SUSTAINABLE DEVELOPMENT GROUP

SHARE WELL

PowerNex

Software Manual

SUZHOU SHARE WELL AUTOMATION TECHNOLOGY CO.,LTD

catalogue

catalogue

catalogue

Safety Precautions [Please Comply Absolutely]

This manual is the basic version of the software user manual, and subsequent versions may contain changes to specifications and operations. These changes may include minor adjustments or significant revisions, as well as entirely new chapters and modules not included in this manual.

To enhance the reliability, design, and functionality of the product, any changes to the information in this manual will be made without prior notice. The information in this manual does not constitute a commitment by the manufacturer. The manufacturer shall not be liable for any direct, indirect, special, incidental, or consequential damages arising from the use of the product or documentation, even if the possibility of such damages has been communicated.

Safety Instructions

To prevent injury to individuals and damage to property, the following statements are made regarding mandatory compliance:

Distinctions and explanations regarding the degree of harm and damage that may arise from improper use of this product:

 Danger: This symbol indicates content that could likely result in death or serious injury.

 This symbol indicates content that could likely cause injury or property damage.

Compliance with the following matters is indicated using the graphical symbols

below:

 This graphical symbol indicates actions that must not be performed.

 This graphical symbol indicates actions that must be implemented.

For items that do not meet "Caution" or "Danger," but are still critical for users to strictly adhere to, relevant notes will be documented in the appropriate places.

safety instruction

Danger

- Do not use the product in wet areas, environments with corrosive or flammable gases, or near combustible materials.
- Do not place flammable materials around the controller.
- Before approaching the robot, ensure that it is in a disabled state, or that the controller is in emergency stop mode. In emergencies, failure to stop the robot in time may result in personal injury or equipment damage.
- When teaching within the robot's operating range, please adhere to the following: Do not enter the robot's movement range.;
- Follow the operational steps and requirements in the respective manuals.
- Consider contingency plans in case the robot suddenly moves toward your position; ensure that a safe retreat area is in place, just in case.;
- Accidental entry into the robot’s movement range or collision with the robot may lead to personal injury. Additionally, in the event of any abnormalities, immediately press the emergency stop button.

Attention

Before performing robot teaching operations, check the following items. If any anomalies are found, timely repairs or other necessary measures should be taken

- Verify that all wiring is correctly connected and ensures that it will not entangle or pull during the robot's movement.
- Check for any abnormal movements or vibrations and unusual noises from the robot during operation.
- Check for any abnormal movements or vibrations and unusual noises from the robot

during operation.

- Check for any abnormal movements or vibrations and unusual noises from the robot during operation.

- Check for any abnormal movements or vibrations and unusual noises from the robot during operation.

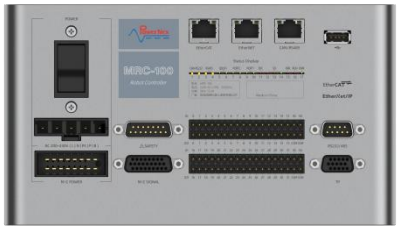
Introduction

This chapter introduces the PowerNexOS software, the installation of PowerNexOS software, and the system requirements for software operation. It is recommended that first-time users of this product read the "Safety Precautions" section before proceeding.

1.1 Overview of PowerNexOS Software

The robot system mainly consists of the human-machine interaction software PowerNexOS, controllers, and the robot body. PowerNexOS is one of the human-machine interaction software options. Compared to a teaching pendant, this software has advantages in programming and debugging. In contrast, the teaching pendant excels in operation and teaching functions. Both can be used in combination depending on the specific application scenario.

PowerNexOS software



PowerNex controller

PowerNex noumenon

1.2 Installation and Operation of PowerNexOS

1.2.1 Hardware Installation

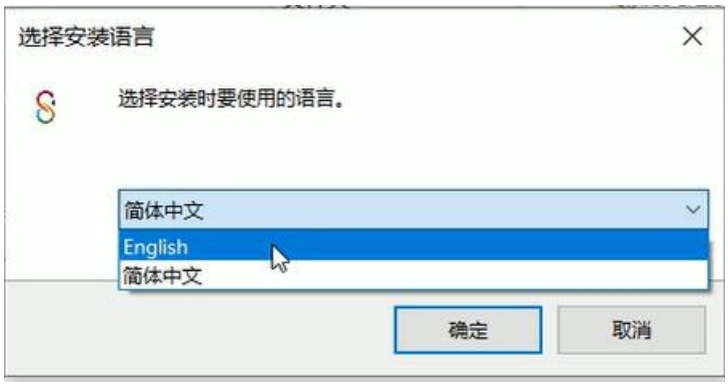
The PowerNexOS software is designed for the MRC-60 and MRC-100 controllers.

It is used to configure and debug the controllers and robots. You will need a personal computer with a Windows 7 or higher operating system, which can connect to the controller via Ethernet.

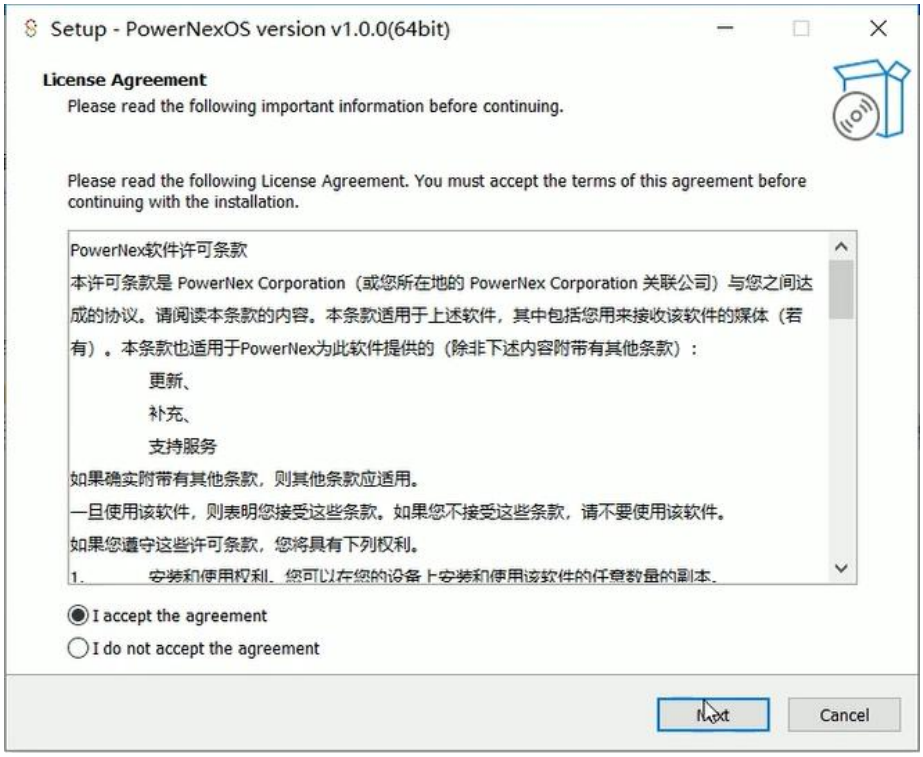
1.2.2 Software Installation

【Note】: It is recommended to disable the firewall during installation and operation to avoid installation or operational failures.

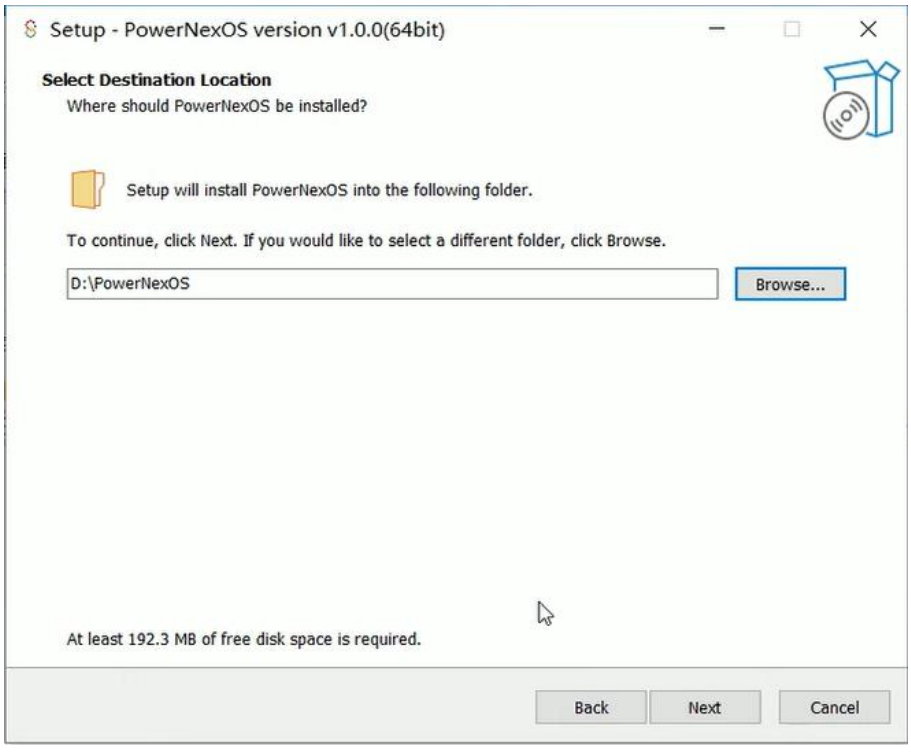
- Run the PowerNexOS.exe installation file, select the language for installation, and click "OK".



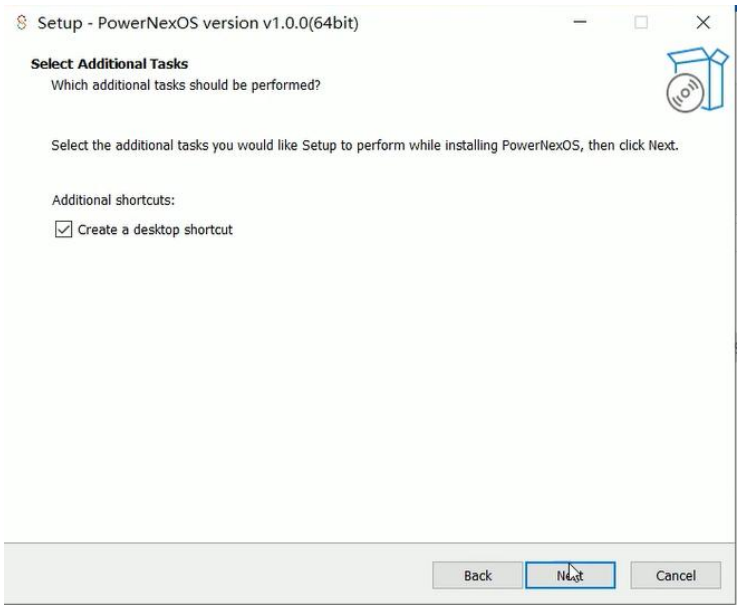
- Accept the installation license agreement and click "Next".



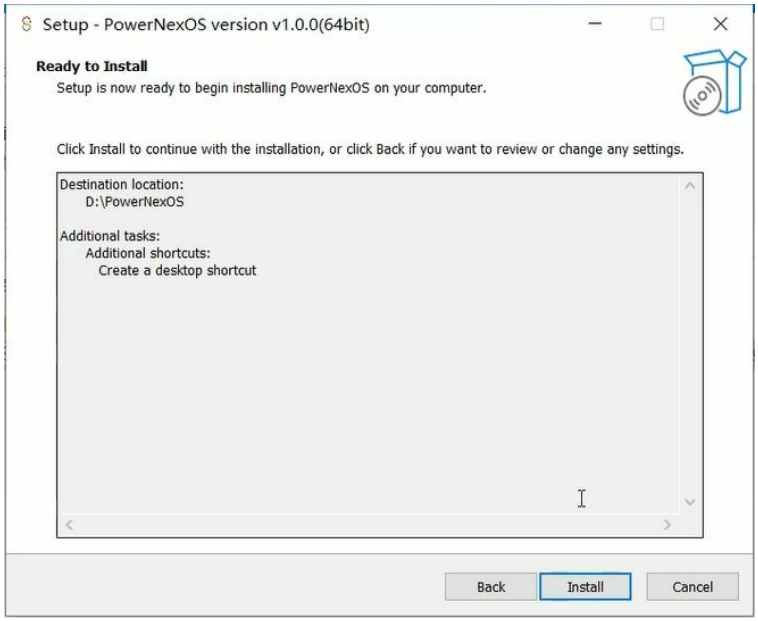
- Choose the installation path for the PowerNexOS software and click "Next".

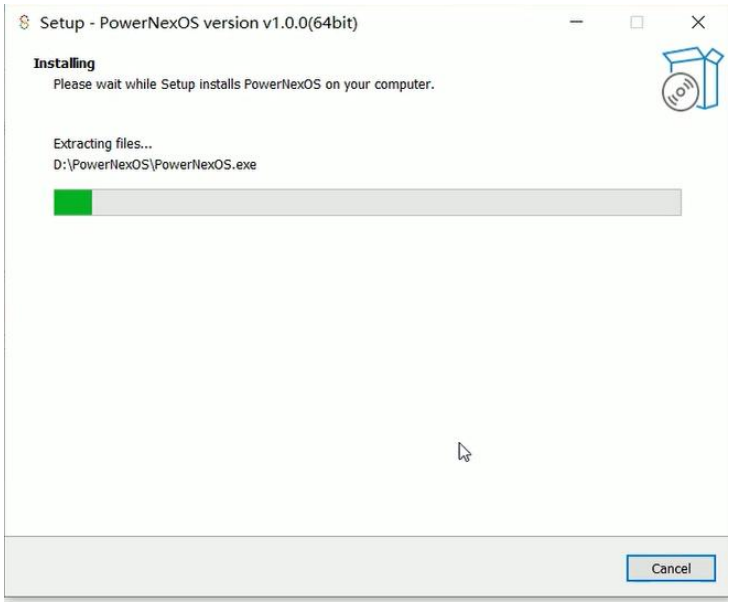


- Select whether to "Create a desktop shortcut" and click "Next" to proceed to the "Installation" page.



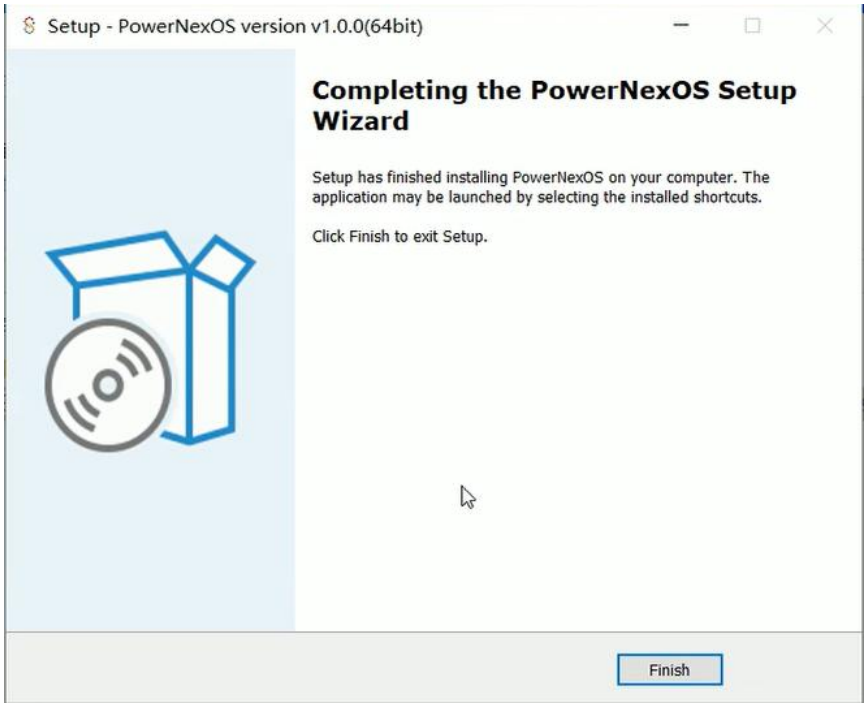
- Click the "Install" button to enter the "Installing" page.





【Note】: The installation process may take a relatively long time. Please be patient and wait for the installation to complete.

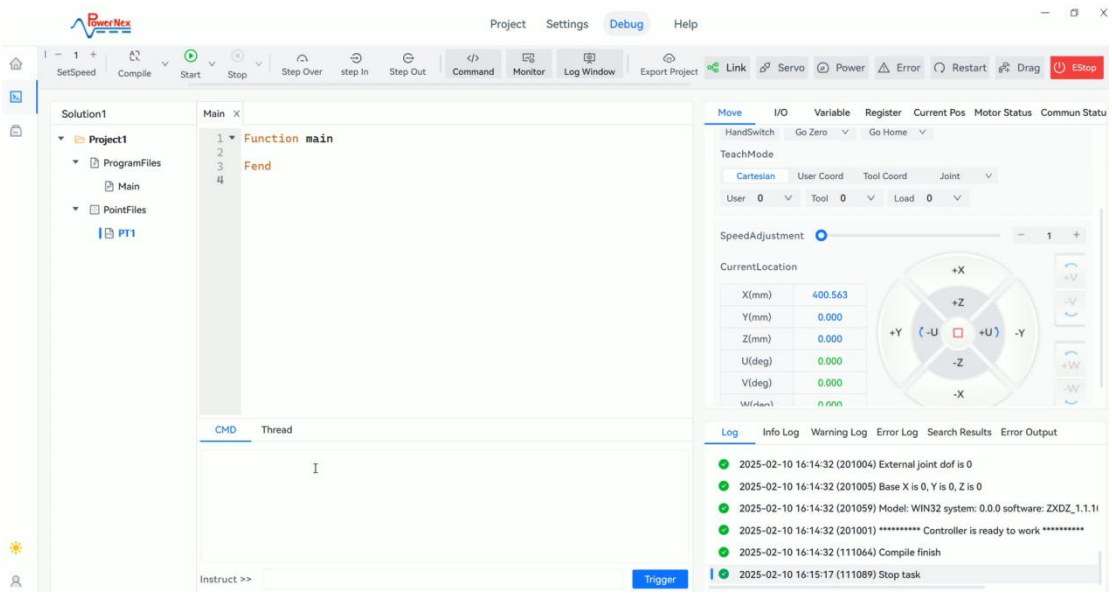
- Click "Finish" to launch PowerNexOS.



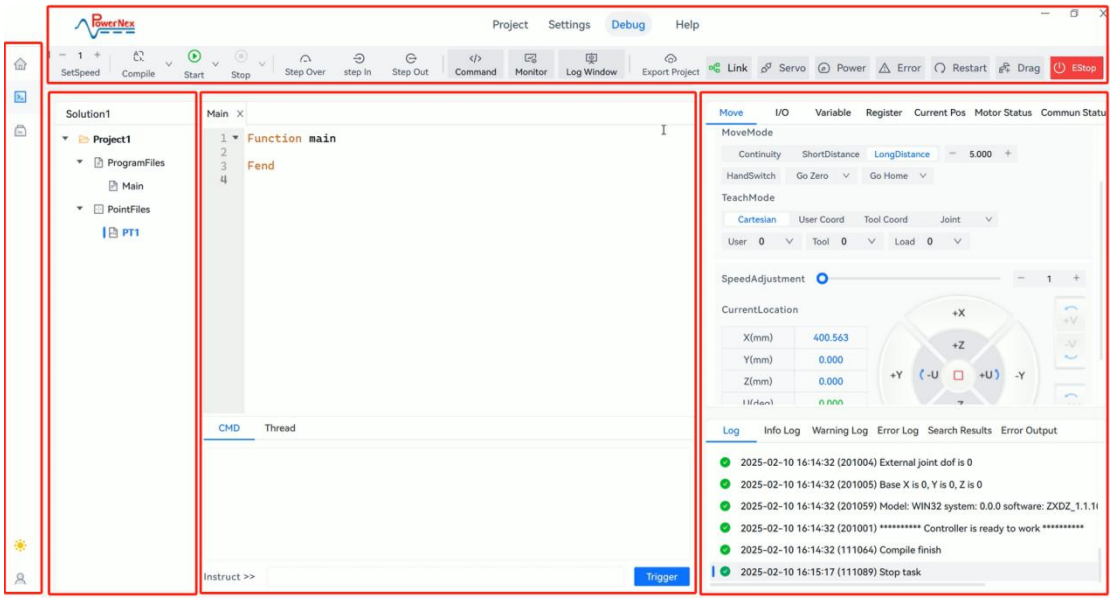
Chapter 1: Software Interface

1.1 Home Page

When you open the PowerNexOS software, the first screen that appears is the home page. You can select functions such as Home, Robot, and Process Package in the upper-left corner for debugging. In the project interface, you can create a project plan by selecting the "New Project" button in the function bar, or open a project by selecting a path to debug the program. In the process package interface, you can debug functions such as conveyor belt tracking and vision calibration. User permission login, theme switching, and settings are located in the lower-left corner of the interface, where you can perform permission login, interface theme switching, and functional settings. The connection button is located in the upper-right corner of the home page interface, allowing you to connect to the controller for on-site debugging of the robot or connect to a virtual robot for offline debugging. The servo button, power button, error button, restart button, and emergency stop button next to the connection button are used to control the robot's motor status and clear systematic errors.



1.2 Software Framework Introduction



The software interface is primarily composed of seven major frameworks:

Navigation Bar: Located on the left side of the software, it is used to switch between different sections of the software, such as Home, Robot, and Permission Switching modules.

Title Bar: Located at the top of the interface, it is used to switch between the main functional sections of the software.

Menu Bar: Located below the menu bar on the left, it displays the functional buttons under the selected menu.

Status Bar: Located below the menu bar on the right, it is used to view the status or access tools.

Project Area: Located to the right of the navigation bar, it is used to manage the project plans opened in the software.

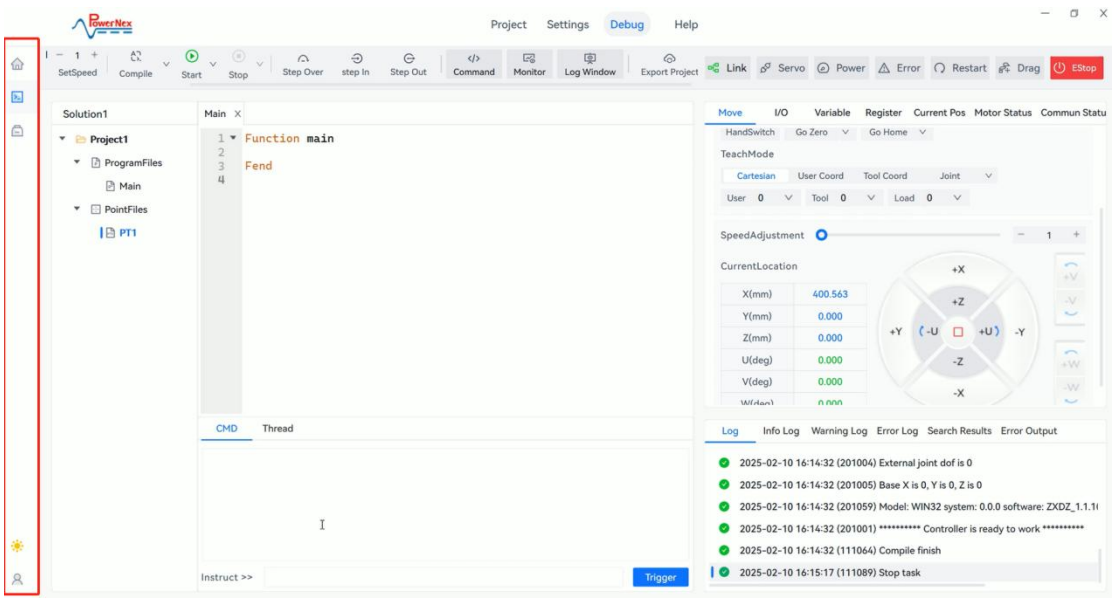
software interface

Program Area: Located in the middle of the software interface, it is used to write program files.

Auxiliary Area: Located on the right side of the software, it is used to monitor the current position status or signal monitoring of the robot during reading or debugging.

1.2.1Navigation Bar

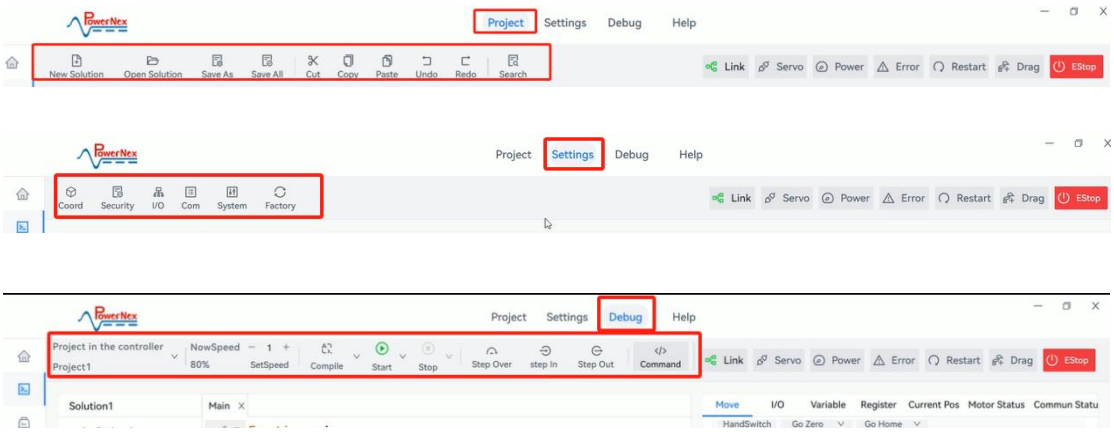
The Navigation Bar is located on the left side of the software interface. It is used to switch between different sections of the software, such as Home, Robot, Process Package, Theme Switching, and Permission Switching modules. This bar allows users to quickly navigate and access the editing and debugging tools of each module, thereby improving efficiency and convenience.



