

EtherNet/IP Debugging Manual of ER Series Industrial Robot

RCS2 V1.04

ESTUN Automation Co., Ltd.
ESTUN Robotics Engineering Co., Ltd.

- This document is ISO compliant and will be updated in line with changes to the IPD system-

Revision History

S/N	Version	Revision Date	Content	Reviser
1	V1.00	2021.06.24	Create a new document	Zang Xiujuan
2	V1.01	2021.07.28	Add the content of “Section 2.2 Function configuration”	Zang Xiujuan
3	V1.02	2022.02.10	Modify the content of “Section 2.2 Function configuration”	Zang Xiujuan
4	V1.03	2022.07.15	Add Demo_EIP_Heartbeat function block for communication verification; add the descriptions for “Chapter 4 Configuration of EIP Communication Data Size Modification”	Han Zicong
5	V1.04	2022.10.27	Modify the “Section 2.2.2 Teach pendant configuration”	Zang Xiujuan

TABLE OF CONTENTS

Preface	1
Target Audience.....	1
Precautions.....	1
Safety Instructions	1
Chapter 1 Function Overview.....	2
Chapter 2 Protocol Introduction	3
2.1 Main indicators	3
2.2 Function configuration.....	3
Chapter 3 Instructions for Debugging	8
3.1 EtherNet/IP Scanner.....	8
3.2 EtherNet/IP Adapter.....	14
Chapter 4 Configuration of EIP Communication Data Size.....	17
4.1 Modifying the EipCfg.ini configuration file.....	17
4.2 Modifying EDS files.....	17

Preface

This manual is applicable to the RCS2 control system version V1.29.00 and above, and it provides a guide for debugging the EtherNet/IP functionality of ESTUN ER Series Robots.




Target Audience

This manual is intended for technical support personnel specifically working with ESTUN Robotics.

Precautions

- During the installation and debugging of these components, operators must strictly follow the instructions and explanations provided in this document.
- Relevant responsible personnel must ensure that the application or use of the mentioned products fulfils all safety requirements, including applicable laws, regulations, guidelines, and standards.
- While this document has been carefully prepared, the products described herein are subject to continuous updates and advancements. We may not always verify that the performance data, standards, or other characteristics described herein are consistent with the physical products after each update.
- Technical or editorial errors may occur in this document. We reserve the right to modify the document information without prior notice. If the data, diagrams and text descriptions in this document have not been modified for a product that has been changed, we will not make any specific statement to this effect.
- No modifications to the hardware or software configuration other than those specified in the text file are permitted, and ESTUN shall not be liable for any consequences resulting therefrom.
- The unit of measurement in the illustrations is in millimeters (mm) unless otherwise stated.

Safety Instructions

 Warning	Injury Failure to comply with the safety instructions associated with this symbol may jeopardize personal life and health safety.
 Caution	Danger to Environment and Equipment Failure to comply with the safety instructions associated with this symbol may pose significant risks to the environment and equipment safety.
 Note	Notes or Tips This symbol indicates that the information is provided to help you better understand the Safety Instructions.

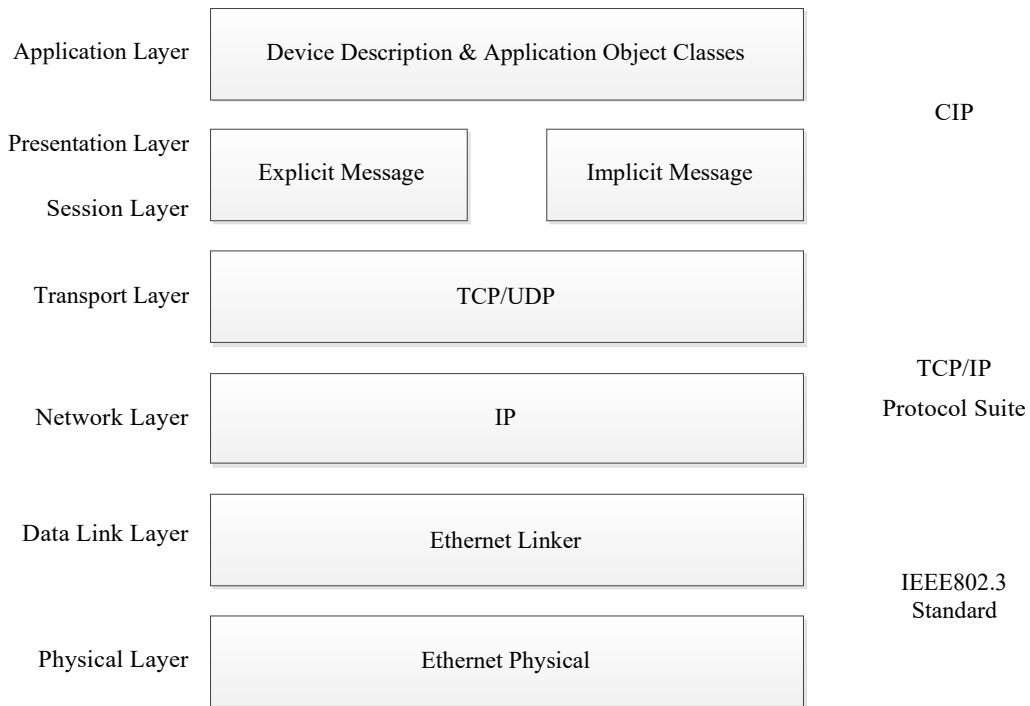
Chapter 1 Function Overview

The EtherNet/IP interface refers to a communication method in which an external logic controller (such as a PLC acts as an EtherNet/IP Scanner) and communicates with a robot (acting as an EtherNet/IP Adapter) using the standard EtherNet/IP protocol.

Chapter 2 Protocol Introduction

EtherNet/IP (Ethernet Industrial Protocol) is a protocol system designed for industrial applications. It is the latest addition introduced by two major industrial organizations, ODVA (OpenDeviceNet Vendors Association) and CI (ControlNet International). Similar to DeviceNet and ControlNet, EtherNet/IP is a network based on the Control and Information Protocol (CIP), utilizing the same object library and adhering to consistent industry specifications, ensuring good consistency and interoperability.

EtherNet/IP is an object-oriented protocol that ensures the effective transmission of both implicit real-time I/O information and explicit information (including configuration parameter settings, diagnostics, etc.) on the network. It utilizes standard Ethernet and TCP/IP technologies to transport CIP communication packets. By combining the widely used Ethernet and TCP/IP protocols with the universal and open application layer protocol CIP, the architecture of the EtherNet/IP protocol is formed. The layers of the protocol structure are as follows:



EtherNet/IP Protocol OSI Model

2.1 Main indicators

1. The robot has EtherNet/IP Adapter functionality, supporting a maximum of 256 bytes for implicit data input/output, and the communication data size can be adjusted.
2. In the Multiprog programming environment, users can access the data area through the provided interface and parse the data for command/status interaction.
3. The communication cycle is set by the EtherNet/IP Scanner.
4. The device's EDS file is provided by the manufacturer.

2.2 Function configuration

Currently, there are two versions of the EtherNet/IP module functionality in the system: trial version and standard version. By default, the EtherNet/IP functionality is not enabled in the standard robot products upon factory delivery.

Before using the EtherNet/IP functionality on-site, users need to activate it. The system supports two activation methods:

1. Configuration in the PLC project
2. Configuration on the teach pendant

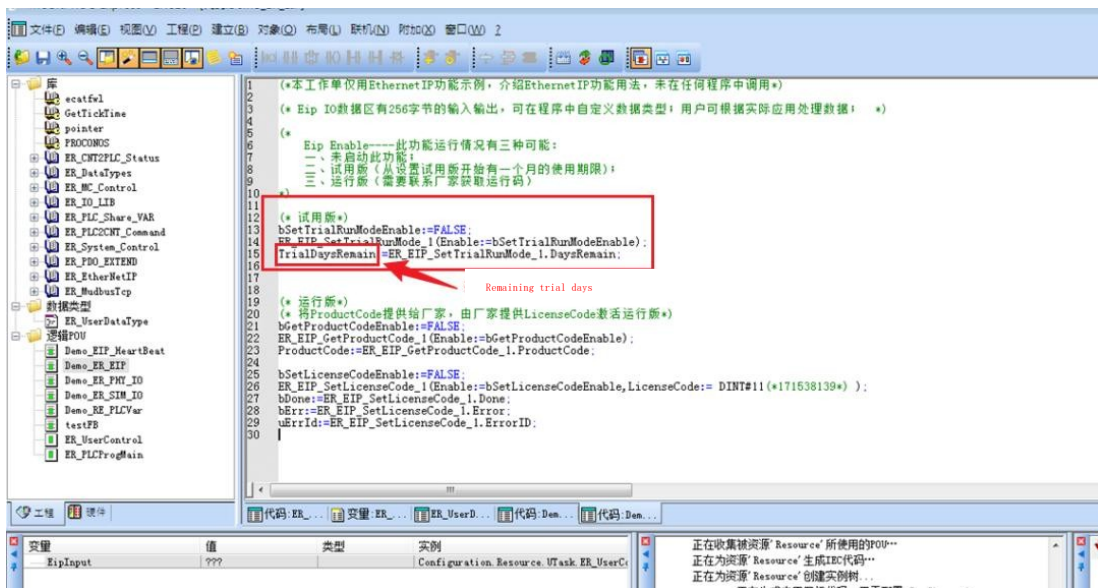
2.2.1 PLC configuration

Support is provided in the Multiprog template project to enable the trial version or directly register as the standard version by calling the function blocks. The detailed instructions are as follows:

1. Trial Version

In the Multiprog project, the EtherNet/IP trial version can be enabled by using the ER_EIP_SetTrialRunMode function block. The trial period is 30 days (counting from the first call to this function block), and the EtherNet/IP functionality will be automatically disabled after the trial period expires.

