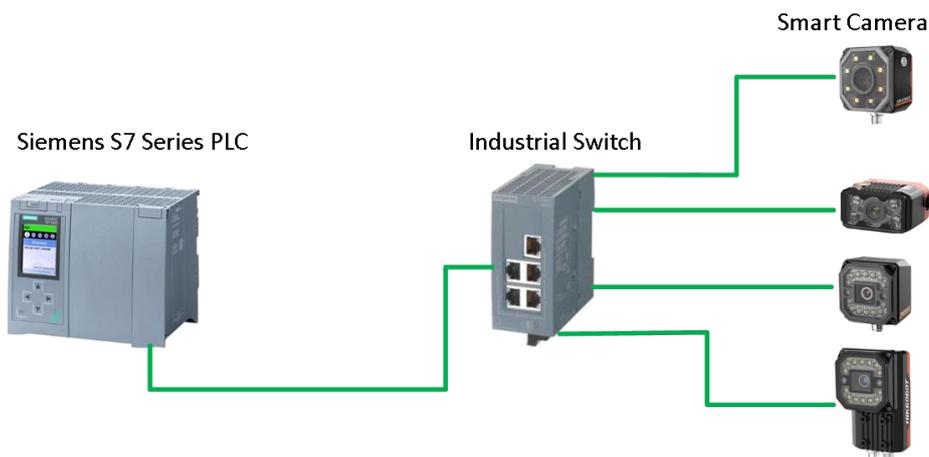


Figure 2-1 Hardware Wiring

2.3 Smart Camera Settings

Before using the smart camera's Profinet function to communicate with other devices, you need to set the smart camera first via the SCMVS client software.

Before you start:

- Make sure that the PC has installed the SCMVS client software.

- Check the device's firmware version.

Note

Regarding SC2000Pro series smart camera, you need to operate it via the web, and operations are similar to those via the SCMVS client software.

Steps

1. Log in the device via the SCMVS client software.
2. Click **Communication** on the menu bar.
3. Click **+** to add Profinet communication.
4. Switch on **○** to enable Profinet communication.
5. Enter **Device Name** according to actual demands.

Note

Make sure that the device name is unique.

6. Set **Result Module** according to actual demands.

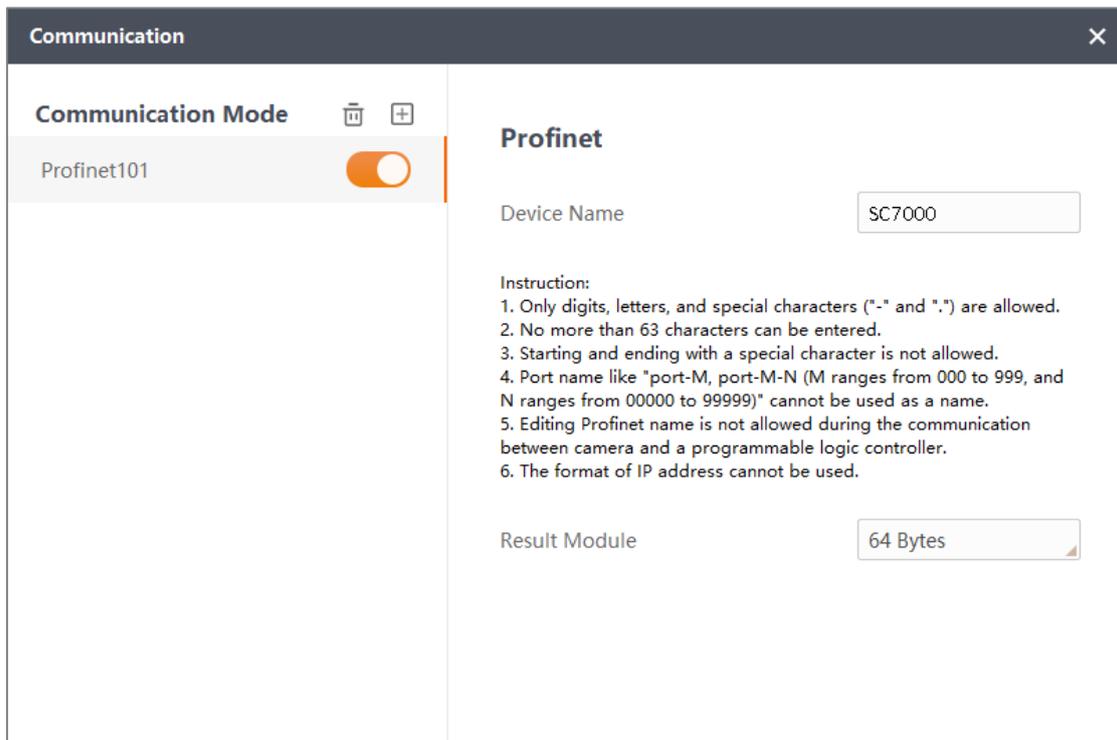


Figure 2-2 Set Communication Parameters

7. Create a project and set parameters for cameras, base images, and vision tools according to actual demands.

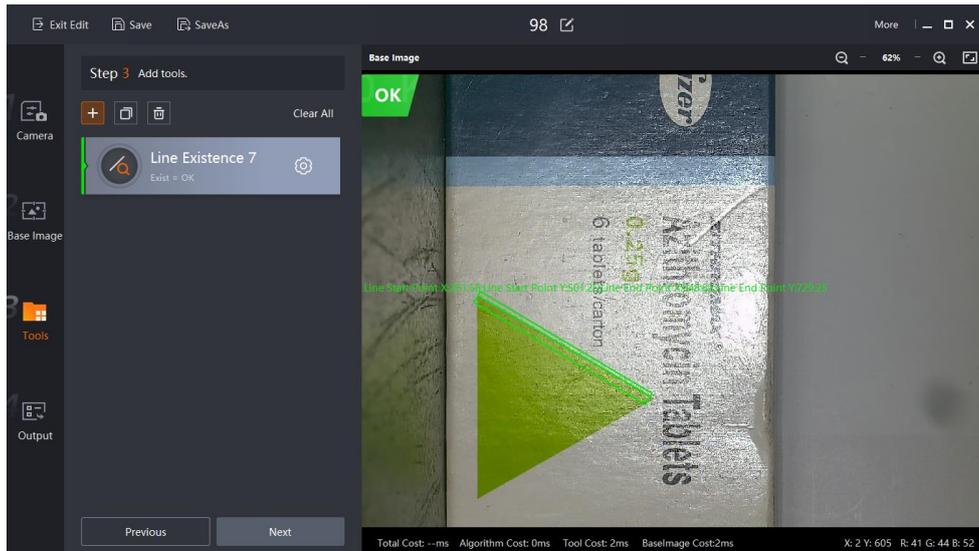


Figure 2-3 Line Existence

Note

Here we take the tool of line existence as an example.

8. Go to **Output** → **Tool Results** → **Edit** to set the outputted content.

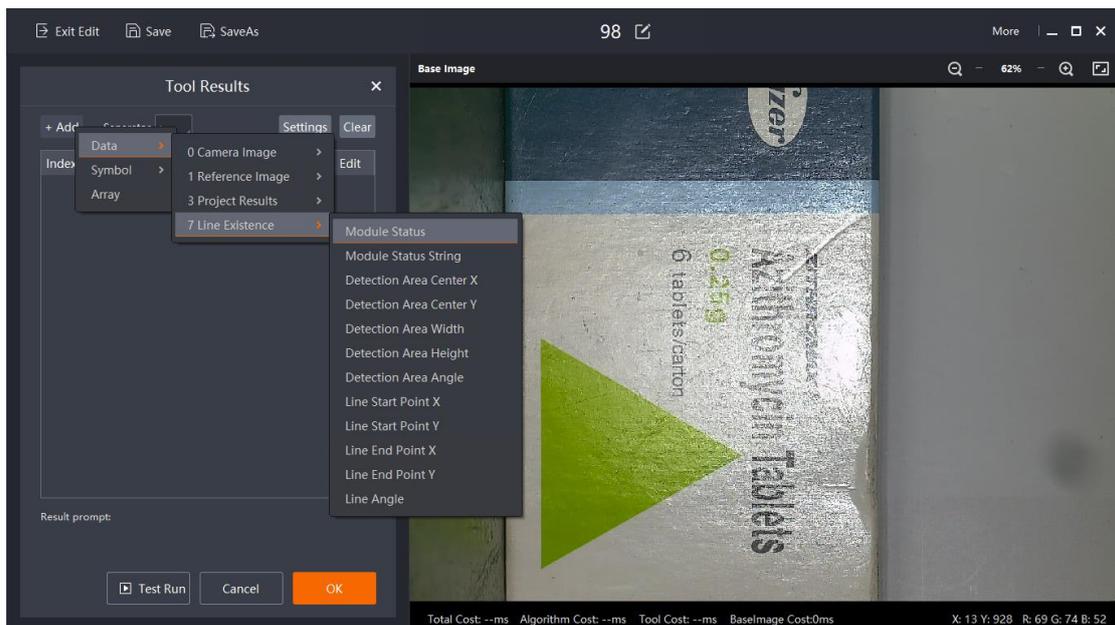


Figure 2-4 Set Tool Results

Note

Here we take the module status and module status string of line existence as an example to introduce tool results.

9. Save the project.
10. If you want to switch projects via communications, go to project management, select **Trigger Communication** as **Mode**, enter **Communication String**, and **Switch Return String** according to actual demands.

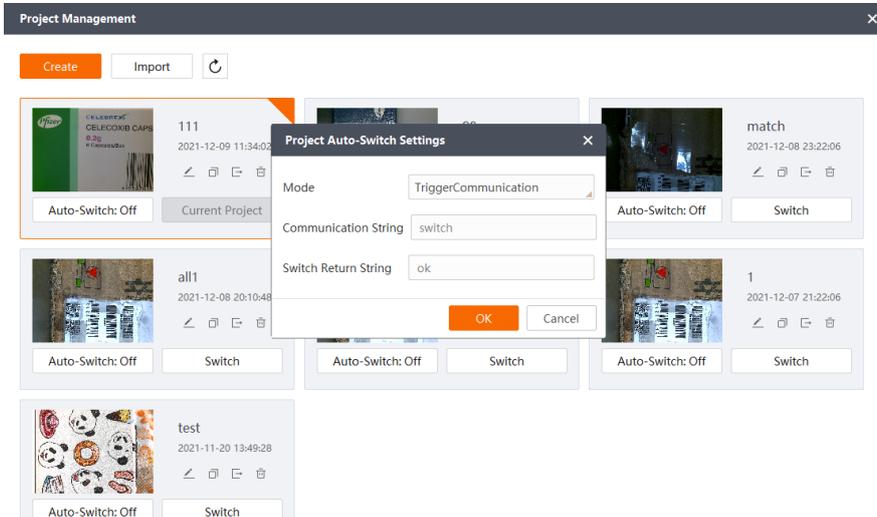


Figure 2-5 Project Switch Settings

2.4 PLC Settings

This section takes Siemens S7 series PLC as an example to explain how to set PLC.