1.2.2 Title Bar）、Menu Bar

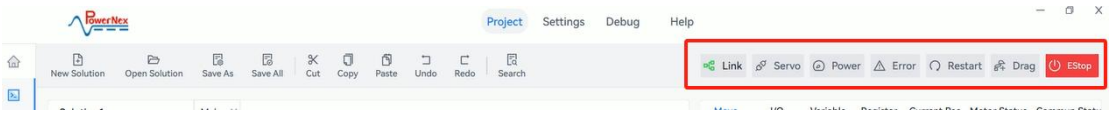
The Title Bar is situated at the very top of the interface and is used to switch

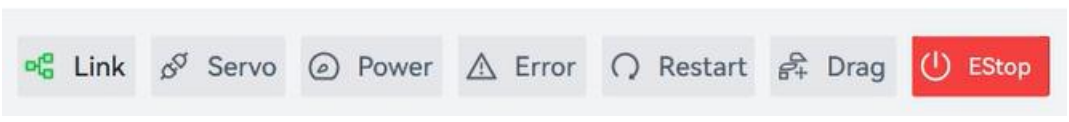
between the main functional sections of the software, such as "Project," "Settings," "Debug" and "Help." The Menu Bar is located below the Title Bar on the left side. It displays the detailed function options after selecting a title, used to perform specific operations or access particular functional modules.。



1.2.3 Status bar

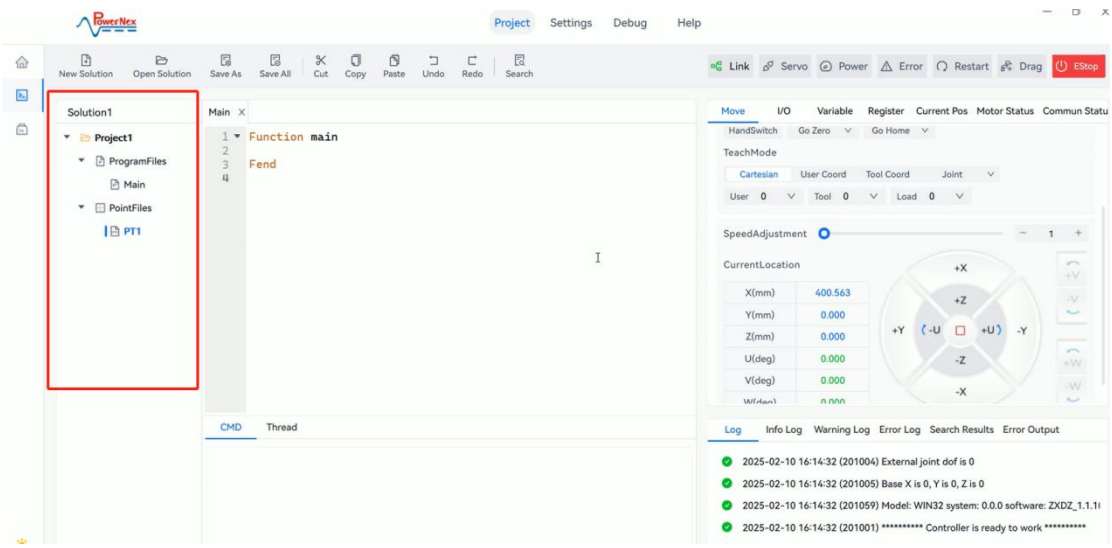
The Status Bar is located at the lower-right corner of the interface. The content in this area remains unchanged when switching between different pages of the Menu Bar. It displays the status and tools that need to be monitored constantly, such as the robot's connection status, motor power-on status, high/low power status, alarm status, as well as functions for restarting the controller, dragging, and emergency stop.





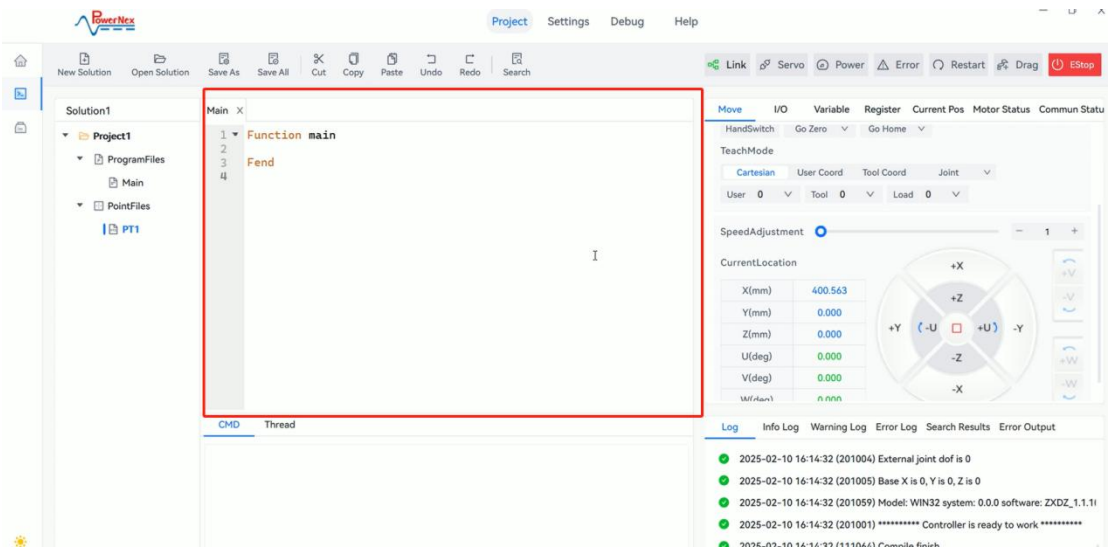
1.2.4 Project area

The Project Area is located to the right of the Navigation Bar and is used to manage the engineering projects opened in the software. Multiple project plans can be stored within an engineering project. By right-clicking on a selected project, it can be set as the active project. A project includes program files and position files. Multiple program and position files can be stored in a single project, but it is important to note that the program file named "Main" must exist as the main program.



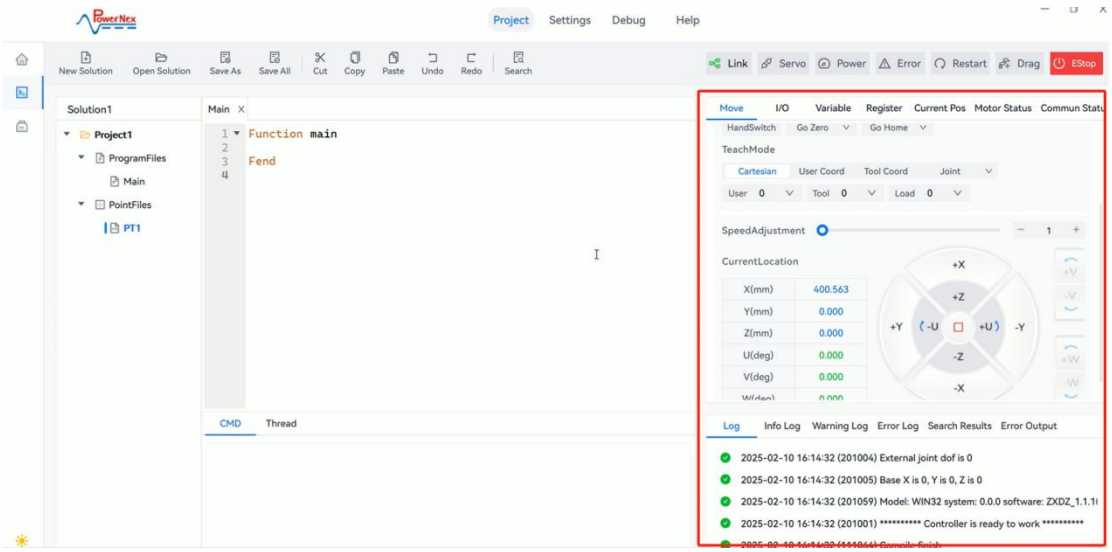
1.2.5 Program area

Double-clicking the Main file will display its content in the Program Area, where users can edit the content of the program file.



1.2.6 Auxiliary area

The Auxiliary Area is located on the right side of the software interface. It is used to monitor and debug the robot's jogging functions, I/O status, program variable monitoring, register status, motor status, and communication status. It also displays the software output logs, information, warnings, errors, and search results in the lower-right corner, efficiently assisting users in viewing prompt messages and locating query results.



1.2.3 Operating Environment

■ **Hardware Requirements**

Item	Minimum	Recommended
CPU	i3 or equivalent performance CPU	i5 or equivalent performance CPU and above
Memory	4GB	8GB and above
Disk Space	60GB (with more than 5GB free space)	200GB and above
Display Resolution	1024×768	1920×1080
Graphics Card	Supports OpenGL 2.0	Supports OpenGL 2.0 and above

■ **Requirements for Portable Version Software**

Item	Requirement
Operating System	Windows 7 and above
.NET Framework	.NET Framework 4.5 and above
VC++ Runtime	Microsoft Visual C++ 2008 Redistributable SP1 (x86) Microsoft Visual C++ 2010 Redistributable (x86) Microsoft Visual C++ 2012 Redistributable (x86)
Other	Microsoft Excel Application
Item	Requirement

■ **Requirements for Installed Version Software**

Item	Requirement	Item
Operating System	Windows 7 and above	Operating System
Other	Microsoft Excel Application	Other

1.2.4 Execution Permissions

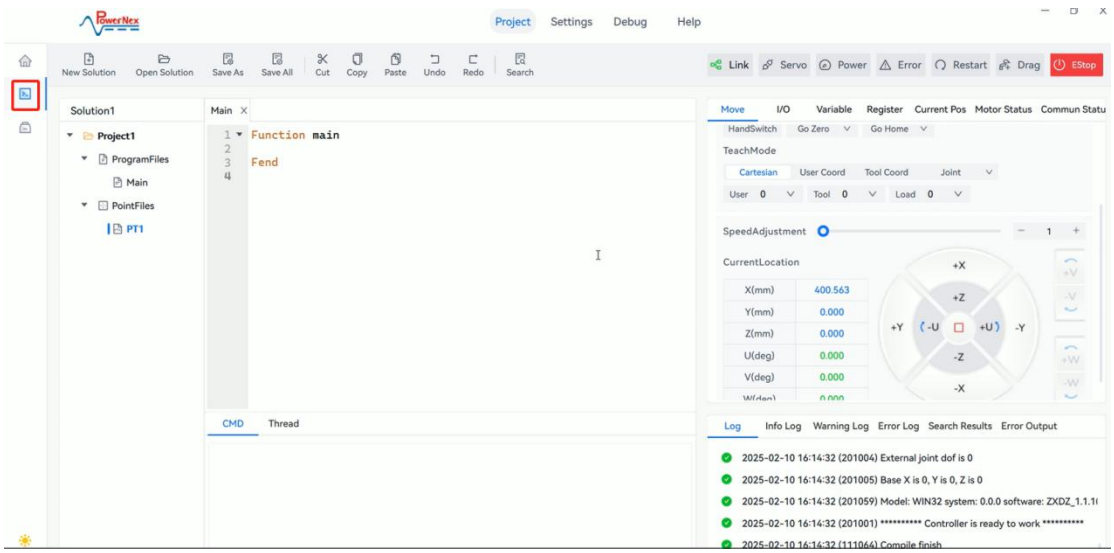
Users must run PowerNexOS with administrative privileges. Failure to do so will prevent connection to the virtual robot, but will not affect the connection and debugging of the controller.

Chapter 2: Navigation Bar Operations

2.1 Home

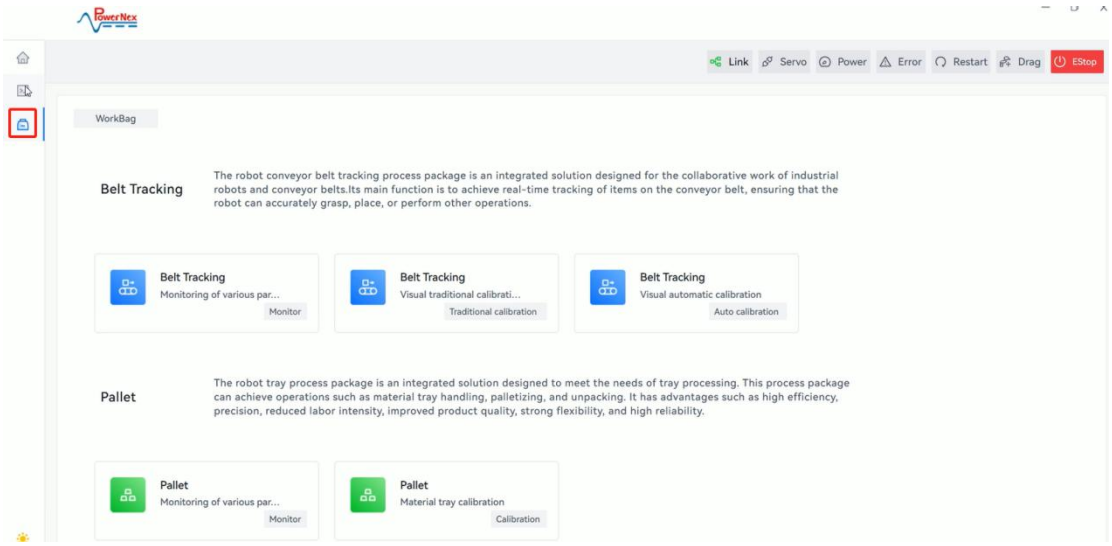
2.2 Robot

When the "Robot" button in the upper-left corner of the Navigation Bar is clicked, the interface switches to the debugging screen, enabling users to toggle between different interfaces.



2.3 Process Package

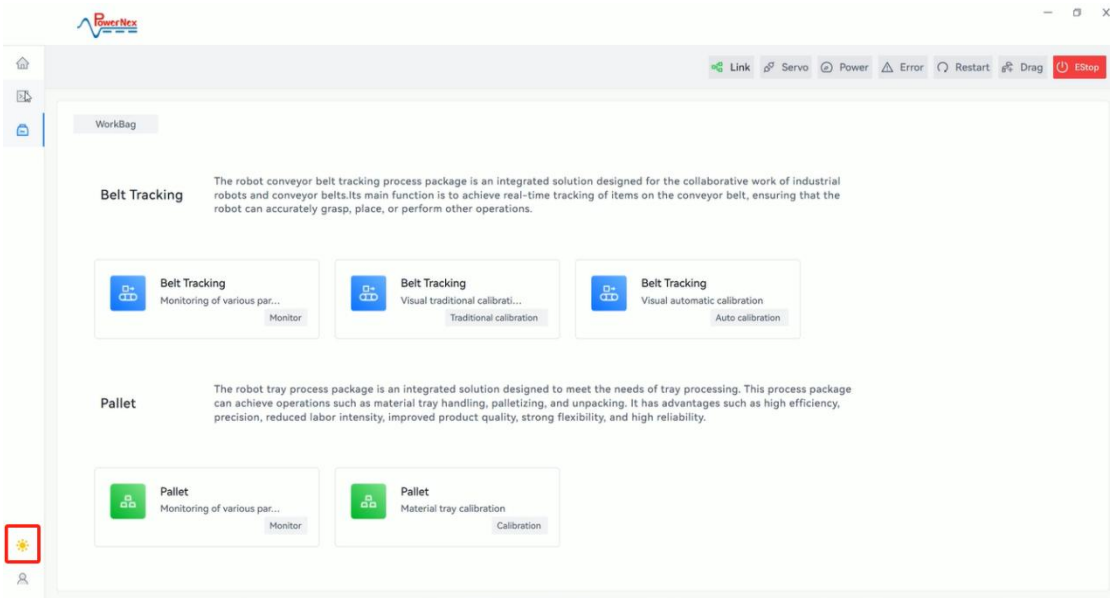
When the "Process Package" button in the upper-left corner of the Navigation Bar is clicked, the interface switches to the process package settings screen, enabling users to toggle between different interfaces.



2.4 Theme Switching

When the "Theme Switching" button in the lower-left corner of the Navigation Bar is clicked, the interface toggles between light and dark themes.

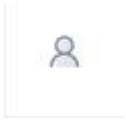
Suggestion: It is recommended to restart the software after switching themes.

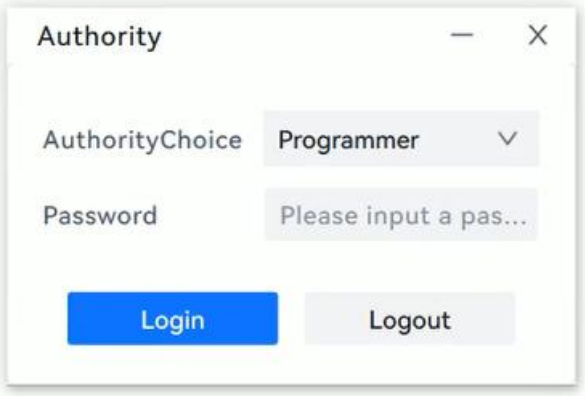


2.5 User Permissions

PowerNexOS supports three login permissions: Programmer, Administrator, and Manufacturer. Each permission level has different operational privileges, with the hierarchy being: Manufacturer > Administrator > Programmer.

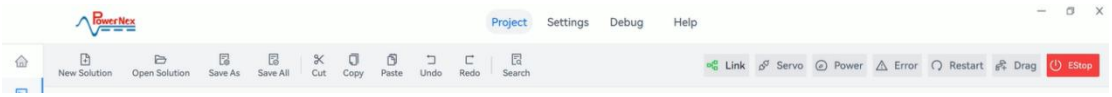
Operation Method:

Click the "Permissions" button in the lower-left corner.  In the pop-up window, select the desired permission level, enter the password, and click "Login" to switch to the corresponding user permission.



Chapter 3: Menu Bar and Function Bar Operations

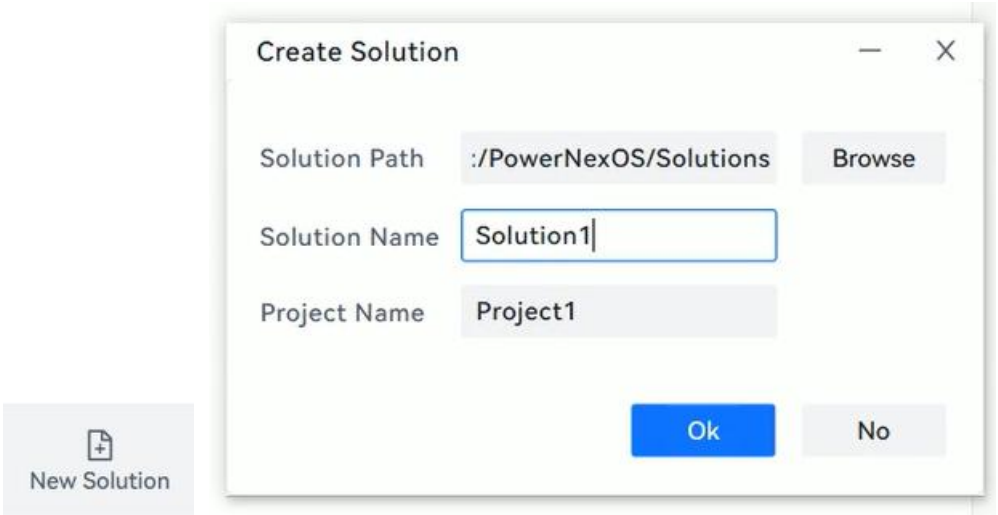
3.1 Project Interface



The Project Interface contains functional buttons related to engineering projects, such as New Project, Open Project, Save As, Save All, Cut, Copy, Paste, Undo, Redo, and Search.

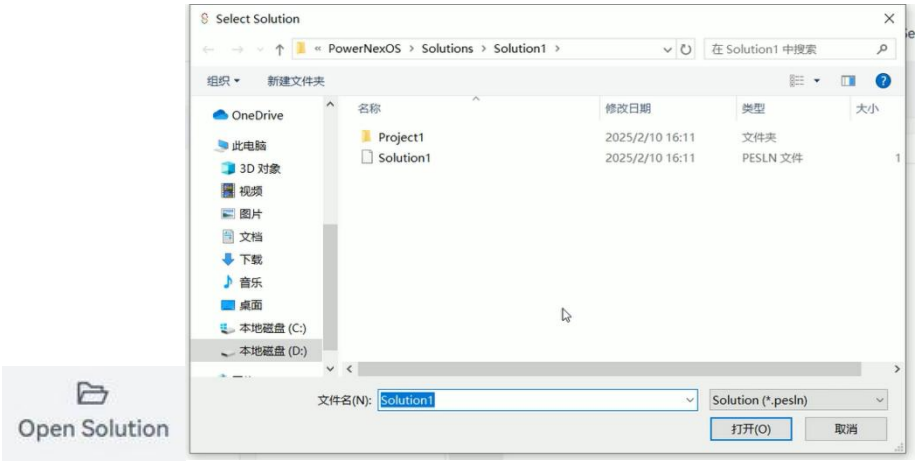
3.1.1 New Project

The first button in the Function Bar is "New Project." When clicked, a window will pop up. Users can select the file path on the computer where the project will be stored, customize the project name, and define the project title. After clicking "OK," the project file will be generated in the specified path, and the project information will be updated in the Project Area of the software.



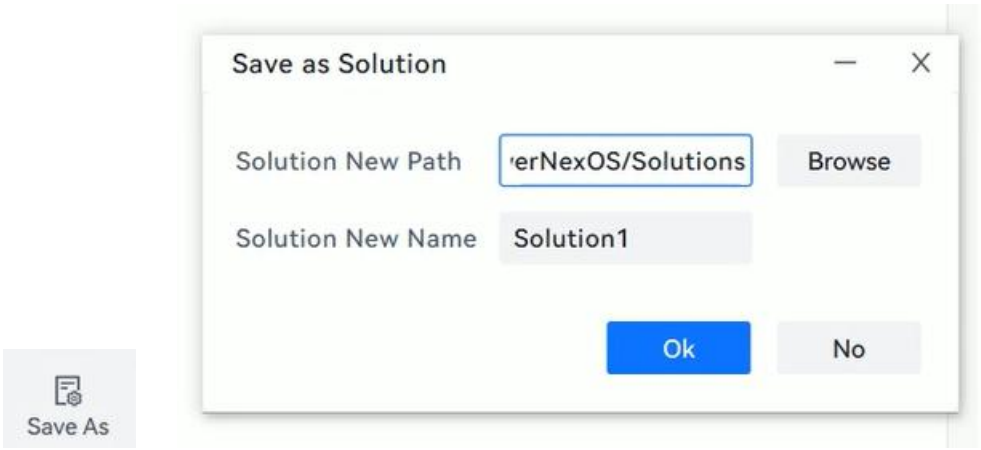
3.1.2 Open Project

The second button in the Function Bar, "Open Project," brings up a window when clicked. Users can select a project file stored on their computer to open. After clicking "Open," the Project Area in the software will update with the information of the opened project.



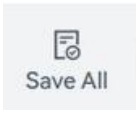
3.1.3 Save As

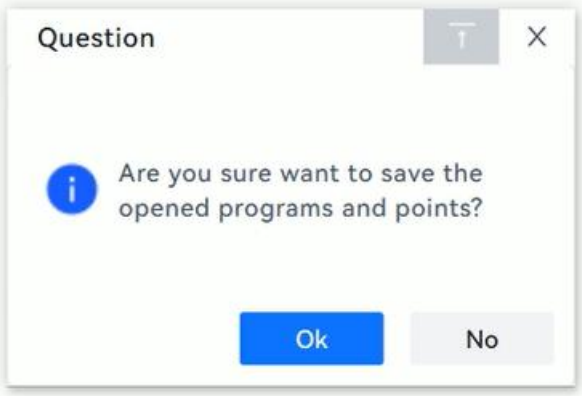
The third button in the Function Bar, "Save As," brings up a window when clicked. Users can select a new path on their computer to save the current project information and define a new project name. After clicking "OK," the current project information will be saved and closed. The software will automatically open the newly saved project file for users to debug or modify.



3.1.4 Save All

The fourth button in the Function Bar, "Save," brings up a prompt when clicked, asking users if they want to save the program files and position files within the project. If "OK" is clicked in the prompt, the files will be saved. If "Cancel" is clicked, the save operation will be aborted.





3.1.5 Cut

The fifth button in the Function Bar, "Cut," is used for modifying programs. When users select a single line or multiple lines of the program and click "Cut," the selected content will be placed on the clipboard. When "Paste" is clicked, the clipboard content will be inserted at the current cursor position. Shortcut: **Ctrl + X**.

3.1.6 Copy

The sixth button in the Function Bar, "Copy," is used for modifying programs. When users select a single line or multiple lines of the program and click "Copy," the selected content will be mirrored to the clipboard. When "Paste" is clicked, the clipboard content will be inserted at the current cursor position. Shortcut: Ctrl + C.

3.1.7 Paste

The seventh button in the Function Bar, "Paste," is used for modifying programs. When clicked, it inserts the content from the Windows clipboard at the current cursor position. Shortcut: Ctrl + V.

3.1.8 Undo

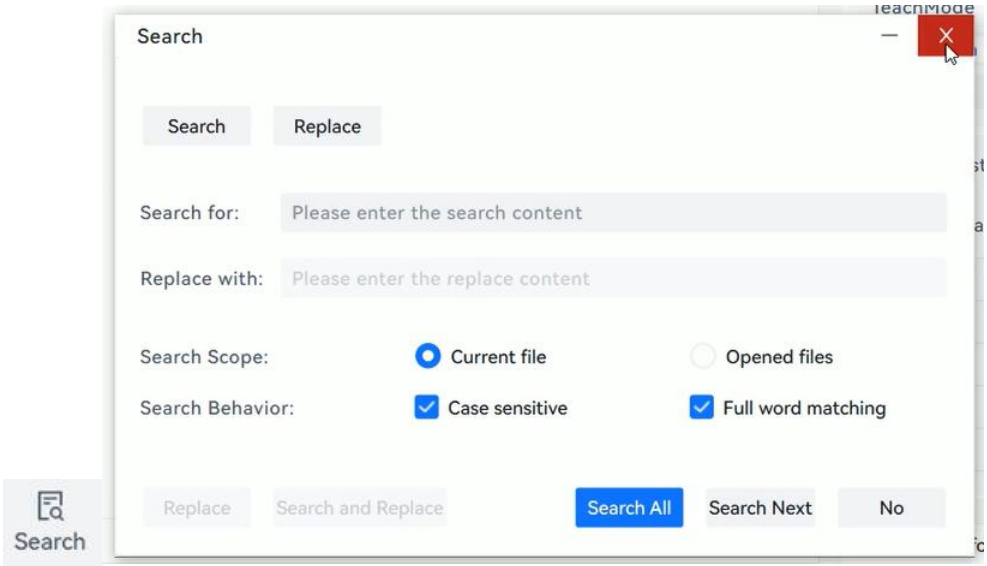
The eighth button in the Function Bar, "Undo," is used for modifying programs. When clicked, it reverts the last operation performed in the program area. Shortcut: Ctrl + Z.

3.1.9 Redo

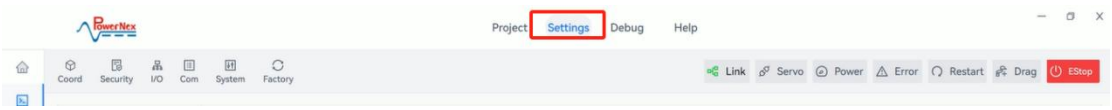
The ninth button in the Function Bar, "Redo," is used for modifying programs. If users have used the "Undo" function to revert an operation, they can use "Redo" to restore the undone content or operation. This function is invalid if no "Undo" operation has been performed.

3.1.10 Search

The tenth button in the Function Bar, " Search," brings up a window when clicked. It allows users to quickly search for or replace content within the program area. Users can choose the search scope, either "Current File" or " Opened Files," and can also select options such as "Case Sensitive" and "Full Word matching." Shortcut: Ctrl + F.

