EtherCat Communication Function Manual

Revision History

Version	Revised Content	Revised by	Applicable Version(s)	Revised Time
V1.0	New	Fu Guocheng		2025.7.04
V1.1	Add EtherCAT Application Examples	Wang Guofang		2025.8.25

Preface

About This Manual

This manual introduces the support and operation of YiKingSoft software for external bus communication functions.

The current manual is applicable to all official releases from version 3.1.9 (9.9ru2) and later.

Prerequisites

Before using this software, users should have basic knowledge of external communication protocols and be familiar with industrial robots and related technical terminology.

Target Audience

- Robot Operators
- Robot Product Technical Personnel
- Robot Technical Service Personnel

Symbols and Conventions

Format	Meaning
< >	Text enclosed in angle brackets "< >" indicates a button name, for example, "Click the <OK> button".
【】	Text enclosed in square brackets "【 】" indicates window names, menu names, and data tables, for example, "The【User Login】window pops up".
/	Multi-level menus are separated by "/", for example, "System Settings/Modbus Settings".

More Manuals

Document Name	Document Number
YiKingSoft Software User Manual	RP-KZ-01-RJ1-001-C-V3.1

Kingkong Command Manual – Common Commands	RP-KZ-01-ZL1-018-C-V1.3

Contents

Chapter One Overview of EtherCAT Communication Applications

1.1 EtherCat Communication Protocol

EtherCAT (Ethernet for Control Automation Technology) is a high-performance industrial Ethernet communication protocol. By means of its unique “on-the-fly” processing mechanism, it achieves communication cycles at the microsecond level.

The protocol adopts a master-slave architecture, in which data frames are read and written by slave devices in real time during transmission. A single frame can serve multiple nodes, resulting in communication efficiency exceeding 90%.

Compared with traditional fieldbuses, EtherCAT offers higher speed (100 Mbps), lower latency, and better cost performance, and has become a mainstream communication standard in the field of industrial automation.

1.2 EtherCat Communication Application Scenarios

When the robot controller operates as an EtherCAT communication slave, the external master station can control the robot to perform operations. The main functions are as follows:

- Allows an external master device to control the robot for teaching operations, I/O control, and starting or stopping command programs.
- Allows the external master device to monitor the robot status.
- Allows users to create external communication variables (custom addresses) to communicate with external devices, enabling complex robot control.

Chapter Two EtherCAT Communication Configuration

2.1 EtherCat Slave Configuration

2.1.1 Hardware Selection

To use the EtherCAT communication function, please contact the manufacturer in advance to select the corresponding hardware options, and replace the controller's corresponding topology file.

2.1.2 Configuration Procedure

(1) Open the YiKingSoft software and click the <Parameter Settings> button to enter the 【Parameter Settings】 interface, as shown in Figure 2-1.

(2) Click "External Communication Settings/Bus Communication Settings" to enter the 【Bus Communication Settings】 interface, as shown in Figure 2-2.

(3) On the 【Bus Communication Settings】 page, enable communication and select the communication mode. Choose "EtherCAT" as the configuration option, as shown in Figure 2-2.

(4) On the 【EtherCAT】 page, configure the relevant communication data types, as shown in Figure 2-3.

(5) On the 【External Behavior Control】 page, advanced configuration options can be set to meet higher-level requirements, as shown in Figure 2-4.

(6) After all parameters are configured, click the <Save> button and restart the software to apply the settings.