Steps

1. Run TIA Portal, click **Create new project**, enter parameters, and click **Create**.

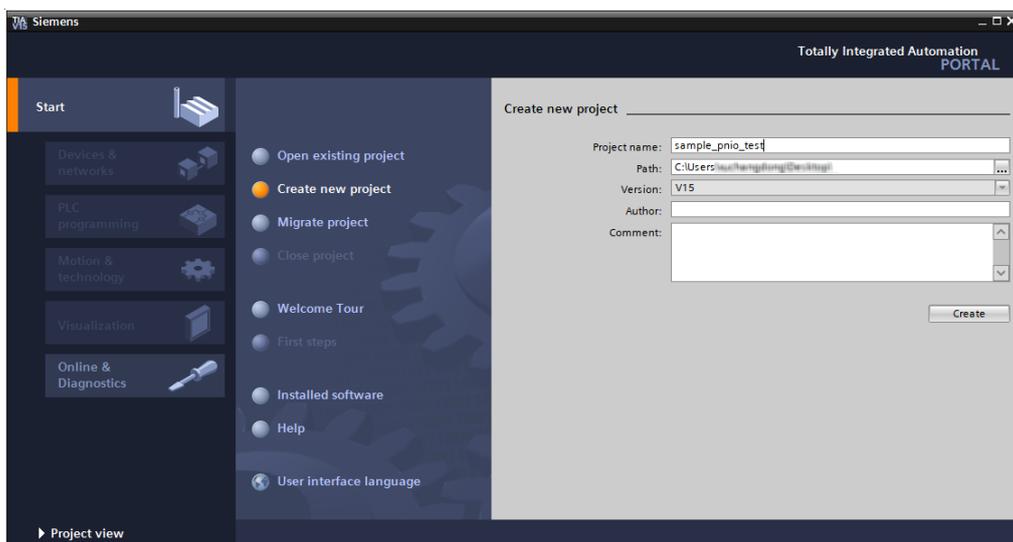


Figure 2-6 Create New Project

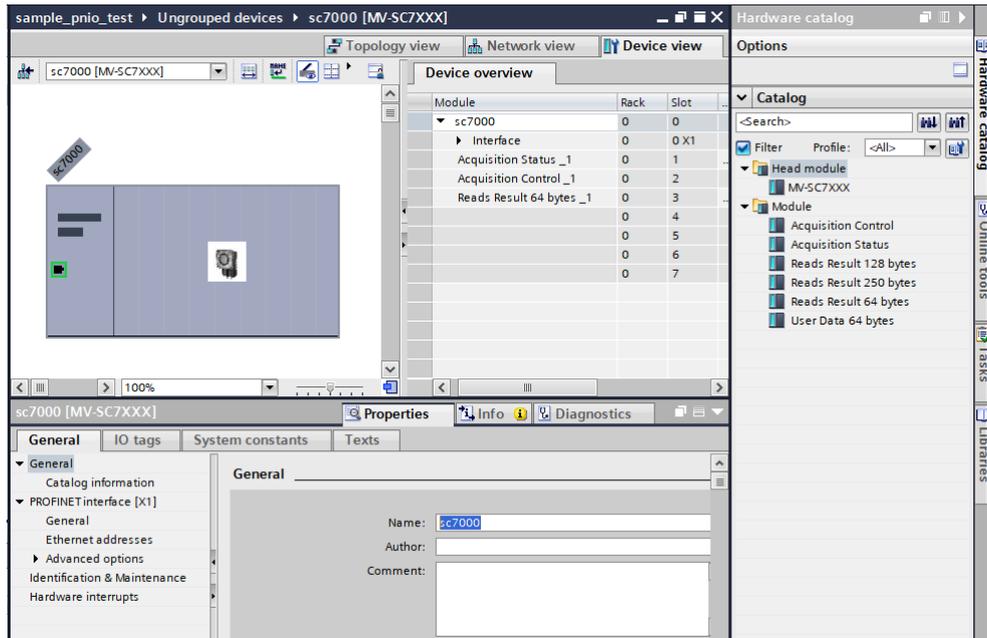


Figure 2-9 Set Device Name

Note

The name configured here should be the same with that in the SCMVS. Otherwise, communication error may occur.

6. Set the subnet that the interface is networked with, and the IP address.

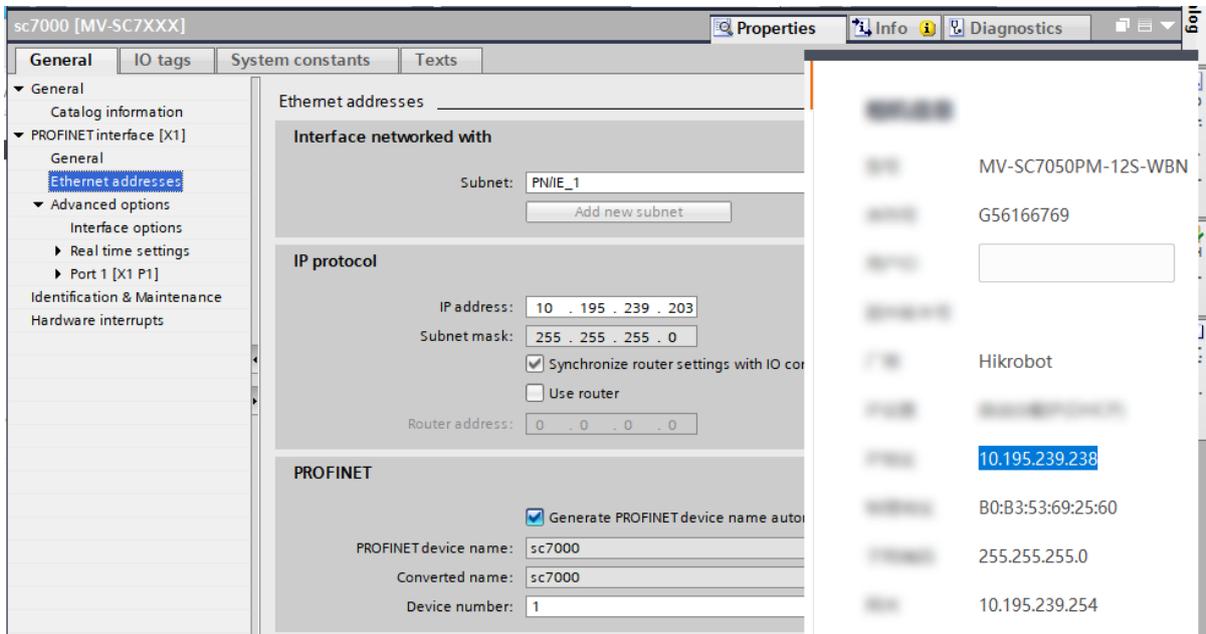


Figure 2-10 Set IP Address

Note

The device's IP address should be different with that in the SCMVS. Otherwise, communication error may occur.

7. Set the device's IO cycle and other parameters.

Note

Increase the value of **Accepted update cycles without IO data** in Watchdog time if the device goes offline.

8. Recompile and load the PLC program. When the PLC is connected to the device, the indicator of the PLC and device will be solid green.

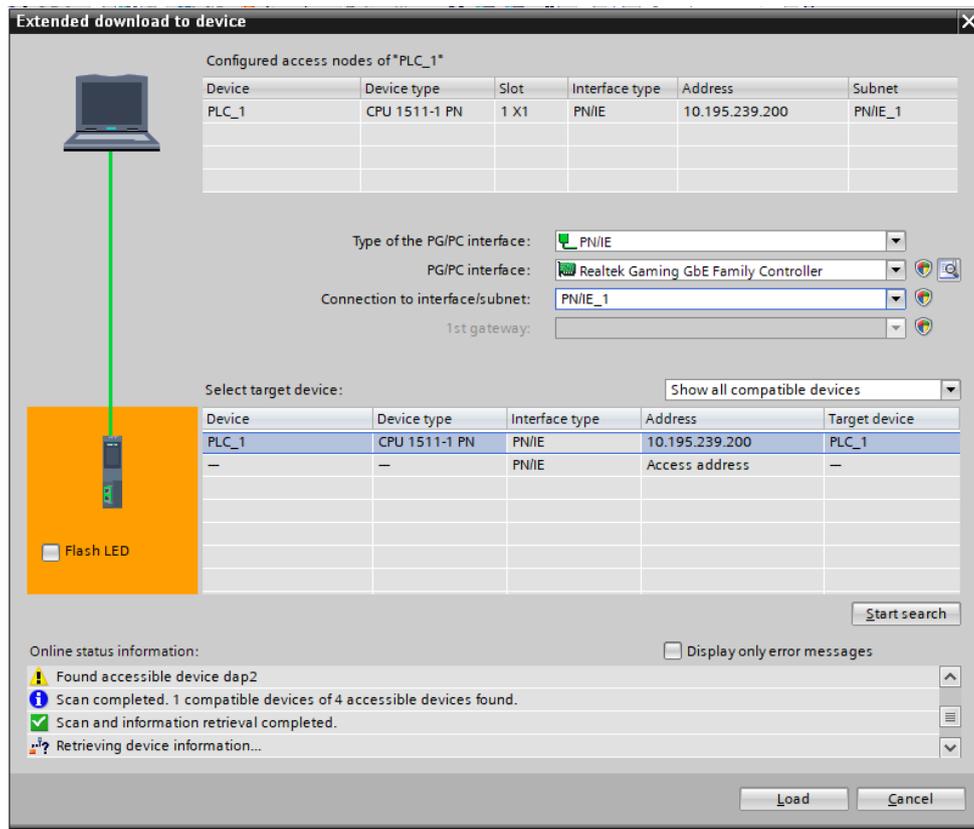


Figure 2-11 Load PLC Program

2.5 Device Module Definition

The device has six configurable modules as shown below:

- Acquisition Status Module
- Acquisition Control Module
- Reads Result 64 Bytes Module
- User Data 64 Bytes Module

- Reads Result 128 Bytes Module
- Reads Result 250 Bytes Module

Acquisition Status Module

The acquisition status module sends the information about the current status of the device from the device to the PLC.

- Slot No.: 1
- Size: 1 byte

Table 2-1 Acquisition Status Module Description

Bit	Name	Description
0	Trigger Ready	The device is ready to receive new trigger signal. When the Trigger Enable is set and the device is ready to receive next trigger signal, the Trigger Ready is set.
1	Trigger Ack	The device has already received the trigger signal.
2	Acquiring	The device is acquiring images.
3	Decoding	The device is recognizing decodes on images.
4	Results Available	The device outputs new results. When the PLC is set to Result Ack, the Results Available will be cleared.
5	Results Timeout	Results are not gotten when the time is out, and internal timeout is 6 sec. When the PLC is set to Result Ack, the Results Timeout will be cleared.
6	Reserved	Reserved.
7	General Fault	An internal error is generated.

Acquisition Control Module

The acquisition control module sends images from the PLC to the device.

- Slot No.: 2
- Size: 1 byte

Table 2-2 Acquisition Control Module Description

Bit	Name	Description
0	Trigger Enable	The PLC controls the device's trigger enable function via this bit.
1	Trigger	When following conditions are met, the PLC sets this bit to trigger the device to acquire an image and run the algorithm once. <ul style="list-style-type: none"> • Trigger Enable is set. • The device is not currently acquiring images and running algorithms. • Trigger Ready is set.
2	Results Ack	After the PLC reads the trigger result, it sets this bit to notify the device, and the device clears Results Available and Results Timeout after confirmation.
3	Execute Command	Execute the command specified in the User Data area once on the rising edge.
4-6	Reserved	Reserved.
7	Clear Error	Clear error status.

Reads Result 64 Bytes Module

The reads result 64 bytes module reads and sends the device's processing result from the device to the PLC.

- Slot No.: 3, configurable
- Size: 65 bytes

Table 2-3 Reads Result 64 Bytes Module Description

Bit	Name	Description
0	Result Length	Result Data contains the actual length of data.
1...64	Result Data	The result output by the device, and its length is 64 bytes. <ul style="list-style-type: none"> • When the actual result is smaller than 64 bytes, the spare bytes will be filled with 0. • When the actual result is greater than 64 bytes, the extra bytes will be cut off.

User Data 64 Bytes Module

The user data 64 bytes module sends user data from the PLC to the device.

- Slot No.: 4, configurable
- Size: 65 bytes

Table 2-4 User Data 64 Bytes Module Description

Bit	Name	Description
0	User Data Length	User Data contains the actual length of data.
1...64	User Data	The data or commands sent to the device.

Reads Result 128 Bytes Module

The reads result 128 bytes module is the result data and is sent to the PLC by the device.

- Slot No.: 3, configurable
- Size: 129 bytes

Table 2-5 Reads Result 128 Bytes Module

Bit	Name	Description
0	Result Length	Result Data contains the actual length of data.
1...128	Result Data	The result output by the device, and its length is 128 bytes. <ul style="list-style-type: none"> • When the actual result is smaller than 128 bytes, the spare bytes will be filled with 0. • When the actual result is greater than 128 bytes, the extra bytes will be cut off.

Reads Result 250 Bytes Module

The reads result 250 bytes module is the result data and is sent to the PLC by the device.

- Slot No.: 3, 5, 6, 7, configurable
- Size: 251 bytes

Table 2-6 Reads Result 250 Bytes Module

Bit	Name	Description
0	Result Length	Result Data contains the actual length of data.
1...250	Result Data	The result output by the device, and its length is 250 bytes. <ul style="list-style-type: none"> When the actual result is smaller than 250 bytes, the spare bytes will be filled with 0. When the actual result is greater than 250 bytes, the extra bytes will be cut off.

2.6 Trigger Test

Communication Sequence Diagram

The communication sequence diagram of PLC and the device is shown below.