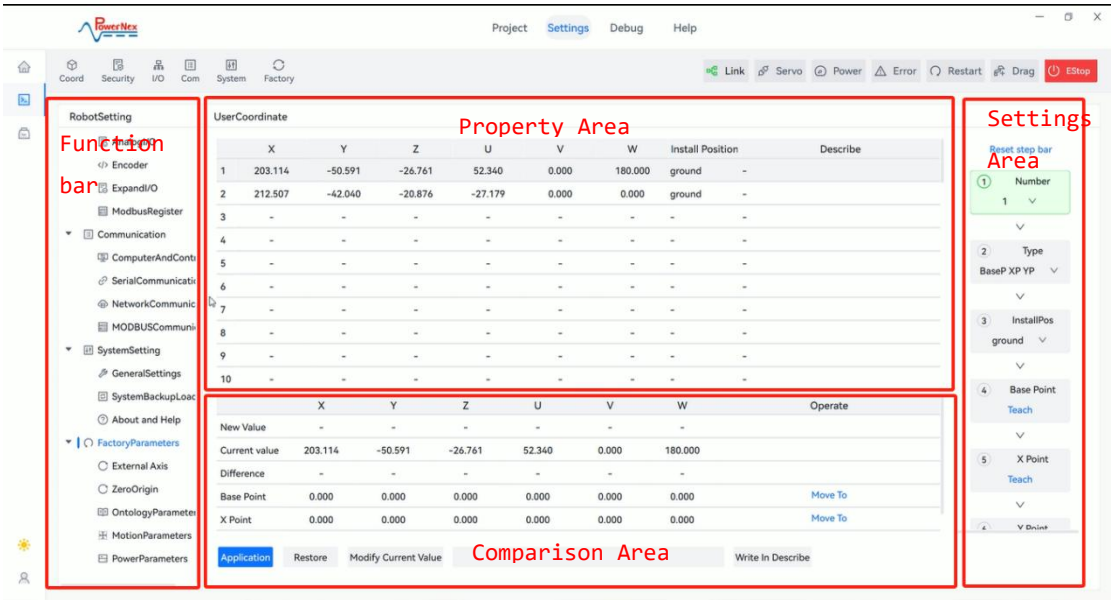


3.2 Settings Interface



The Settings Interface includes various debugging functions, such as coordinate systems, safety parameters, I/O, and communication.

3.2.1 Coordinate Systems



The first button in the Function Bar, "Coordinate Systems," allows users to select and configure user coordinate systems, tool coordinate systems, and payload settings from the left-hand function bar.

The interface is divided into the following areas:

Function Bar: Located on the left side, providing options for users to select the desired functions.

Property Area: Located in the middle, listing the function numbers and information for users to select.

Comparison Area: Located at the bottom of the middle section, allowing users to compare the current coordinate system settings with the previous data.

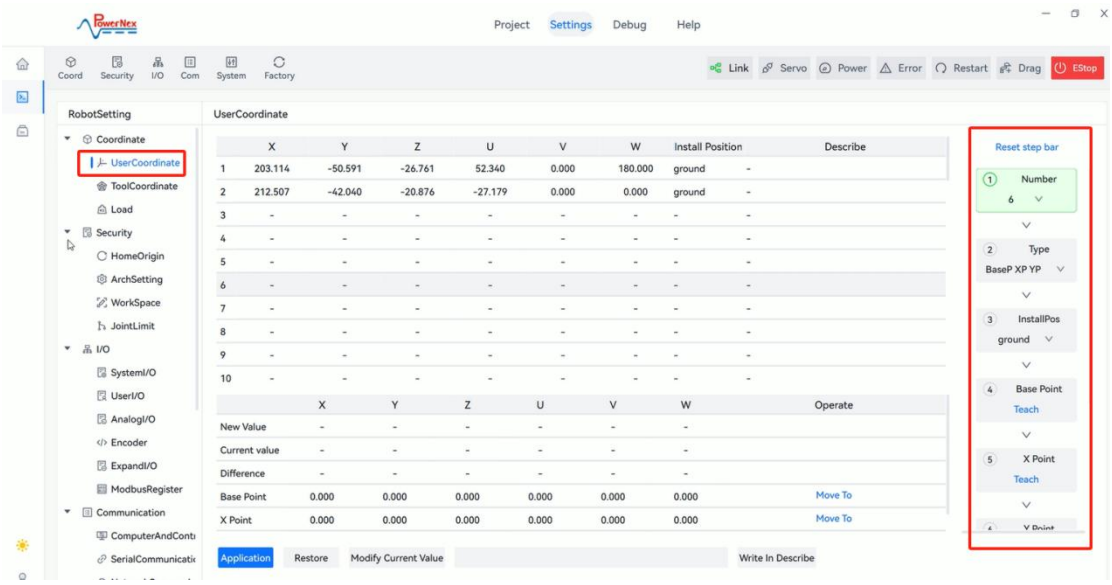
Settings Area: Located on the right side, where users can configure coordinate systems and payload information.

- **User Coordinate System**

The robot system supports up to 65 user coordinate systems. User 0 is the world coordinate system of the robot and cannot be modified. Users can define and set user coordinates 1 to 64.

Function of User Coordinate System: In the world coordinate system, with the robot base as the origin and the XY directions defined by the factory settings, the XYZU position of the robot's end-effector is calculated by comparing the end-effector's position with the base. The user coordinate system allows users to redefine the origin and XY directions. After switching to a user coordinate system, the current coordinate calculation is based on the user-defined origin and XY directions.

Teaching Method for User Coordinate System:



1. Select the number of the user coordinate to be defined.
2. Choose the method for defining the user coordinate:

   BaseP XP YP: Define the origin, X-direction point, and Y-direction point to create the user coordinate system using three points.

   XP1 XP2 YP: Define two X-direction points and one Y-direction point to create the coordinate system using three points.

3.Default selection is "ground" Other options are temporarily unavailable。

ground: Use the world coordinate system as the reference for teaching the user coordinate.

4. Define the origin or the first X-direction point according to the selected method. Click "Teach" to enter the jogging interface.
5. Define the first or second X-direction point according to the selected method.
6. Define the Y-direction point according to the selected method. Click "Teach" to enter the jogging interface.
7. Select whether the current coordinate system is tilted to define whether the user coordinate is planar.
8. Click "OK" to complete the teaching of the user coordinate.
9. Click the blue "Application" button to write the user coordinate information.

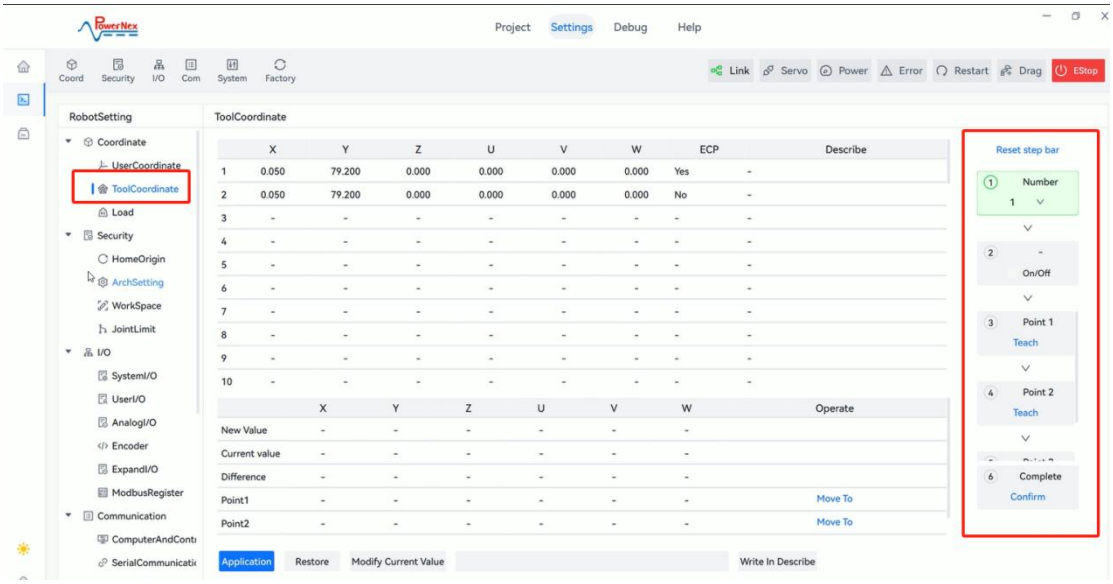
Note: When using the user coordinate system for movement, the reference position of XYZU may be changed! Pay attention to the motion trajectory planning.

- **Tool Coordinate System**

The robot system supports up to 65 tool coordinate systems. Tool 0 is the world coordinate system of the robot and cannot be modified. Users can define and set tool coordinates 1 to 64.

Function of Tool Coordinate System: In the world coordinate system, the tool center is defined as the position of the robot's end-effector. The current position of the end-effector is calculated by comparing it with the reference position.

Teaching Method for Tool Coordinate System:



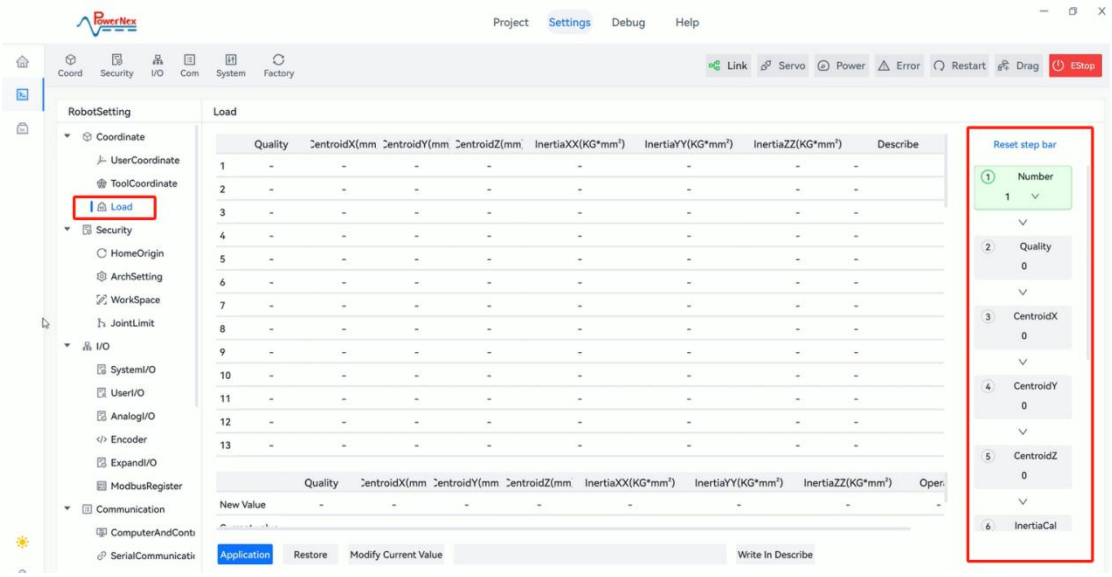
1. Select the number of the tool coordinate to be defined.
2. Choose whether to enable the external tool coordinate function.
3. After aligning the tool with a fixed reference point, teach Point 1 and click "Teach" to enter the jogging interface.
4. Rotate the tool approximately 120° clockwise or counterclockwise and use the jogging function to realign the tool with the reference point to teach Point 2. Click "Teach" to enter the jogging interface.
5. Rotate the tool another 120° in the same direction and realign it with the reference point to teach Point 3. Click "Teach" to enter the jogging interface.
6. Click "OK" to complete the teaching of the tool coordinate.
7. Click the blue "Application" button to write the tool coordinate information.

- **load**

The robot system supports up to 65 load options. Load 0 represents the no-load state and cannot be modified. Users can define and set load options 1 to 64.

Function of Load: The system adjusts motor parameters based on the load information entered by the user. It dynamically adjusts the motor torque and acceleration/deceleration ratios according to different motion paths. If the robot is loaded with an object but the load information is not set or enabled, it may cause motor overload alarms during high-speed motion.

Teaching Method for Load:



1. Select the load number to be defined.

2. Enter the mass (weight) of the load.

3. Enter the X-coordinate of the load's center of gravity. This data can be measured using a 3D drawing (optional for center of gravity and inertia).

4. Enter the Y-coordinate of the load's center of gravity. This data can be measured using a 3D drawing (optional for center of gravity and inertia).

5. Enter the Z-coordinate of the load's center of gravity. This data can be measured using a 3D drawing (optional for center of gravity and inertia).

6. Enter the inertia XX of the load. This data can be obtained through inertia calibration in Step 9 (optional for center of gravity and inertia).

7. Enter the inertia YY of the load. This data can be obtained through inertia calibration in Step 9 (optional for center of gravity and inertia).

8. Enter the inertia ZZ of the load. This data can be obtained through inertia calibration in Step 9 (optional for center of gravity and inertia).

9. Click "Calculate." A pop-up window will appear, prompting you to select a similar load model and enter the XYZ dimensions of the load.

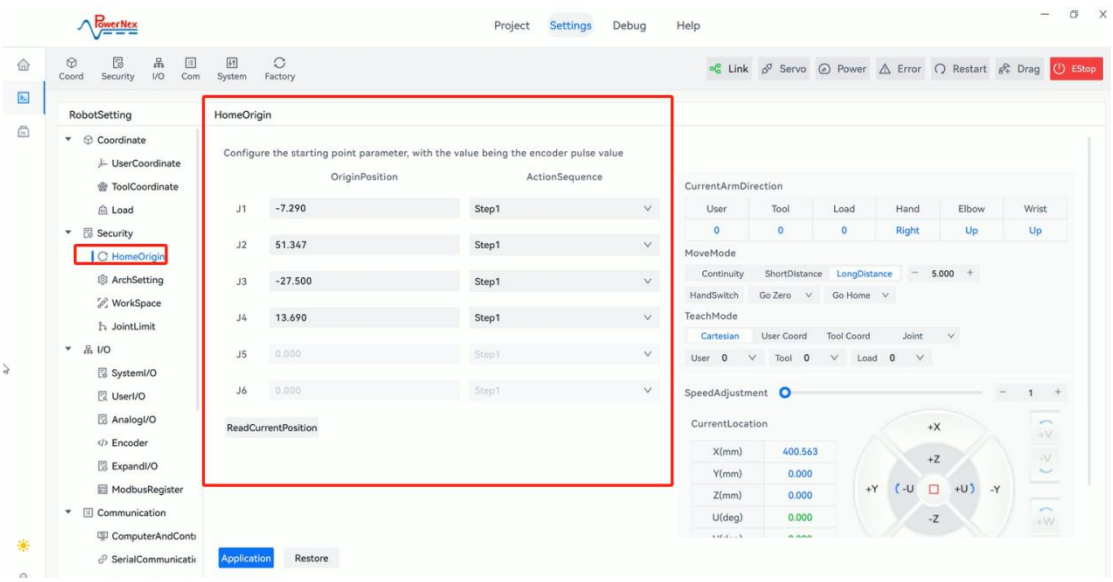
10. Complete the load teaching by clicking "OK."

11. Click the blue "Application" button to write the load information.

3.2.2 Security Parameters

- **Home Origin**

Select the Home Origin under Security in the function bar to define the standby position. Users can set the joint positions for the Home point and the joint movement sequence for returning to the Home point.



Method for Setting the HomeOrigin::

1. Move the robot to the desired Home point position and click "Read Current Position." The system will save the current position as the Home point.

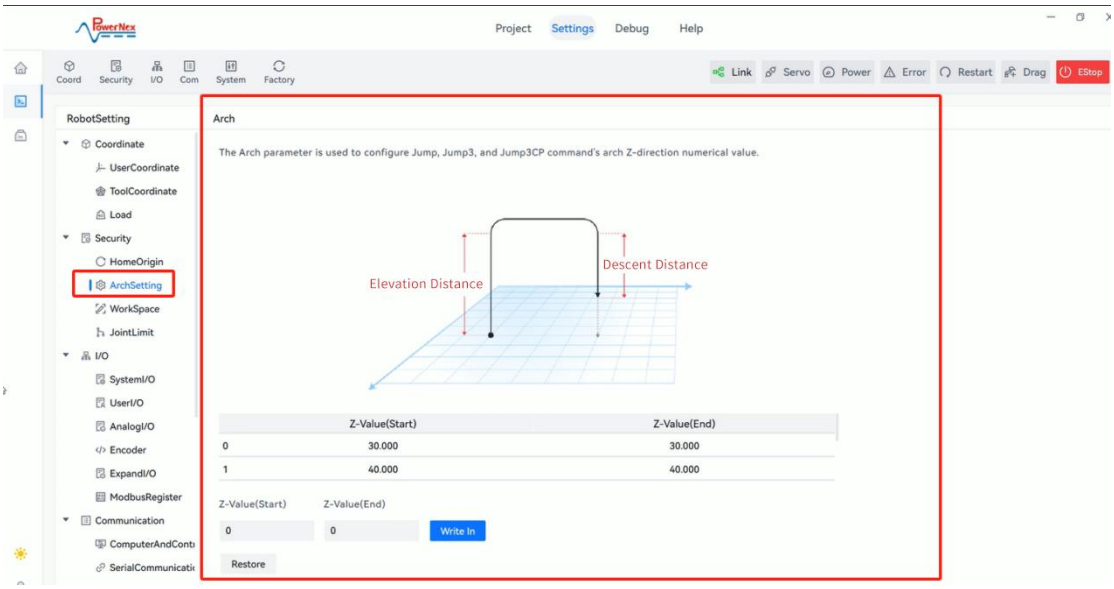
2. Set the movement sequence for each axis, ensuring that the axes move in ascending order (i.e., Axis 1 moves first, followed by Axis 2, etc.).

3. Complete the Home point setting by clicking "Application."

- **Arc Setting**

The arc movement command (JumP) requires defining the Z-direction value for the arc. If not defined, the default setting (Arc Setting 0) will be used. Therefore, users need to adjust the arc settings according to the actual application scenario.

Method for Arc Setting:



1. Select the arc number to be defined (0 to 7).

2. Enter the transfer distance value in the Z-Value (Start) input box (the distance in mm from the current position to start the transfer).

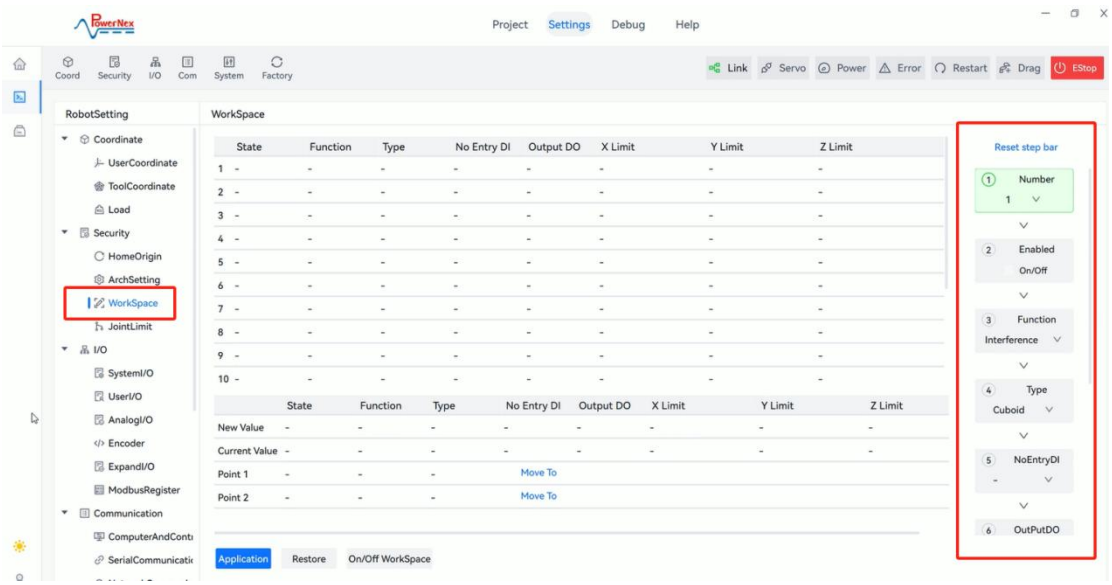
3. Enter the approach distance value in the Z-Value (End) input box (the distance in mm to start vertical descent when approaching the target point).

4. Complete the arc setting by clicking "Write in."

- **Workspace**

The robot system supports up to 16 workspaces for user configuration. The primary function of a workspace is to define a specific area within the robot's range of motion and set its behavior, such as restricting robot movement within or outside the defined area.

Method for Workspace Setting:



1.Select the workspace number to be defined (1 to 16).

2.Enable or disable the current workspace.

3.Choose the function of the current workspace:

Interference Area: Outputs a user-specified signal when

the robot's end-effector enters the area.

Workspace: Restricts the robot's end-effector to move only within the defined area. Motion stops and an alarm is triggered if the robot exceeds the defined range.

Forbidden Zone: Allows the robot's end-effector to move freely within any range, but stops motion and triggers an alarm if it enters the defined forbidden zone.

4.Select the type of space to teach (cuboid or cylinder). The teaching methods differ: a cuboid is defined by two diagonal points, while a cylinder is defined by a center point, radius, and height.

5.Set the DI pin to prohibit entry into the space. If the current space is set as an interference area or workspace, the robot will not enter the space when the specified DI pin is active. A hyphen indicates no restriction.

6.Set the DO pin to output a signal when the robot enters the defined space. A hyphen indicates no signal output.

7.Define the first diagonal point for a cuboid or the center of a cylinder.

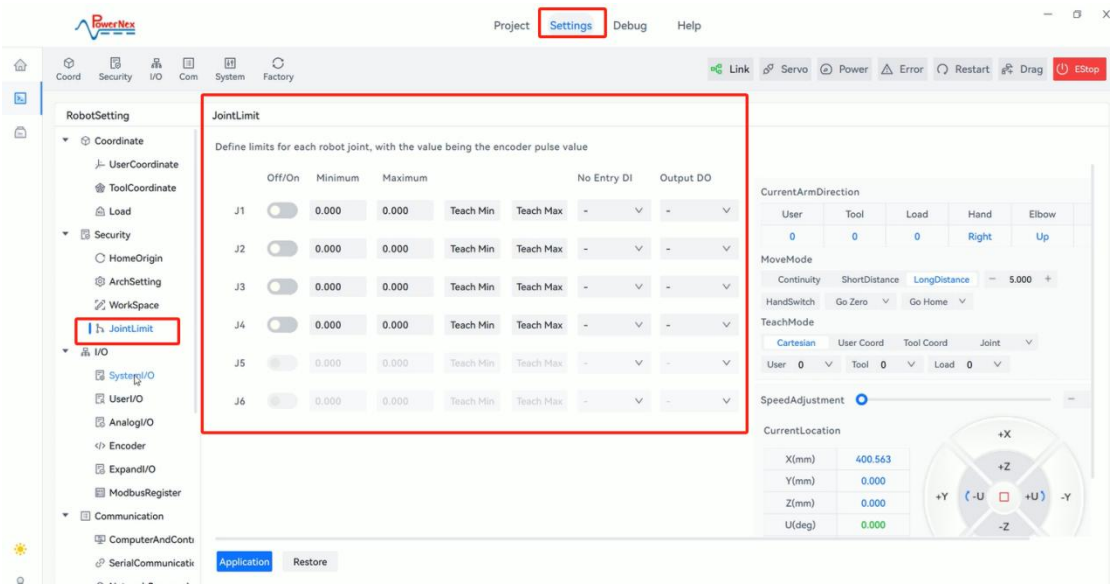
8.Define the second diagonal point for a cuboid or the radius of a cylinder.

9.Complete the cuboid definition or set the height of the cylinder.

10.Complete the workspace teaching by clicking "Finish"

- **Axis Limitation**

The robot is shipped with predefined soft limits for each axis to prevent collisions. However, if the predefined soft limits still pose a high risk of collision in actual applications, users can set secondary limits within the predefined range to further restrict the axis movement.



On/off: Whether to enable secondary limitation for the axis.

Minimum Value: The smallest angle the axis can move to.

Maximum Value: The largest angle the axis can move to.

No Empty DI: The limitation takes effect when the signal is input. If the option is set to "-", it is enabled by default.

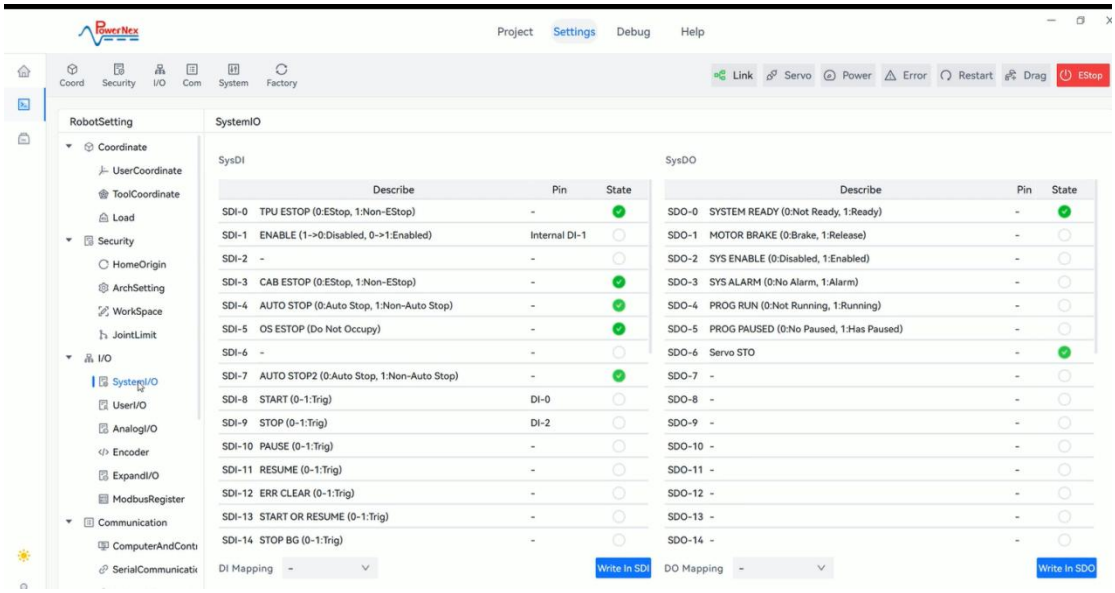
Output DO: Outputs a specified DO pin when the axis reaches the minimum/maximum value.

3.2.3 I/O

- **System I/O**

System I/O is used to display the status of system functions, rather than the actual wired functional pins. Users can configure the actual wired pins (User I/O) in the desired System I/O to trigger system functions via buttons.

Method for Setting System I/O:



1.Take System DI as an example; the same applies to System DO.

2.Select the required System I/O function, for example, SDI-12 Alarm Clear.