Figure 2- 1

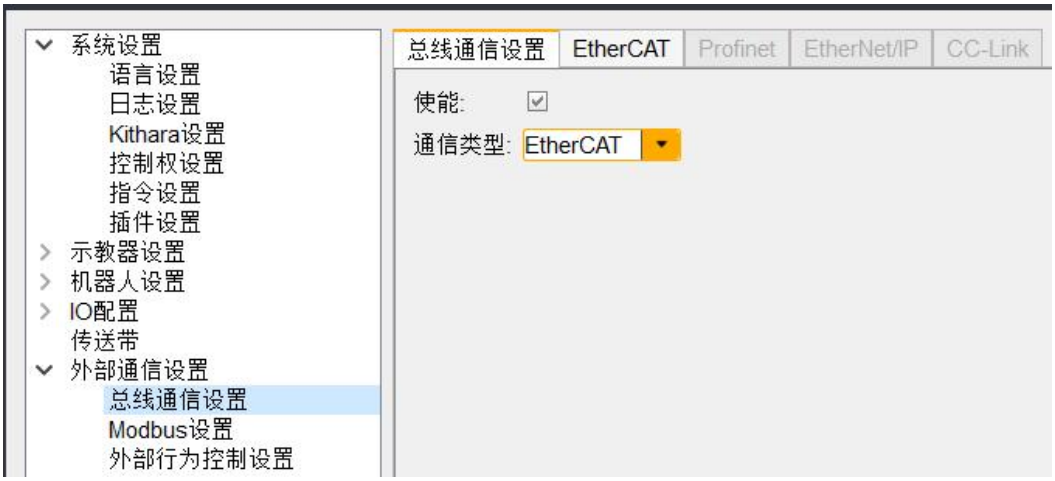


Figure 2- 2

Figure 2- 3

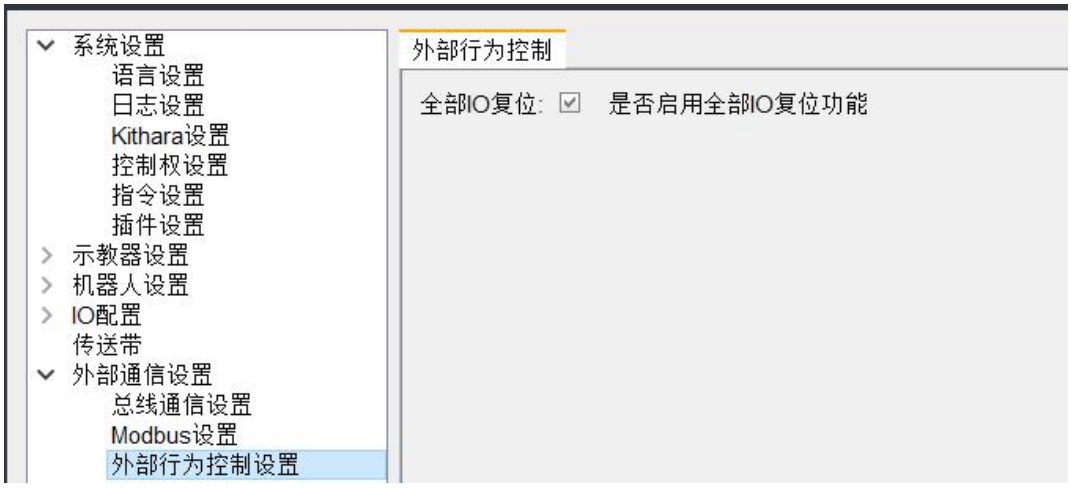


Figure 2- 4

2.1.3 Bus Communication Configuration Parameters

Name	Functions and Parameters
Enable	Enable Slave
Communication Type	EtherCat、Profinet、EtherNet/IP、CC-Link

2.1.4 External Behavior Control Configuration Parameters

Name	Functions and Parameters
Slave Integer Data Format	ABCD indicates sequential order, BADC indicates byte-wise reversal, CDAB indicates word-wise reversal, and DCBA indicates reverse order. Select the appropriate format according to actual requirements.
Slave Floating-Point Data Format	The selection method is similar to that described for the slave integer data format and should be chosen based on actual needs.
Slave Character Data Format	Intel uses little-endian byte order, while Motorola uses big-endian byte order.

2.1.5 External Behavior Control Configuration Parameters

Name	Functions and Parameters
Reset All I/O	The external master station operation function code supports the “Reset All I/O” function. As this function may be easily misoperated by users and could cause safety issues, it is disabled by default and must be manually enabled by the user.

- After completing the settings, click the <Save> button to save the configuration.
- Restart YiKingSoft to make all changes take effect.

2.2 EtherCAT Variables

2.2.1 Variable Management (Add, Delete, Modify, and Query)

(1) Open the YiKingSoft software and click the <Robot Programming> button to enter the 【Robot Programming】 interface.

(2) Click the “Variables” tab on the right, then locate the External Communication tab. As shown in the figure below, this interface allows users to add, delete, modify, and view external communication variables.

(3) Click the <Add> button to enter the variable creation interface. On the【Basic】tab, complete the editing of the variable's basic information. The interface is shown in the figure below.

(4) On the 【EtherCAT】 tab, complete the editing of external communication–related information. The interface is shown in the figure below.

(5) Click the <Add> button to add the variable, or click the <Cancel> button to discard the current operation.

(6) Double-click a variable item in the “EtherCAT” tab to modify the corresponding variable.

2.2.2 Basic Tab Parameters for Variable Editing

Name	Function			
Name	Enter the variable name. The name may consist of letters, numbers, and underscores, must start with a letter, must not be a reserved keyword, and must not duplicate an existing variable name.			
Type	Variable	Storage	Range	Precision
	Bool Type	1bit	False/True	
	Int Type	1word	-32768—32767	
		2word	-2147483648—2147483647	
	Double Type	2word	-/+3.4e38	Displays 4 decimal places, with a maximum of 7 significant digits.

		4word	-/+1.7e308	Displays 4 decimal places, with a maximum of 16 significant digits.
	String Type	Custom Length		
Number of Variables	Number of variables to add			
Value	Initial value of the variable: enter an appropriate value according to the variable type			
Applicable Range	false: Do not set the variable's value range. The Maximum and Minimum fields cannot be entered. true: Set the variable's value range. The Maximum and Minimum fields can be entered.			
Minimum	Maximum Value			
Maximum	Minimum Value			
Common Attributes	false: This variable is not visible in the variable interface during program execution. true: This variable is visible in the variable interface during program execution.			

2.2.3 External Communication Tab Parameters for Variable Editing

Name	Function
Address Type	Input/Output

Address	Enter Address
Bit Address	(Visible when the variable type is set to Boolean) Enter the bit address (range: 0–15). Example: For a Boolean variable test_bool with Address = 2 and Bit Address = 10, the status of bit 10 at Address 2 corresponds to the status of test_bool.
Address Length	(Visible when the variable type is not Boolean) Select or enter the number of addresses. For INT variables: Currently supports 16-bit and 32-bit integers; you can choose 1 or 2 addresses. For DOUBLE variables: Currently supports 32-bit and 64-bit doubles; you can choose 2 or 4 addresses. For STRING variables: The length must be defined according to the length of the string to be sent or received, with one character occupying one byte.