Example program to enable the trial version:



Note: In the above diagram, TrialDaysRemain indicates the remaining trial days available for use (unit: days).

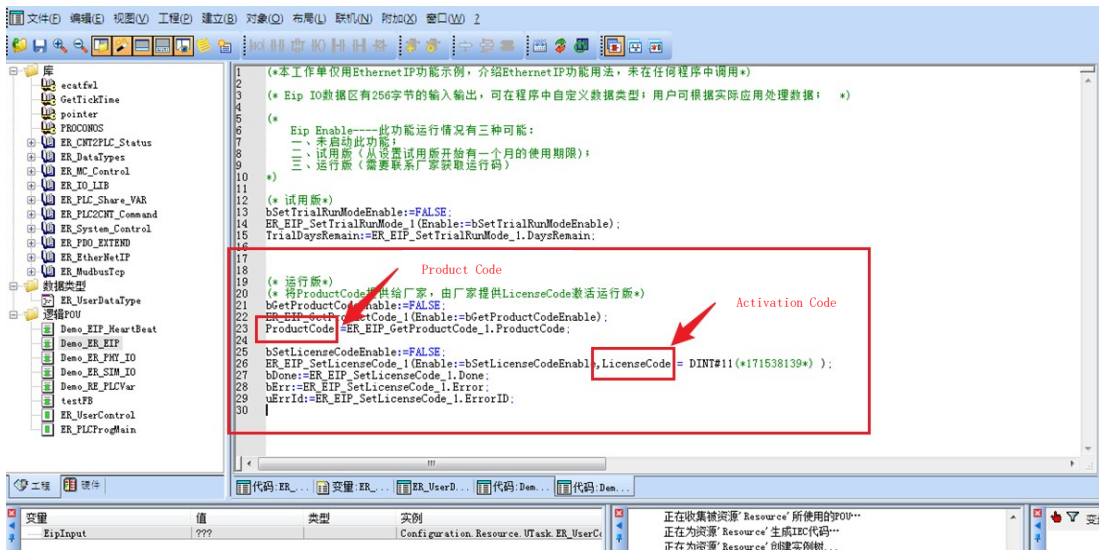
2. Standard Version

The standard version has no time limit. In the Multiprog project, you can register as the standard version by following these steps:

- 1) Use the ER_EIP_GetProductCode function block to obtain the product serial number and send it to the manufacturer. The manufacturer will provide the corresponding product license code.
- 2) Use the ER_EIP_SetLicenseCode function block to register the obtained product license code into the EtherNet/IP module. After successful registration, restart the controller for the changes to take effect.

Note: The product license code needs to be registered only once. Registering it multiple times may cause the license code to become invalid.

Example program to register the standard version:



Note: The assignment of the activation code DINT#11 in the diagram is for demonstration purposes only. Please use the actual activation code in practice.

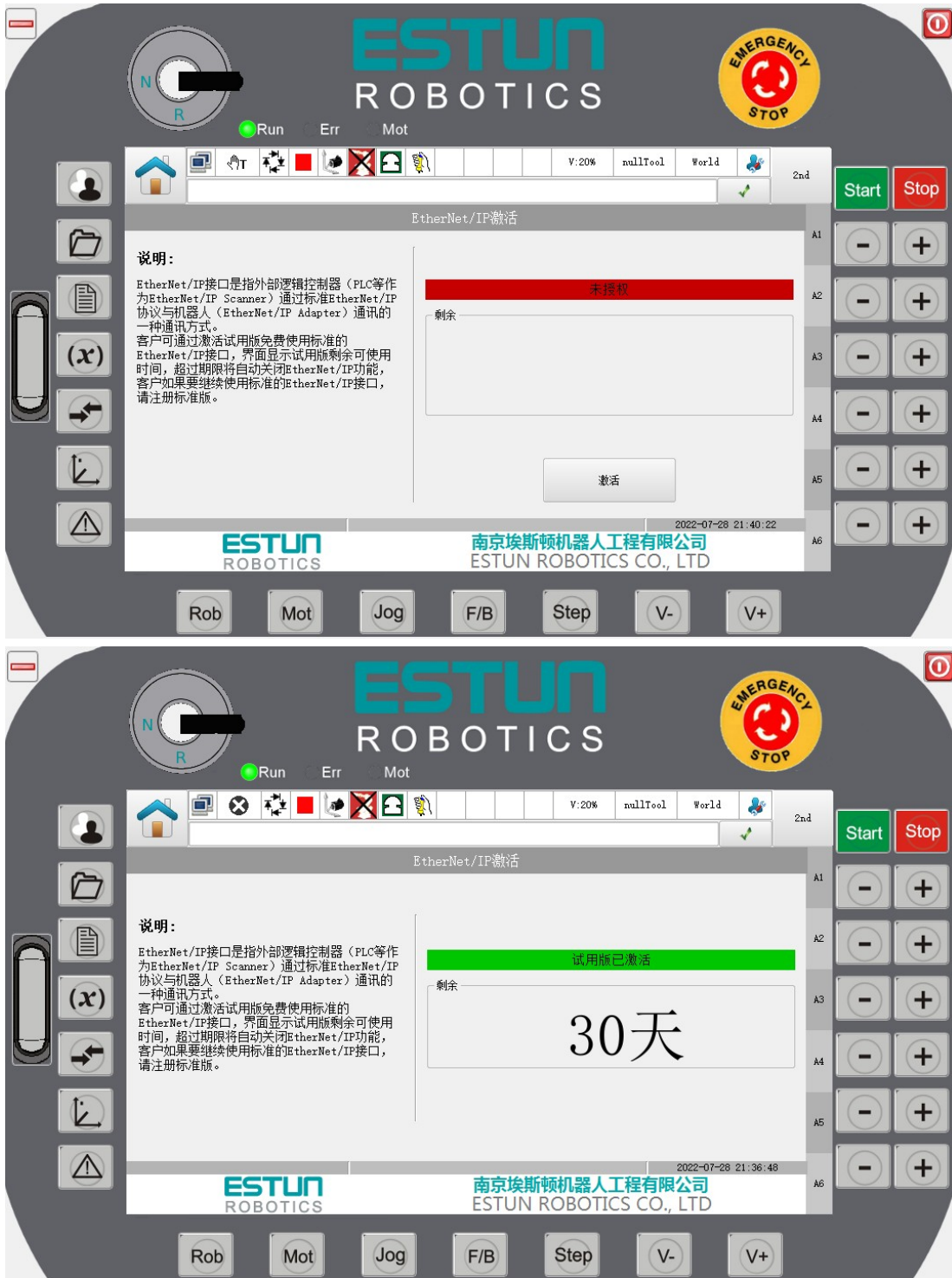
2.2.2 Teach pendant configuration

Support is available to enable the trial version or directly register it as a standard version on the teach pendant.



1. Trial Version

Under the Admin permission of teach pendant, access the software licensing in advanced settings. Click on the trial version to enter the trial version interface. After clicking the activate button, the interface will display the remaining trial days. The trial period is 30 days (counting from the first established communication), and the EtherNet/IP functionality will be automatically disabled after the trial period expires.