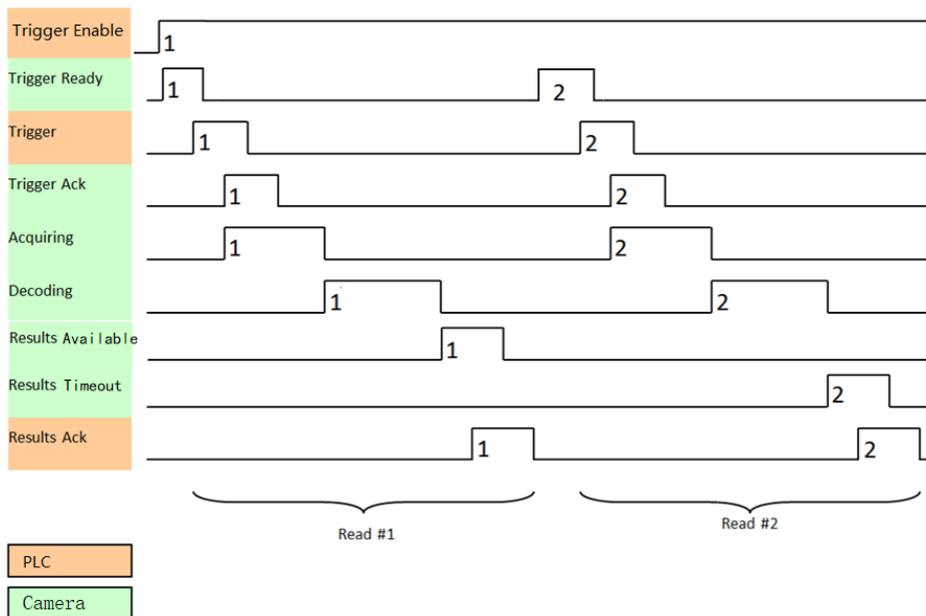


Figure 2-12 Communication Sequence Diagram

The logic of the above sequence diagram is shown below:

1. PLC sets Trigger Enable signal Q0.0. After the device is ready, set Trigger Ready signal I0.0.
2. After detecting the device's Trigger Ready signal, the PLC sends Trigger signal Q0.1 and controls the device to operate once.
3. The device starts to acquire images and runs the algorithm after receiving Trigger signal.
 - If the result of the algorithm tool is outputted correctly, set the Results Available signal I0.4 and put contents of the configuration result into the address starting from QB2.
 - If the result output times out, set the Results Timeout signal I0.5 and clear the start

address of QB2.

Note

For tools like character recognition, the result will be outputted after the character is recognized. If the character is not recognized, the result will not be outputted, and the result output will time out at this time. If you need to return results quickly, you can use the exception output function.

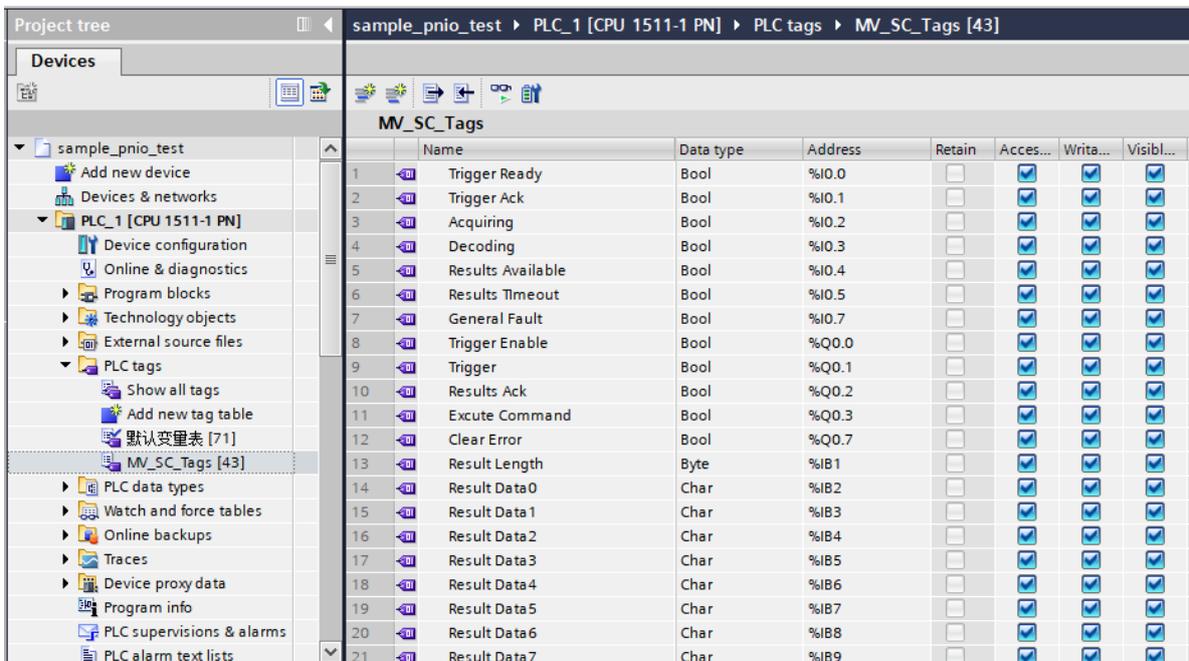
4. After detecting Results Available, the PLC starts to read results from QB2.
5. After reading results is finished, set Results Ack signal Q0.2, and notify the device.
6. After the trigger flow is finished, PLC sends Trigger signal again and next round of trigger flow begins.

Create Variables

Create a new PLC variable according to the address assigned to the device according to the device configuration, and you can view the current value of each module of the device online.

Overview of addresses								
Filter: <input checked="" type="checkbox"/> Inputs <input checked="" type="checkbox"/> Outputs <input type="checkbox"/> Address gaps <input checked="" type="checkbox"/> Slot								
	Type	Addr. fr...	Addr. to	Size	Module	Rack	Slot	Device name
	I	0	0	1 Bytes	Acquisition Status _1	0	1	sc7000 [MW-SC7XXX]
	O	0	0	1 Bytes	Acquisition Control _1	0	2	sc7000 [MW-SC7XXX]
	I	1	65	65 By...	Reads Result 64 bytes _1	0	3	sc7000 [MW-SC7XXX]

Figure 2-13 Overview of Address



	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...
1	Trigger Ready	Bool	%I0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	Trigger Ack	Bool	%I0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	Acquiring	Bool	%I0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	Decoding	Bool	%I0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	Results Available	Bool	%I0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	Results Timeout	Bool	%I0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	General Fault	Bool	%I0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	Trigger Enable	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	Trigger	Bool	%Q0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	Results Ack	Bool	%Q0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	Excute Command	Bool	%Q0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
12	Clear Error	Bool	%Q0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
13	Result Length	Byte	%IB1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
14	Result Data0	Char	%IB2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
15	Result Data1	Char	%IB3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
16	Result Data2	Char	%IB4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
17	Result Data3	Char	%IB5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
18	Result Data4	Char	%IB6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
19	Result Data5	Char	%IB7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
20	Result Data6	Char	%IB8	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
21	Result Data7	Char	%IB9	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figure 2-14 Create PLC Variable

Ladder Diagram

The PLC triggers the device to run the project, and related ladder diagrams are shown below.

- Enable Trigger



Figure 2-15 Enable Camera

- Send Trigger Signal

The trigger signal can be generated by adding an edge signal before Trigger Ready according to actual demands. For example, when the PLC detects the photoelectric input signal, it sends a trigger signal once. The figure below describes that a trigger signal is outputted every second via the internal system clock.

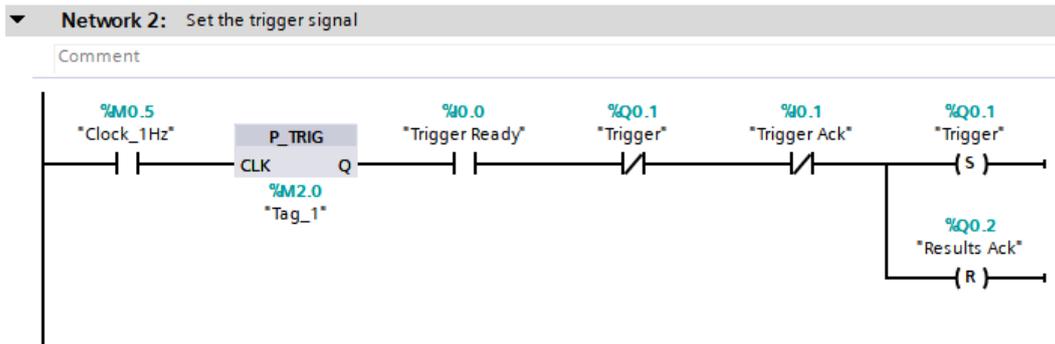


Figure 2-16 Send Trigger Signal

- Clear Trigger Signal

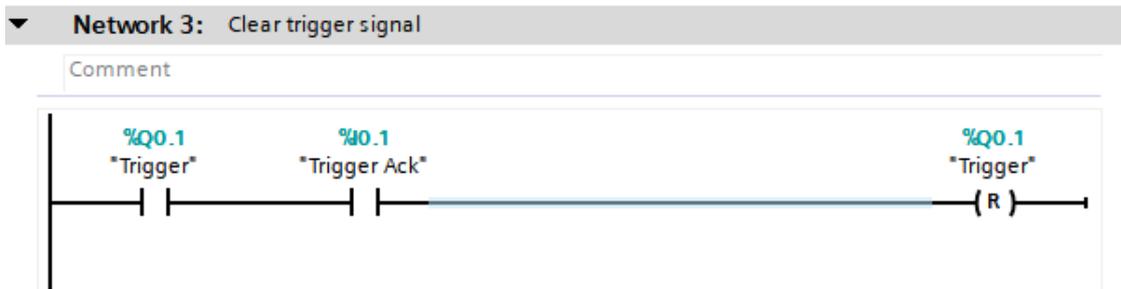


Figure 2-17 Clear Trigger Signal

- Get Device Results

Create a DB data block to store the reading result, add the result variable as shown below. The first byte is the result length, and the result data is stored starting from the second byte.

Name	Data type	Offset	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
Static		0.0	16#0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Result Length	Byte	0.0	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Result Data0	Char	1.0	**	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Result Data1	Char	2.0	**	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Result Data2	Char	3.0	**	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Result Data3	Char	4.0	**	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Result Data4	Char	5.0	**	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Result Data5	Char	6.0	**	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Result Data6	Char	7.0	**	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Result Data7	Char	8.0	**	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Result Data8	Char	9.0	**	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Result Data9	Char	10.0	**	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Result Data10	Char	11.0	**	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Result Data11	Char	12.0	**	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Result Data12	Char	13.0	**	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Result Data13	Char	14.0	**	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Result Data14	Char	15.0	**	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Result Data15	Char	16.0	**	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Result Data16	Char	17.0	**	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Figure 2-18 Add Result Data Block

When the device feedbacks the Results Available or Timeout signal, it means that the result data has been updated, and the PLC reads the result data to the DB data block. After the reading is finished, set the Results Ack signal to acknowledge that the device has finished reading the result data.

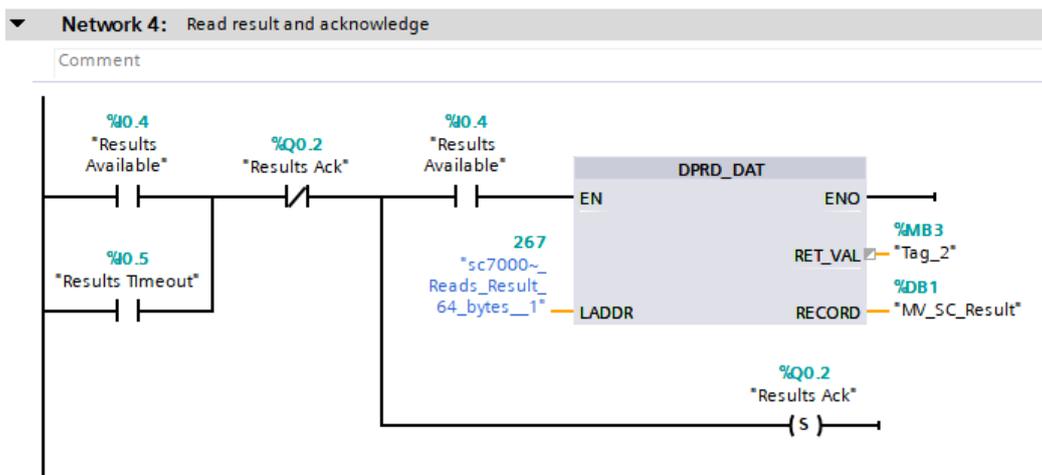


Figure 2-19 Get Results

View Results

Through the monitoring and option tab, you can monitor the status of the variables of each module of the device and view the result output. You can also add trace records of the

variables via the Trace tab to view the control sequence.