2.3 EtherCat Variable Usage

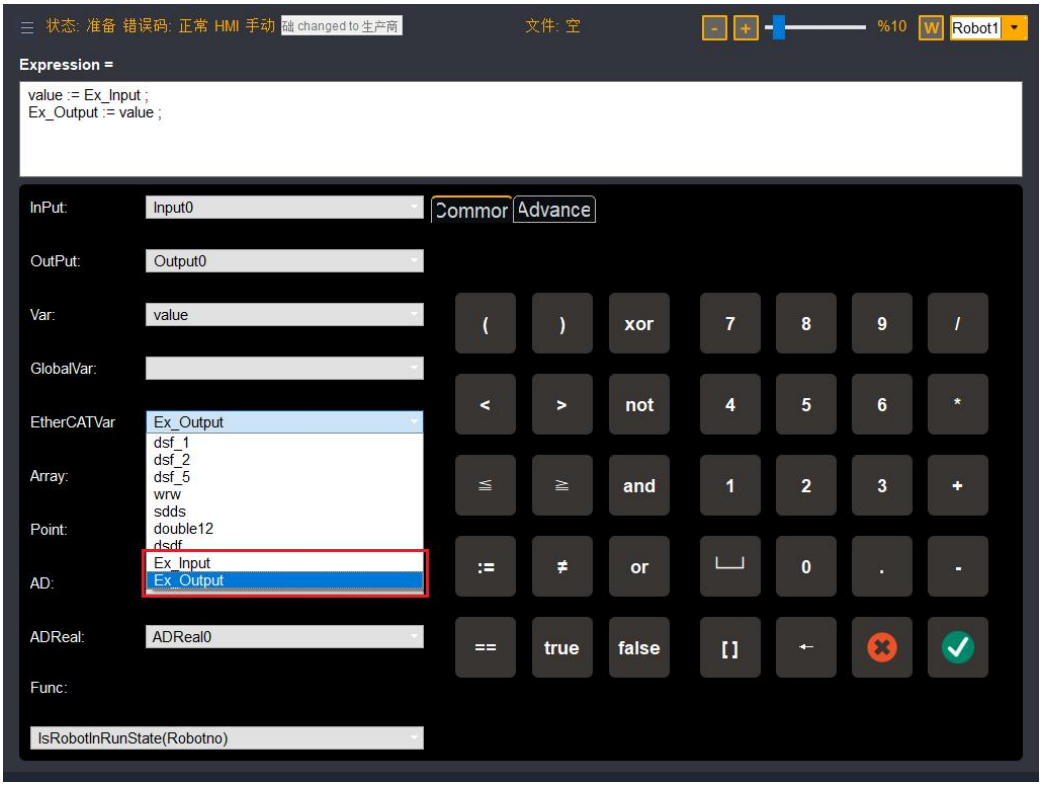
2.3.1 Variable Value Monitoring

After the external communication is established, the current values can be refreshed in real time on the 【EtherCAT】 Variable Editing interface.

When the robot is in automatic operation mode, variable values can be monitored in real time under 【Debug】 → 【Runtime Variables】.

2.3.2 Variable Operations

Expressions support read and write operations on external communication variables. The expression editing interface is shown in the figure below.


状态: 准备 错误码: 正常 HMI 手动
文件: 空
%10
Robot1
Expression =
value := Ex_Input ;
Ex_Output := value ;
InPut:
Input0
Commor
Advance
OutPut:
Output0
Var:
value
GlobalVar:
EtherCATVar
Ex_Output
dsf_1
dsf_2
dsf_5
wrw
sdds
double12
dsdf
Array:
Point:
AD:
Ex_Input
Ex_Output
ADReal:
ADReal0
Func:
IsRobotInRunState(Robotno)
( ) xor 7 8 9 /
< > not 4 5 6 *
≦ ≧ and 1 2 3 +
:= ≠ or 0 . -
== true false [ ]

Chapter Three External Communication Control Application

3.1 Read Robot Status

The robot's operational status is acquired via EtherCat communication, where the robot functions as an EtherCat slave. The external master station retrieves the robot's status information by reading the corresponding addresses output by the slave.

EtherCat output addresses feature both default and custom states. You can view the default state or configure custom states through the Robot Software/Function Operation/EtherCat Output Address Configuration interface.

3.1.1 Default Output Address Configuration

By default, all output addresses are set to robot 1. To get other robot statuses or information, use custom configuration.

Default output address settings

(1) Select the status row to set. For example, selecting location X displays the current status row information below.

(2) Modify its parameters as needed. The default state supports the following changes:

Position X: Address length (1 or 2);

Position Y: Address length (1 or 2)

Position Z: Address length (1 or 2)

Location RX: Address length (1 or 2)

Location RY: Address length (1 or 2)

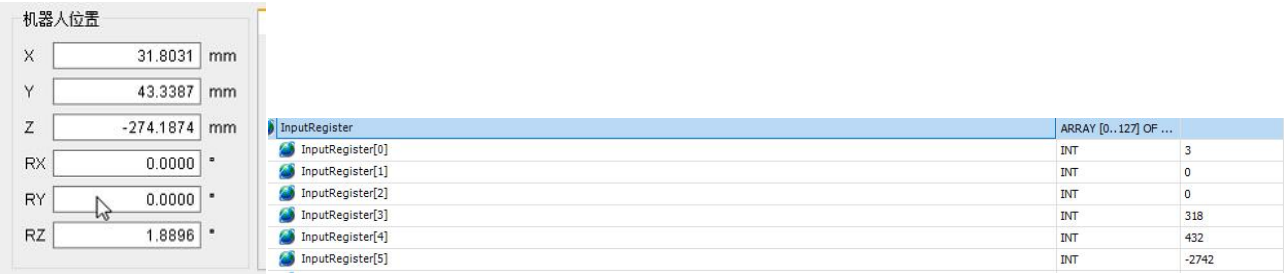
Position RZ: Address length (1 or 2)

(3) Save the changes after modifying a status line.