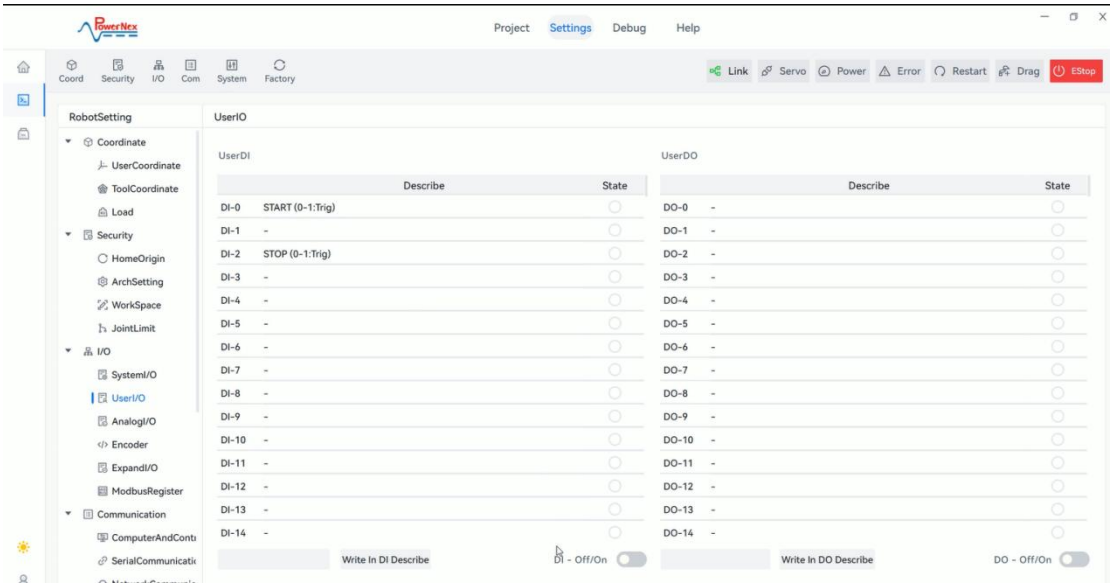
3.In the DI mapping section at the bottom of the interface, select the actual wired User I/O pin, for example, DI1.

4.Click "Write in SDI" to complete the I/O mapping. The status of User DI1 will then be equal to the status of the SDI-12 Alarm Clear function.

- **User I/O**

User I/O is used for level signal interaction and can be used for communication or to control system functions. Users can connect I/O pins according to their actual needs. The current number of User I/O channels is 1024, of which 32 I/O channels can be connected via actual wiring, while the remaining channels are for expansion or communication purposes.

Method for Setting User I/O:



1、Take User DO as an example; the same applies to System DO.

2、Select the User DO number to be controlled, for example, DO-1.

3、In the DI mapping section at the bottom of the interface, select the actual wired User I/O pin, for example, DI1.

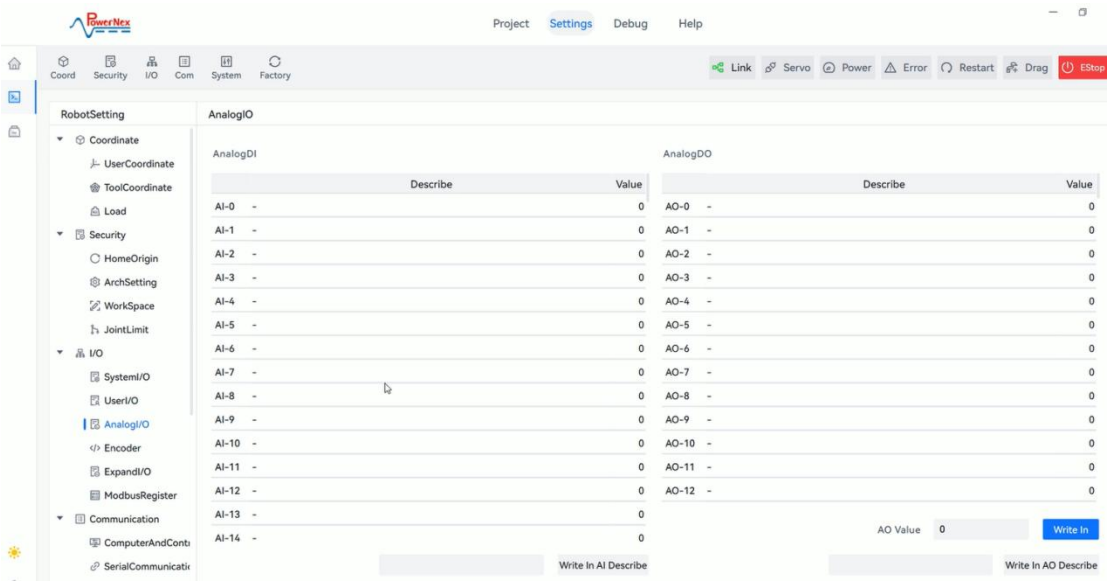
4、Click "DO-On/Off" to control the DO state.

Note: When connecting to a physical robot, it is not possible to force the output of a DI signal. If a DO pin is configured with a System DO function, the system controls this DO, and the user cannot output the DO pin independently.

- **Analog I/O**

Analog signals are values that can be exchanged via external communication, ranging from -10000 to 10000. The value type must be integer, not floating-point. The current number of analog I/O channels is 256.

Usage Declaration for Analog I/O:



1.The left side represents analog input, which is provided to the robot by external sources.

2.The right side represents analog output, where users set the value and output it to external devices via communication.

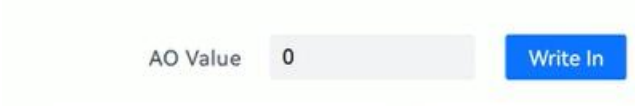
3.The value of analog input can only be monitored and cannot be simulated.

Method for Writing Analog Output (AO):

1.Select the AO number to be controlled, for example, AO-0.

2.Enter the desired output value in the "AO Value" input box at

the bottom of the interface and click "Write in".



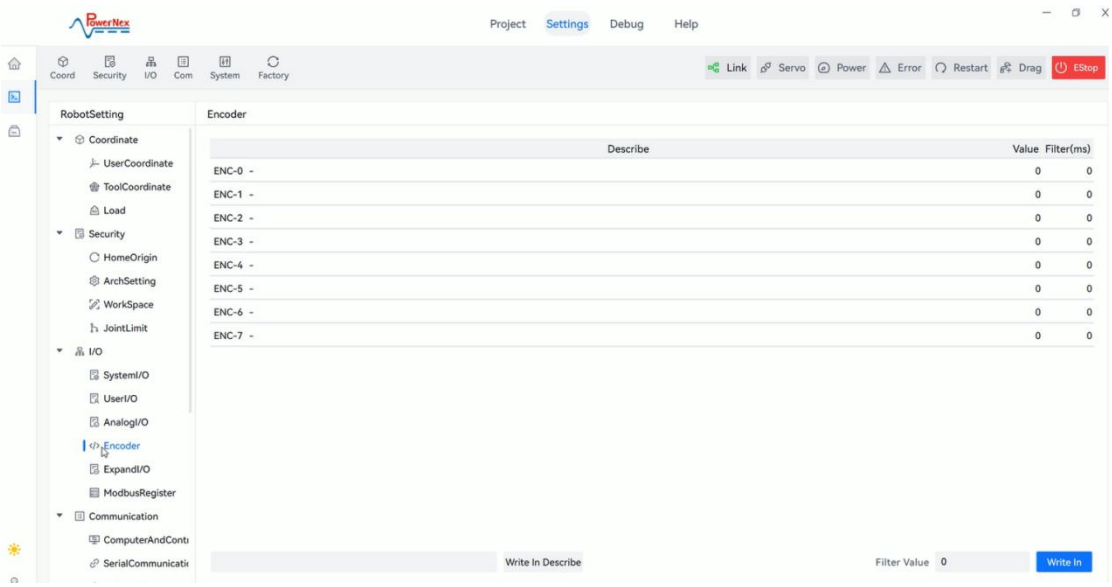
Note: Analog I/O is generally used for communication. Establish communication before data interaction.

The allowed value type for analog I/O is integer, with a range of -10000 to 10000.

- **Encoder**

Encoders typically refer to pulse feedback from encoders. The current system supports up to 8 encoder channels.

Usage Declaration for Encoders:



1.The current interface is only used to monitor the values fed back by the encoder channels.

2.By default, the encoder feedback values represent real-time status.

3.If the encoder feedback values are not balanced, use a filter

to average the values and reduce interference.

Method for Setting the Filter:

1.Select the encoder channel to be configured, for example, ENC-0.

2.Enter the desired filter value in the "Filter Value" input box at the bottom of the interface and click "Write in."。

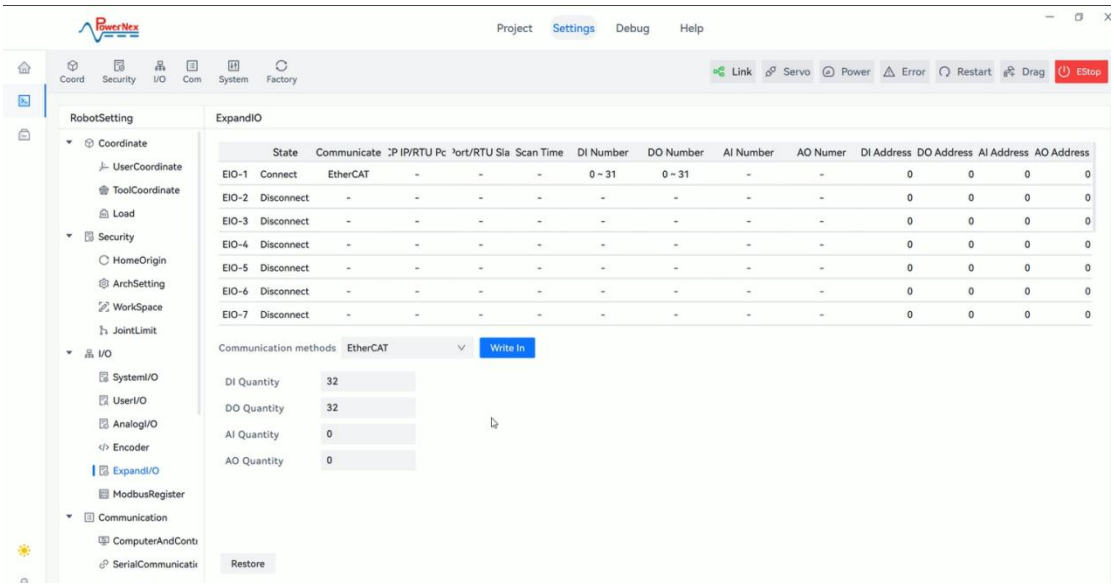
Note: The filter value setting range is 0 to 1000 ms.

For example, with a filter value of 100 ms, the encoder value is the average feedback over 100 ms.

- **Extended I/O**

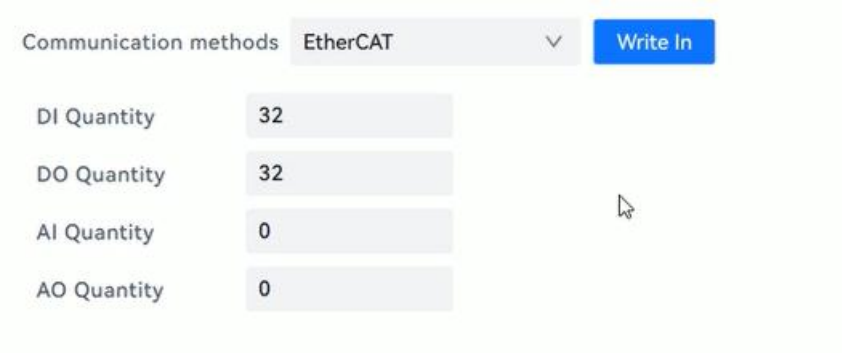
Extended I/O is used to configure the signal sources for User I/O or Analog I/O. The signal sources include EtherCAT, Registers, MODBUS-TCP Master, MODBUS-RTU Master, EtherNet/IP Adapter, EtherCAT Gateway (Single Byte), and EtherCAT Gateway (Double Byte).

Usage Declaration for Extended I/O:



1.Select the Extended I/O number to be used (ENC-1 to ENC-16).

2.Choose the communication method to be used. (The following example uses a hard-wired I/O card.)



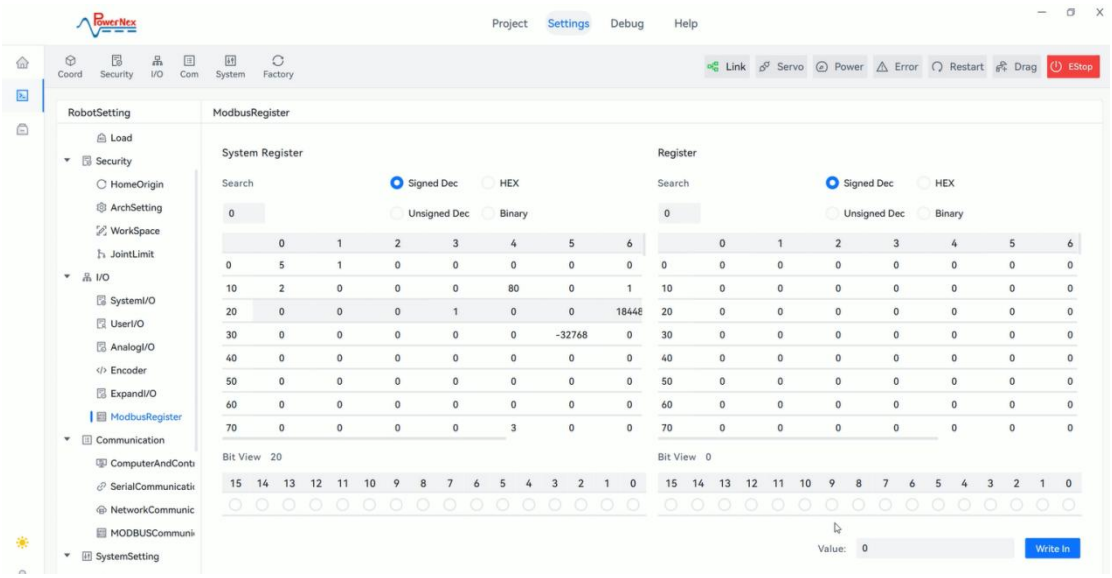
3.In the quantity interface, enter the number of DIO and AIO channels to be configured.

4.Complete the communication configuration by clicking "Write in"

- **Modbus Register**

Modbus registers are used to store and monitor the values of system register addresses and user register addresses. The system register addresses range from 0 to 1023, totaling 1024 addresses, with default hexadecimal values ranging from 0x0000 to 0x03FF. These addresses store various system functions and should be used with caution. The user register addresses range from 0 to 10239, totaling 10240 addresses, with default hexadecimal values ranging from 0x0400 to 0x27FF.

Usage Declaration for Modbus Register:



1.The content on the left side of the interface represents system registers, while the content on the right side represents user registers.

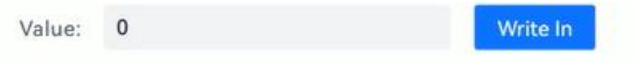
2. The search box in the interface allows users to input a register address, and the software will automatically jump to the corresponding register location.

3. Users can switch between different types of values to be monitored according to their needs.

4. The bottom of the interface displays the bit value of the currently selected register address.

Method for Writing to Modbus Register Addresses:

Select the register address to be written to, enter the value in the write box at the bottom right of the interface, and click "Write in" 。

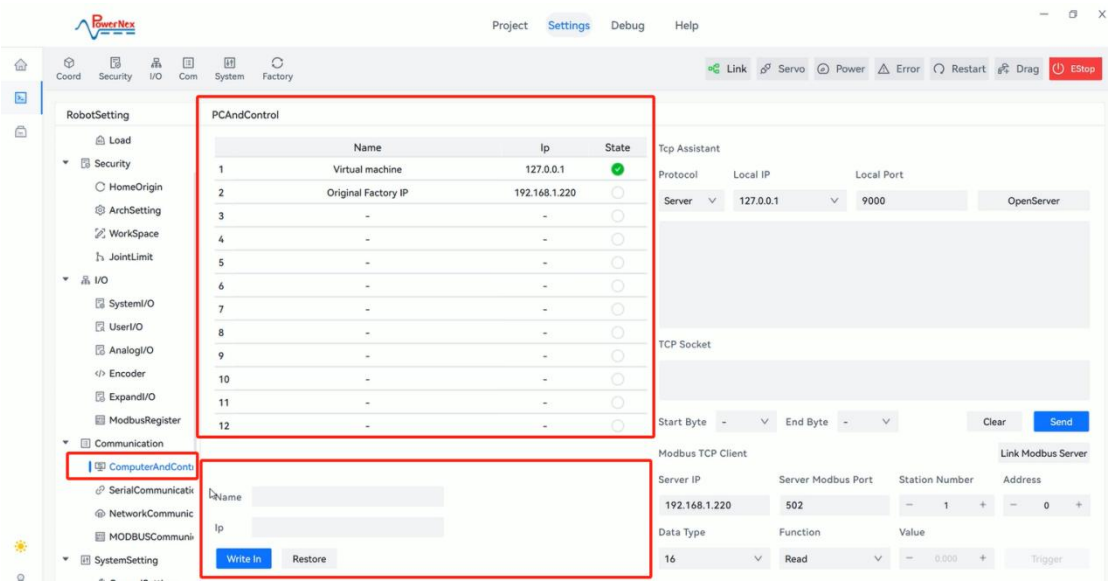


3.2.4 Communication

- **PCandControl**

Monitor or configure the name and IP address of the controller, as well as the network debugging assistant interface.

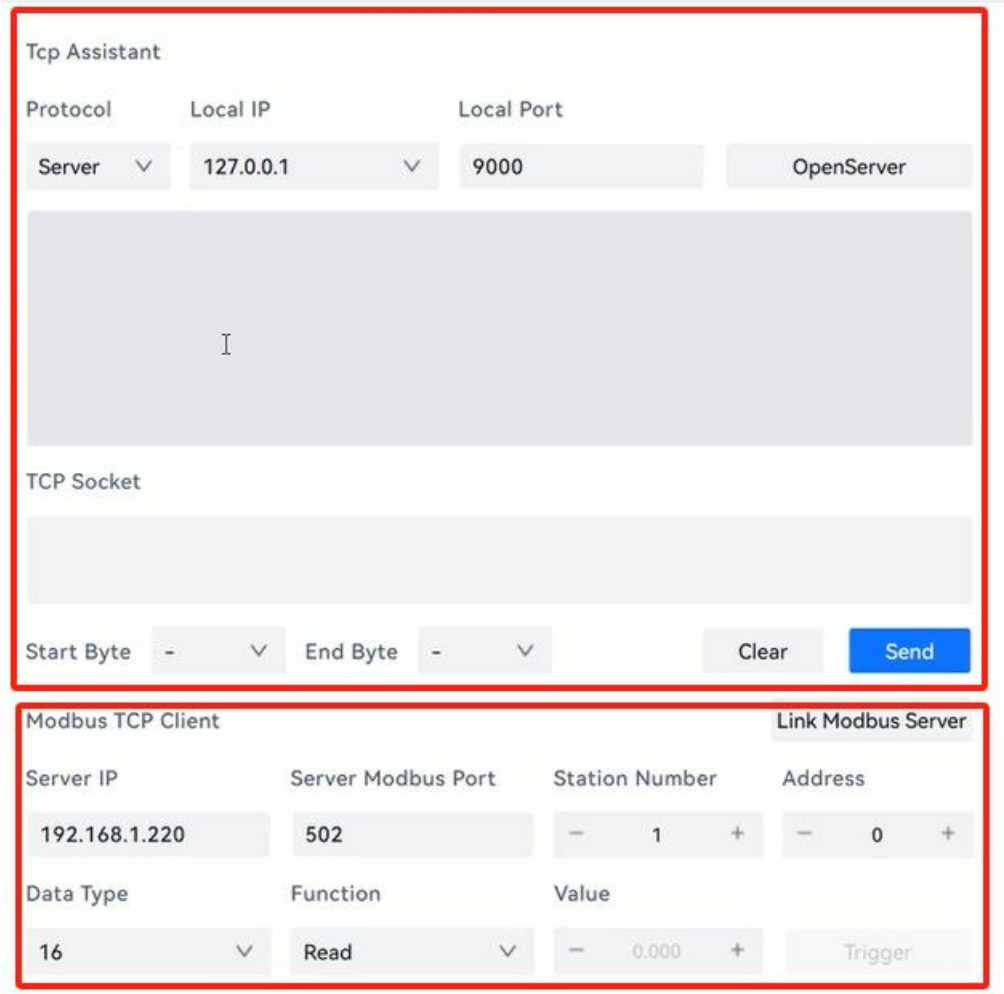
Usage Declaration for PC&Control:



1.Select the device name to be modified.

2.Modify the IP address of the selected device in the lower-left corner of the interface and click "Write in"

Network Debugging Assistant:



1. The upper section is a TCP/IP protocol-free communication assistant, used for opening a virtual server or client to debug communication programs or simulate communication with a host/PLC for external

communication tests.

2. The upper section is a TCP/IP protocol-free communication assistant, used for opening a virtual server or client to debug communication programs or simulate communication with a host/PLC for external communication tests.

3. The lower section is a Modbus communication assistant, supporting only the Modbus TCP protocol. It is used for direct communication with the controller to read and write to specified addresses.

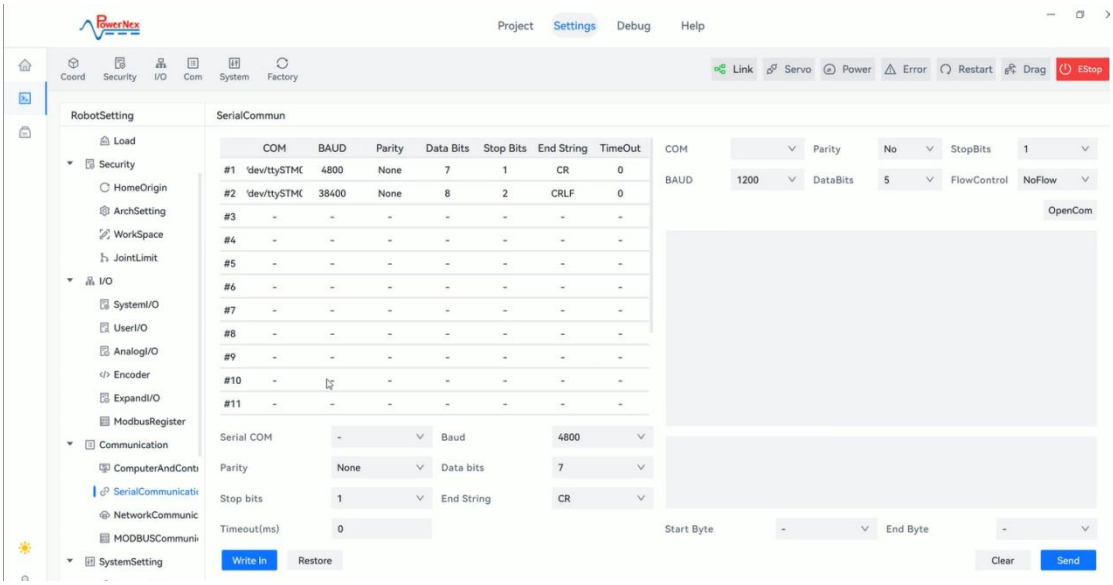
4. The lower section is a Modbus communication assistant, supporting only the Modbus TCP protocol. It is used for direct communication with the controller to read and write to specified addresses.

For specific usage examples, refer to Section 7.2 Network Communication Example or Section 7.4 Network Debugging Assistant.

- **SerialCommun**

Serial communication (Serial Communication) is supported in the PowerNexOS system, including protocol-free serial communication and Modbus-RTU communication. A total of 16 communication channels are available for user use.

Usage Declaration for Serial Communication:



1.Select the channel number to be used (#1 to #16).

2.Configure the selected channel number at the bottom of the interface.

Choose the COM port number, communication baud rate, parity bit, data bits, stop bits, communication terminator, and timeout settings.

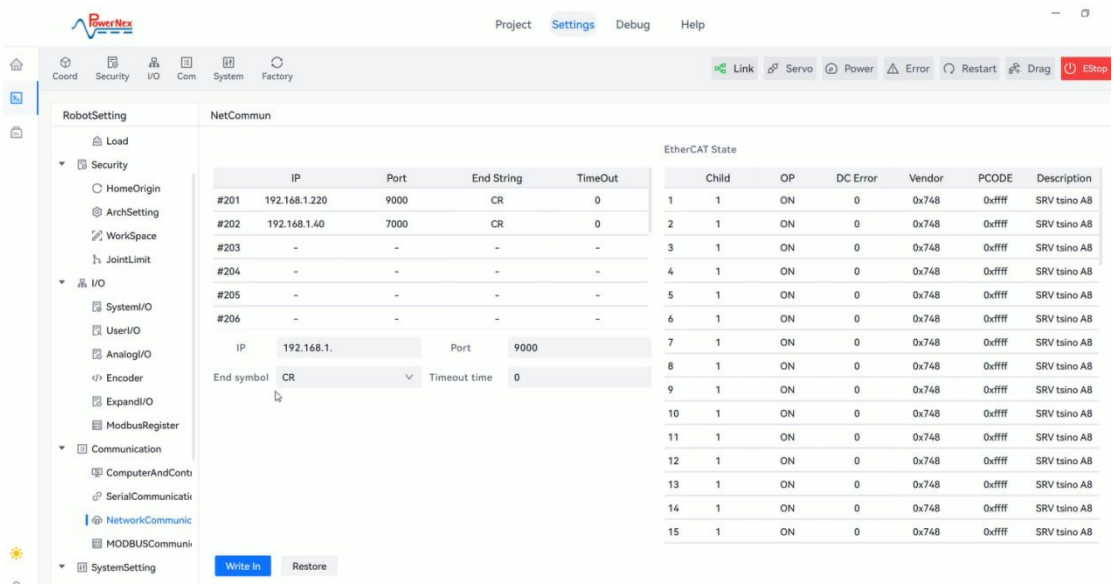
3.Click "Write in" after configuration is complete.

4.For examples on opening serial communication and sample code, refer to Appendix 7.1 Serial Communication Example

- **Net Commun**

Network communication (TCP/IP protocol-free communication) is supported in the PowerNexOS system, including TCP/IP Socket protocol and Modbus-TCP communication. A total of 16 communication channels are available for user use.

Usage Declaration for NetCommun:



1.Select the channel number to be used (#201 to #216).