	Name	Data type	Offset	Start value	Monitor value	Retain	Accessible f...	Writa...
1	Static					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	Result Length	Byte	0.0	16#0	16#05	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	Result Data0	Char	1.0	''	'1'	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	Result Data1	Char	2.0	''	'.'	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	Result Data2	Char	3.0	''	'0'	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	Result Data3	Char	4.0	''	'k'	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	Result Data4	Char	5.0	''	'.'	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	Result Data5	Char	6.0	''	'\$00'	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	Result Data6	Char	7.0	''	'\$00'	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	Result Data7	Char	8.0	''	'\$00'	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	Result Data8	Char	9.0	''	'\$00'	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
12	Result Data9	Char	10.0	''	'\$00'	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
13	Result Data10	Char	11.0	''	'\$00'	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
14	Result Data11	Char	12.0	''	'\$00'	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
15	Result Data12	Char	13.0	''	'\$00'	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
16	Result Data13	Char	14.0	''	'\$00'	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figure 2-20 View Results

2.7 Project Switching Test

Hardware Configuration

Add User Data 64 bytes module to the device, and recompile the hardware configuration and download it to the PLC.

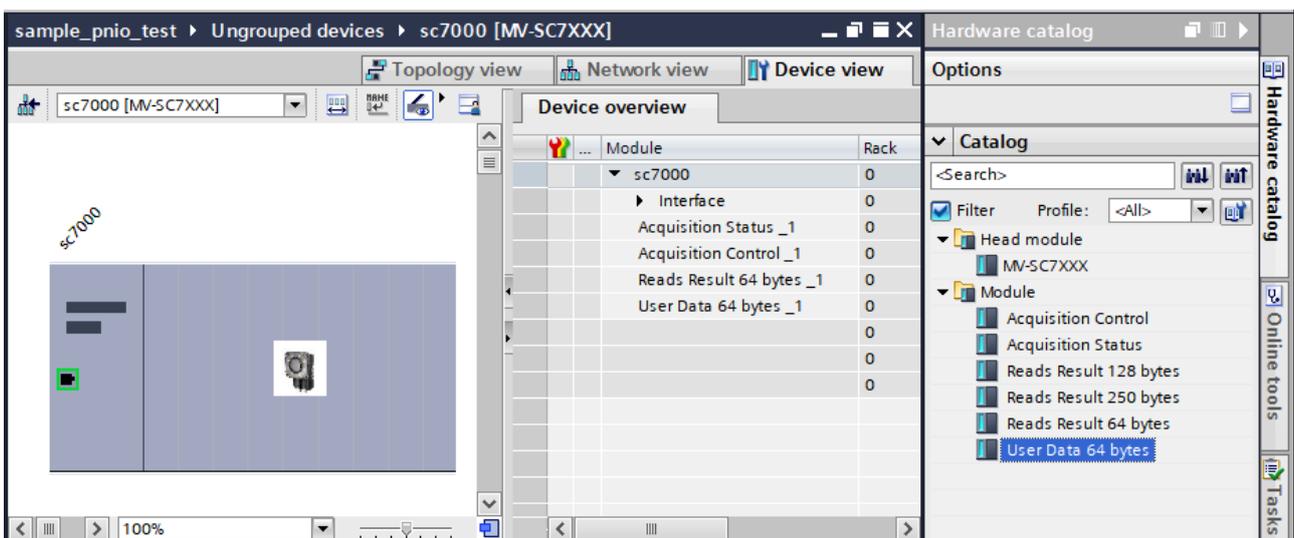


Figure 2-21 Add User Data Module

Ladder Diagram

Create a DB data block to store the reading result, add the command variable as shown below. The first byte is the command length, and command character is stored starting from the second byte.

Name	Data type	Offset	Start value	Retain	Accessible f...	Writa...	Visible in ...
Static				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Command Length	Byte	0.0	12	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Command Data0	Char	1.0	's'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Command Data1	Char	2.0	'w'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Command Data2	Char	3.0	'i'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Command Data3	Char	4.0	't'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Command Data4	Char	5.0	'c'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Command Data5	Char	6.0	'h'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Command Data6	Byte	7.0	''	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Command Data7	Char	8.0	't'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Command Data8	Char	9.0	'e'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Command Data9	Char	10.0	's'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Command Data10	Char	11.0	't'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Command Data11	Char	12.0	'i'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Command Data12	Char	13.0	''	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Command Data13	Char	14.0	''	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figure 2-22 Add Command Data Block

The related ladder diagram of project switching test is shown below.

- Send Switching Command

Write "command length + command character" to User Data 64 bytes module.

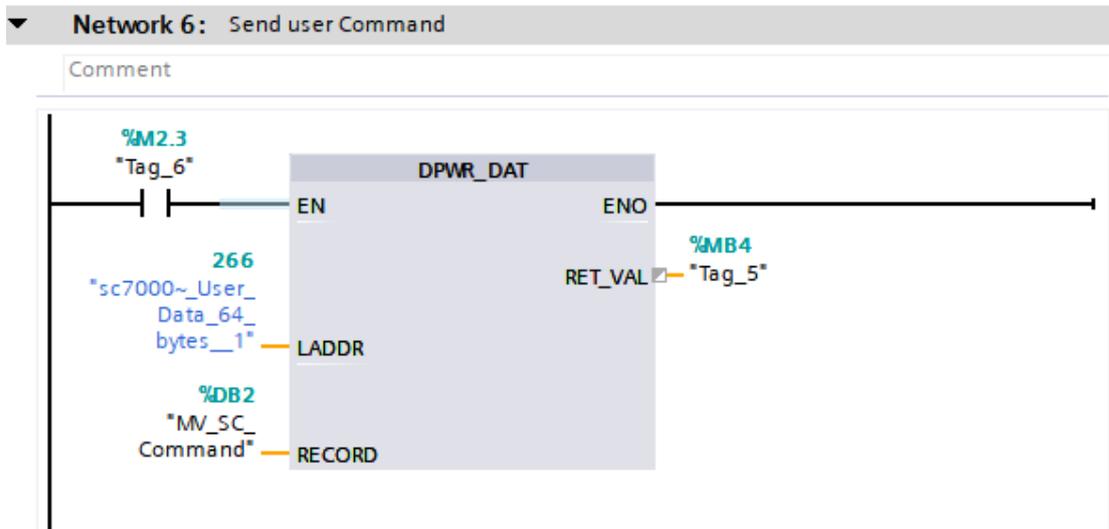


Figure 2-23 Send Switching Command

- Execute Switching Command

Write a rising edge to the Executive Command of the Acquisition Control Module to trigger

the execution of the command.

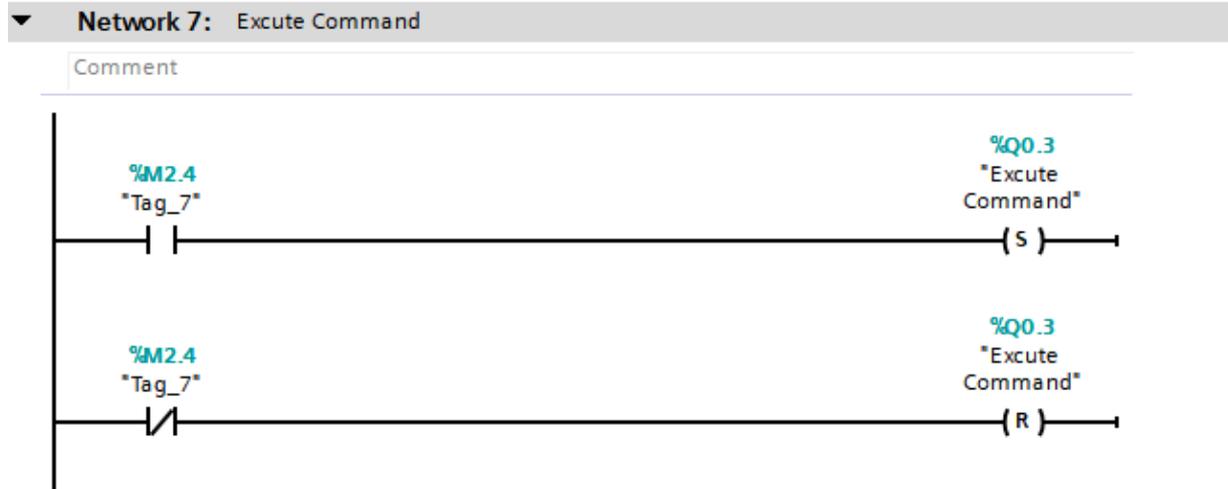


Figure 2-24 Execute Switching Command

Chapter 3 ModBus

3.1 Introduction

Modbus is a request/response protocol whose services are specified by function codes. Port 502 is reserved for ModBus communication. It uses standard EtherNet hardware and software to exchange I/O data and diagnostic information.

Note

This chapter takes Siemens S7 series PLC as an example to explain how to communicate with smart cameras via ModBus communication protocol. For other devices, refer to the user manual provided by the manufacturer and make related settings by reading contents of this chapter.

3.2 Hardware Wiring

The wiring of PLC controller, ModBus TCP test tool, and the smart camera is shown below.

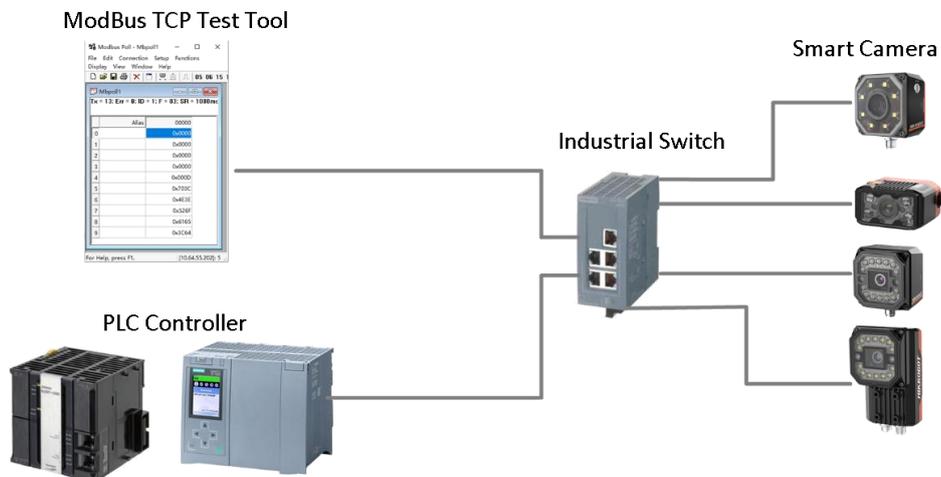


Figure 3-1 Hardware Wiring

3.3 Smart Camera Settings

Before using the smart camera's ModBus function to communicate with other devices, you need to set the smart camera first via the SCMVS client software.

Before you start:

- Make sure that the PC has installed the SCMVS client software.

- Check the device's firmware version.

Note

Regarding SC2000Pro series smart camera, you need to operate it via the web, and operations are similar to those via the SCMVS client software.

Steps

1. Log in the device via the SCMVS client software.
2. Click **Communication** on the menu bar.
3. Click **+** to add ModBus communication.
4. Switch on to enable ModBus communication.
5. Keep the default parameter settings.

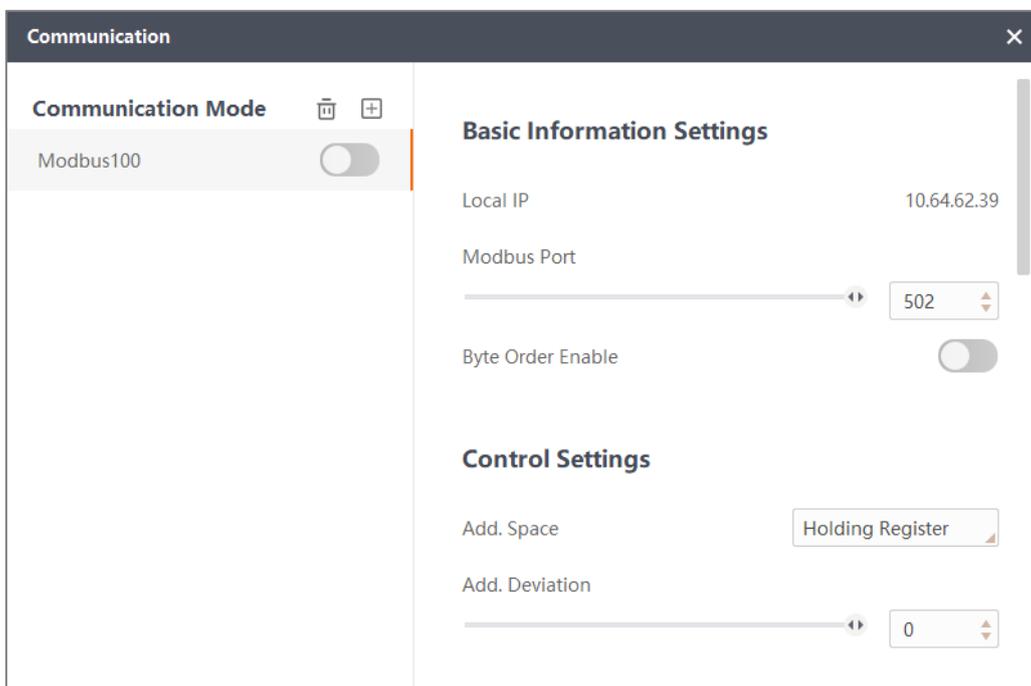


Figure 3-2 Set Communication Parameters

Note

Byte order is only invalid for float type results.