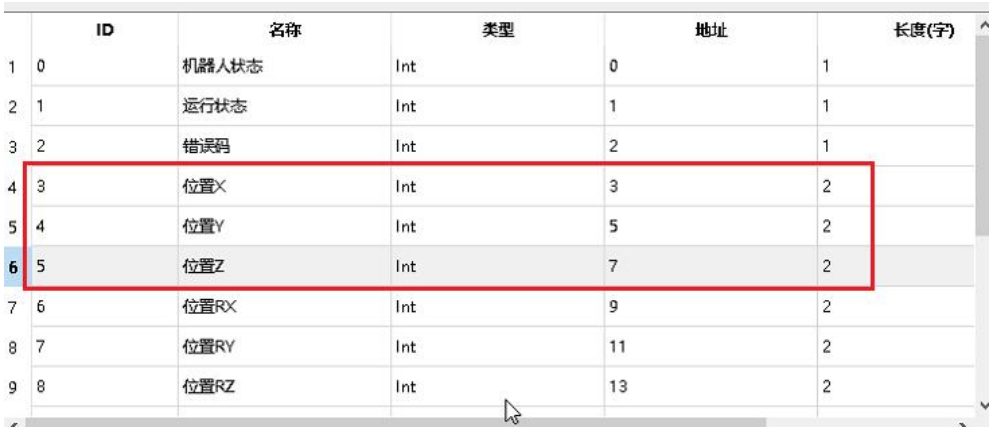
(4) Restart to apply after all settings are completed.

Feature Instructions

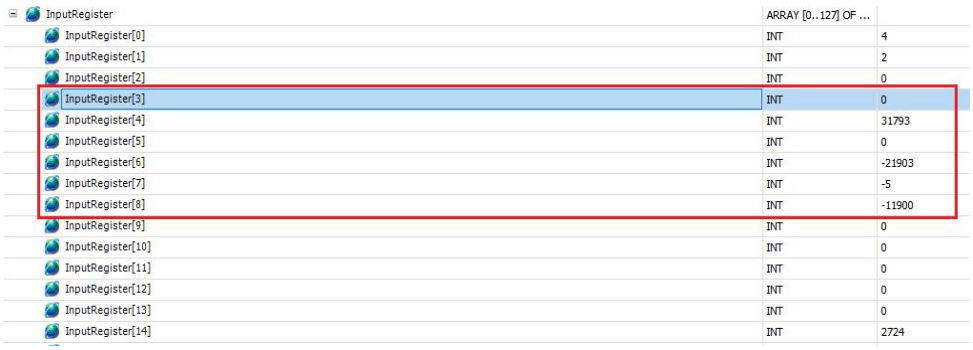
- The default address is the last external slave address in the address allocation table.
- The robot's position information is stored as a signed integer in the input address. If the address length is 1, the position information unit is 0.1 mm. If the address length is 2, the position information unit is 0.001 mm.
- If the address length of the location information is 1, the corresponding address data is read and converted to a signed integer type.



- If the address length of the location information is 2, read the corresponding two address data and convert them to a signed integer type according to the slave integer type format. For example:

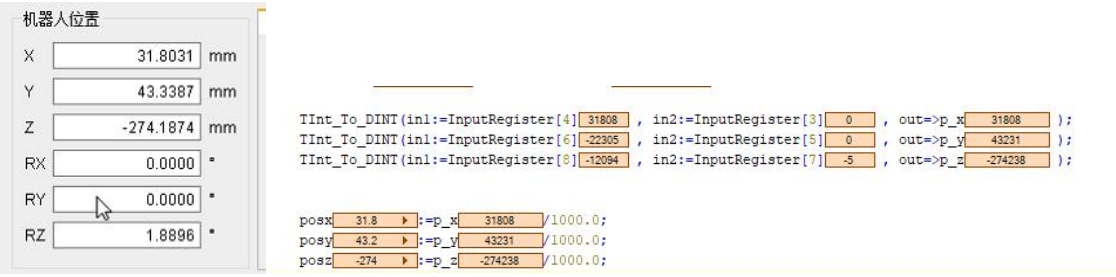


	ID	名称	类型	地址	长度(字)
1	0	机器人状态	Int	0	1
2	1	运行状态	Int	1	1
3	2	错误码	Int	2	1
4	3	位置X	Int	3	2
5	4	位置Y	Int	5	2
6	5	位置Z	Int	7	2
7	6	位置RX	Int	9	2
8	7	位置RY	Int	11	2
9	8	位置RZ	Int	13	2



InputRegister	ARRAY [0..127] OF ...	
InputRegister[0]	INT	4
InputRegister[1]	INT	2
InputRegister[2]	INT	0
InputRegister[3]	INT	0
InputRegister[4]	INT	31793
InputRegister[5]	INT	0
InputRegister[6]	INT	-21903
InputRegister[7]	INT	-5
InputRegister[8]	INT	-11900
InputRegister[9]	INT	0
InputRegister[10]	INT	0
InputRegister[11]	INT	0
InputRegister[12]	INT	0
InputRegister[13]	INT	0
InputRegister[14]	INT	2724

As shown in the figure above, positions X, Y, and Z are configured with a 2-byte address length and a starting address of 3. Specifically, position X reads data from input addresses 3 and 4, position Y reads data from input addresses 5 and 6, and position Z reads data from input addresses 7 and 8. These values are then combined according to the integer type format ABCD and converted into signed integers.