2.Configure the selected channel number at the bottom of the interface. Set the communication IP address for the current channel. For server mode, enter the controller's IP address; for client mode, enter the server's IP address. Also, set the communication port number, terminator (Carriage

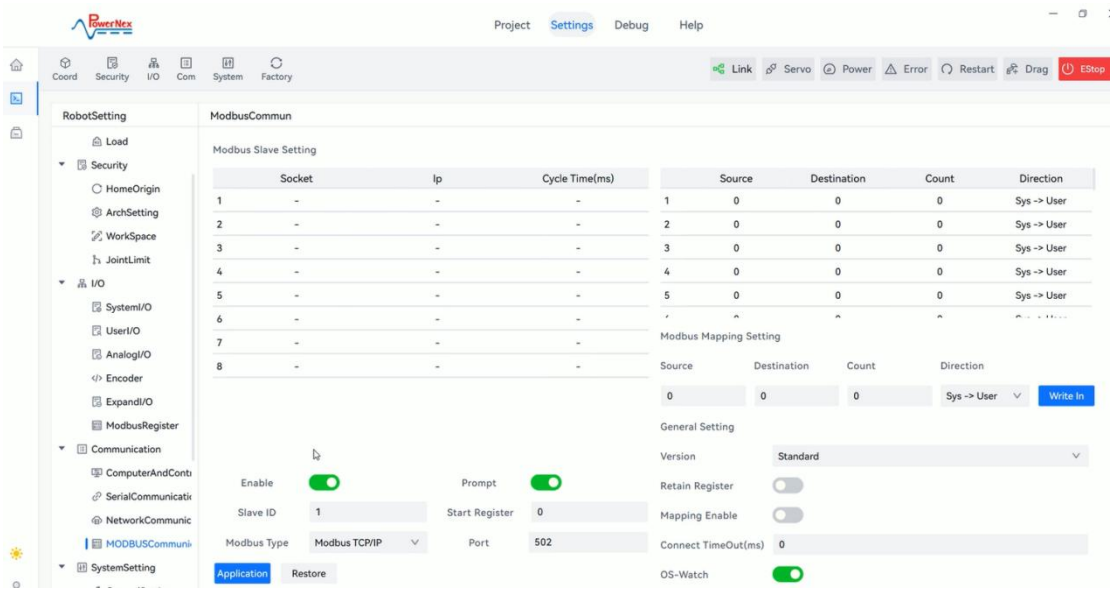
Return CR, Line Feed LF, Carriage Return Line Feed CRLF), and timeout settings. A timeout value of 0 indicates no timeout. If no interaction occurs within the specified time, the communication on the current port will be disconnected.

3..Click "Write in" after configuration is complete.

4.For examples on opening network communication and sample code, refer to Appendix 7.2 Network Communication Example

- **ModbusCommun**

Modbus communication refers to the Modbus communication protocol. The PowerNexOS system supports Modbus TCP and Modbus-RTU. A total of 8 communication channels are available for user use.



1.Select the Channel Number (1 to 8): Choose the communication channel you intend to use.

2.Configure the Selected Channel: In the interface below, enable or disable the selected channel and configure whether to display messages

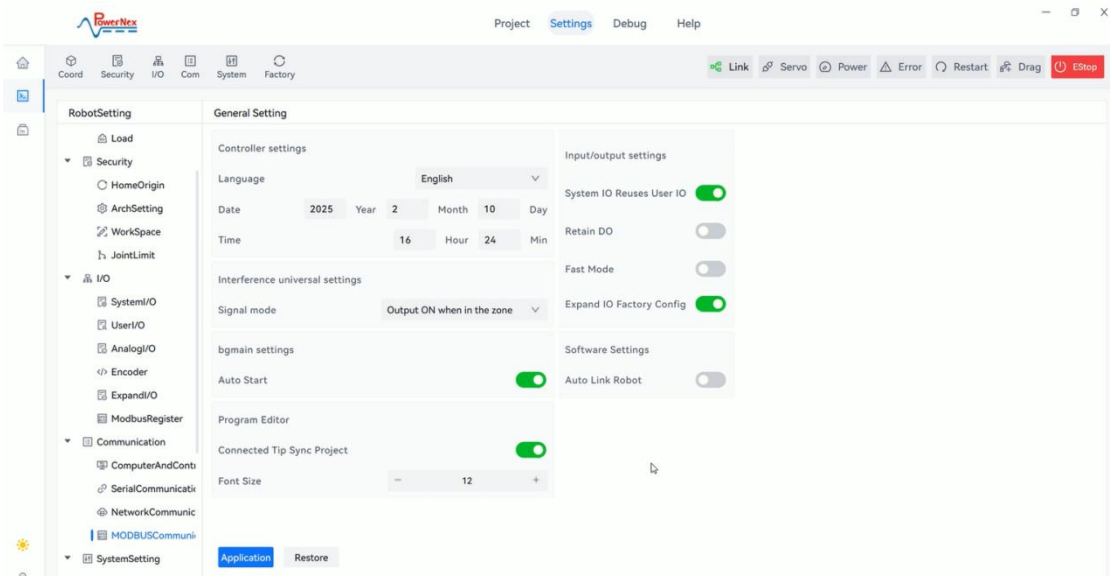
on the output panel. Set the device station number, starting register address, communication protocol type, and communication port for the selected channel.

3.Apply the Configuration: Click "Application" after completing the settings.

4.Modbus Communication Example: Refer to Appendix 7.3 for Modbus Communication Examples.

3.2.5 System Settings

- **General Settings**



Controller Settings: Set the controller language to Chinese/English and configure the controller's time and date.

Interference universal Settings: Configure the output mode of the DO signal in the workspace.

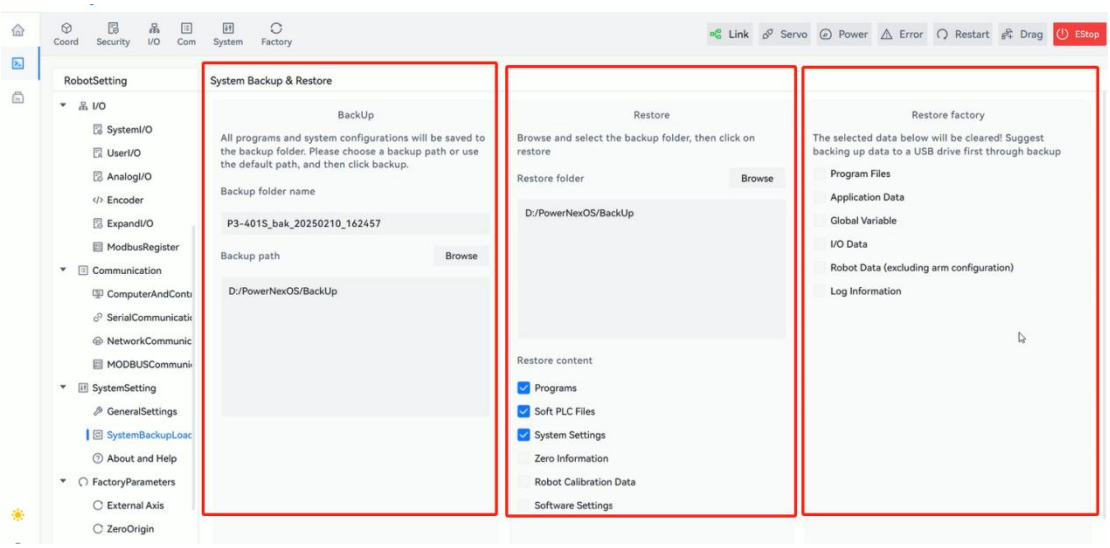
Program Editor: Adjust the font size in the program area.

Input/Output Settings: Configure the I/O pin mode and retention settings.

bgmain Settings: Set whether the bgmain background program should start automatically upon controller startup. Refer to 7.9 for background.

Note: After making settings, click "Application" to save the changes.

- **System Backup and Restore**

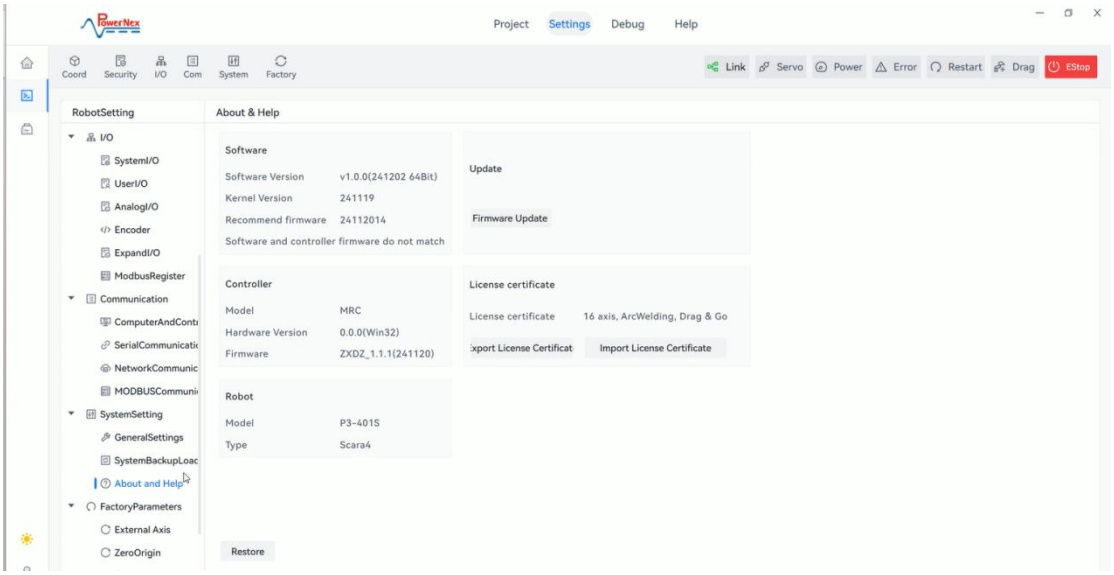


Backup: Save programs and system configurations to a user-specified folder. The folder path is defined by the user.

Restore: Use backup files to restore programs, soft PLC files, and system settings. Zero-point information, robot calibration data, and PowerNex software settings should be operated under the guidance of the manufacturer.

Factory Reset: Clear data for the selected items. Backup your data before performing this operation.

- **About and Help**



Software: Displays the current software version and the recommended controller firmware version. A mismatch between the controller firmware and software versions will be indicated.

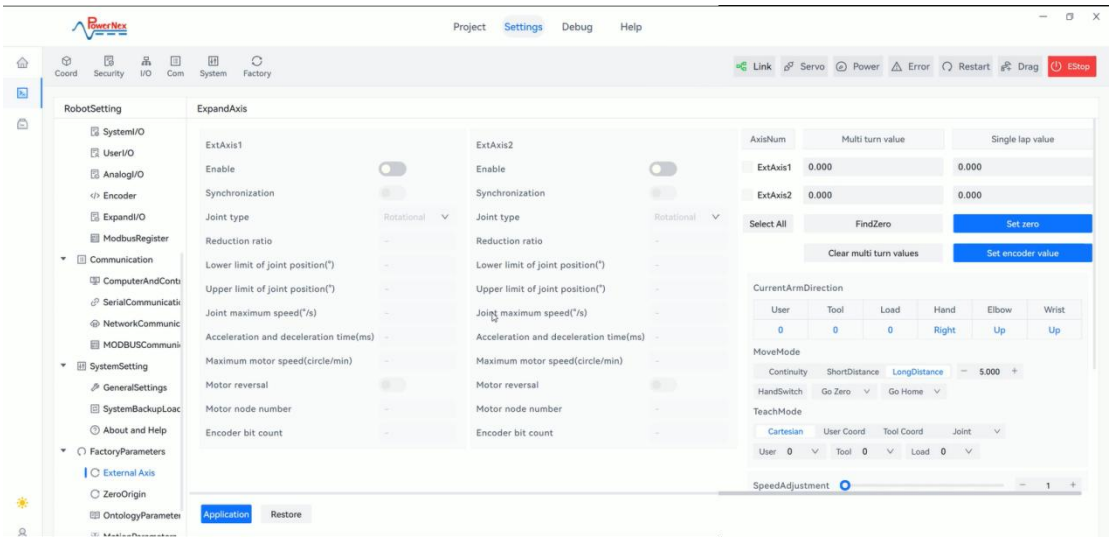
Robot: Displays the model of the robot currently connected.

Controller: Displays the model, hardware version, and firmware information of the controller.

Update: Used for updating the controller firmware. Refer to [7.8 for detailed instructions on firmware updates]

3.2.6 Factory Parameter

- **External Axes**

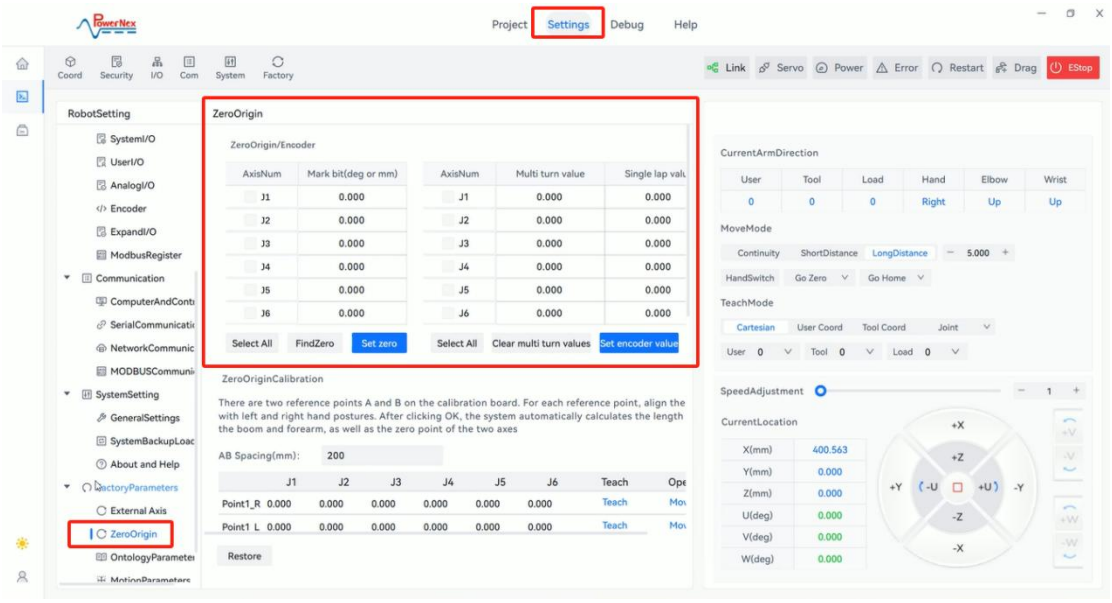


Note: The external axes interface requires administrator privileges to access. Please log in with administrator permissions before entering the interface.

PowerNexOS supports up to two external axes.

- **ZeroOrigin**

The robot zero-point setting interface is generally used for re-teaching or recovering the origin. Do not change the robot zero point unless there is an abnormal situation.



ZeroOrigin Teaching Method:

Step 1: Push the first and second axes of the robot to an approximately straight position, and push the third axis to its highest point while roughly aligning the fourth axis to the front direction.

Step 2: In the robot's zero-point interface, select the member axes that need to be taught and click "Clear Multi-turn Values."

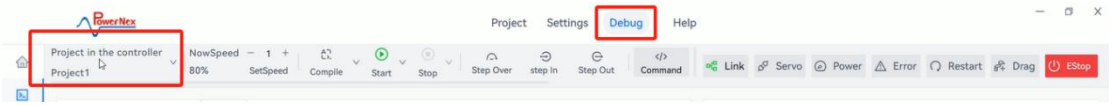
Step 3: Scan the QR code on the robot base to obtain the single-turn value information of the member axes and enter it into the single-turn value list.

Step 4: Click "Set Encoder Values" to complete the zero-point teaching.

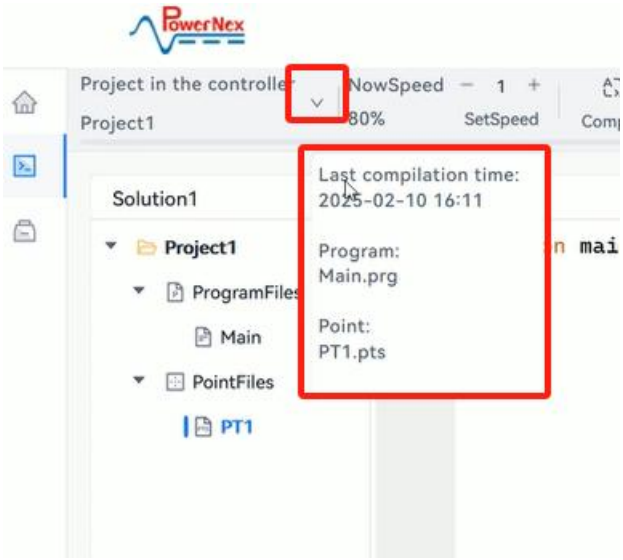
Step 5: Perform a full-axis zero-point return in the jogging interface.

3.3 Debug

3.3.1 Current Active Project

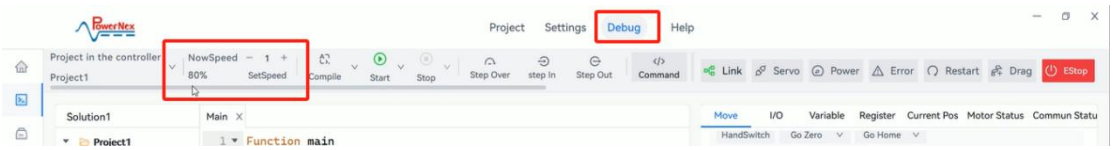


In the debugging interface, the first item in the function bar is the current active project, that is, the program project in the controller, not the active project specified in the software. When you click "Compile" or "Run," the system will download the active project specified in the software to the controller and execute it.



Click the drop-down menu on the right to display the time of the last compilation and the content stored in the current controller.

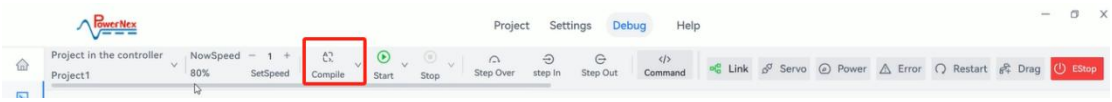
3.3.2 NewSpeed/SetSpeed



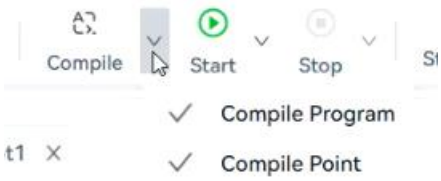
In the debugging interface, the second item in the function bar indicates

the current speed display and speed setting, which means the speed of the program running in the controller is associated with the jogging speed. The setting range for the operating speed is from 1% to 100%.

3.3.3 Compile

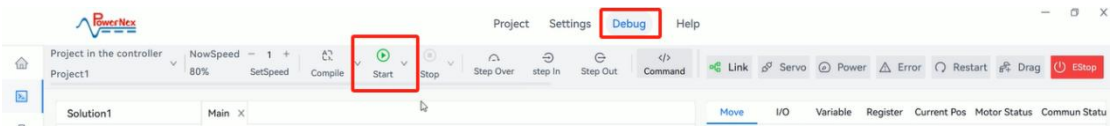


In the debugging interface, the third button in the function bar is the "Compile" button. It is used to check whether there are any errors in the program of the current active project. If no errors are found during the check, a pop-up window will appear indicating that the compilation is complete, and the program will be downloaded to the controller.

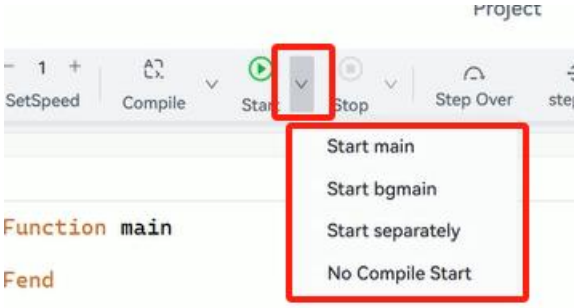


The dropdown menu also allows you to select downloading the program alone or downloading the point table alone.

3.3.4 Start

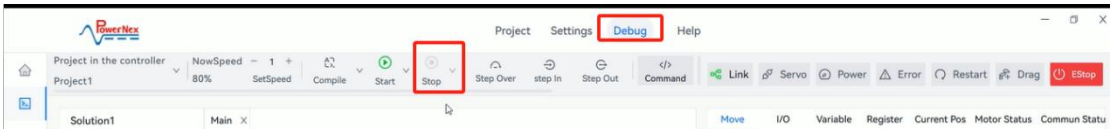


The fourth button in the function bar of the debugging interface is the “Start” button. It is used to launch all programs in the solution. When the Start button is clicked, the current active project will be compiled and downloaded to the controller. If the compilation fails, an alarm will be triggered. When the program begins to run, the Start button switches to a Pause button, which can be clicked to pause the program.



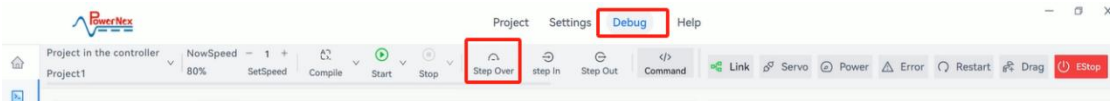
In the drop-down menu, you can select to run the following options: “Start main (foreground task)” / “Start bgmain (background task)” / “Start individually (specified Function)” / “Start without compilation (Start the current active project without downloading to the controller).”

3.3.5 Stop



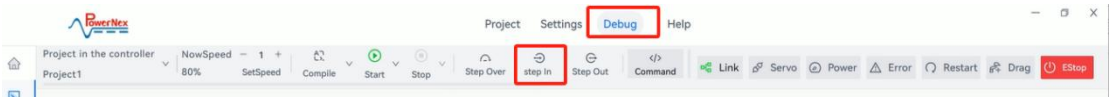
The “Stop” button is used to terminate all tasks or to selectively stop the foreground/background program in the drop-down menu.

3.3.6 Step Over



The sixth button in the function bar of the debugging interface is the “Step Over” button. It is used when the program is at a breakpoint to execute the current breakpoint statement or function call and maintain the breakpoint state at the next statement line.

3.3.7 Step In



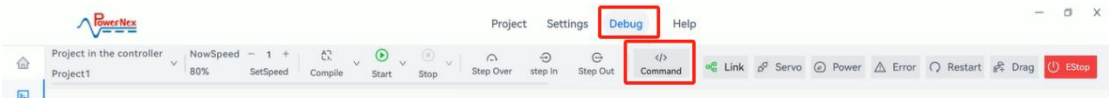
The seventh button in the function bar of the debugging interface is the “Step In” button. It is used when the program is at a breakpoint to execute

the current breakpoint statement or function call and maintain the breakpoint state at the next statement line. Note: If the current breakpoint statement is a function call, the statements within the function will also maintain the breakpoint state.

3.3.8 Step Out

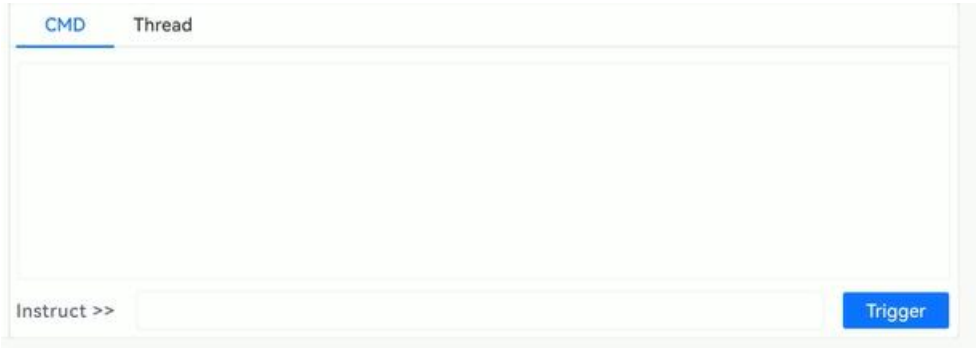
The eighth button in the function bar of the debugging interface is the “Step Out” button. It is used when the program is at a breakpoint to execute the current breakpoint statement or function call and exit the breakpoint state of the current statement.

3.3.9 Command



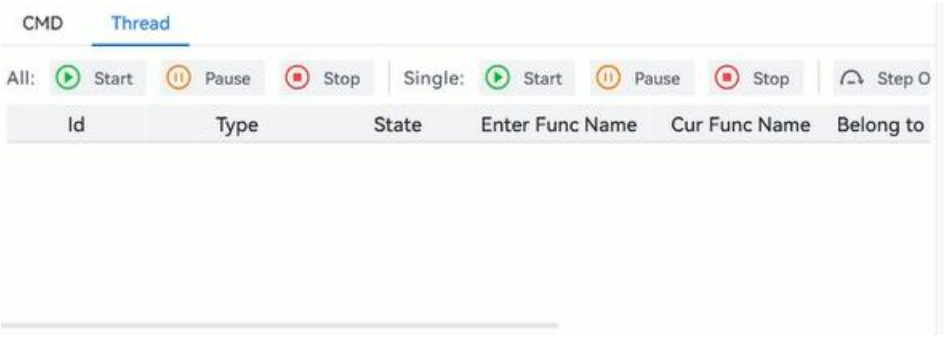
In the debugging interface, the ninth position on the toolbar is the "Command" button. Upon clicking it, a command line and thread monitoring panel will pop up at the bottom of the software. It is used to execute command statements or encapsulated functions, as well as to monitor all task processes.

Usage Instructions for the Command:



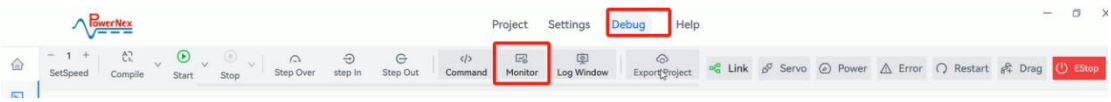
Enter the statements to be executed or the encapsulated functions to be called in the command input box below. The command will be sent and executed when you click "Trigger" or press the Enter key on the keyboard.