6. Create a project and set parameters for cameras, base images, and vision tools according to actual demands.

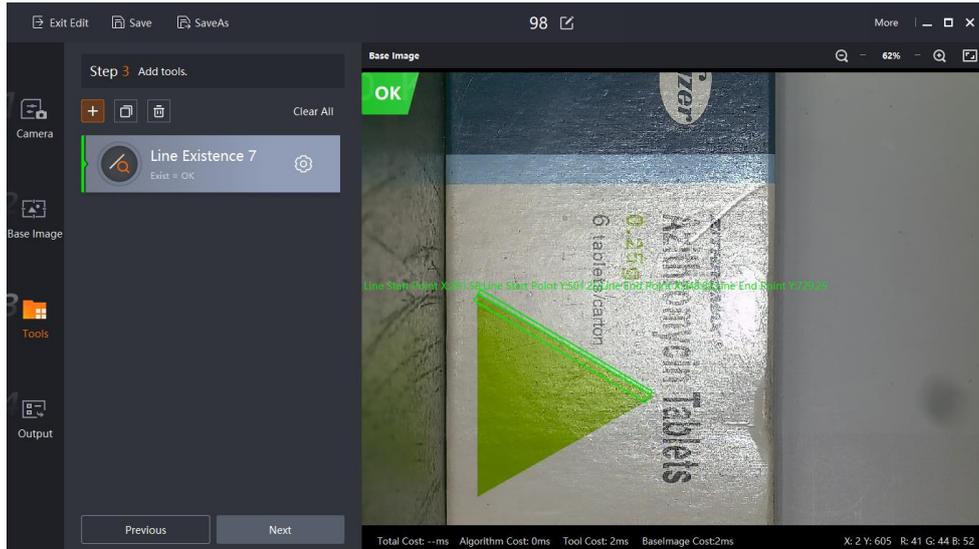


Figure 3-3 Line Existence

Note

Here we take the tool of line existence as an example.

7. Go to **Output** → **Tool Results** → **Edit** to set the outputted content.

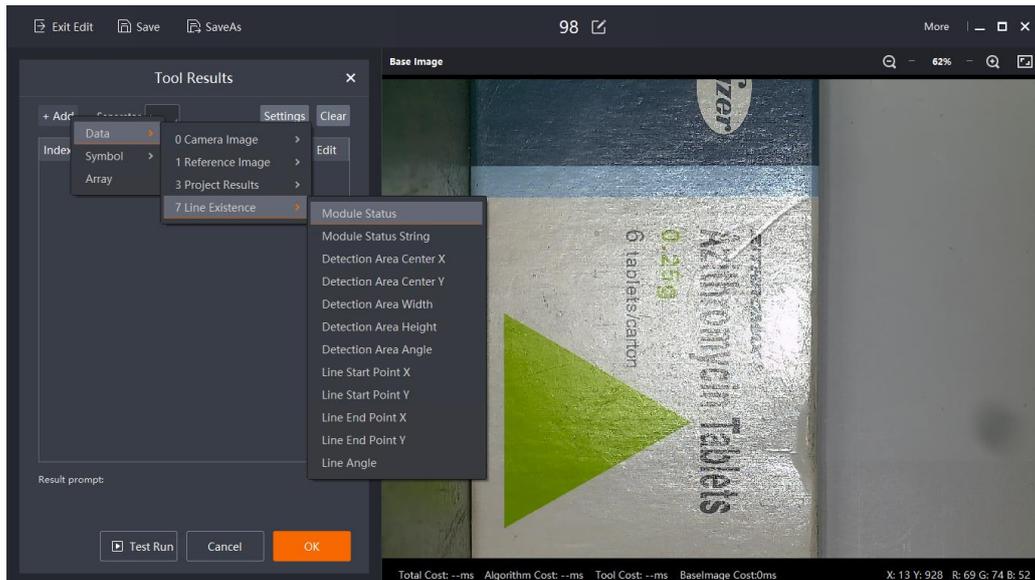


Figure 3-4 Set Tool Results

Note

Here we take the module status and module status string of line existence as an example to introduce tool results.

8. Save the project.

9. If you want to switch projects via communications, go to project management, select **Trigger Communication** as **Mode**, enter **Communication String**, and **Switch Return String** according to actual demands.

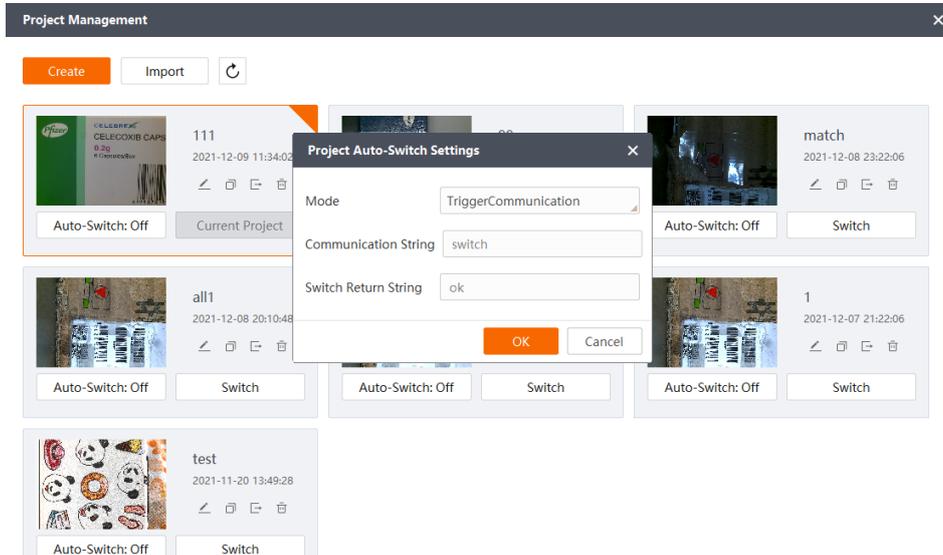


Figure 3-5 Project Switch Settings

3.4 ModBus TCP Test Tool

The ModBus TCP test tool can be used to test whether the ModBus communication between devices is normal. When the device is working in server mode, ModBus Poll test software can be used. ModBus Poll software works in client mode and will actively connect to the device.

Steps

1. Enter connection setup parameters according to actual demands.

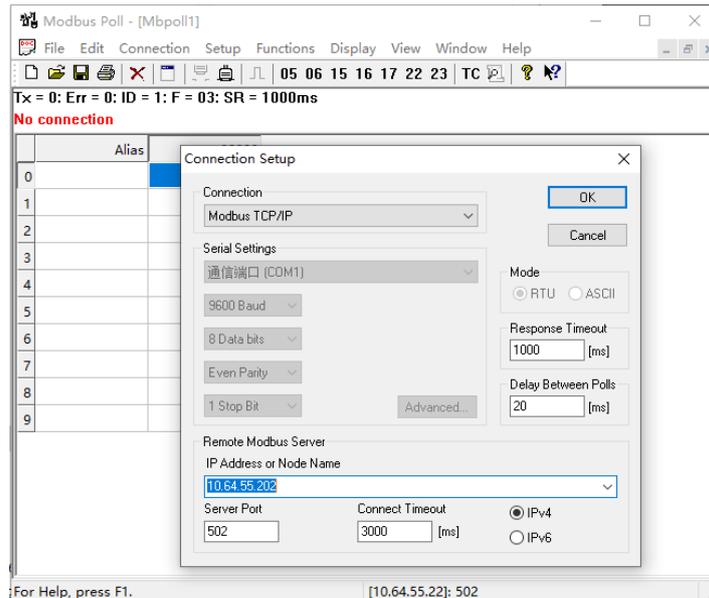


Figure 3-6 Set IP Address and Port

Note

The server port is set as 502.

2. Poll and read the holding registers of the device via the 03 command, start reading from address offset 0, and set an appropriate number of read registers and polling read interval time.

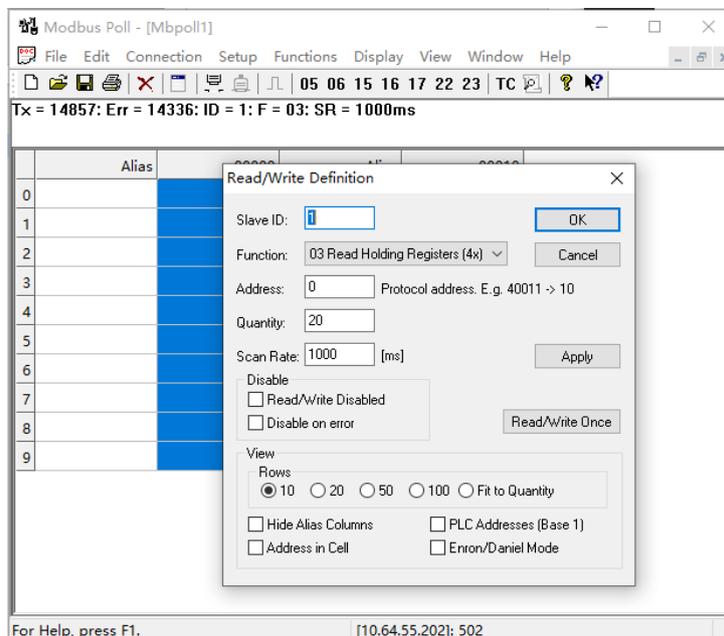


Figure 3-7 Set Polling and Reading Holding Register

3. Alternately write 3 and 5 to the register of offset address 0 in the control area to trigger the device to run the project once, and return the project result in the result area.

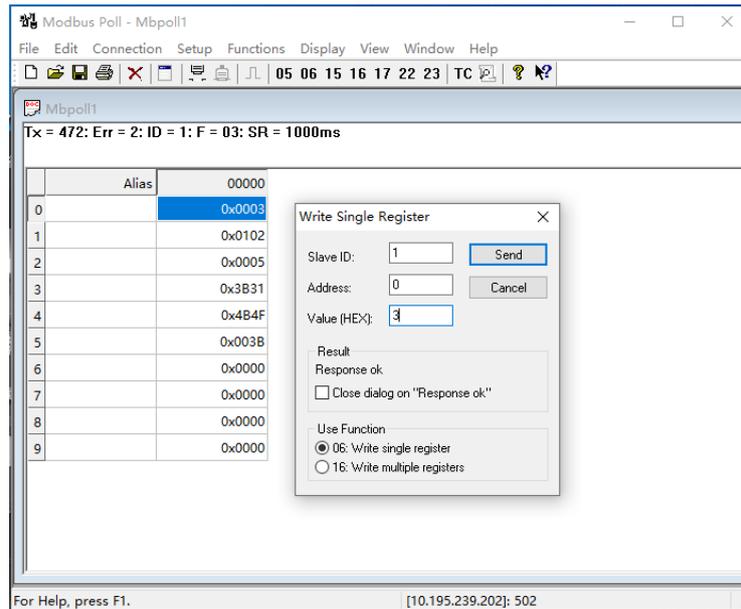


Figure 3-8 Write Single Register

3.5 PLC Settings

This section takes the ModBus TCP settings of Siemens S7 series PLC as an example to explain how to set PLC's ModBus communication.