3.1.2 Customize Input Address Configuration

As shown in the figure below, the controller provides customizable input address configuration, allowing users to configure the target status information in consecutive addresses according their needs.

The provided robot information includes:

Robot status, operational status, error codes, position X, position Y, position Z, position RX, position RY, position RZ, manual/automatic mode, robot speed, input IO, output IO, heartbeat, current command number, etc.

The above states match the default state. Refer to the output address definitions in the slave address allocation table of the final chapter to understand the meaning of each state value.

The address must be continuous for custom status configuration.

Living example ：

As shown in the figure, three states have been configured: robot state, position X, and error code. To set the next address to position Y, follow these steps:

(1) Select the first unspecified row.

	ID	名称	类型	地址	长度(字)	机器人序号
1	0	机器人状态	Int	0	1	1
2	3	位置X	Int	1	1	1
3	2	错误码	Int	2	1	1
4	-1	未指定	Int	0	0	0
5	-1	未指定	Int	0	0	0

(2) Change the name, bot ID, and address length.

(3) Click Save button.

(4) Setup complete.

(5) To enable custom status, select Custom Status and click Save, as shown in the figure below.

Initialize

To initialize a configuration, select the row to be initialized, change the name to unspecified, and click Save. If other configured address states exist after this address, they will also be initialized.

3.2 Robot Automatic Mode Control

When the robot acts as an EthrerCat slave with the control authority of the bus communication unit, the external master station can control the robot's functions such as running programs, error reset, enabling, and speed adjustment by writing the input address.

The first 25 characters of the input address are system addresses for robot control, where 0-9 are the robot control area, enabling functions such as start-stop programs in the remote automatic state of the robot.

View and switch control rights as shown in the figure below.

注意

- Since the software monitors the bit transition in the address, to execute the same function consecutively, assign a non-functional code value (default 0 recommended) to the corresponding address bit after the first execution, then repeat the function.
- When the external master station remotely controls the robot, the robot performs a legitimacy check, and any illegal operation will be invalidated. The legitimacy check typically includes aspects such as 'control authority', 'manual-automatic working mode', 'robot status', and 'whether functions are enabled'.

3.2.1 Start/Stop and Open

The robot allows the user to control the opening of the robot instruction program through external communication. After the instruction program is successfully opened, the current program's instruction number can be queried in the "Input Address" robot status area.

- Instruction number address: Specifies the instruction number for the robot's loading or execution. Refer to the' Input Address 'section in the' Robot Slave Address Allocation Table' chapter for details.
- Start/Stop/Address: Controls the program's startup, shutdown, and address access functions. For specific addresses, refer to the' Input Addresses 'section in the' Robot Slave Address Allocation Table' chapter.
- External startup command storage location: Cmds\ExStartCmd folder.
- External startup command format: Only programs matching the external command format can be correctly recognized and loaded. The interface setup is shown below, with Cmd_XXX format as the default.

Figure 3-1 [Parameter Settings/Robot Settings]

- External boot instruction loading: Specifies the loading mode for external boot instructions.

  "Before Startup": The default setting requires the command program to be opened before the command is executed. Otherwise, the command will fail to start due to a mismatch between the command number and the program number.

  "On Startup": The program loads upon instruction execution. Before the startup command is issued, no prior program execution is required. The startup instruction follows either the specified instruction number upon receiving the startup signal or the currently active program.

Figure 3-2 [Parameter Settings/Robot Settings]

- Start confirmation: Checks the "start confirmation signal" when the external start command program is

executed to ensure the safety of robot startup. The default is "Enable", meaning the startup signal is only valid within 5 seconds after receiving the "start confirmation" signal.

Figure 3-3 [Parameter Settings/Robot Settings]

- Operation legitimacy: The program must be valid when the bus communication unit is in control, automatic mode, and ready state.

Robot 1 start-stop open instance

(1) Write the external startup command program: Cmds\ExStartCmd\Cmd_001.xml

(2) The robot is cut to the control of the "bus communication unit", and the robot is in the "ready state" in the "automatic" working mode.

(3) Set the start instruction number: Write 1 to output address 0, representing instruction 1.

(4) To activate the command program: Enter 103 (robot ID*100+3) at address 1 to launch the program. The robot's [Operation Interface] will then display the loaded program name and its command details.

Enter the address via the robot to check the currently loaded instruction number.

(5) Start confirmation: Enter the function code 1001 (robot number * 1000 + 1) in the address field on the 25th, and complete the next step within 5 seconds.

(6) Initiate the command program: Write 101 (robot ID*100+1) to input address 1 to execute the command program. The robot's [Run Interface] will then display the current status of the program.

(7) Stop instruction program: Write 102 (robot number*100+2) to input address 1, which indicates the stop instruction program. The robot's instruction execution ends, and it enters the' ready state'.

Start or stop instances on multiple machines simultaneously

(1) Write the external startup command program: Cmds\ExStartCmd\Cmd_001.xml

(2) The robot is cut to the control of the "bus communication unit", and the robot is in the "ready state" in the "automatic" working mode.

(3) Set the start instruction number: Write 1 to address 0 to represent instruction 1.

(4) To activate the instruction program: Enter 3 as the address in field 1 to launch the program. The robot's [Operation Interface] will then display the loaded program's name and details.