Usage Instructions for Thread Monitoring:

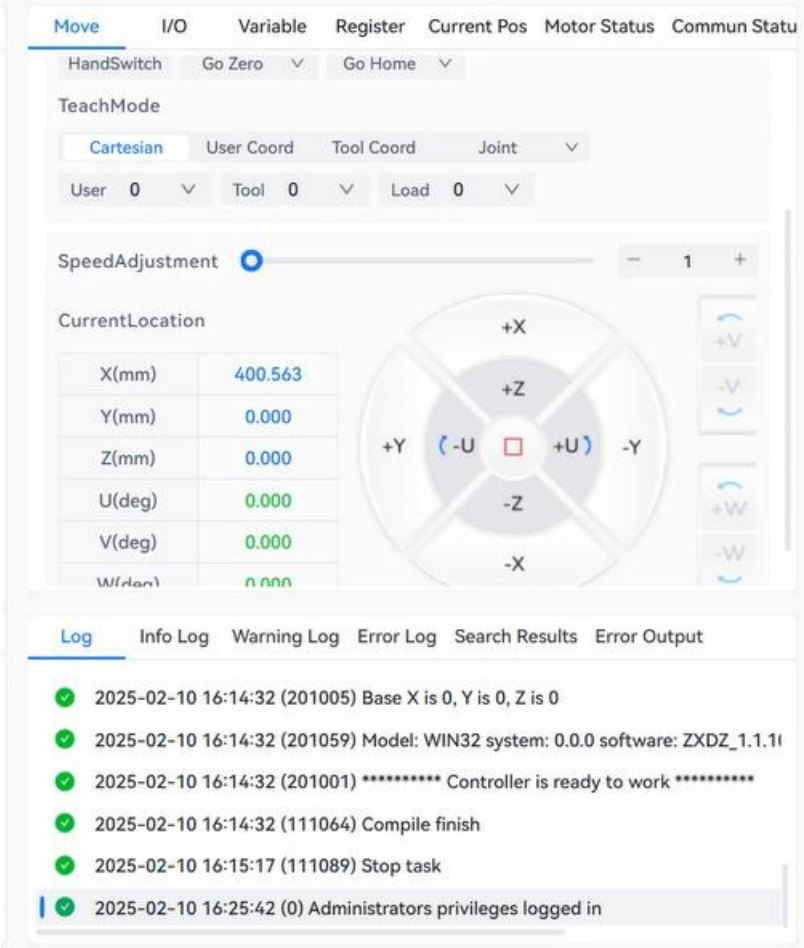


In this functional panel, you can monitor all task threads that are in the execution/pause/alert state. You can also set the status of individual or all task threads.

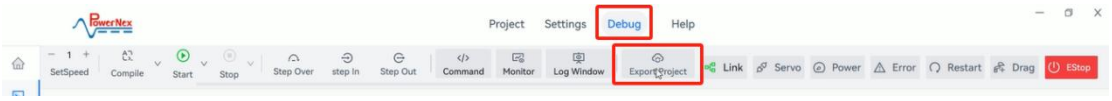
3.3.10 Monitor



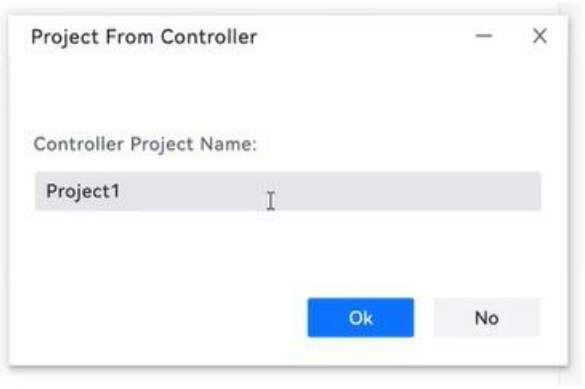
The auxiliary panel on the right side of the project/debugging interface, where you can choose whether to display it. By default, it is set to be displayed.



3.3.10 ExportProject



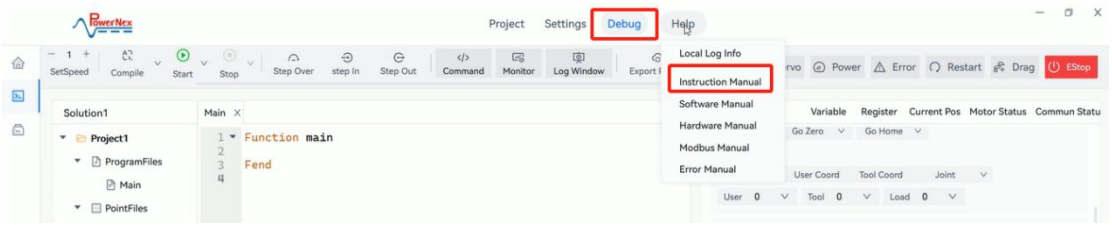
Export the scheme from the controller to the software, which includes program files and point files. After clicking the button, a pop-up window will appear. Please name the exported scheme in the pop-up window and click "OK." After the operation is completed, the exported program will be stored in the current project path and displayed in the project area.



Note: After the program in the controller is exported, the software will set the exported scheme as the active project.

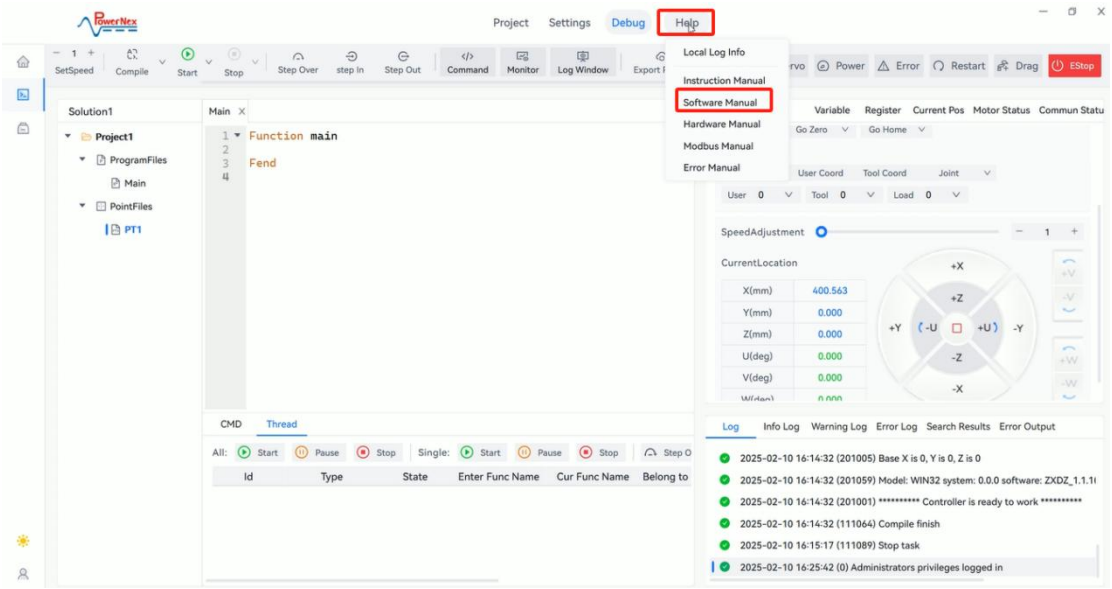
3.4 Help

3.4.1 Instruction Manual



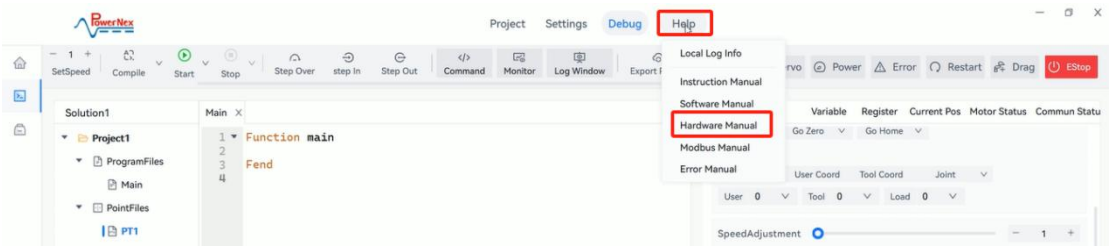
In the Help menu, the first option is “Instruction Manual.” Clicking on it will automatically open the instruction manual.

3.4.2 Software Manual



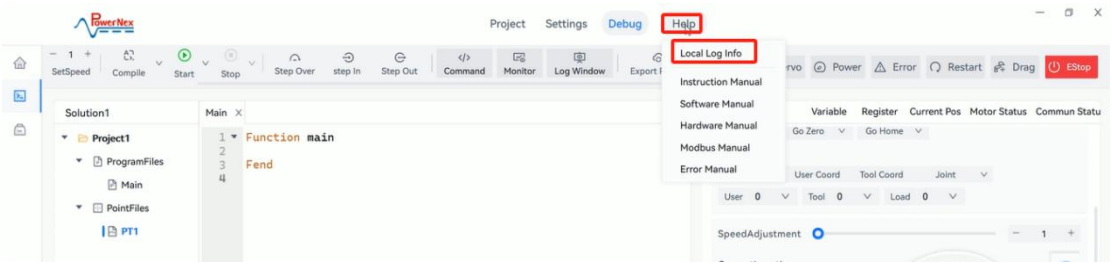
In the Help menu, the second option is “Software Manual.” Clicking on it will automatically open the software manual (i.e., this document).

3.4.3 Hardware Manual



In the Help menu, the third option is “Hardware Manual.” Clicking on it will automatically open the hardware manual.

3.4.4 Local Log Info

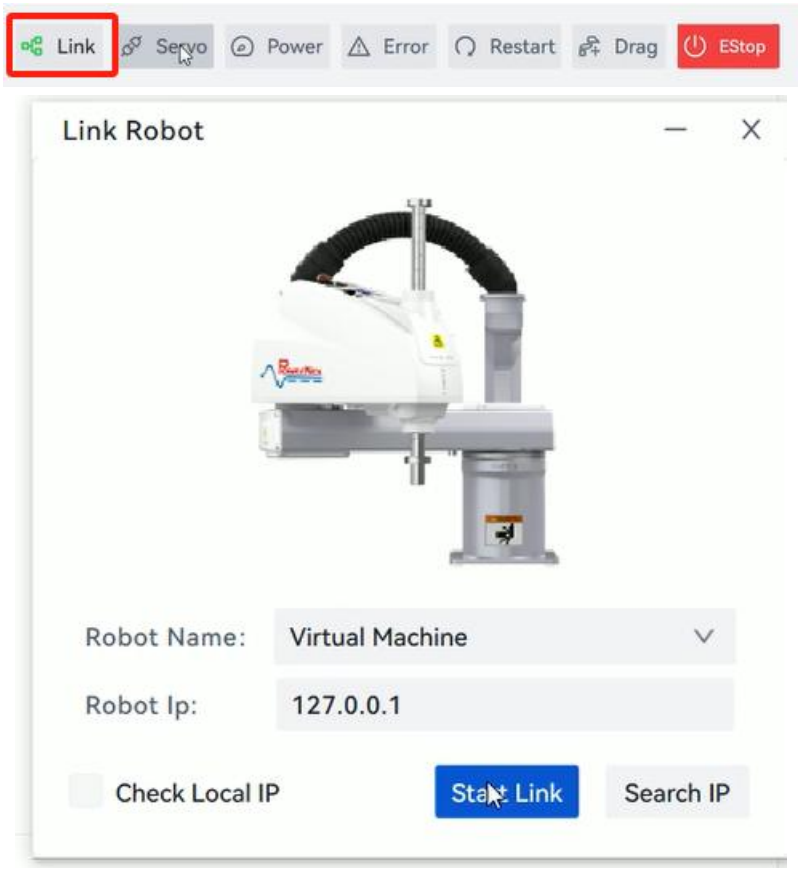


In the Help menu, the sixth option is “Local Log Info.” Clicking on it will open the local log text file stored in the local path. The file type is TXT format.

Chapter 4: Status Bar Operation Applications

4.1 Link/Disconnecting the Controller

- Link

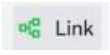


Click the “link” 

button in the upper right corner of the main interface. In the pop-up window, select the original factory IP or virtual machine for connection (default physical machine: 192.168.1.220) (Virtual machine: 127.0.0.1). You can also search for connectable robotic arm information by clicking the “Search IP” button below. After confirming the IP address, click “Link.” If the robot’s IP address has been changed, you can also enter the robot’s IP address in the Robot IP window and click “Start Link” to complete the operation.

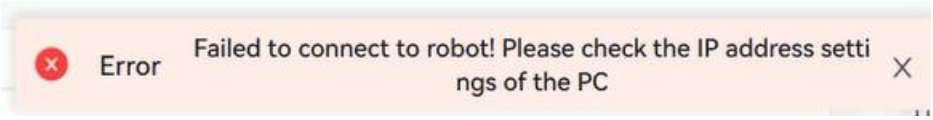
Note: The “Search IP” function can search across subnets, but the computer and controller must still be in the same subnet for connection.

Operation Results:

- If the connection is successful, the “Link” button in the upper right corner will turn green: 

.Meanwhile, the log in the lower right corner of the software will display the message “Controller is ready”.
- If the connection fails, the connection button in the upper right corner will remain gray: 

. The software will also pop up a message box indicating that the robot cannot be connected.



- **Disconnecting the Controller**

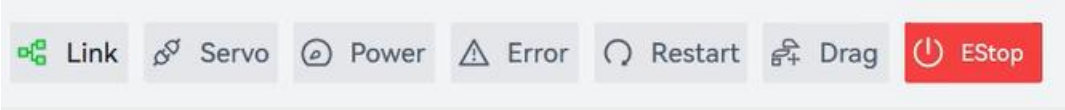
To disconnect the controller, simply click the green “Link” button. When the button turns gray, it indicates that the connection has been terminated. The “Output Module” in the lower right corner of the software will display the message “Disconnected from Controller”.

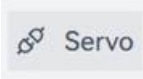
- **Troubleshooting for Connection Issues:**
  - Check if the hardware connections are normal.
  - Ensure that the PC and the target robot controller are on the same network and that the robot controller is reachable via Ping.

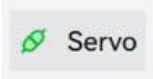
- Verify if the target robot controller is connected to another PowerNexOS or teaching pendant. If it is, disconnect it and then connect with the current software.
- Check if the Ethernet cable being used is a Category 6a cable.
- If the PC and controller are in the same subnet, uncheck the option to verify the local IP.

- **Conditions That May Cause Disconnection:**
  - Strong network interference.
  - High CPU usage on the PC (exceeding 50%).
  - Physical network disconnection, such as a disconnected Ethernet cable or power loss to a router or switch.
  - Significant version differences between the software and the controller.

4.2 Servo Button

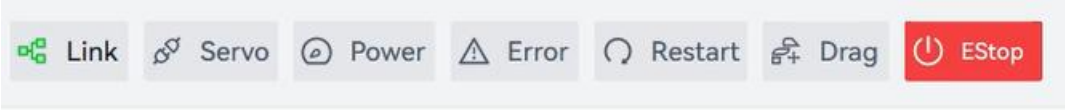


Braking State: 

;Enabled State: 

.

4.3 Power Button

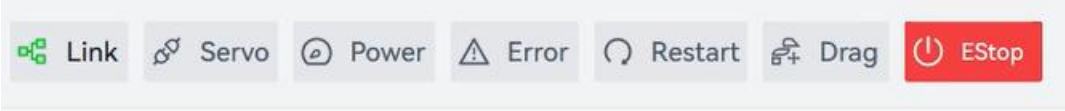


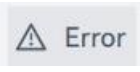
Low Power: 

;High Power: 

.

4.4 Error Button

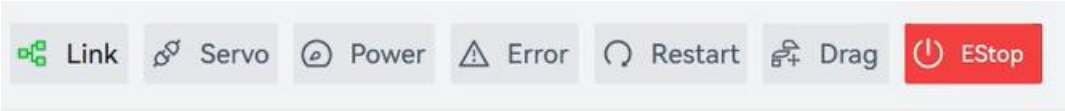


When the “Error” button is clicked 

, a pop-up window will appear

.

4.5 Restart Button

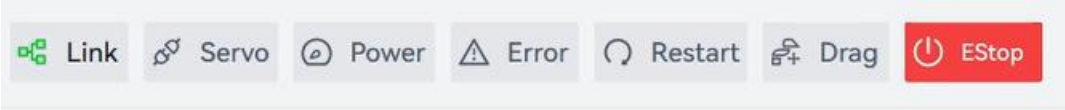


When the “Restart” button is clicked 

,a pop-up window will appear 

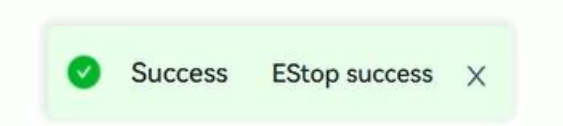
.

4.6 Emergency Stop Button



When the “EStop” button is clicked 

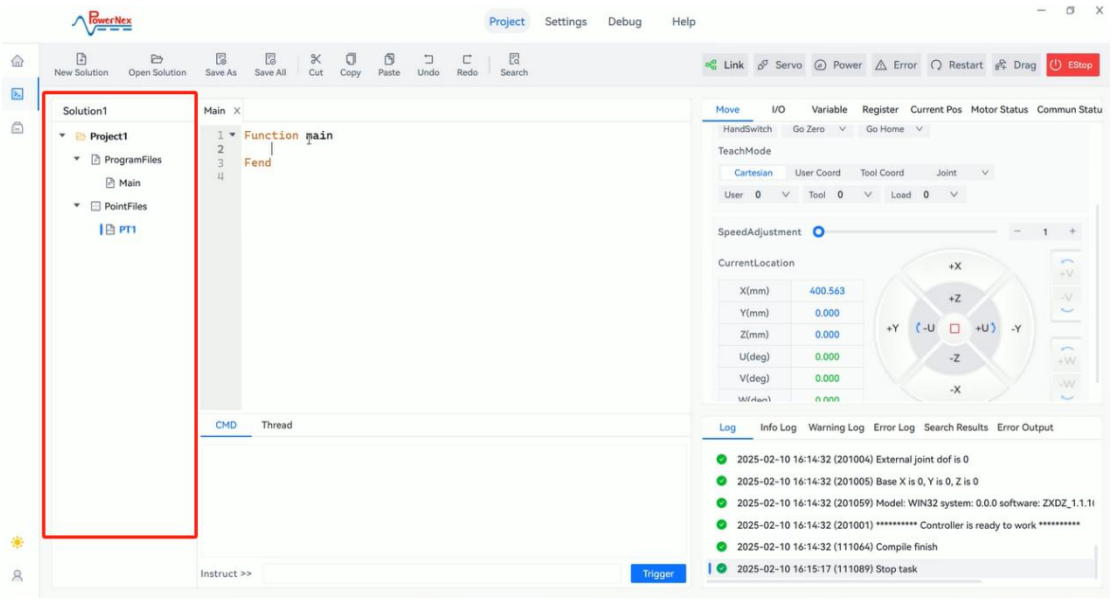
, the following pop-up message will be displayed.



Chapter 5 Project Area Operations and Applications

5.1 Project Engineering

The project area is located to the right of the navigation bar and is used to manage the engineering solutions opened by the software. An engineering solution can store multiple schemes. After selecting a scheme, right-clicking with the mouse allows you to set it as the active project. A scheme includes program files and point files. One scheme can store multiple program files and point files. However, it should be noted that the program file named "Main" must exist as the program main.

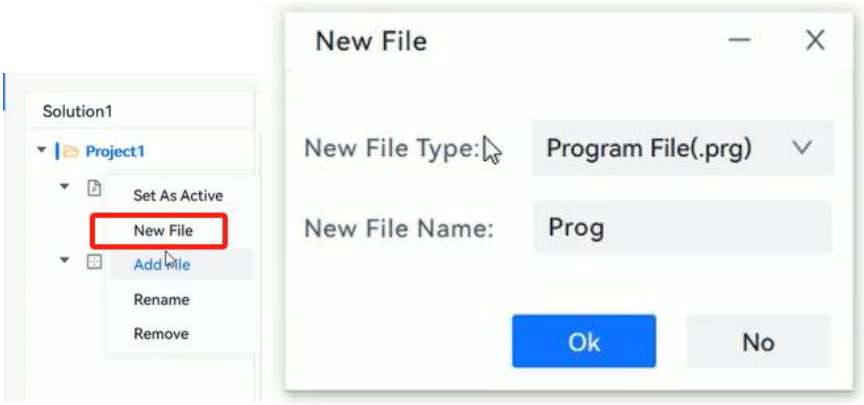


- Example from the diagram text: "New System"; This area is the engineering area used to manage all schemes in the engineering folder.
- Example from the diagram text: "Project1"; This area is the project scheme area, used to store the program files and point files of this scheme.
- Example from the diagram text: "ProgramFiles"; This option is the program file area, used to store program files or to add/delete program files.

- Example from the diagram text: "PointFiles"; This option is the point file area, used to store point files or to add/delete point files.

5.2 Creating New Program Files

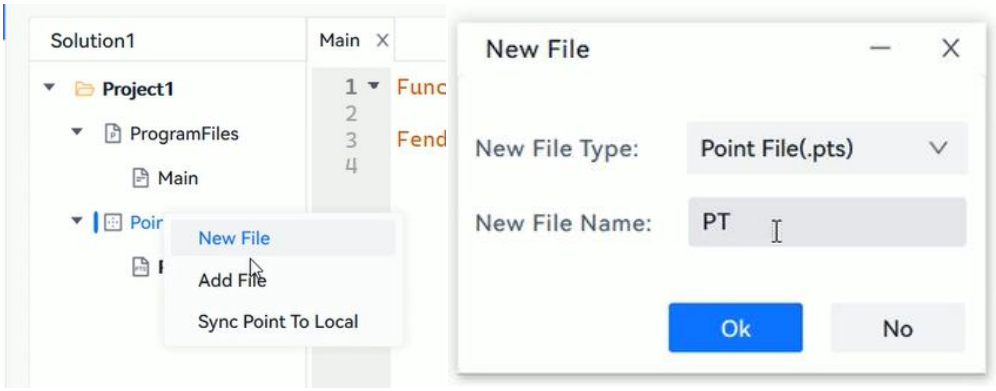
After right-clicking on the program file, a menu bar will be displayed. Users can create a new program file or add an existing program file here.



After clicking "New File," the software will display a pop-up window. Users need to choose whether to add a new program file or a point file and customize the name of the new file. After completing the settings, click "OK" to finish the creation. The new file will be generated in the program file or point file area.

5.3 Creating New Point Files

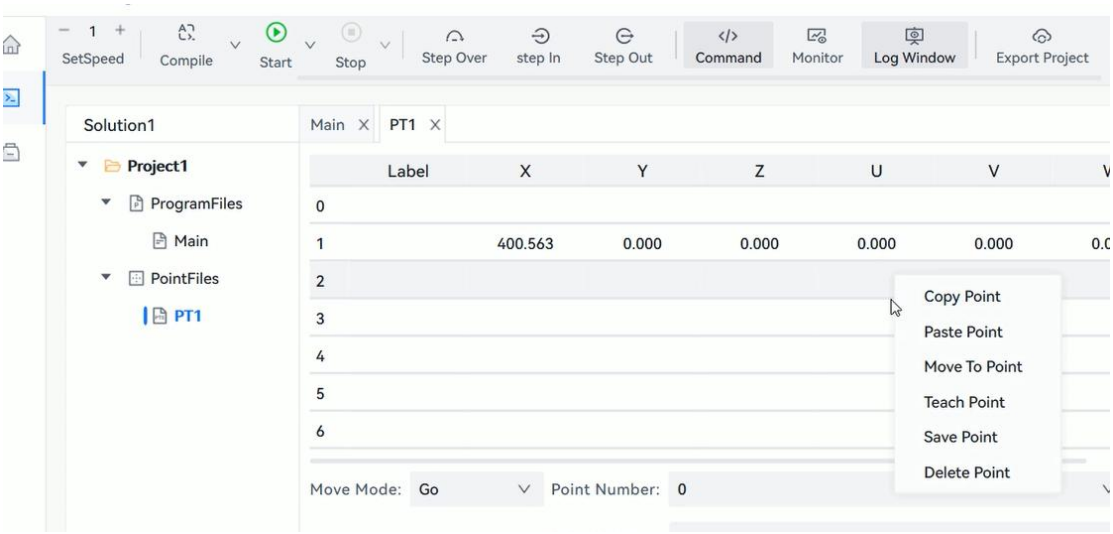
After right-clicking on the point file, a menu bar will be displayed. Users can create a new point file or add an existing point file here.



After clicking "New File," the software will display a pop-up window. Users need to choose whether to add a new point file or a program file and customize the name of the new file. After completing the settings, click "OK" to finish the creation. The new file will be generated in the program file or point file area.

5.4 Point File Description

Double-click to open the point file. A single point file can store up to 1,000 points. Each point data includes (point number, point label, X coordinate, Y coordinate, Z coordinate, U coordinate, V coordinate, W coordinate). You can also modify and operate on individual point data, as shown in the menu list that appears after right-clicking on a point in the figure below.



Copy Point: Copy the point information, which can be pasted into

another point number.

Paste Point: Paste the copied point information into the selected point number.

Move to Point: Move to the selected point coordinates in a point-to-point manner according to the robot's current set movement speed.

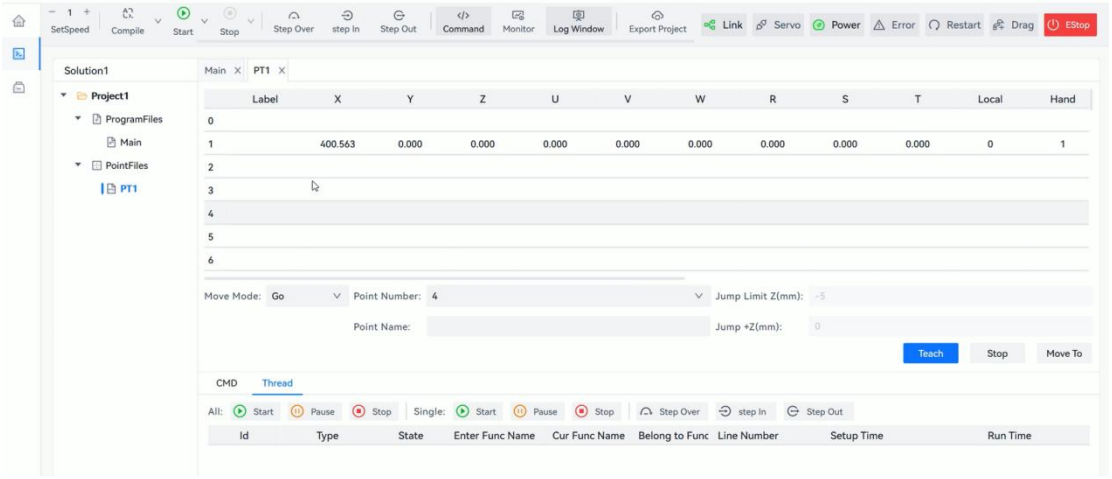
Teach Point: Store the current coordinate information in the selected point number.

Save Point: Save all data in the point table.

Delete Point: Delete the selected point number.

5.5 Point Teaching

Double-click to open the point file. The teaching and motion bars are located at the bottom of the interface.



Teaching Method:

Step 1: Select the point number (0 to 999) that needs to be taught and enter the "Point Name" (optional).