Steps

1. Run TIA Portal, click **Create new project**, enter parameters, and click **Create**.

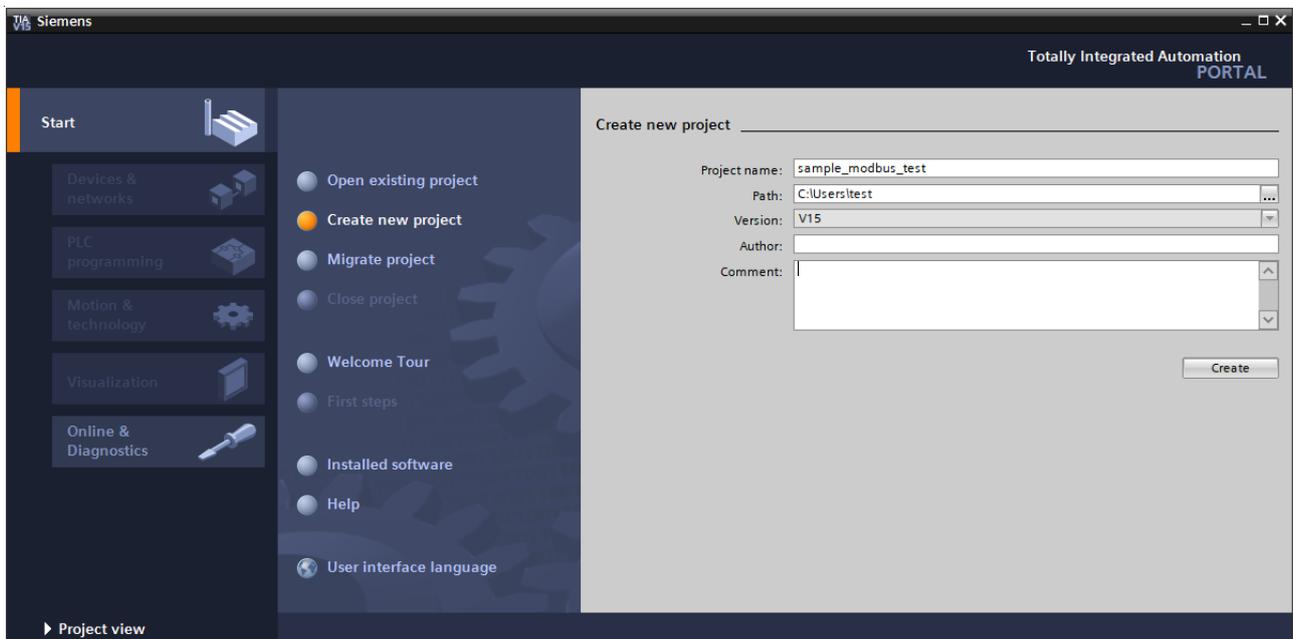


Figure 3-9 Create New Project

2. Add the corresponding PLC controller.

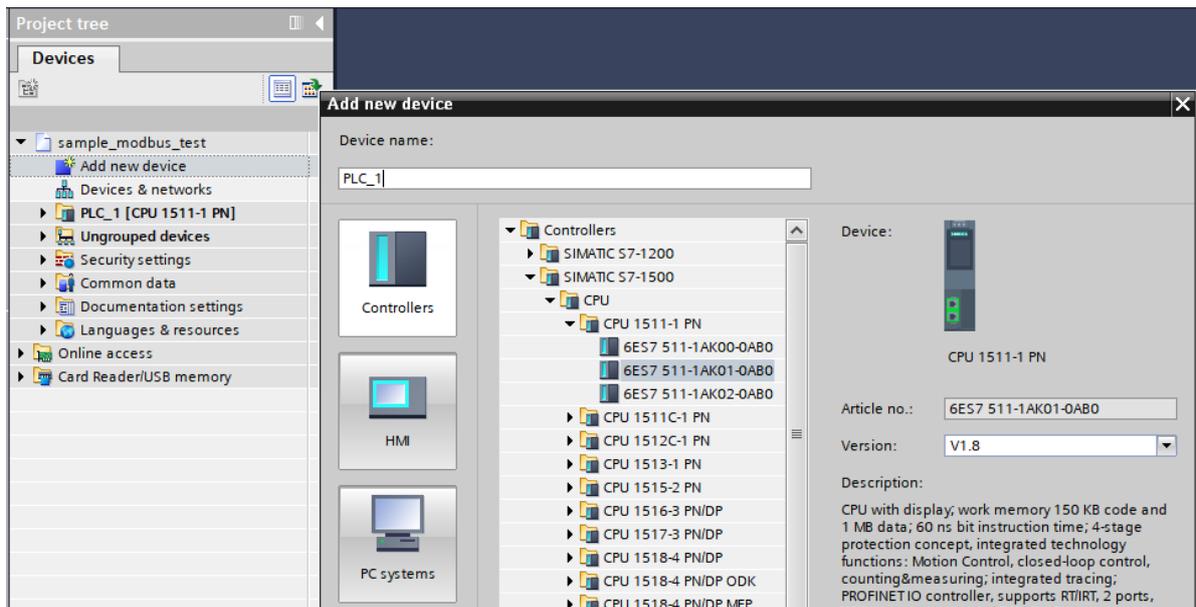


Figure 3-10 Add PLC

3. Set the IP address of the PLC controller.

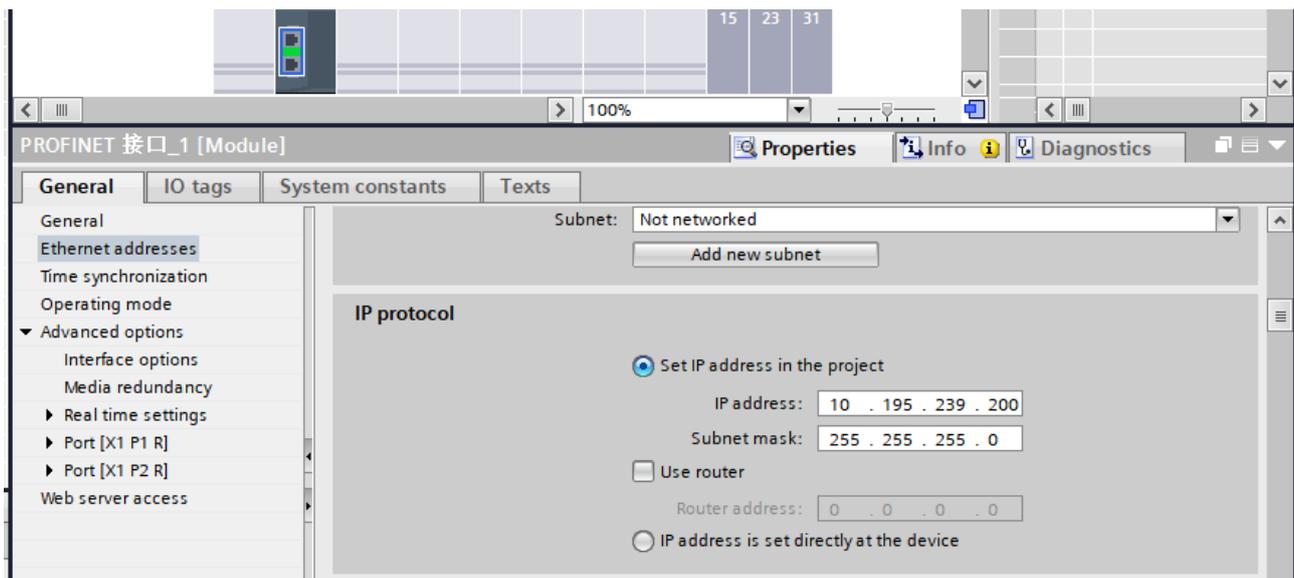


Figure 3-11 Set IP Address

4. Go to **Communication** → **Others** → **MODBUS TCP**, and add two MB_CLIENT commands.

Note

Two MB_CLIENT commands: One is used to write to register, and the other one is used to read register.

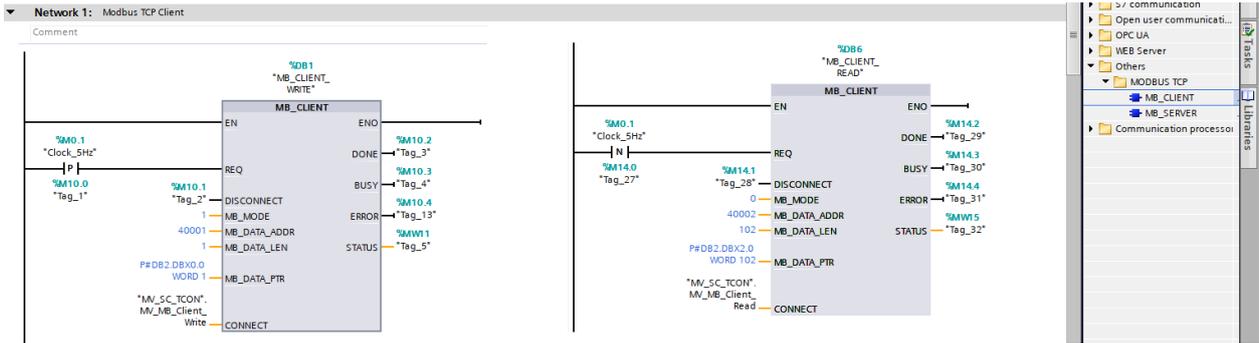


Figure 3-12 Add MB_CLIENT Commands

5. MB_DATA_PTR points to the pointer of the data register, create a new MV_SC_REG data block to store the communication data of the device's register, and add variables to the data block.

Name	Data type	Offset	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...
1	Static								
2	Control	0.0	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3	Status	2.0	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
4	Result	4.0			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
5	Result Length	4.0	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
6	Result Data	6.0			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

Figure 3-13 MV_SC_REG Data Block

6. Create a new MV_SC_TCON data block at the CONNECT pin, and create two TCON_IP_v4 type variables (MV_MB_Client_Write, MV_MB_Client_Read), and expand variables to assign values to them.

Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...	Comment
1	Static								
2	MV_MB_Client_Write	TCON_IP_v4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
3	InterfaceId	HW_ANY	64	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			HW-identifier of IE-interface submodule
4	ID	CONN_OUC	16#1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			connection reference / identifier
5	ConnectionType	Byte	16#0B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			type of connection: 11=TCP/IP, 19=UDP (17=TCP)
6	ActiveEstablished	Bool	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			active/passive connection establishment
7	RemoteAddress	IP_V4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			remote IP address (IPv4)
8	ADDR	Array[1..4] of Byte		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			IPv4 address
9	ADDR[1]	Byte	16#0a	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			IPv4 address
10	ADDR[2]	Byte	16#c3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			IPv4 address
11	ADDR[3]	Byte	16#ef	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			IPv4 address
12	ADDR[4]	Byte	16#c6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			IPv4 address
13	RemotePort	UInt	502	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			remote UDP/TCP port number
14	LocalPort	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			local UDP/TCP port number
15	MV_MB_Client_Read	TCON_IP_v4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
16	InterfaceId	HW_ANY	64	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			HW-identifier of IE-interface submodule
17	ID	CONN_OUC	16#2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			connection reference / identifier
18	ConnectionType	Byte	16#0B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			type of connection: 11=TCP/IP, 19=UDP (17=TCP)
19	ActiveEstablished	Bool	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			active/passive connection establishment
20	RemoteAddress	IP_V4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			remote IP address (IPv4)
21	RemotePort	UInt	502	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			remote UDP/TCP port number
22	LocalPort	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			local UDP/TCP port number

Figure 3-14 MV_SC_TCON Data Block

Note

The ID value should be different.

7. Repeat steps above to configure other pins.
8. Compile project and download it to the PLC controller.

3.6 Device Register Definition

The device only uses holding registers and is divided into four areas: control, status, result, and command.

Control

The PLC sends commands via control area and controls the device to acquire image, and run projects.

The control area controls the device to get images from PLC to the device.

- Control area location: holding register, fixed offset (0 by default).
- Control area size: 1 register (2 bytes)

Table 3-1 Control Area Definition

REG/bit	Name	Description
REG0.0	Trigger Enable	The PLC controls the device's trigger enable function via this bit.
REG0.1	Trigger	When following conditions are met, the PLC sets this bit to trigger the device to acquire an image and run the algorithm once. <ul style="list-style-type: none"> • Trigger Enable is set. • The device is not currently acquiring images and running algorithms. • Trigger Ready is set.
REG0.2	Results Ack	After the PLC reads the trigger result, it sets this bit to notify the device, and the device clears Results Available and Results Timeout after confirmation.
REG0.3-7	Reserved	Reserved.
REG0.8	Execute Command	Execute the command specified in the commands area once on the rising edge.
REG0.3-14	Reserved	Reserved.
REG0.15	Clear Error	Clear error status.

Status

The status area feedbacks the device's current status from the device to PLC.