Enter the address via the robot to check the current loaded instruction number for each robot.

(5) Start confirmation: Enter each robot's start confirmation code (robot number * 1000 + 1) in the address field on the 25th, and complete the next step within 5 seconds.

(6) To initiate the instruction program: Enter 1 at address 1 to execute the program. The robot's [Operation Interface] will then display the status of each instruction program.

(7) Stop instruction program: Write 2 to the input address 1 to execute the stop instruction program. The robot instruction ends execution, and all robots transition to the 'ready state'.

3.2.2 Instruction Pause and Resume

- Pause and resume address: controls the pause and resume of the robot's instruction program. Refer to the' Input Address 'section in the' Robot Slave Address Allocation Table' chapter for details.

3.2.3 Error Reset

- Error reset address: Used to reset the robot's error. For details, see the' Input Address 'section in the' Robot Slave Address Allocation Table' chapter.

3.2.4 Emergency Stop Reset

- Emergency stop reset address: This address resets the robot's emergency stop status. For details, refer to the' Input Address 'section in the' Robot Slave Address Allocation Table' chapter.

3.2.5 Automatic Speed Control

The robot's running speed can be controlled automatically through external communication, and the real-time running speed of each robot can be queried in the robot status area of "output address".

- Speed value address: Specifies the robot's speed value. The default is 0, and the valid range is [1,100]. For

details, see the Input Address section in the Robot Slave Address Allocation Table.

- Speed control address: controls the automatic speed function. For details, see the input address section in the 'Robot Slave Address Allocation Table' chapter.
- Control validity: Custom mode with no cruise control command.

3.3 Robot Manual Mode Teaching

When the robot operates as a slave and is in “Remote Manual” mode, the external master station can control the robot to perform teaching operations and other functions by writing to the input addresses.

Control Requirement: Bus communication control authority or “Multi-Point Teaching” must be enabled.

Multi-Point Teaching: When bus communication control authority is not active, this option determines whether the user is allowed to perform robot teaching operations via the master station. It is disabled by default. The configuration interface is shown below:

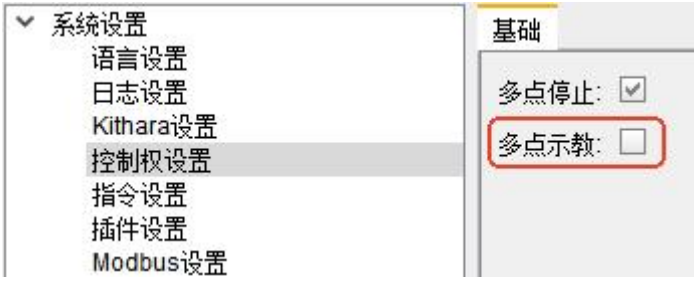


图 3- 4 【系统设置/控制权设置】

3.3.1 Teaching Function Code

Output addresses 25–30 correspond to addresses used for controlling the robot in manual mode. By writing a function code to input address 25 and supplying parameters through input addresses 26–30, the robot can be controlled.

Since the software detects edge changes in the addresses, to execute the same function consecutively, the corresponding address bit should be set to a non-functional code (recommended default value: 0) after each execution, and then the function can be executed again.

3.3.2 Robot Jog and Incremental Motion

Before introducing Robot Jog and Incremental Motion, the speed settings in Remote Manual mode should be explained.

- Remote Manual mode uses a separate speed setting, which can be modified remotely via function codes.

The speed settings are independent for each robot, and the YiKingSoft speed bars do not display Remote Manual mode speeds synchronously.

- The speed in remote manual mode is not saved. When switching from other modes to remote manual mode, the speed in remote manual mode will be the initial value of 10.

Speed adjustment has two modes:

(1) Fine-tuning: Speed increases or decreases by 1.

(2) Speed level adjustment: Speed is divided into 8 levels: 1, 2, 5, 10, 25, 50, 75, and 100. Using the level adjustment will increase or decrease the speed from the current level to the adjacent level.

Robot Jogging

- Robot jogging refers to the robot continuously moving in a certain direction at its current speed until it stops.
- Robot jogging can only be used in remote manual mode.
- Use input addresses 25 and 27.

Robot Jogging Operation Steps:

(1) Check if the manual/automatic status in the output address is 2 (remote manual). If it is not remote manual, write the mode switching function code (101) in input address 25 to switch to remote manual.

(2) Write the enable function code (robot number * 1000 + 1) in input address 25 to switch the robot status to the paused state in the running state. This corresponds to robot status 4-running state and running state 2-pause state in the output address.

(3) Enter the current jogging coordinate system in input address 27: 0-world coordinate system, 1-base coordinate system, 2-TCP coordinate system, 3-Joint coordinate system.

(4) After entering the corresponding jogging function code in input address 25, the robot starts moving. You can check the jogging coordinate system in output address 26: 0-world coordinate system, 1-base coordinate system, 2-TCP coordinate system, 3-Joint coordinate system.

(5) Change the value of input address 25 to stop the robot.

Robot inch movement

- Robot inching refers to the robot moving to a specified position at the current speed and stopping after reaching the specified position.
- Robot inching can only be used in remote manual mode.
- To control robot inching from the master station, input addresses 25, 26, and 27 are required.

Robot Inching Operation Steps

(1) Check if the manual/automatic status in the output address is 2 (remote manual). If it is not remote manual, enter the mode switching function code (101) in input address 25 to switch to remote manual.