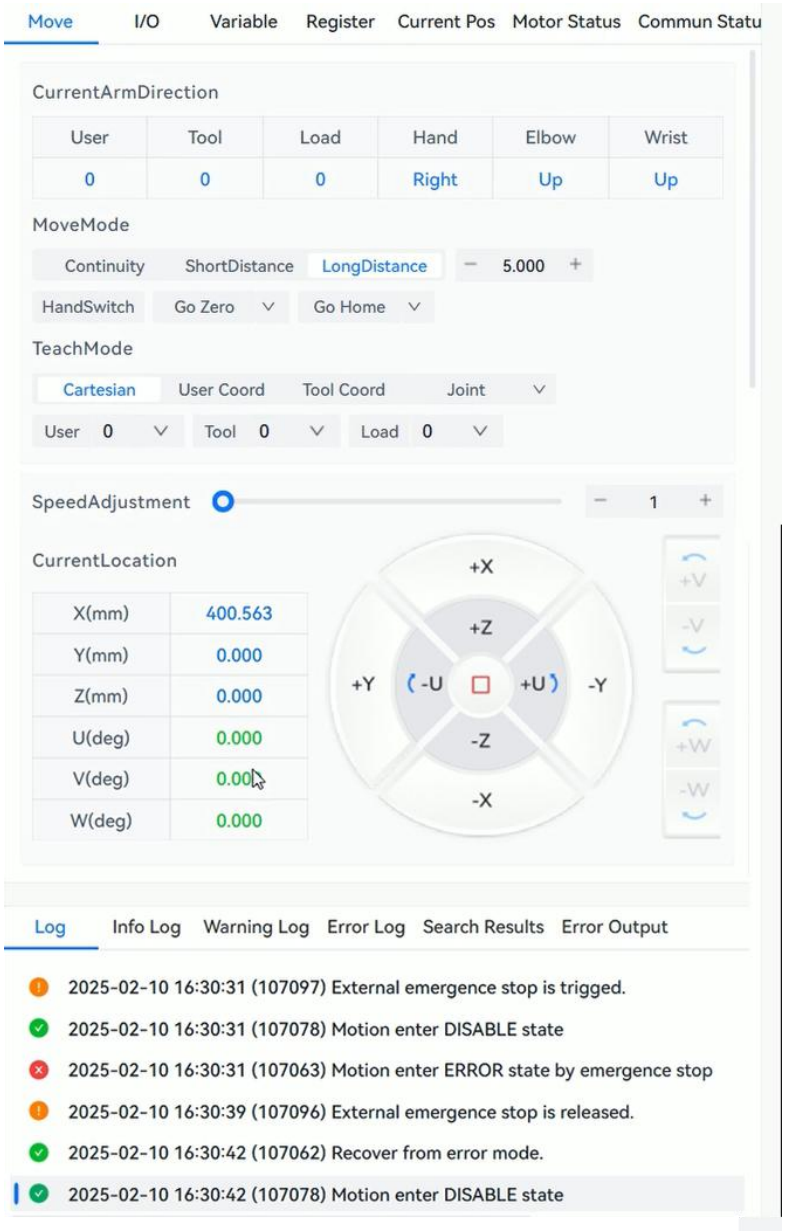
Step 2: Click the blue "Teach" button.

Step 3: Right-click on any point number in the point file and select "Save Point."

Chapter 6: Auxiliary Zone Operation Application

When the software is launched, the auxiliary zone interface is opened by default. If the auxiliary zone interface is not open, it can be manually opened from the debugging interface. The auxiliary window is located on the right side of the software and is used for monitoring and debugging the robot's jogging function, I/O status, program variable status, register status, motor status, and communication status.

Auxiliary Zone Interface:



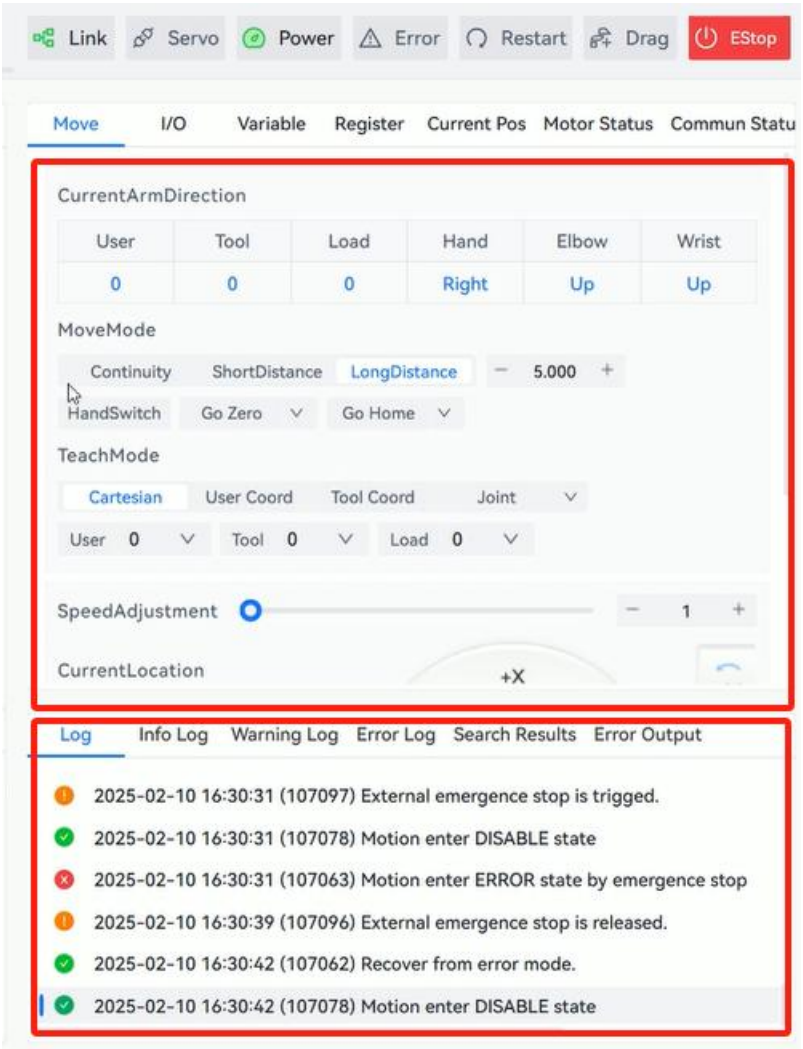
Method to Open the Auxiliary Zone:

Select “Debug” from the menu bar, and then choose “Monitor” from the function bar to open or close the auxiliary panel.

6.1 Move

The move interface is divided into two main sections: the move panel and the log panel. The move panel is used to move the robot, with two motion

modes: continuous motion mode and fixed-distance mode.



Current Arm Pose: Displays the user coordinate system, tool coordinate system, payload number, left/right-handedness, elbow pose (six-axis), and wrist pose (six-axis) of the robot's end-effector center.

Move Mode: Select the desired motion mode, which can be continuous, short-distance, or long-distance. When the mode is continuous, the robot will continue to move as long as the direction button is pressed; otherwise, the motion stops. For short-distance and long-distance modes, the robot moves once according to the set distance value, and multiple

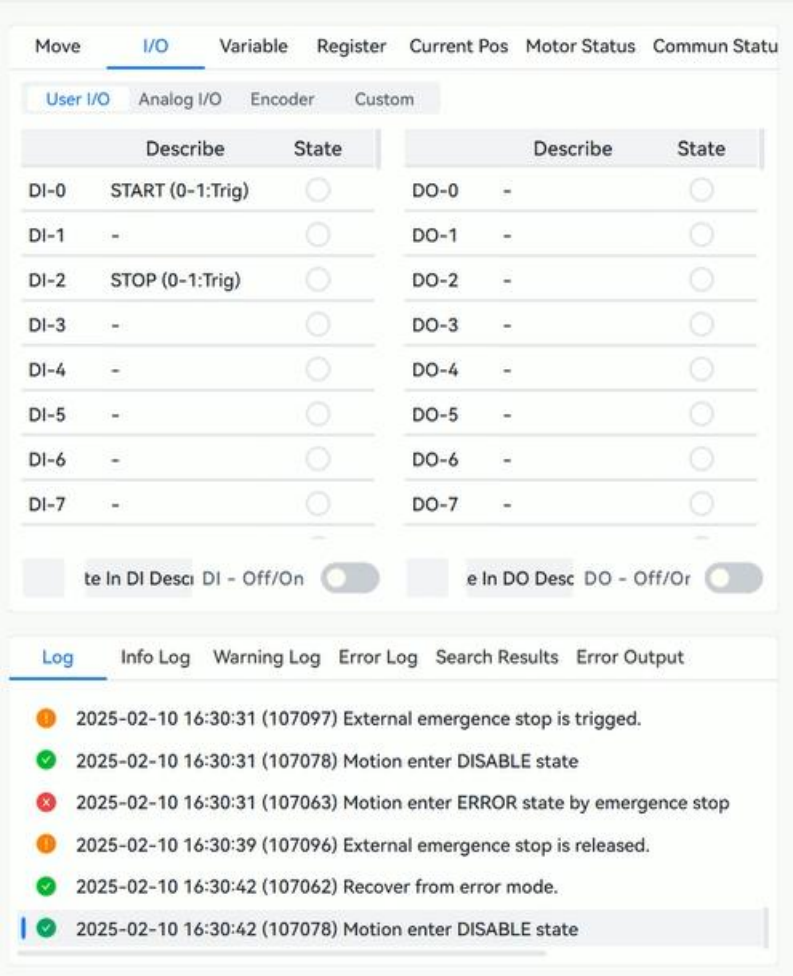
movements require multiple button presses.

Speed: The speed ratio of the robot's motion, effective only for the move function.

Current Position: Displays the current coordinates. For a four-axis robot, only the XYZU coordinates are shown, with a directional wheel button on the right side.

6.2 I/O

The I/O interface is divided into four sections: “User I/O,” “Analog I/O,” “Encoder,” and custom “I/O.” These sections allow for real-time monitoring or control of I/O outputs.



User I/O: Monitors the status of DI (Digital Input) or controls the output status of DO (Digital Output).

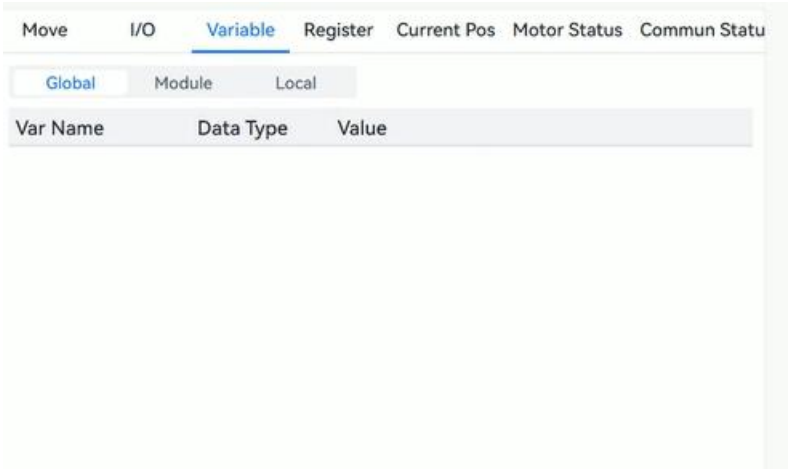
Analog I/O: Monitors the values of analog inputs (AI) or sets the output values for analog outputs (AO).

Encoder: Monitors the values returned by external encoders to the controller and sets the filtering values for monitoring.

Custom I/O: Defines the channels of user I/O, analog I/O, and encoders to be monitored.

6.3 Variables

The variable interface is divided into three sections: “Global module local”. Users can monitor or modify the variable information used in the active project here.



Global Variables: Monitors variables defined by the Global function in the program.

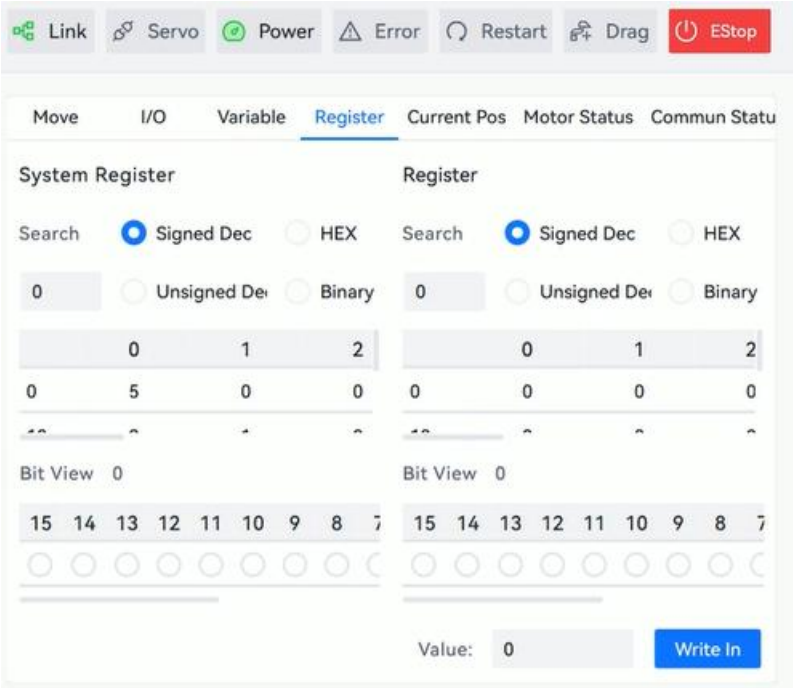
Module Variables: Monitors variables within a specified module that are not inside functions.

Local Variables: Monitors variables within a specific function of a specified module, which must be visible at a breakpoint in the function.

Modify Value: After selecting the variable to be modified, enter the desired value in the input box below and click “Write” to complete the modification of the variable's value.

6.4 Registers

The register interface is divided into two sections: “System Register” and “Registers.” Users can monitor the values of system registers or user registers here and also modify the values within the registers.



System Register Address Range: Decimal (0 to 1023), Hexadecimal (0x0000 to 0x0399)

Register Address Range: Decimal (0 to 10239), Hexadecimal (0x0400 to 0x27FF)

Signed Decimal: Decimal values with ± signs, range (-32768 to 32768)

Unsigned Decimal: Decimal values starting from 0, range (0 to 65535)

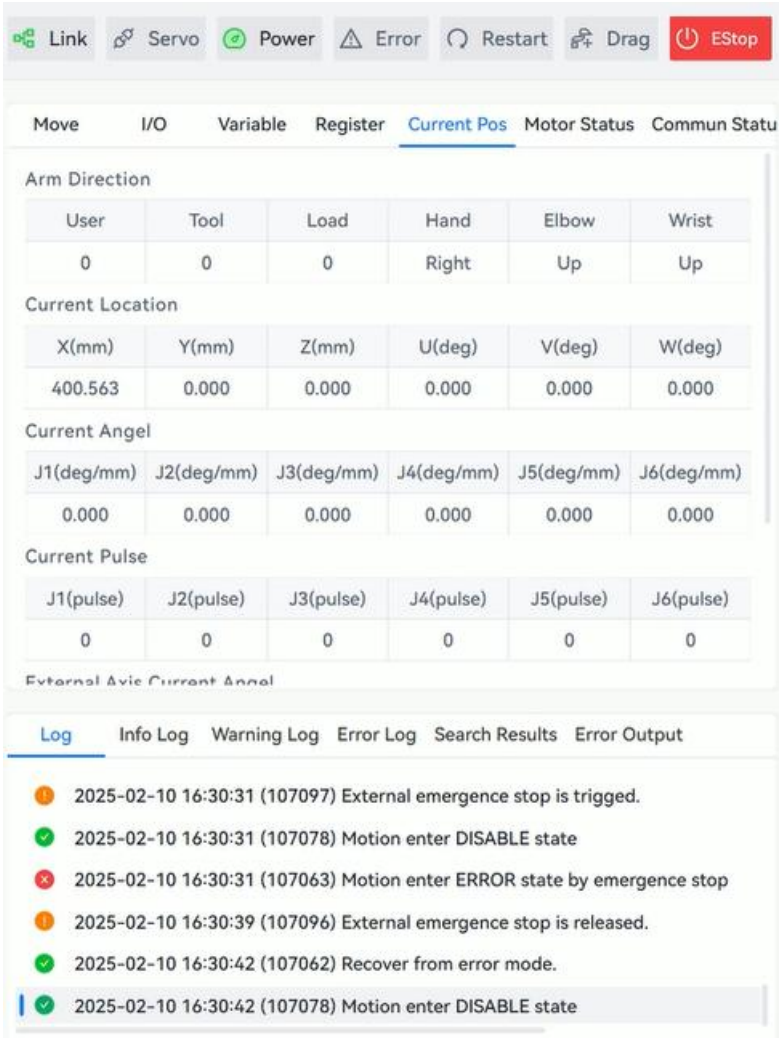
Hexadecimal: Values are taken in units of words, with a maximum value

of 0xFFFF.

Binary: Values are taken in units of bits, with each register having 16 bits.

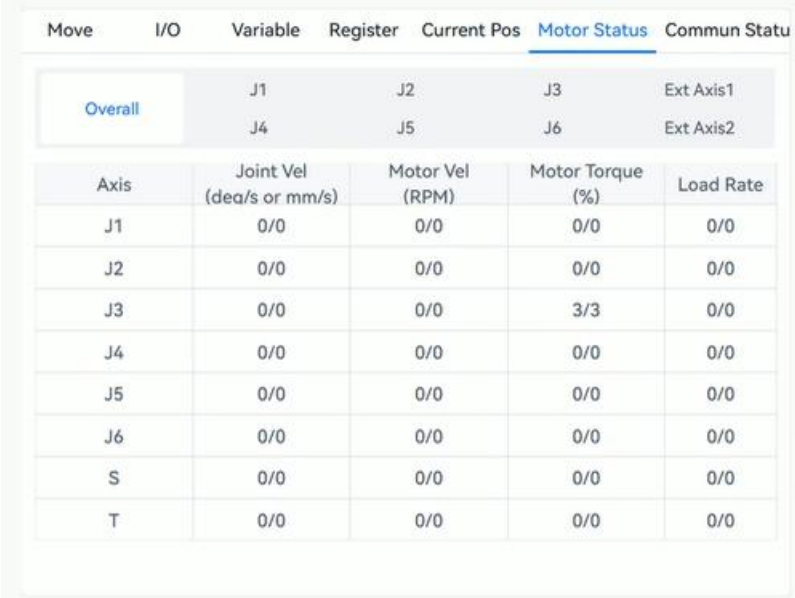
6.5 Current Position

Displays the current robot coordinates, including the user coordinate system, tool coordinate system, payload number, current gesture, elbow pose, wrist pose, as well as the world coordinates, joint coordinates, and single-joint pulse positions.



6.6 Motor Status

Displays the current status of all motors (joint velocity, motor speed, motor torque, load rate) and detailed information for individual motors.



Joint Vel: The motor speed of the specified joint and the actual movement speed, displayed in millimeters per second.

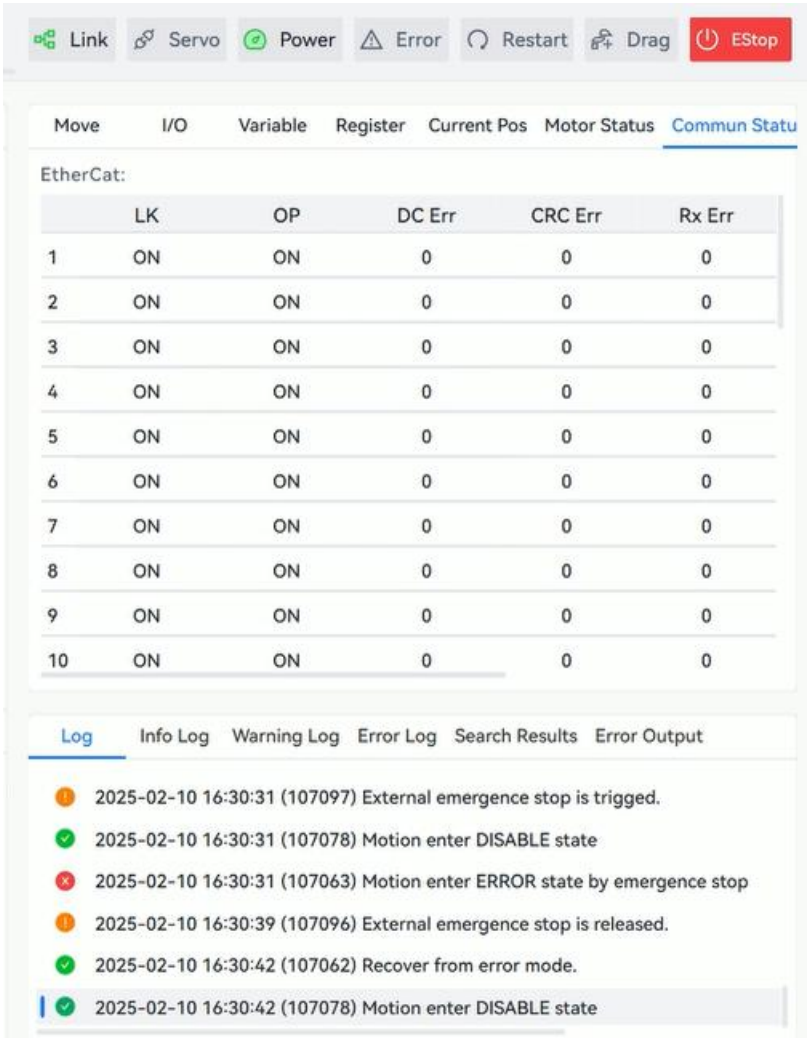
Motor Vel: The current rotational speed of the specified joint, displayed in revolutions per minute.

Motor Torque: The torque generated by the motor's rotation, displayed as current torque/limit torque.

Load Rate: Function reserved, not yet activated.

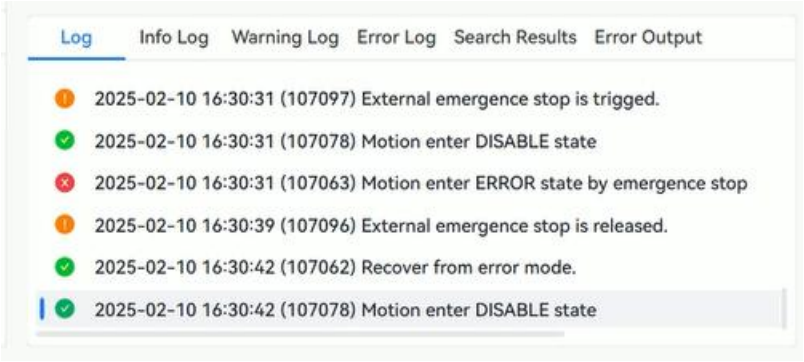
6.7 Commun Status

Displays the connection status of the current controller, driver, I/O board, and third-party peripherals.



6.8 Log

The log panel is located at the bottom right of the auxiliary zone. It is used for entering software operation logs, information prompts, warning messages, error messages, and search results.



Log: Outputs the content of the information panel, warning panel, and error panel.

Information Log: Outputs software prompts and information printed in the program.

Warning Log: Outputs warning messages prompted by the software.

Error Log: Outputs error messages generated by the servo system.

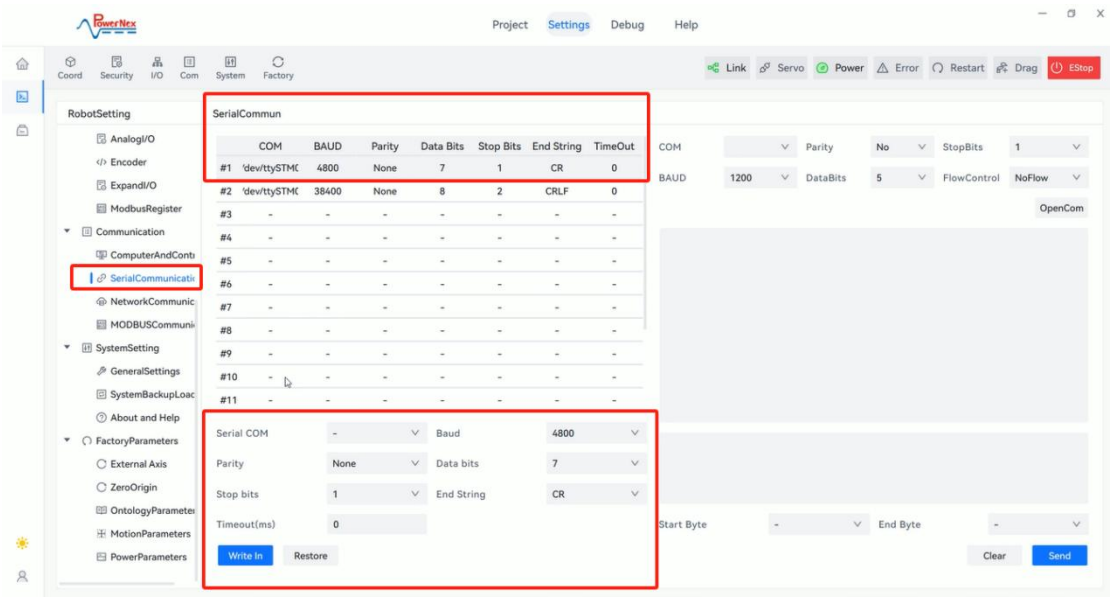
Search Results: Displays the results of the search function, indicating which program modules contain the keywords and the line numbers where the keywords appear within the modules.

Chapter 7 Appendix

7.1 Serial Communication Example

When performing serial communication, it is necessary to configure the communication port and enable it within the program.

Communication Port Configuration Example:



Serial Communication Program Example:

```
Function main

Integer AA

ReConnect:

SetCom #2,38400,8,2,N,CRLF,NONE,NONE,0

OpenCom #2
```

Note: This program is provided as an example only. When conducting

communication tests, please modify the example program according to the actual application!

The communication program must be written within the same function; it is not permissible to separate the establishment of communication from data interaction, as this may trigger alarms!