- Status area location: holding register, fixed offset (1 by default).
- Status area size: 1 register (2 bytes)

Table 3-2 Status Area Definition

REG1/bit	Name	Description
REG1.0	Trigger Ready	The device is ready to receive new trigger signal. When the Trigger Enable is set and the device is ready to receive next trigger signal, the Trigger Ready is set.
REG 1.1	Trigger Ack	The device has already received the trigger signal.
REG 1.2	Acquiring	The device is acquiring images.
REG 1.3	Decoding	The device is recognizing decodes on images.
REG 1.4-7	Reserved	Reserved.
REG 1.8	Results Available	The device outputs new results. When the PLC is set to Result Ack, the Results Available will be cleared.
REG 1.9	Results Timeout	Results are not gotten when the time is out, and internal timeout is 6 sec. When the PLC is set to Result Ack, the Results Timeout will be cleared.
REG 1.10-14	Reserved	Reserved.
REG 1.15	General Fault	The device's internal fault, and you can clear this signal via Clear Error.

Result

The result area stores result data from the device to PLC.

- Result area location: holding register, fixed offset (2 by default).
- Result size: 4 to 500 registers (100 by default).

Table 3-3 Result Area Definition

REG/word	Name	Description
REG2	Result Length	The result area includes invalid data length.
REG3...	Result Data	The result output by the device. <ul style="list-style-type: none"> ● When result data length is smaller than configured result module, the spare bytes will be filled with 0. ● When result data length is greater than configured result module, the extra bytes will be cut off.

Command

The command area stores commands that users send to the device from the PLC to the device.

- Command area location: holding register, fixed offset (500 by default).
- Command area size: 4 to 500 registers (100 by default).

Table 3-4 Command Area Definition

REG/word	Name	Description
REG500	User Data Length	The instruction area includes invalid data length.
REG501...	User Data	Command character.

3.7 Trigger Test

Communication Sequence Diagram

The communication sequence diagram of PLC and the device is shown below.

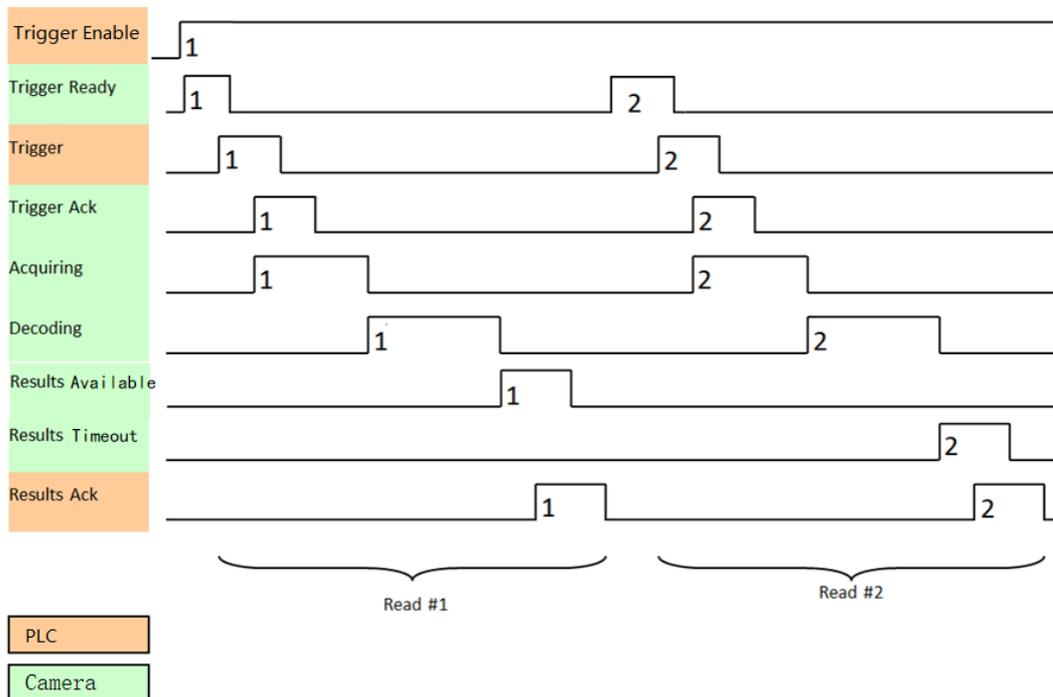


Figure 3-15 Communication Sequence Diagram

The logic of the above sequence diagram is shown below:

1. PLC sets Trigger Enable signal REG0.0. After the device is ready, set Trigger Ready signal REG1.0.

2. After detecting the device's Trigger Ready signal, the PLC sends Trigger signal REG0.1 and controls the device to operate once.
3. The device starts to acquire images and runs the algorithm after receiving Trigger signal.
 - If the result of the algorithm tool is outputted correctly, set the Results Available signal REG1.8 and put contents of the configuration result into the address starting from REG2.
 - If the result output times out, set the Results Timeout signal REG1.9 and clear the start address of REG2.

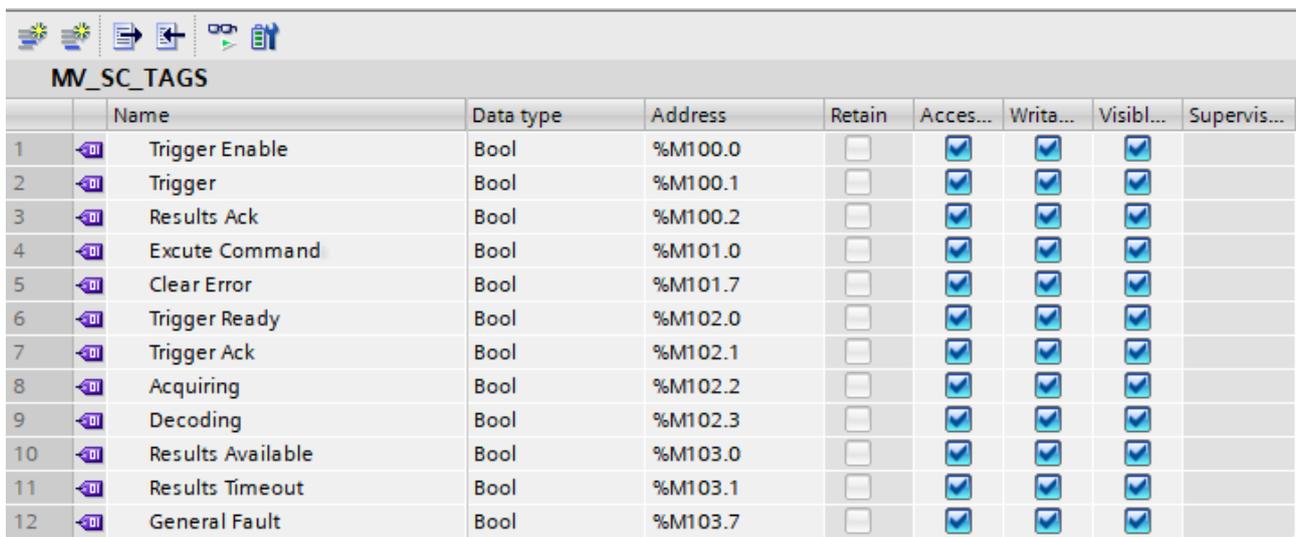
Note

For tools like character recognition, the result will be outputted after the character is recognized. If the character is not recognized, the result will not be outputted, and the result output will time out at this time. If you need to return results quickly, you can use the exception output function.

4. After detecting Results Available, the PLC starts to read results from REG2.
5. After reading results is finished, set Results Ack signal REG0.2, and notify the device.
6. After the trigger flow is finished, PLC sends Trigger signal again and next round of trigger flow begins.

Create Variables

Create variables in accordance with the device's register.



MV_SC_TAGS									
	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Supervis...	
1	Trigger Enable	Bool	%M100.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
2	Trigger	Bool	%M100.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3	Results Ack	Bool	%M100.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
4	Excute Command	Bool	%M101.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
5	Clear Error	Bool	%M101.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
6	Trigger Ready	Bool	%M102.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
7	Trigger Ack	Bool	%M102.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
8	Acquiring	Bool	%M102.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
9	Decoding	Bool	%M102.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
10	Results Available	Bool	%M103.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
11	Results Timeout	Bool	%M103.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
12	General Fault	Bool	%M103.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

Figure 3-16 Create Variables

Ladder Diagram

The PLC triggers the device to run the project, and related ladder diagrams are shown below.

- Send Control and Reading Status

Write MW100 to the control area of the device, read the status area of the device to MW102,

and create variables for each bit of control and status.

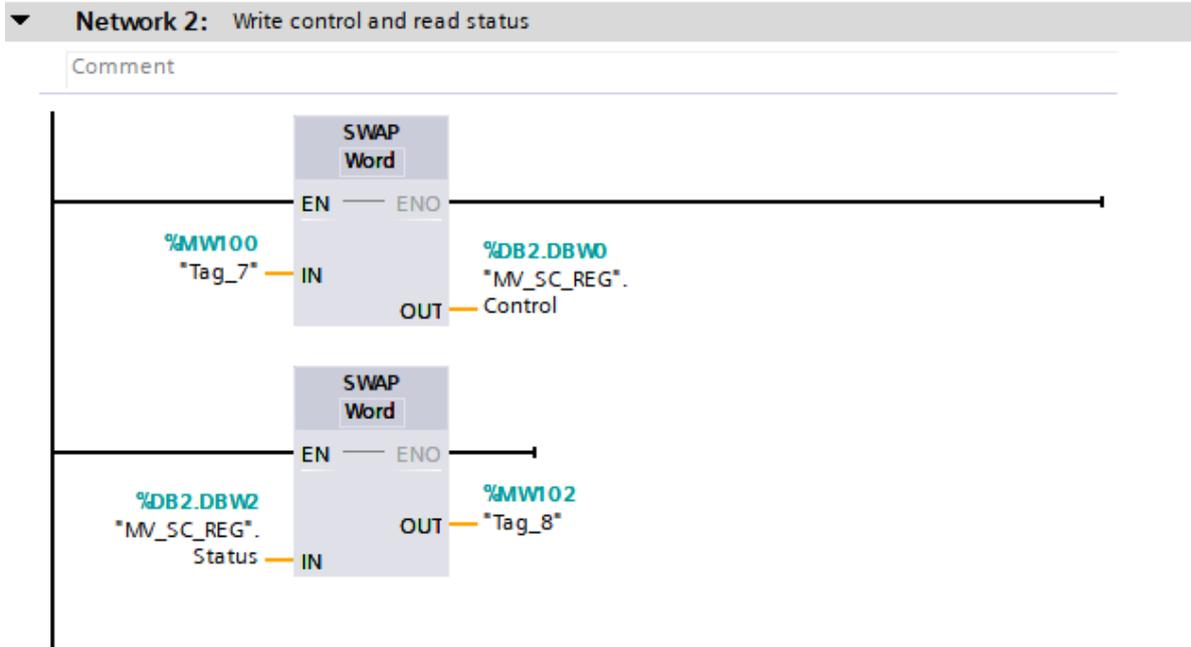


Figure 3-17 Send Control and Reading Status

- Enable Trigger



Figure 3-18 Enable Trigger

- Send Trigger Signal

The trigger signal can be generated by adding an edge signal before Trigger Ready according to actual demands. For example, when the PLC detects the photoelectric input signal, it sends a trigger signal once. The figure below describes that a trigger signal is

outputted every second via the internal system clock.