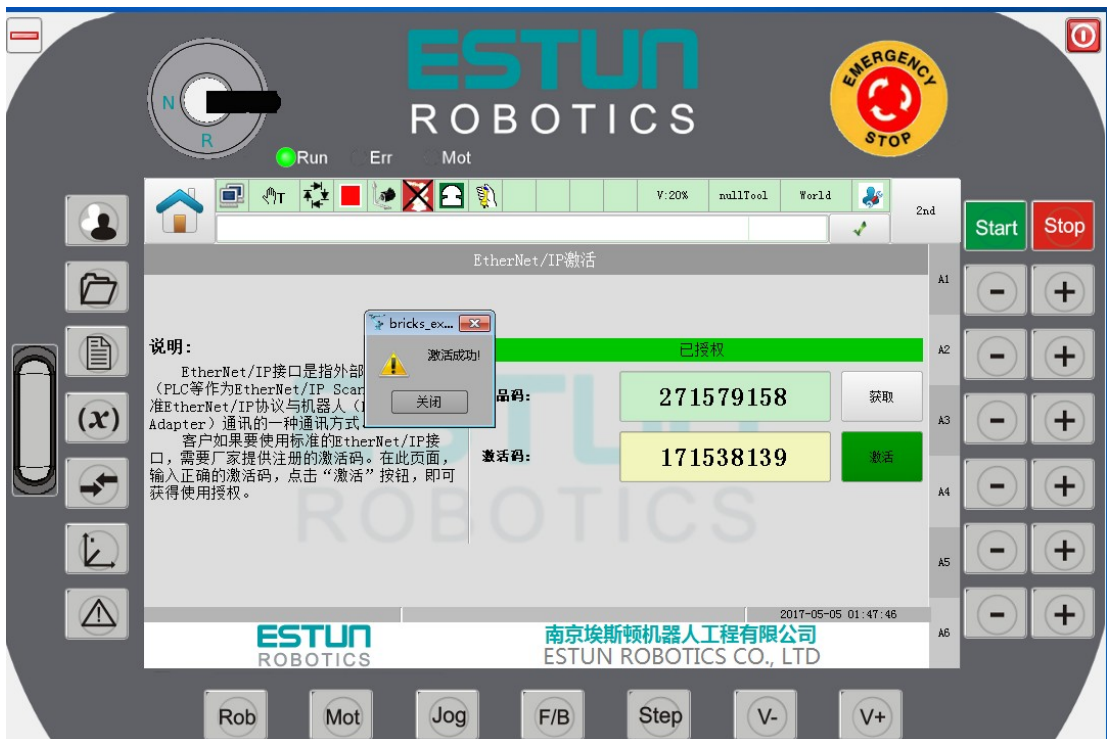


2. Standard Version

To register as the standard version on the teach pendant, follow these steps:

- 1) Under the Admin permission, access the software licensing in advanced settings. Click on the standard version to enter the Standard Version interface. Click the "Get" button on the interface to get the product serial number. Send this serial number to the manufacturer to obtain the corresponding product license code (matching the product serial number). (If the activation status is already activated when entering the interface, this activation configuration process is not necessary)

- 2) Enter the obtained product license code into the activation code editing box on the teach pendant. Click the Activate button. If the entered product license code is correct, a successful activation message will be displayed, and the activation status will be activated. If the entered product license code is incorrect, a successful activation message will be displayed, and the activation status will be deactivated.

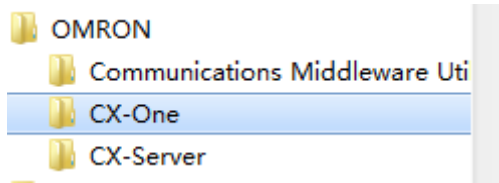


Chapter 3 Instructions for Debugging

In this section, the hardware Omron PLC PA202, CJ2M CPU31 is used as an example of an EtherNet/IP Scanner to illustrate how to use the EtherNet/IP interface to interact with the robot.

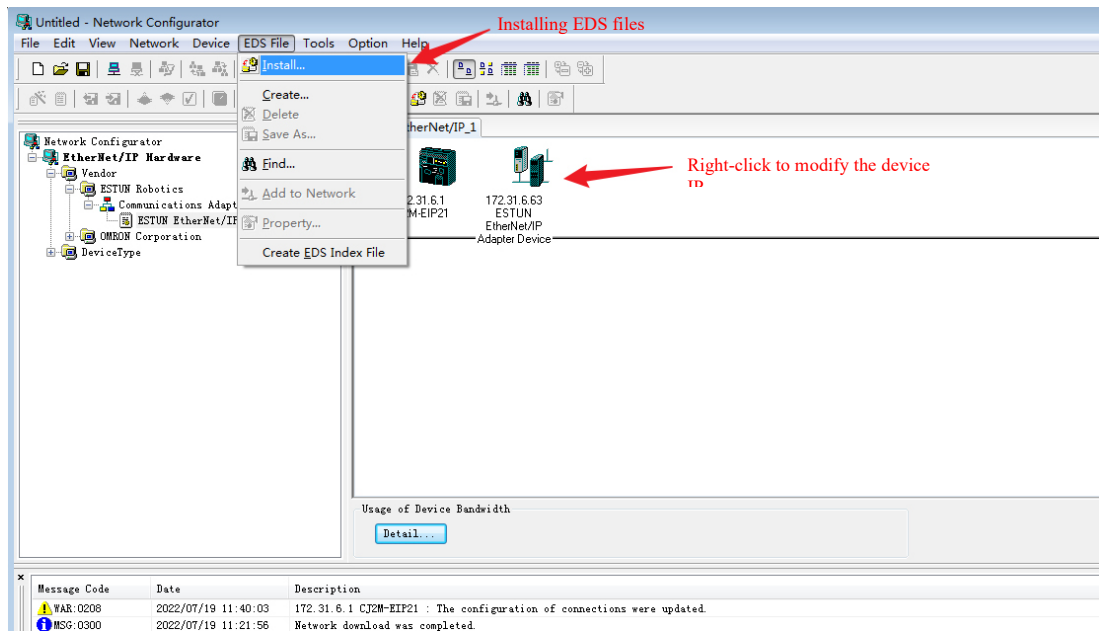
3.1 EtherNet/IP Scanner

1. Install the software corresponding to the Omron PLC. After successful installation, the files should be displayed as shown in the following image:

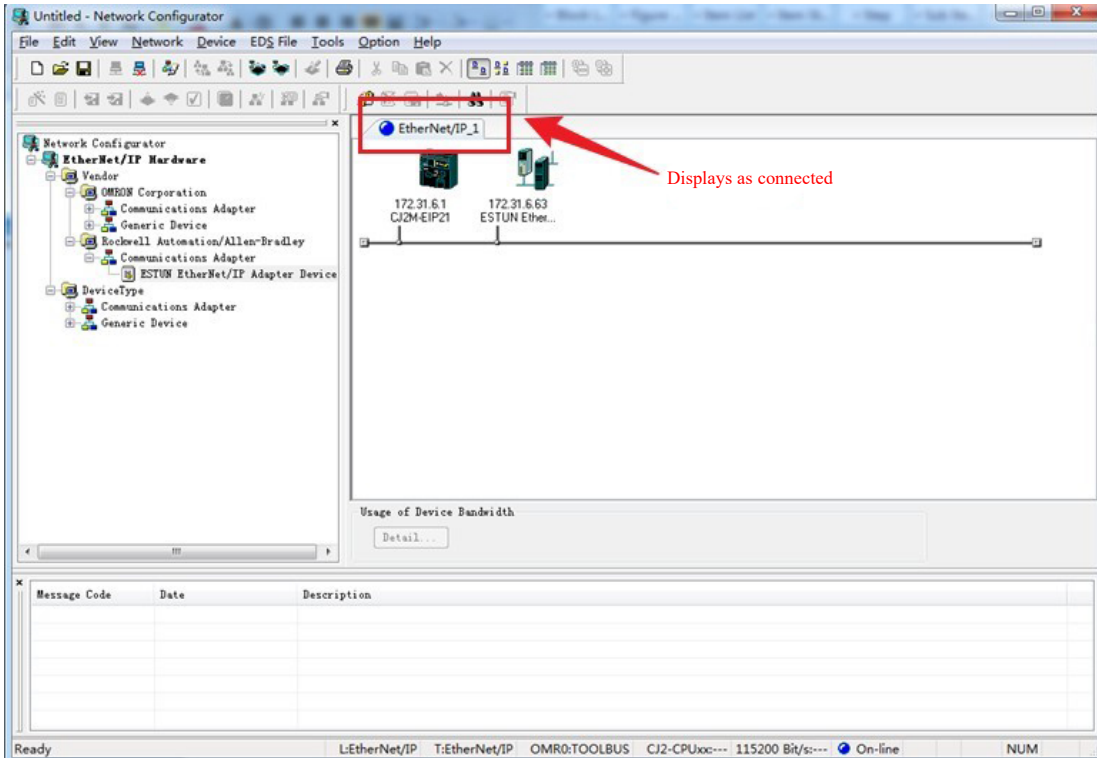


2. Connect the computer with the installed software to the Omron PLC hardware using a USB cable, and then configure the EtherNet/IP settings through the software.