7.2 Network Communication Example

Since direct communication through port settings is not possible, this is only used for information monitoring. Typically, the network communication port is configured and opened within the program. The following is an example.

```
Function main

String CCD_X$,CCD_Y$,CCD_U$

ReConnect:

CloseNet #201                                   Close Port 201

SetNet #201 , "192.168.1.220" , 9000 ,CRLF  Configure port 201

OpenNet #201 As Client                          Open port 201

WaitNet #201, 5                 Wait for connection on Port 201 within 5 seconds

If ChkNet(201) =  -1 Then       Output feedback based on connection status

Print             Vision TCP Communication error, port is open but communication
has not been established

Wait 1

GoTo ReConnect
```

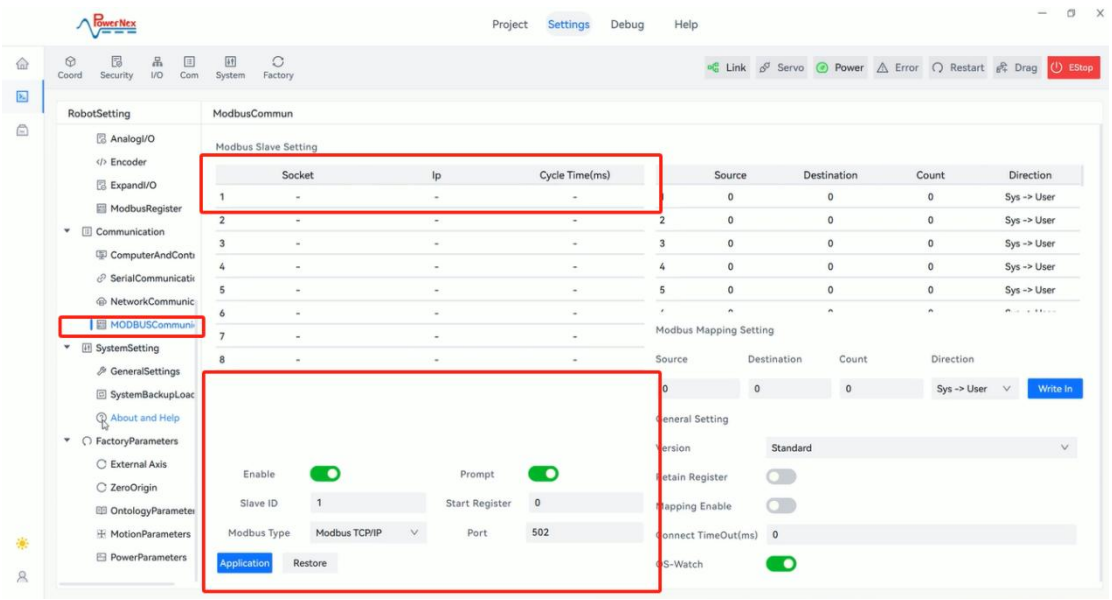
Note: This program is provided as an example only. When conducting communication tests, please modify the example program according to the

actual application!

The communication program must be written within the same function; it is not permissible to separate the establishment of communication from data interaction, as this may trigger alarms!

Suggestion: Use global variables for received content to make the data interaction content globally accessible!

7.3 Modbus Communication Example



Step 1: Select "Modbus Communication" in the settings panel.

Step 2: Choose the number for which communication needs to be configured.

Step 3: In the settings bar below, click "Enable" and configure the slave ID, Modbus type, and port.

Step 4: Click the blue "Apply" button.

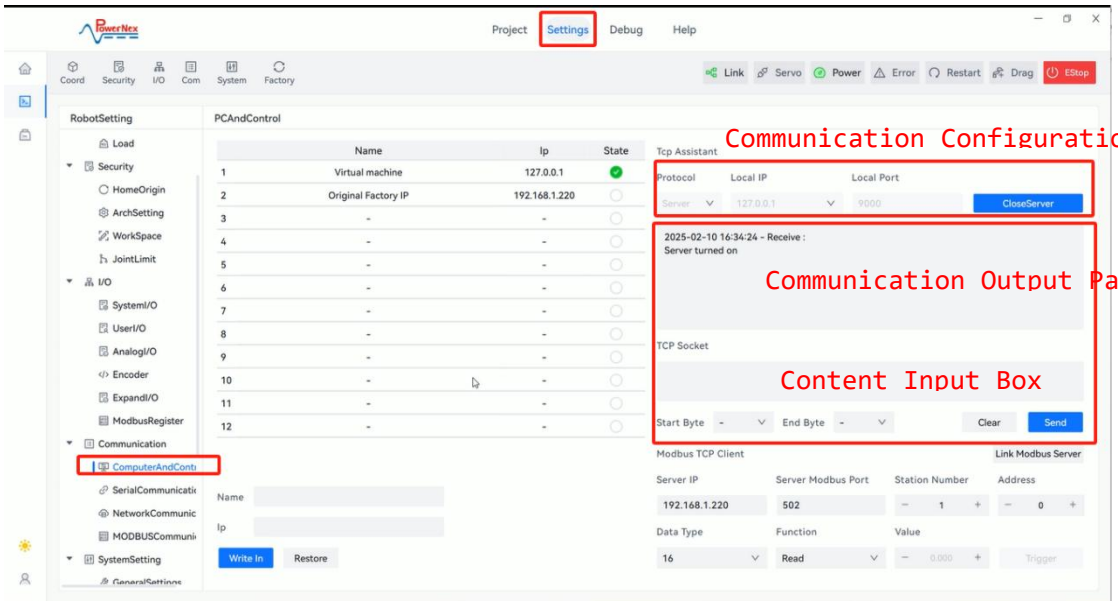
Step 5: Use a higher-level machine/PLC or network debugging assistant to read and write registers.

Note: In the current Modbus communication mode, the controller acts as a slave, while the higher-level machine/PLC or network debugging assistant acts as the master to access the controller. External access to the controller's register addresses for reading and writing is required.

For Modbus communication read/write examples, please refer to Section 7.4 Network Debugging Assistant.

7.4 Network Debugging Assistant

Operation Application of Network Debugging Assistant:



Step 1: Select "PC and Controller" in the settings panel.

Step 2: On the right side of the interface, configure the debugging assistant in the communication settings section. Protocol (server/client), local IP (server IP address), local port (0 ~ 65535), open/close server, or connect/disconnect server.

Step 3: Run the communication program to conduct communication tests.

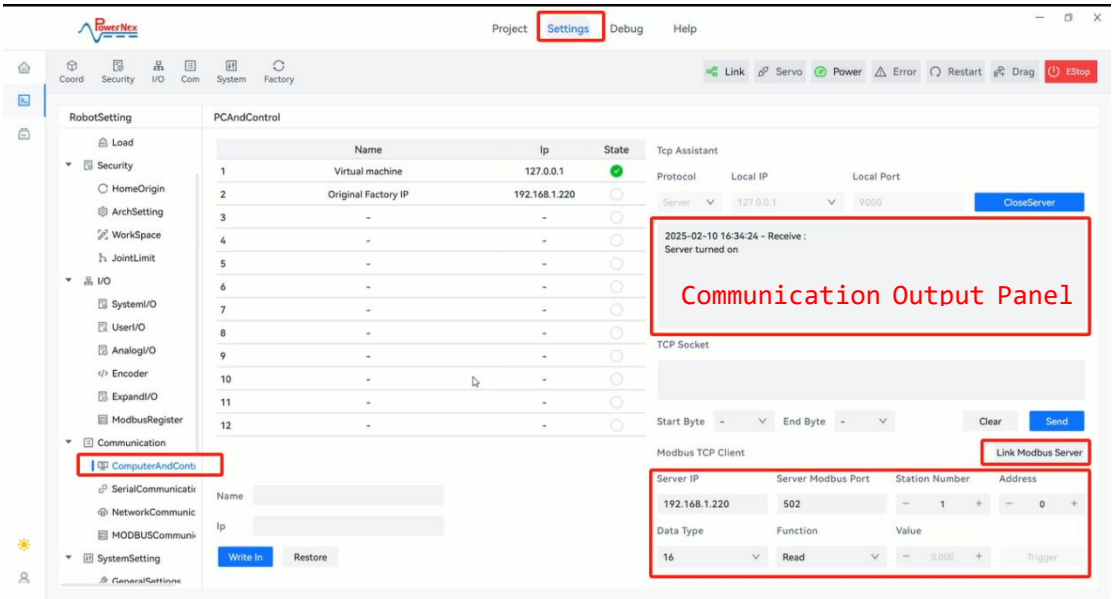
Content Input Box: You can enter the content to be sent in the input box and select the start and end characters required for communication.

Note: All communication feedback will be displayed in the "Communication Output Panel." Please check the panel content promptly when using the debugging assistant.

Pay attention to the set end character (CR carriage return, LF line feed, CRLF carriage return and line feed).

Warning: If the debugging assistant connects to a non-existent server as a client, operations will be restricted.

Modbus TCP Client Operation Application:



Step 1: Select "PC and Controller" in the settings panel.

Step 2: Set the IP, port, and station number of the Modbus slave in the lower right corner of the interface.

Step 3: Click the blue "Connect to Modbus Server" button.

Step 4: Check the "Communication Output Panel" to see if the connection to the server is successful.

Step 5: Set the read/write address, data type, function, value (fill in according to the function), and other content.

Step 6: Click the "Trigger" button to read and write to the Modbus

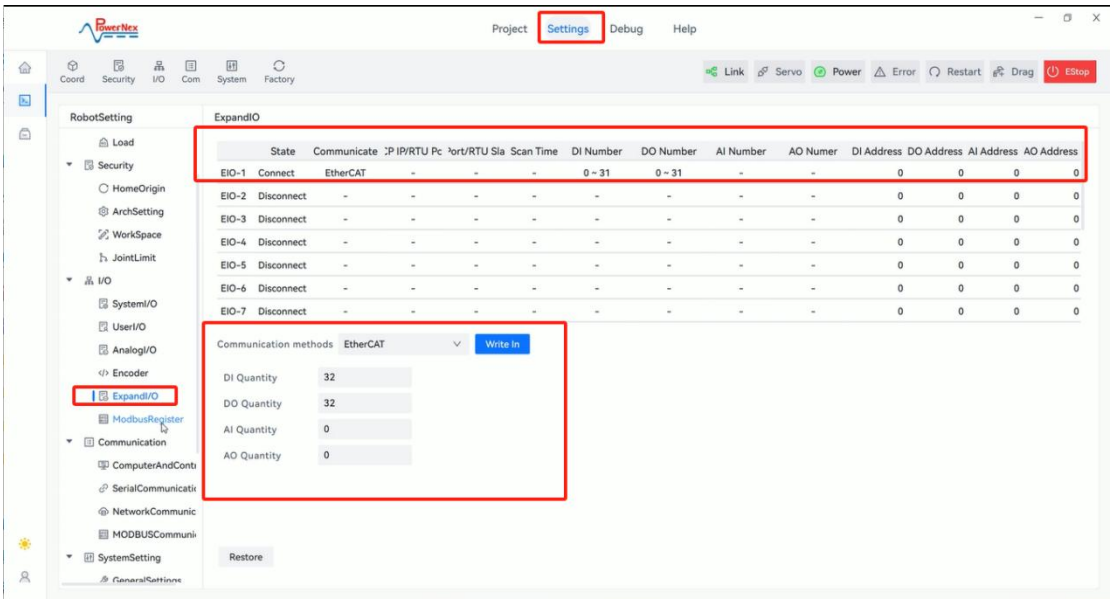
slave register addresses.

Tip: The "Ip" in the lower right corner of the interface is in decimal, not the hexadecimal in the Modbus message.
System register decimal addresses (0 ~ 1023);
User register decimal addresses (1024 ~ 10239).

Note: All communication feedback will be displayed in the "Communication Output Panel."
Please check the panel content promptly when using the debugging assistant.

7.5 Hardwired I/O Placeholder Example

When configuring register mapping I/O, EIP communication, or ProfiNet communication, it is necessary to prioritize the configuration of hardwired I/O interfaces. Otherwise, system errors may occur, causing hardwired signals to become invalid or the controller to crash.



Step 1: Select "Extend I/O" in the settings panel.

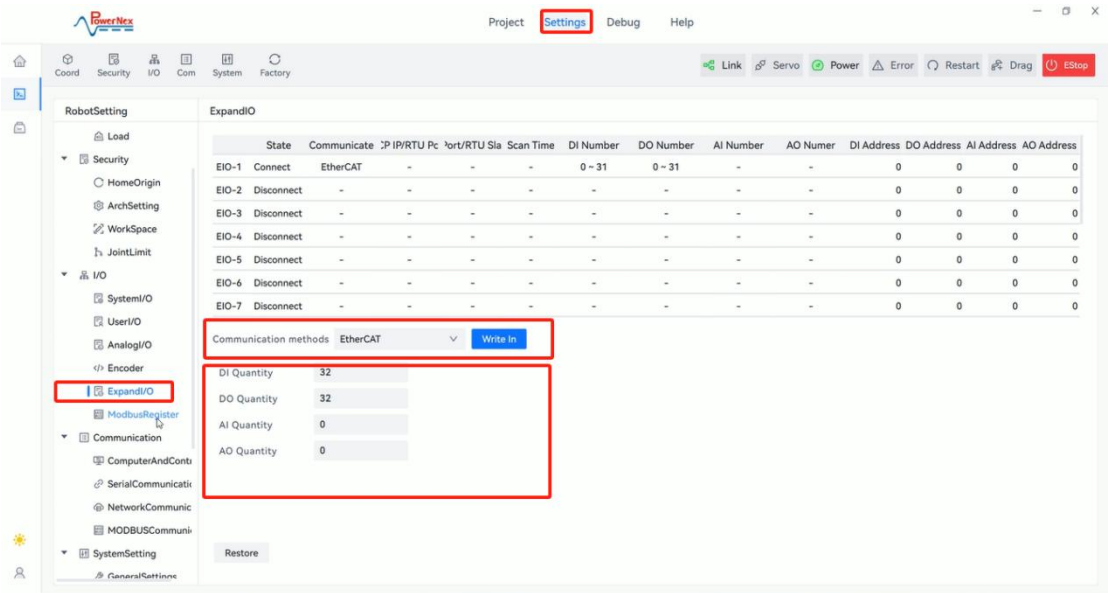
Step 2: Choose a number for configuration, prioritizing "EIO-1."

Step 3: In the settings bar below, select EtherCAT as the communication method and enter 32 for the number of DI and 32 for the number of DO.

Step 4: Click the blue "Write In" button.

7.6 Register Mapping I/O

Before configuring register mapping I/O, please configure the hardwired I/O placeholder first. Otherwise, system errors may occur, causing hardwired signals to become invalid or the controller to crash. For hardwired I/O configuration, please refer to Section7.5 Hardwired I/O Placeholder Example.



Step 1: Select "Extend I/O" in the settings panel.

Step 2: Choose a number for configuration, other than "EIO-1."

Step 3: In the settings bar below, select "Register" as the communication method and enter the number of DI and DO to be mapped.

Step 4: Separate the starting addresses of the DI-mapped registers and the DO-mapped registers to avoid mapping DI and DO to the same address content.

Step 5: Click the blue "Write In" button.

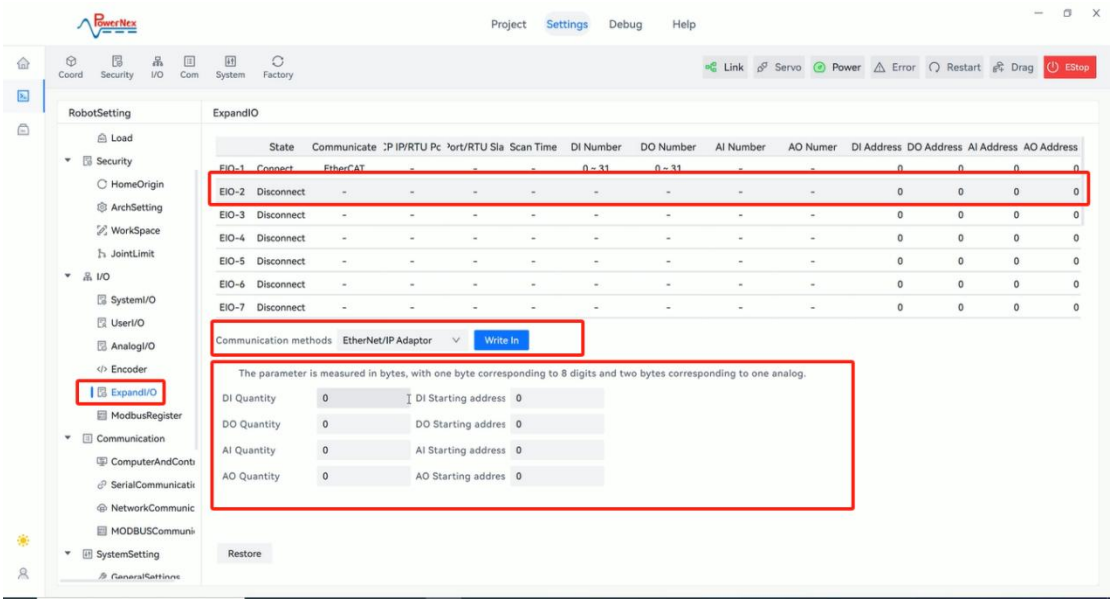
Note: The method of register mapping DIO is one register address corresponding to 16 DI/DO.

If mapping AIO, one register corresponds to one AIO.

The operation method is similar, so please allocate addresses reasonably when configuring.

7.7 EtherNet/IP Communication Configuration

Before configuring EIP communication, please configure the hardwired I/O placeholder first. Otherwise, system errors may occur, causing hardwired signals to become invalid or the controller to crash. For hardwired I/O configuration, please refer to Section 7.5 Hardwired I/O Placeholder Example.



Step 1: Select "Extend I/O" in the settings panel.

Step 2: Choose a number for configuration, other than "EIO-1."

Step 3: In the settings bar below, select "EtherNet/IP Adapter" as the communication method and enter the number of DI, DO, AI, and AO to be mapped.

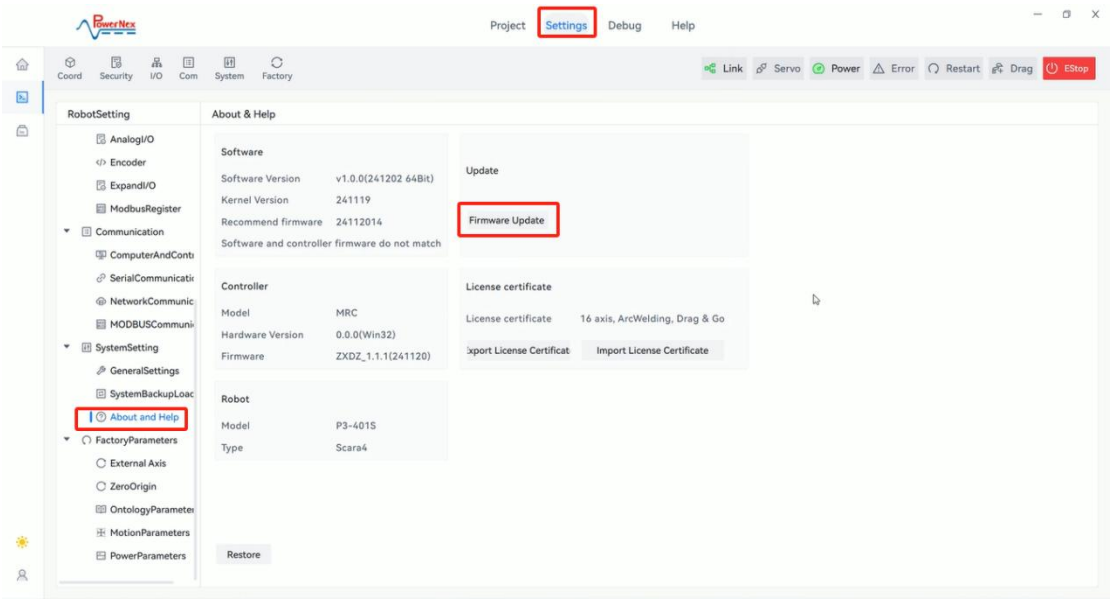
Step 4: Separate the starting addresses of the DI-mapped and DO-mapped addresses to avoid mapping DIO and AIO to the same address content.

Step 5: Click the blue "Write" button.

Note: The EDS file supports a maximum of 480-byte communication groups. There are 1024 DIO channels in total, with 32 hardwired placeholders remaining, so the maximum number of DIO channels for EIP communication configuration is 992.
Since 992 channels / 8 bits (1 byte) equals 124, the remaining 356 bytes can be configured for AIO.
One AIO channel is 2 bytes, so the maximum number of AIO channels for EIP communication configuration is generally 178.

Warning: When configuring, it is essential to separate the starting addresses of DIO and AIO to avoid mapping them to the same address content.

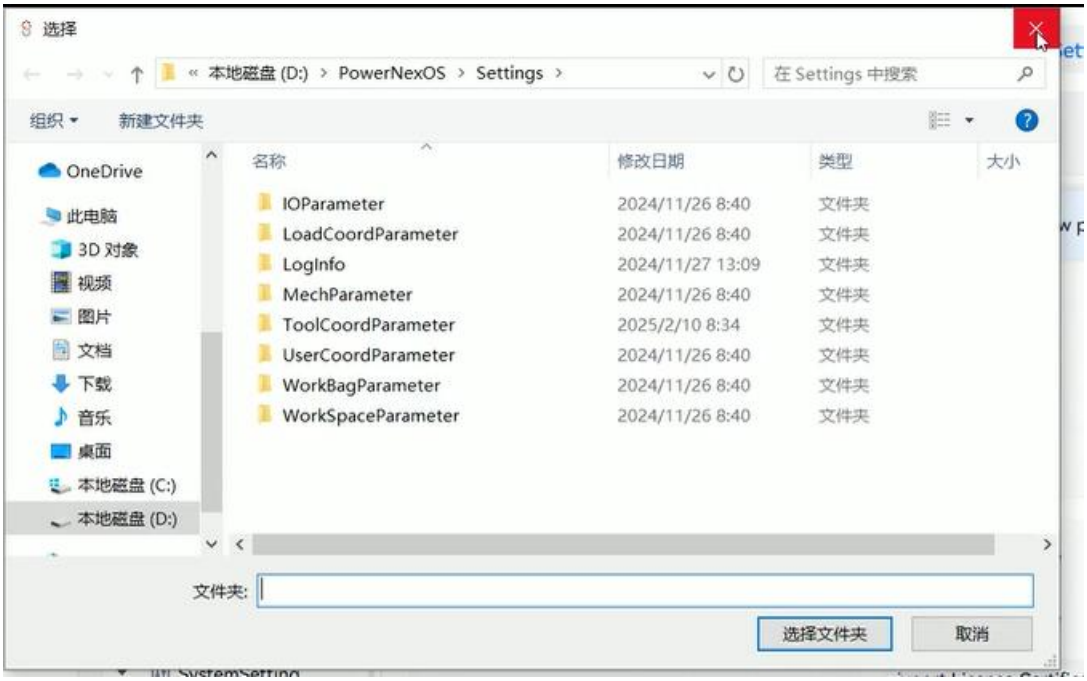
7.8 Firmware Update



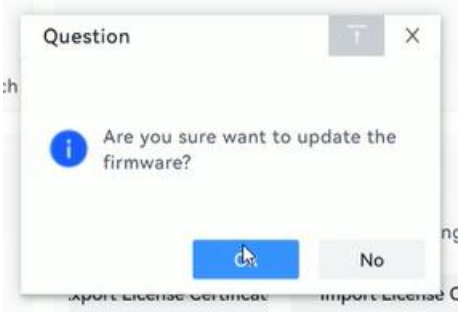
Step 1: Select "About and Help" in the settings panel.

Step 2: Click "Firmware Update."

Step 3: Select the firmware package path in the pop-up window.



Step 4: Click "OK" in the prompt window.

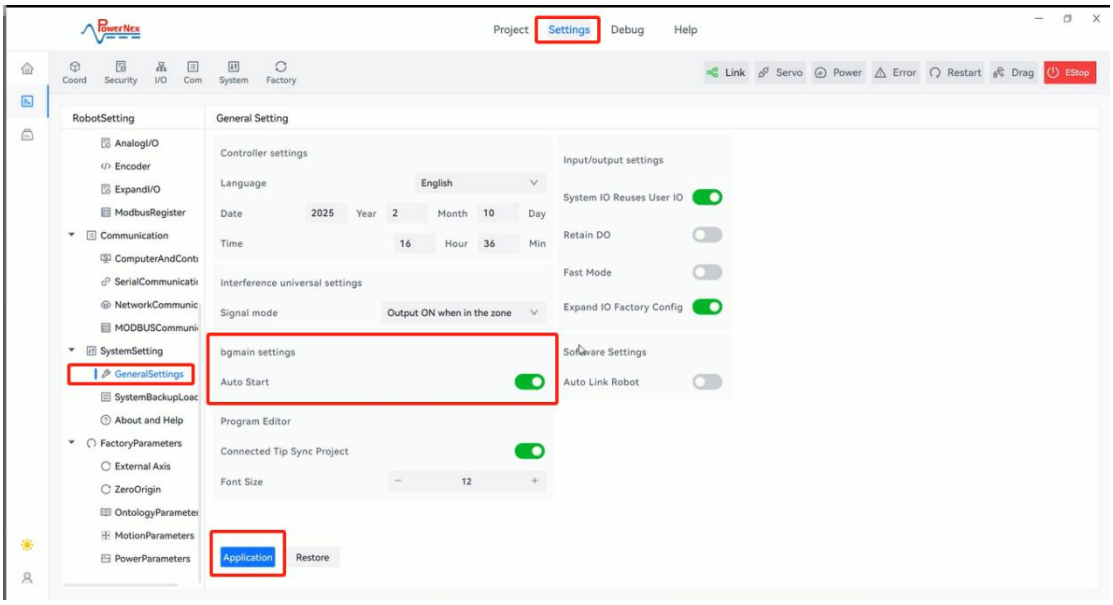


Step 5: Wait for the controller to restart. The firmware update time is approximately 5 to 8 minutes, so please be patient.

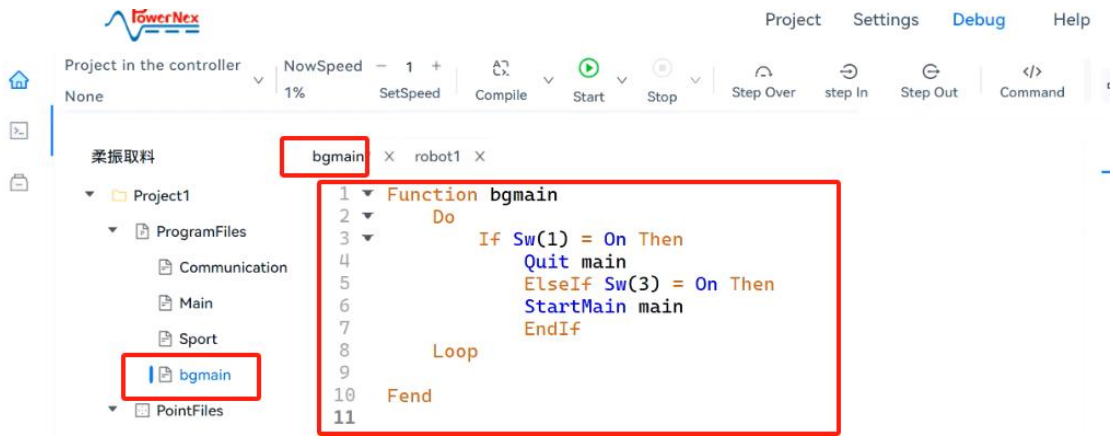
7.9 Background Task

The background task, referred to as bgmain in this software, is the program that automatically runs after the controller is powered on. The content executed in the task is written by the user. Whether the controller automatically runs the program after startup is also chosen by the user.

Enabling Automatic Background Task Execution:



Background Task Program Example:



Note: After completing the bgmain program, it must be saved and built into the controller; otherwise, the controller will not be able to reference the bgmain program upon startup.

Warning: The bgmain program file prohibits the use of motion commands and servo control status.