(2) Enter the enable function code (robot number * 1000 + 1) in input address 25 to switch the robot status to the paused state in the running state. This corresponds to robot status 4 - running state and running state 2 - paused state in the input addresses.

(3) Enter the current inching coordinate system in input address 27: 0 - world coordinate system, 1 - base coordinate system, 2 - TCP coordinate system, 3 - Joint coordinate system.

(4) Enter the movement step size (integer) in input address 26. The maximum value is 10000, and the unit is 0.01 mm.

(5) After entering the corresponding inching function code in input address 25, the robot begins to move. The inching step size (in 0.01 mm) can be viewed at output address 25, and the inching coordinate system (0-world, 1-base, 2-TCP, 3-Joint) can be viewed at output address

(6) Wait for the robot to move the specified step size before stopping. If you need to stop the inching earlier, simply change the value of input address 25.

3.4 Robot Position Variable Operations

Control Requirements: External master station control or "Multi-point Teaching" enabled.

Multi-point Teaching: Whether to allow users to perform robot teaching operations through the master station when not under "Bus Communication Control"; disabled by default. The configuration interface is as follows:

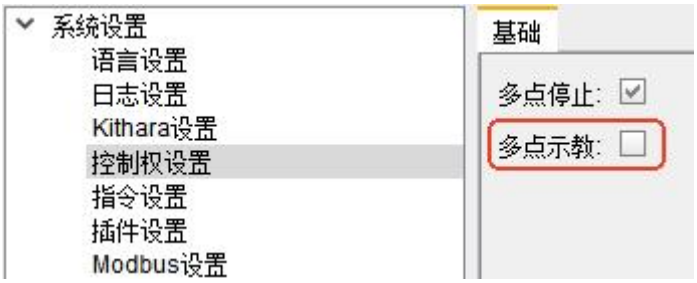


Figure 3-5 [System Settings/Control Settings]

3.4.1 View System Location Point Variables

Allow the master station to remotely view system location point variables.

Operation steps:

(1) Enter the ID (10000-10199) of the system location point variable to be viewed in input address 28.

(2) Check if the output address 28 is the same as the content in input address 28.

(3) If they are the same, the viewing is successful, and the parameters of this system location point variable will be displayed from output addresses 29-48. The specific meaning of the parameters can be found in the robot slave station address allocation table.

(4) If the value in output address 28 is -1, it means the system location point variable ID entered in input address 28 is incorrect, and the viewing fails.

3.4.2 Robot Teaching Position Variables

It supports remote teaching of robot positions, modification of robot positions, and movement to robot positions. The robot must be in remote manual mode. The teach position variable function assigns the robot's

current position to a specific system position variable. Only system position variables with modification permissions greater than or equal to 1 will be taught or modified.

- **Enable location point variables:** You must enable location point variables before using this function, as shown in the figure below.

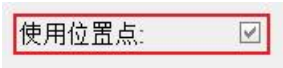
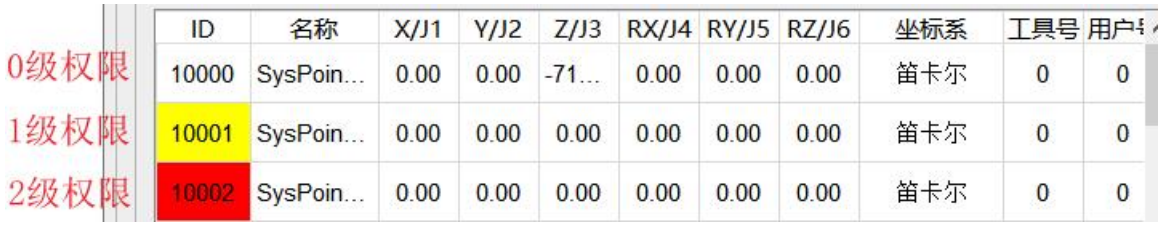


Figure 3-6 [Parameter Settings/Robot Settings]

- **Location Point Variables:** These include local location point variables, system location point variables, and persisted location point variables. For detailed descriptions and usage, please refer to the *YiKingSoft Software Manual*.

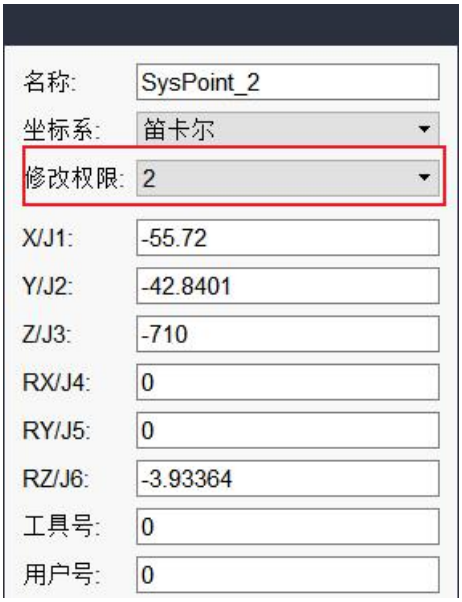
- **Teach Point ID:** Only system location point variables are supported for teaching. The ID range for system location point variables is 10000-10199.