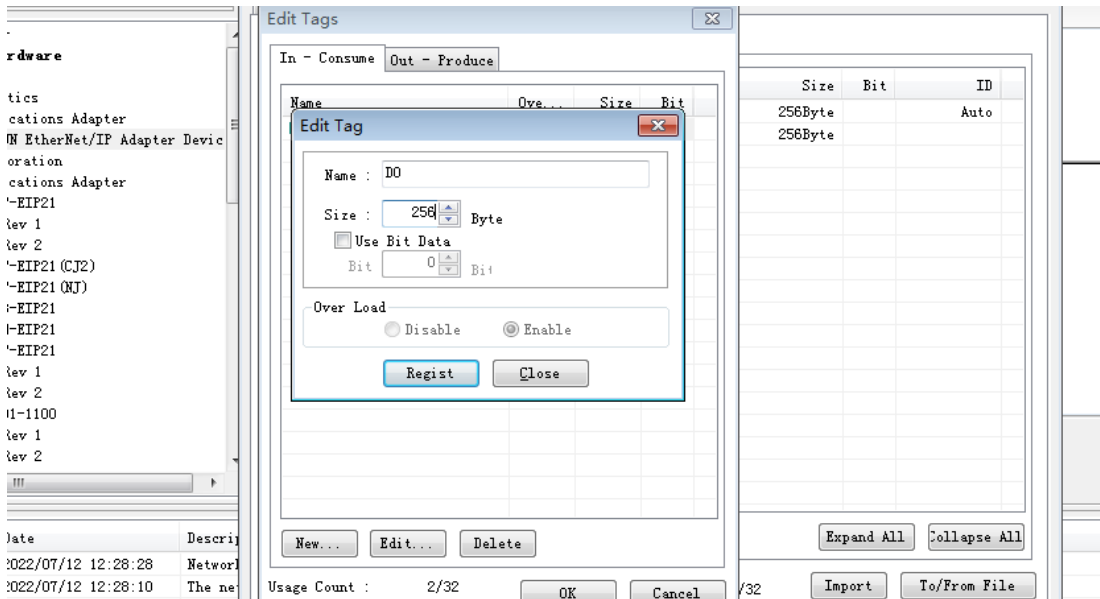
1) Open the Omron Network Configurator software for EtherNet/IP configuration. Install the EDS file for the EtherNet/IP Adapter. Drag the device from the left side to the right side of the EtherNet/IP network and modify the corresponding IP address (set the IP address of the EtherNet/IP Adapter according to the actual device). For example, refer to the following image:

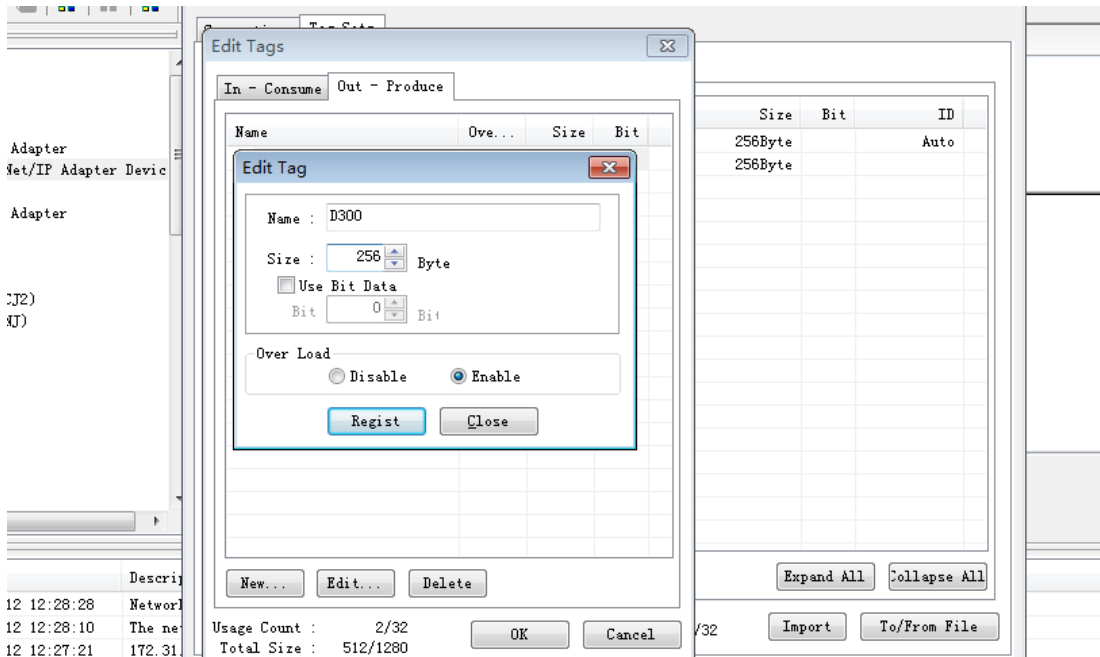


2) Click on "Connect" under the "Network" section to connect to the hardware device. The status in the status bar will show "Online". The process is as follows:

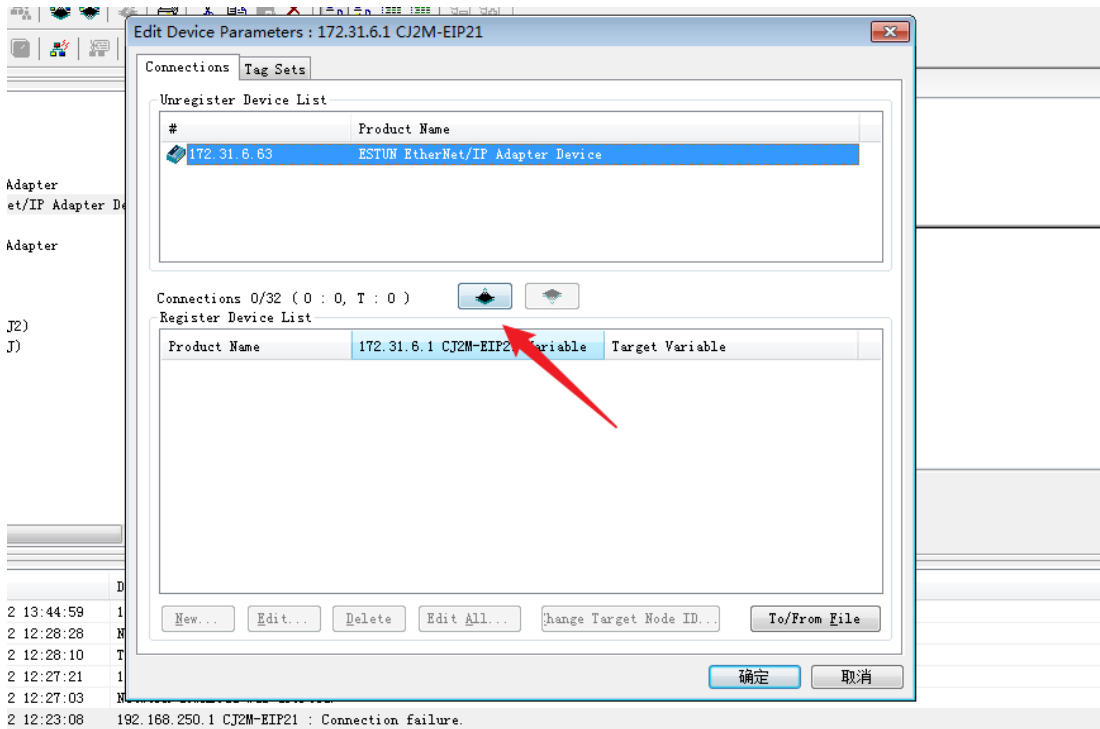


3) Double-click on the CJ2M device to configure the relevant parameters on the Omron PLC side. In this example, choose address D0 in the Omron PLC memory area as the starting address with a size of 256 bytes for inputs, which will receive output data from the robot controller. Choose address D300 as the starting address with a size of 256 bytes for outputs, which will be inputs on the robot controller side.