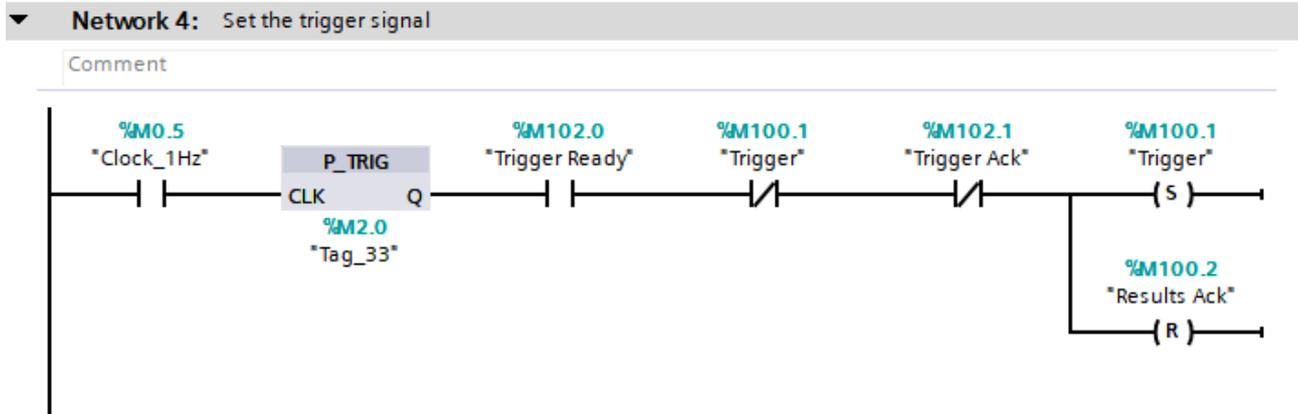


Figure 3-19 Send Trigger Signal

- Clear Trigger Signal

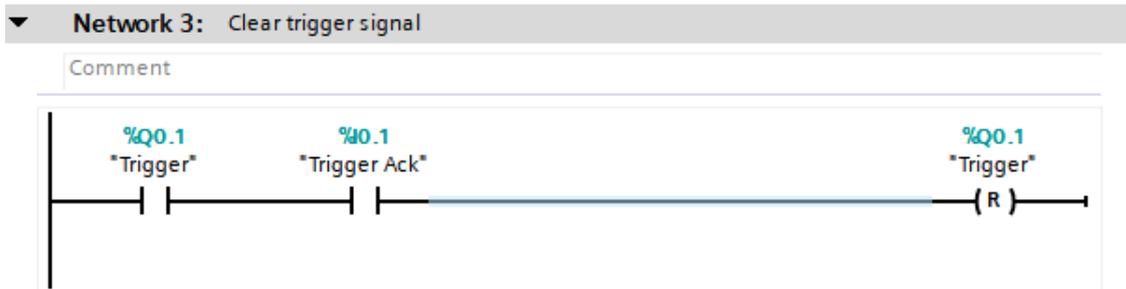


Figure 3-20 Clear Trigger Signal

- Get Device Results

Create a DB data block to store the reading result, add the result array variable as shown below.

MV_SC_RESULT									
	Name	Data type	Offset	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
1	Static				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	result	Array[0..199] of Char	0.0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Figure 3-21 Add Result Array Variable

When the device feedbacks the Results Available or Timeout signal, it means that the result data has been updated, and the PLC reads the result data to the DB data block. After the reading is finished, set the Results Ack signal to acknowledge that the device has

finished reading the result data.

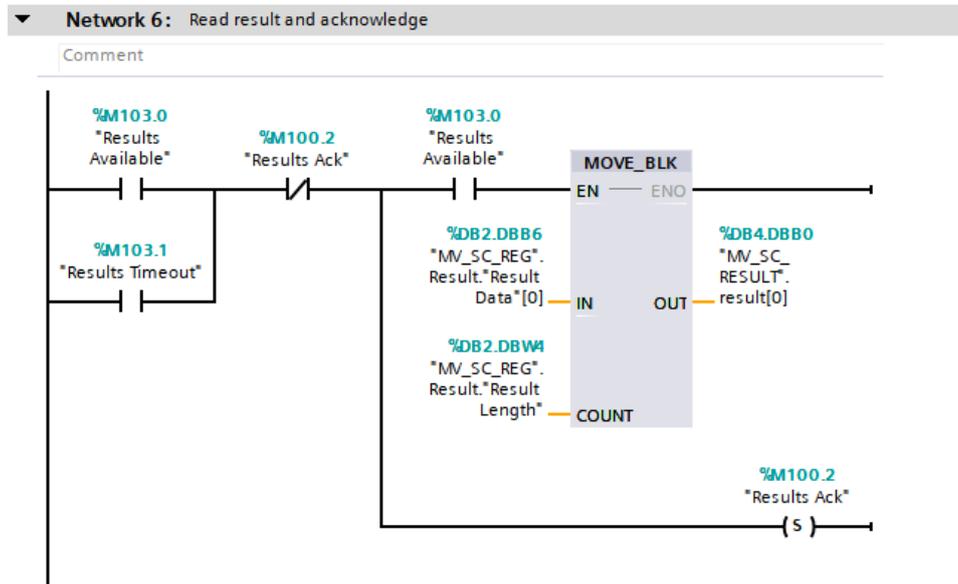


Figure 3-22 Get Device Results

- Clear Error

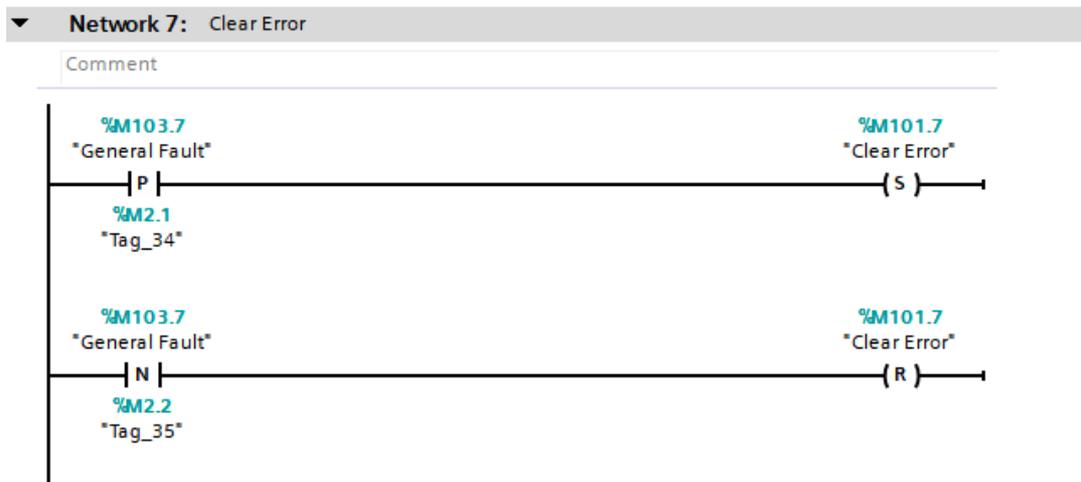


Figure 3-23 Clear Error

View Results

Through the monitoring and option tab, you can monitor the status of the variables of each module of the device and view the result output. You can also add trace records of the

variables via the Trace tab to view the control sequence.

	Name	Data type	Offset	Start value	Monitor value	Retain	Accessible f...	Writa...	Visible in ...
1	Static								
2	Control	Word	0.0	16#0	16#0005	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	Status	Word	2.0	16#0	16#0000	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	Result	Struct	4.0			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	Result Length	Word	4.0	16#0	16#0005	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	Result Data	Array[0..199] of Char	6.0			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	Result Data[0]	Char	6.0	''	''	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	Result Data[1]	Char	7.0	''	'1'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	Result Data[2]	Char	8.0	''	'K'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	Result Data[3]	Char	9.0	''	'O'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	Result Data[4]	Char	10.0	''	'\$00'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
12	Result Data[5]	Char	11.0	''	''	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figure 3-24 View Results

3.8 Project Switching Test

Create a DB data block to store command data, add the command variable as shown below. The first byte is the command length, and command character is stored starting from the second byte.

	Name	Data type	Offset	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
1	Static								
2	Command Length	Word	0.0	12	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	Command Data0	Char	2.0	'w'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	Command Data1	Char	3.0	's'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	Command Data2	Char	4.0	't'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	Command Data3	Char	5.0	'i'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	Command Data4	Char	6.0	'h'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	Command Data5	Char	7.0	'c'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	Command Data6	Char	8.0	't'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	Command Data7	Char	9.0	''	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11	Command Data8	Char	10.0	's'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12	Command Data9	Char	11.0	'e'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13	Command Data10	Char	12.0	'i'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
14	Command Data11	Char	13.0	't'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
15	Command Data12	Char	14.0	''	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Figure 3-25 Add Command Data Block

Note

The byte order of command character should be set according to the figure above.

The related ladder diagram of project switching test is shown below.

- Send Switching Command

Write "command length + command character" to the command area.

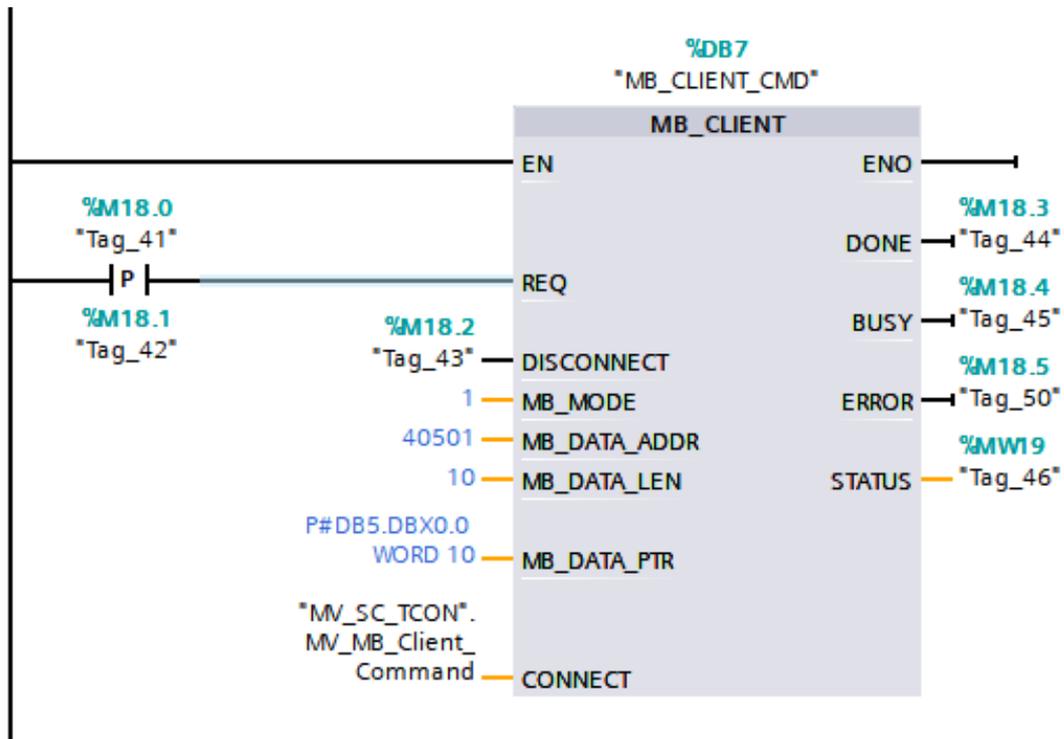


Figure 3-26 Send Switching Command

- Execute Switching Command

Write a rising edge to the Executive Command bit in the control area to execute a switching command for the project.

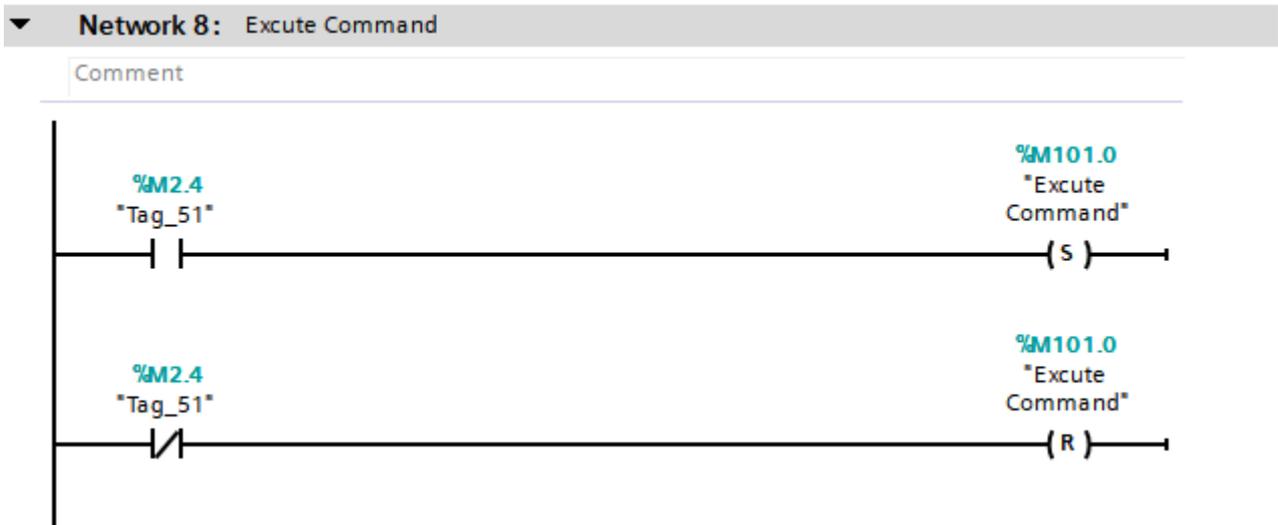


Figure 3-27 Execute Switching Command



See Far, Go Further