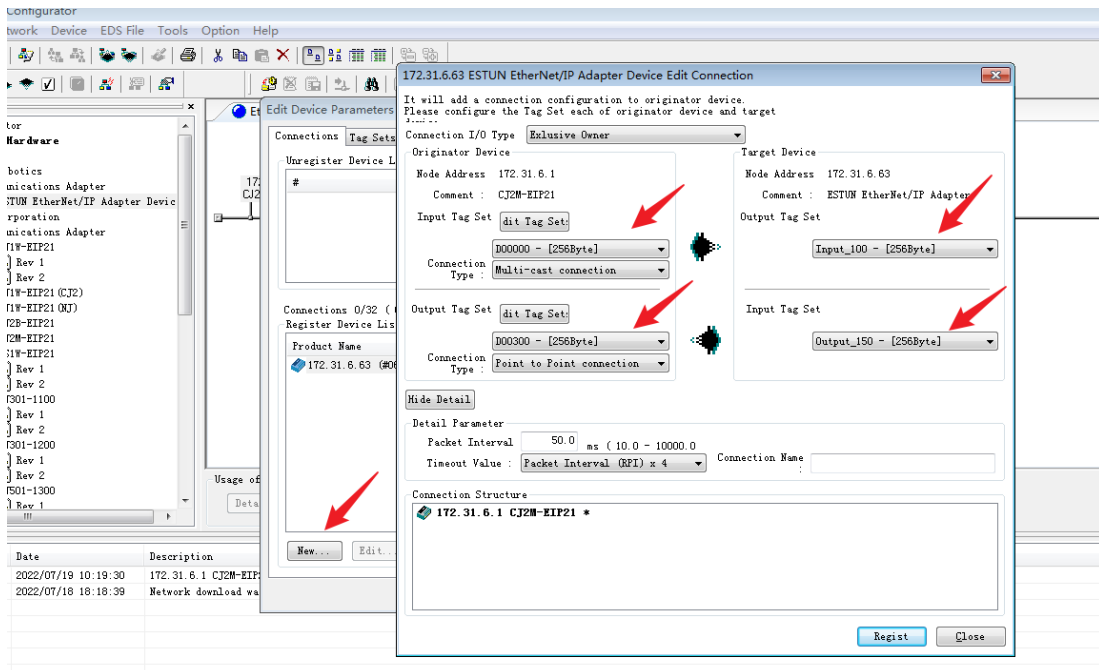




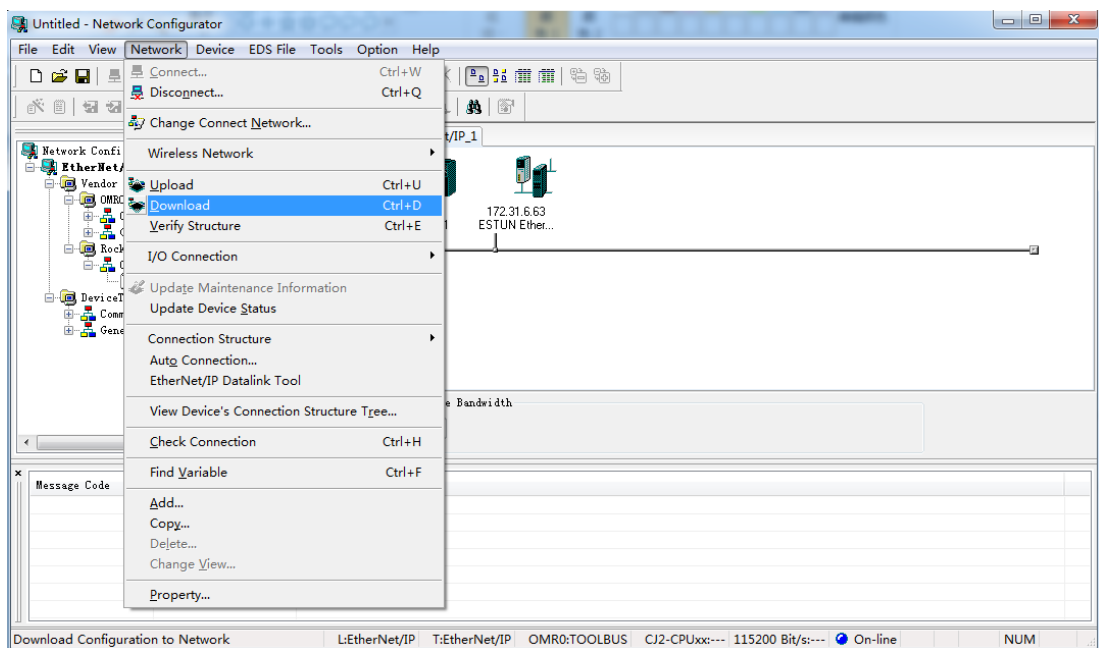
4) Double-click on the Omron PLC icon to register the device you want to connect, which in this case is the robot controller.



Click on "NEW" to create a mapping relationship. In the example, map D00000 to the Input_100 output port provided by the controller (named "Input" because the external PLC treats this interface as an input), and map D00300 to the Output_150 output port provided by the controller. The input and output ports are defined in the controller's EDS file.

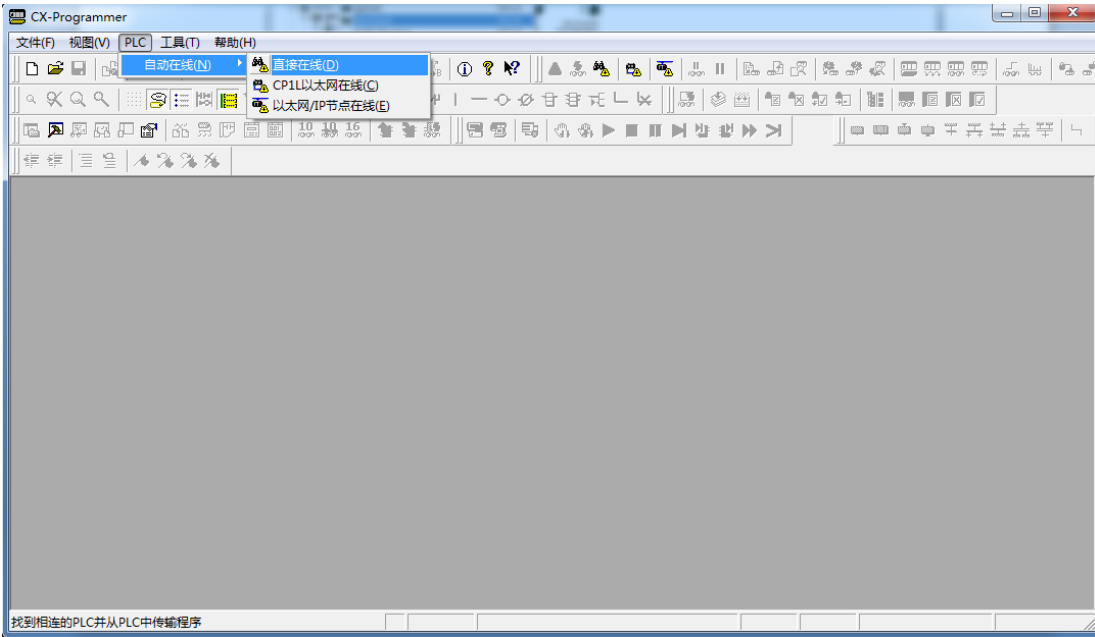


5) Download the relevant configuration to the PLC device by clicking on "Download" in the figure to complete the network configuration. (If the EtherNet/IP function is not enabled on the controller side, an alarm will appear, and communication cannot be established.)

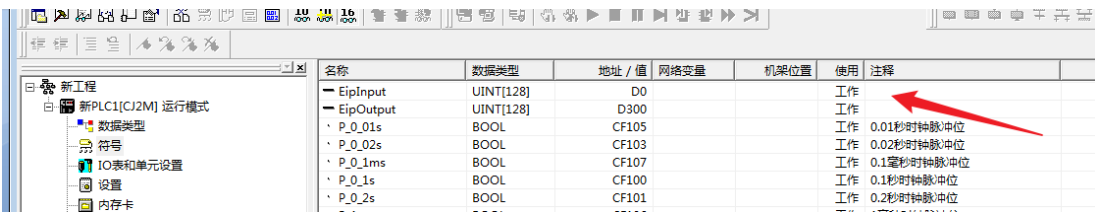


3. Connect the computer with the installed software to the Omron PLC hardware using a USB cable and perform data exchange through the software.

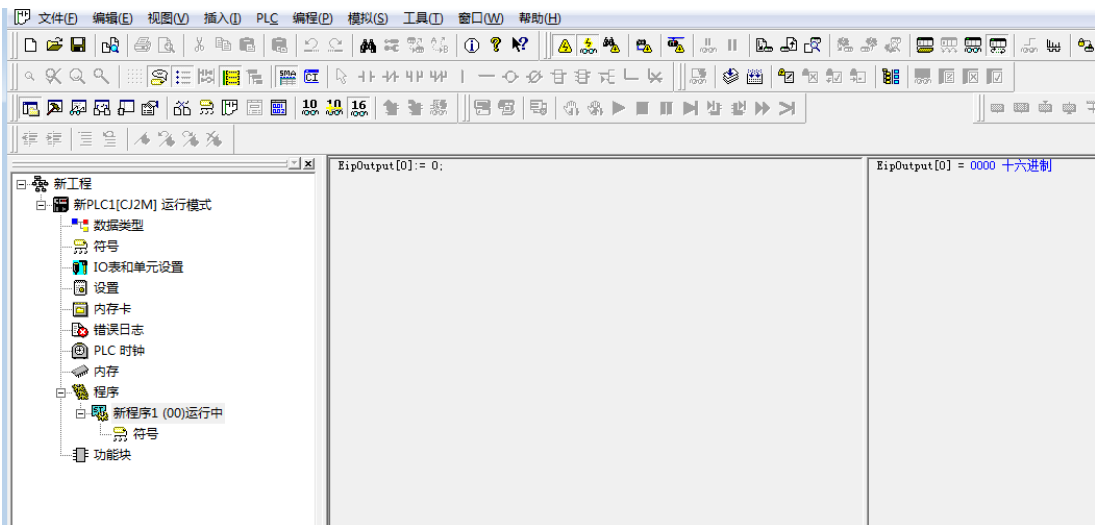
1) Open the Omron PLC programming software CX-Programmer and upload the actual connected PLC program to the programming software. For example, refer to the following image:



Define EipInput and EipOutput as inputs and outputs, and set their starting addresses as D0 and D300, respectively. Establish the mapping relationship. (Note: This Omron PLC device uses a two-byte addressing, and the data type of array elements is selected as UINT, which occupies two bytes. 2 bytes × 128 = 256 bytes.)

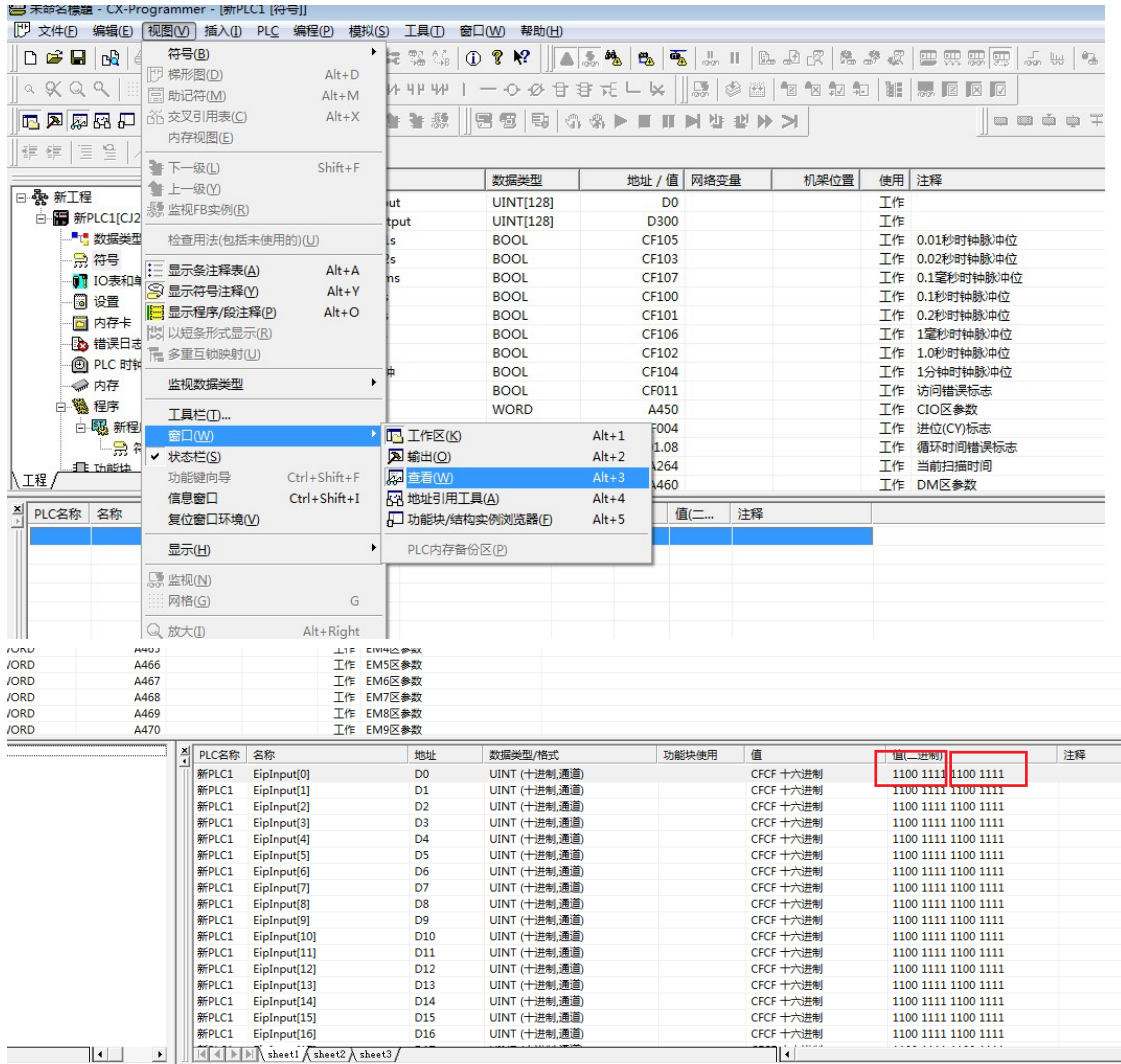


2) In the program editing interface, there is only one statement assigning a value to EipOutput[0], which is not actually used (the programming interface cannot be empty). EipOutput serves as the output interface of the Omron PLC. Users can use this interface to communicate with the robot controller based on their actual requirements. After the previous network configuration, communication has been successfully established.



3) Open the monitoring window and copy EipInput to the monitoring window. You can see that the data in the EipInput array is constantly changing. Since it uses a two-byte addressing, the range of variation is

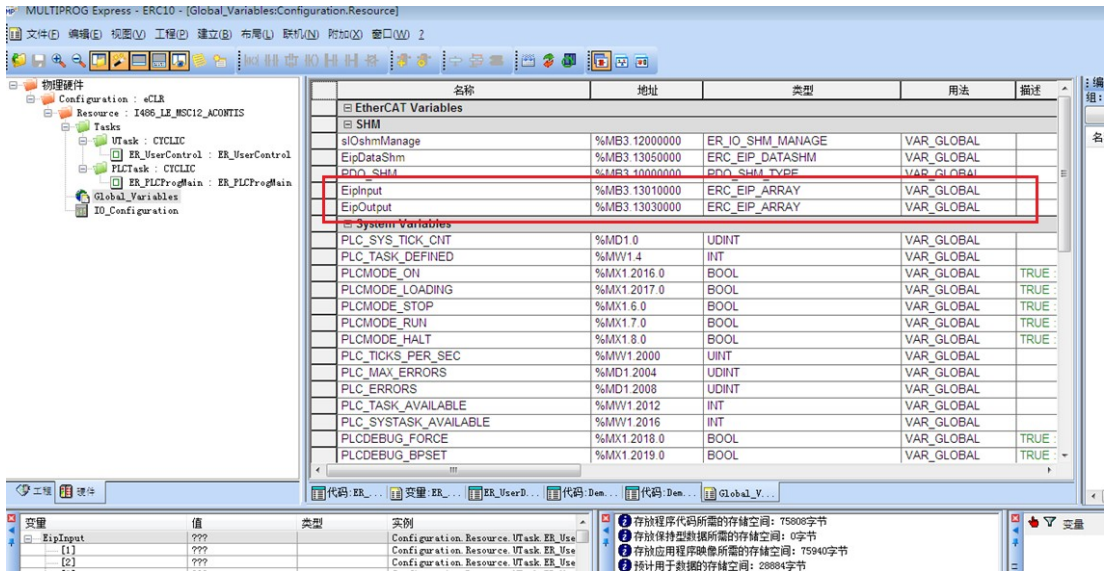
from 0x0000 to 0xFFFF. If you observe the low 8 bits and high 8 bits of a specific data, you will notice that they change in the same way, cyclically accumulating. This behavior aligns with the logic on the controller side, indicating that communication is functioning properly.



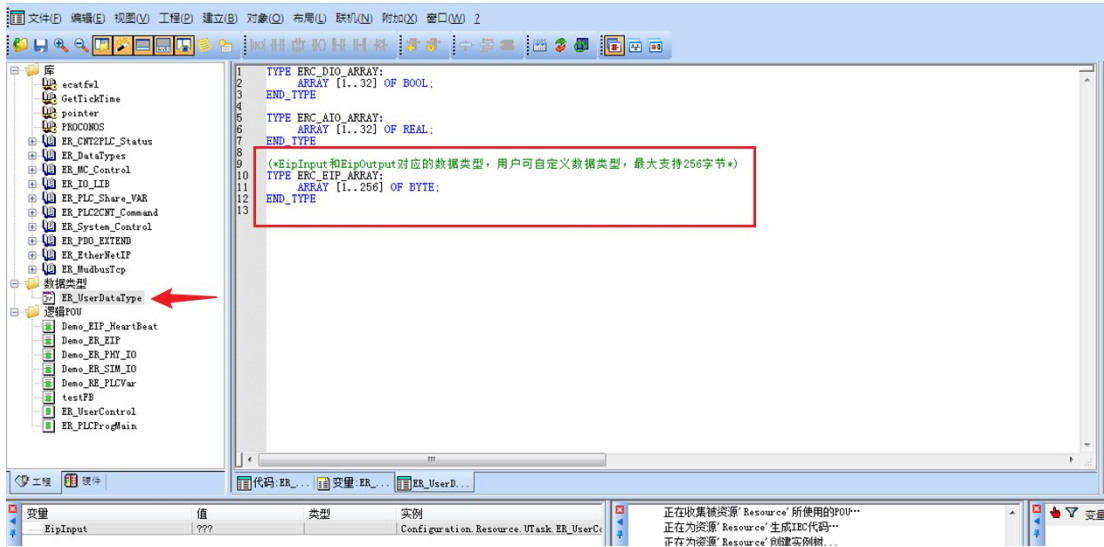
3.2 EtherNet/IP Adapter

Perform the following operations under the condition that the robot control system has started normally:

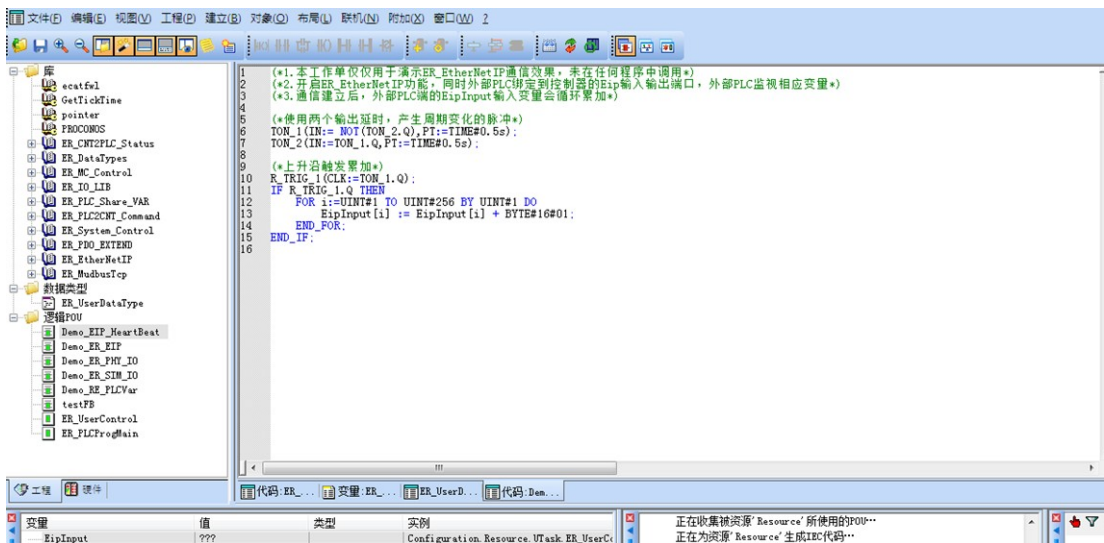
1. Open the Multiprog project and configure the EtherNet/IP functionality (refer to section 2.2 for functional configuration). After enabling the EtherNet/IP functionality, proceed with the following steps.
2. Open the Multiprog project and locate the EtherNet/IP data area (the data type of this area can be defined by the user). Users can write their own code to implement the desired functionality in the User Control Task or create a new Task.



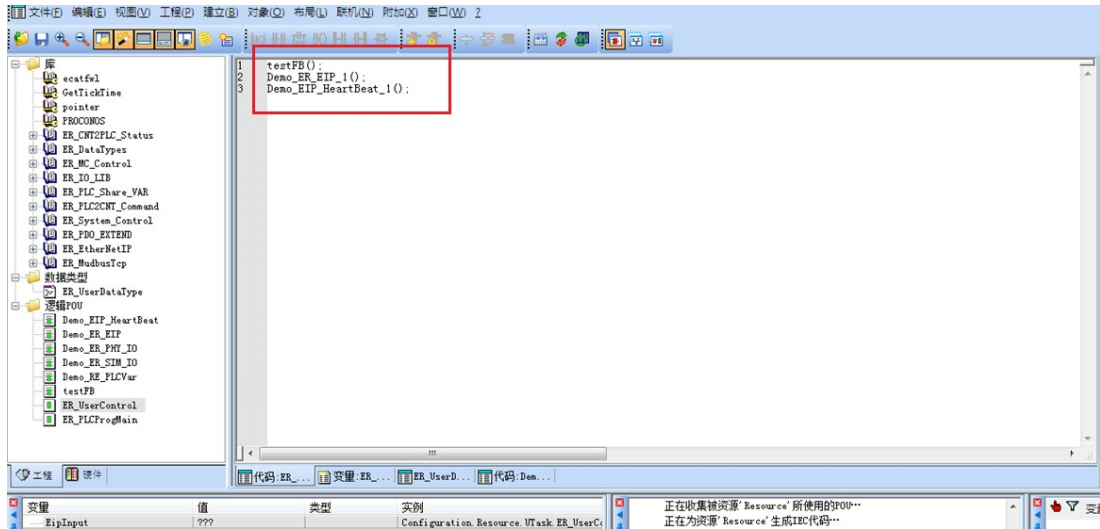
In this example, the ERC_EIP_ARRAY is an array of BYTE type with a size of 256 bytes (this data type can be defined by the user, and the data definitions are in ER_UserDataType), as shown in the figure:



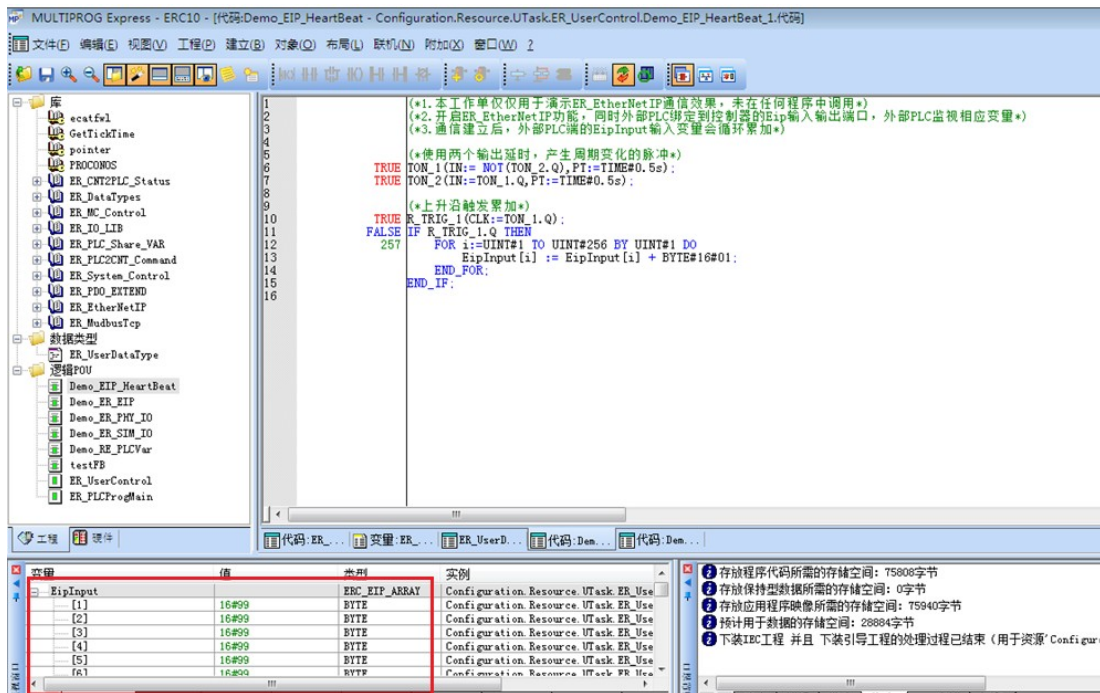
The main logic of Demo_EIP_HeartBeat is as follows:



Demo_ER_EIP and Demo_EIP_HeartBeat are called by default in ER_UserControl. After establishing communication, users can customize the logic and use EipInput and EipOutput to send and receive data.



3. After compiling and debugging the project, download it to the controller. You will see that the output data cyclically accumulates and changes (the figure below shows the monitoring effect on the Multiprog side). If communication with an external PLC is successfully established at this point, the corresponding ports will also show changes.



Chapter 4 Configuration of EIP

Communication Data Size

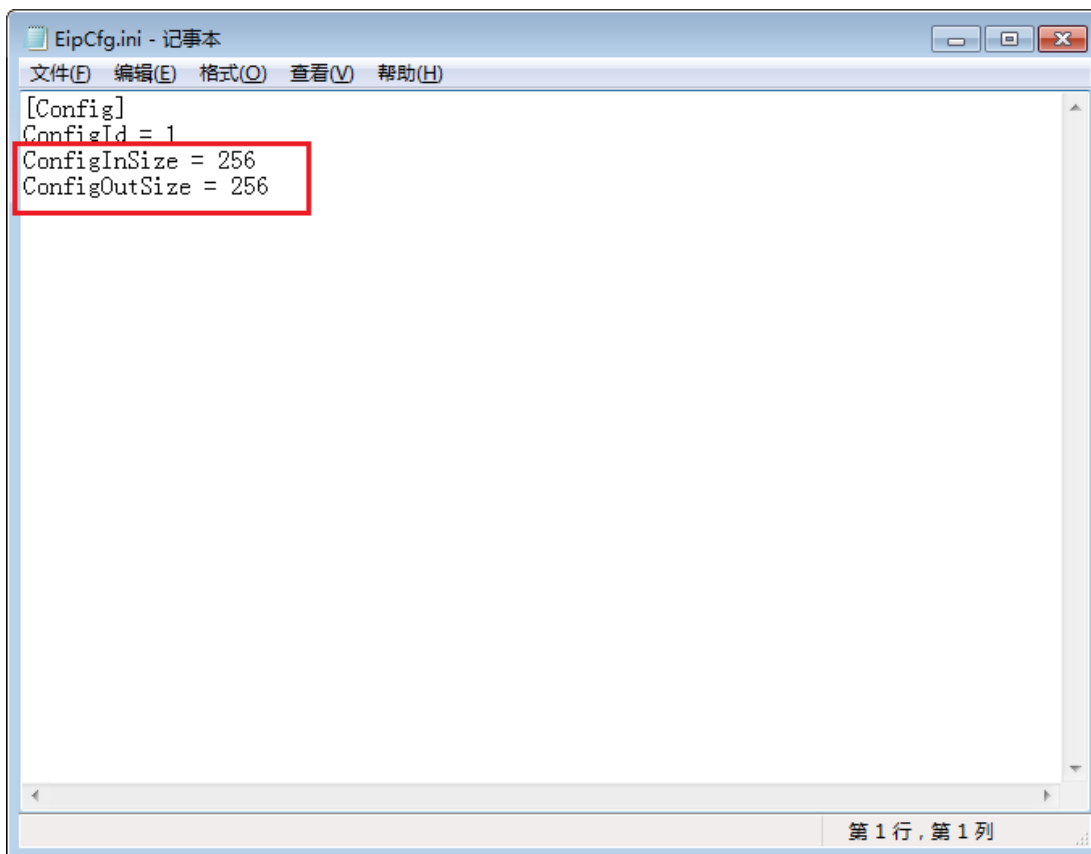
Taking into account actual working conditions, it is not necessary to transmit 256 bytes of data when using EtherNet/IP communication. The EtherNet/IP functionality supports the modification of communication byte size.

In this section, we will continue to use the Omron PLC PA202, CJ2M CPU31 as an example of an EtherNet/IP Scanner. The following are the steps to modify the communication byte size of EtherNet/IP.

4.1 Modifying the EipCfg.ini configuration file

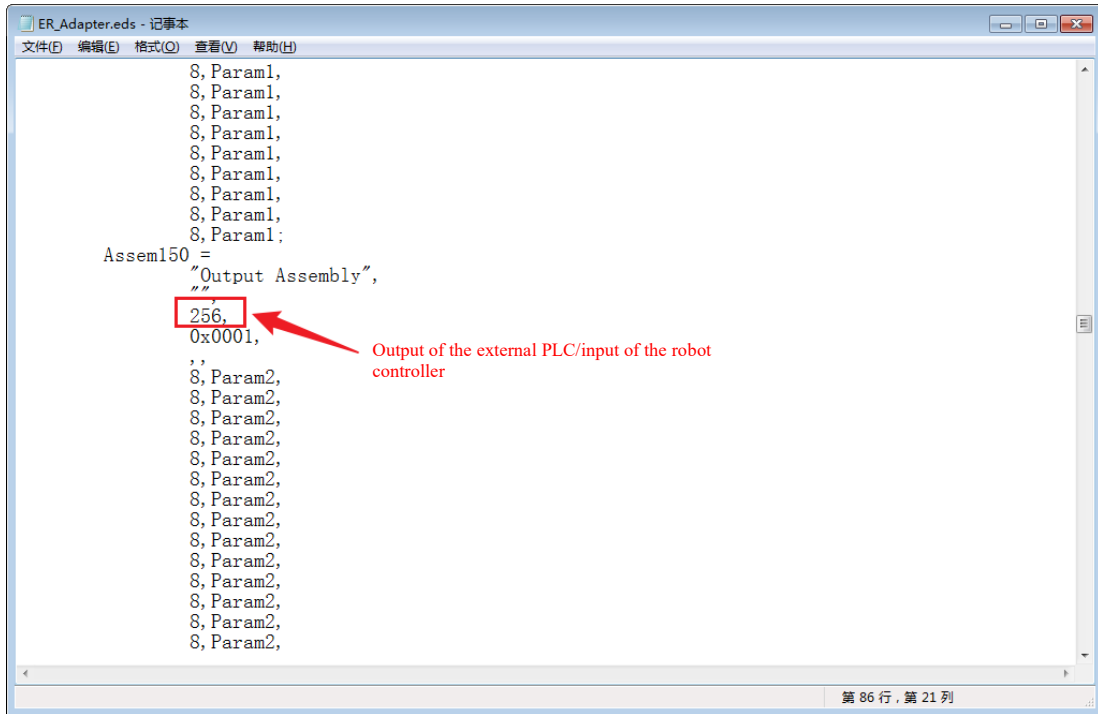
Modify the EipCfg.ini configuration file located in runtime -> rtk -> EipAdapter.

Open the EipCfg.ini configuration file as shown in the figure below. The parameters ConfigInSize and ConfigOutSize under the [Config] section represent the input and output byte sizes (here, "In" and "Out" refer to the direction from the external PLC).



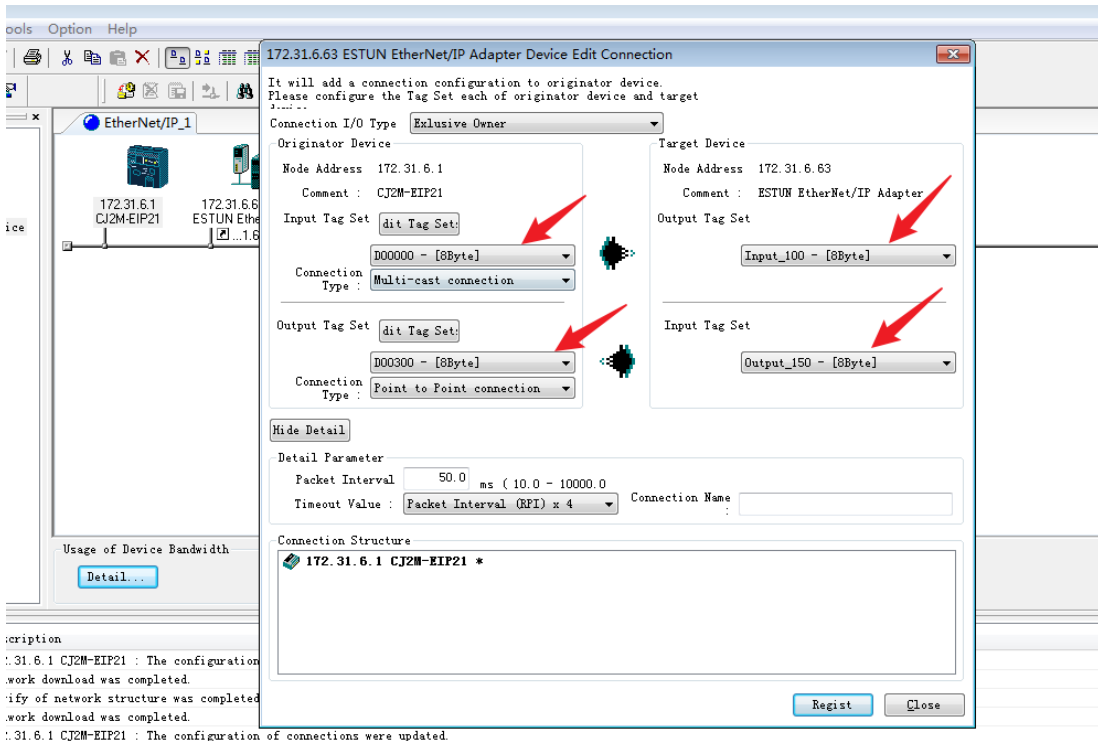
Taking the example of modifying it to 8 bytes (the byte size can be any integer value between 1 and 256), change the assigned values of the two parameters in the above figure to 8. For example, ConfigInSize=8 and ConfigOutSize=8. Save the document and restart the controller.

4.2 Modifying EDS files



After modifying the EDS file, reload it. During network configuration, you will see that the corresponding interfaces have also changed. Additionally, you need to modify the byte size mapped by the external PLC; otherwise, an error may occur. In the figure, it is modified to 8 bytes.

Once the modifications are completed, you can download the changes to the PLC (as described in section 3.1).



(Note: Some PLCs may not require loading the controller's EDS file. In such cases, modifying the configuration parameters in the EipCfg.ini file is sufficient.)