- **Point Modification Permissions:** Remote modification of system location point variables has three permission levels: 0 - External modification is not allowed; 1 - External modification of x, y, z, and rz is allowed only; 2 - Highest permission, all can be modified. See the figure below:

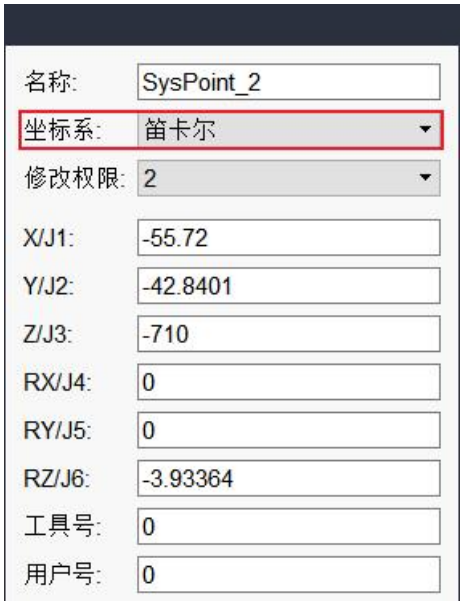


ID	名称	X/J1	Y/J2	Z/J3	RX/J4	RY/J5	RZ/J6	坐标系	工具号	用户号
10000	SysPoin...	0.00	0.00	-71...	0.00	0.00	0.00	笛卡尔	0	0
10001	SysPoin...	0.00	0.00	0.00	0.00	0.00	0.00	笛卡尔	0	0
10002	SysPoin...	0.00	0.00	0.00	0.00	0.00	0.00	笛卡尔	0	0

Permission modification: Double-click the system location variable to view or modify permissions.



- **Point type:** Whether the teaching point type is a Cartesian coordinate system or a joint coordinate system depends on the coordinate system type of the system's position point variables.



- **Operating Steps:**

(1) Move the robot to the teaching point using the jog or inching function.

(2) Enter the system position point variable ID (10000-10099) to be written in input address 28.

(3) Enter the teaching point function code (robot number * 1000 + 20) in input address 25.

(4) Check output address 28. If the value matches the system position point variable ID in input address 28,

and output addresses 29-48 display the parameters of this system position point variable, the writing was successful. The specific parameter meanings can be found in the robot slave address allocation table. If the value in output address 28 is -1, the writing failed.

3.4.3 Robot modifies position variables

Once the robot's current position to a certain system position variable is taught, the system point variables X/J1, Y/J2, Z/J3, RX/J4, RY/J5, and RZ/J6 can be modified individually.

Operation Steps:

(1) Refer to section 3.4.1 to view the system position point variable to be modified.

(2) Enter the system position point variable value with a unit of 0.001 mm into input addresses 29 and 30. The unit of the entered value needs to be changed to 0.001 mm, then converted from a signed integer to 32-bit binary, and split according to the EtherCat slave station integer type format before writing it into the corresponding bytes of input addresses 28 and 30.

(3) Enter the function code (211-216) to modify the system position point variable into input address 28.

(4) Check output address 28. If the value matches the system position point variable ID of input address 28, the modification is successful. Output addresses 29-48 will display the parameters of this system position point variable. The specific meaning of the parameters can be found in the robot slave station address allocation table. If the value of output address 28 is -1, the modification failed.

3.4.4 Variables for moving to a position point

Supports remote control of the robot to move to a specific system position variable using the MoveL method.

Operation steps:

(1) Write the enable function code (robot number * 1000 + 1) to input address 28 to switch the robot's state to the paused state within the running state. This corresponds to robot state 4-running and running state 2-pause in the output address.

(2) Refer to section 3.4.1 to view the system position variable to be modified.

(3) Write the MoveL function code (robot number * 1000 + 30) to input address 25. The robot will then begin moving, and the value of output address 28 will change to -1. When the value of output address 28 matches the value of input address 28 (i.e., when the value of output address 28 becomes the target system position variable ID), the robot has moved to the position described by the target system position variable.

(4) The robot will immediately stop moving when the value of input address 25 changes.

3.5 Robot IO Operations

External communication enables IO operations on the robot. For a single robot, 32 general-purpose outputs can be set.

Control requirements: Bus communication control or "multi-point teaching" must be enabled.

Multi-point teaching: Whether to allow users to perform robot teaching operations via the bus master when not under "bus communication control"; disabled by default. The configuration interface is shown below:

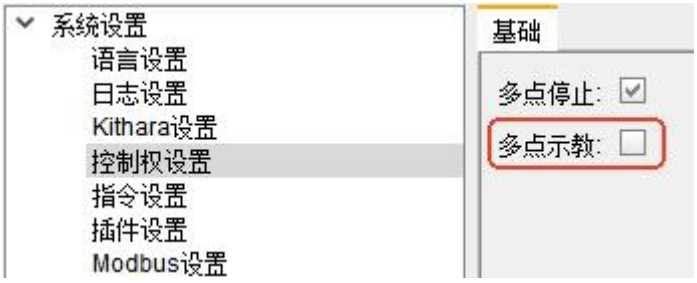


Figure 3-7 [System Settings/Control Settings]

3.5.1 Single IO bit set

Operation Codes:

General Output IO Set to 0: Robot Number * 1000 + 300 + IO Channel Number;

General Output IO Set to 1: Robot Number * 1000 + 400 + IO Channel Number.

Operation Examples:

(1) Write enable code 1300 to input address 25 to set channel 0 of robot 1 to low level;

(2) Write enable code 1400 to input address 25 to set channel 0 of robot 1 to high level;

(3) Write enable code 1301 to input address 25 to set channel 1 of robot 1 to low level;

Write enable code 1401 to input address 25 to set channel 1 of robot 1 to high level;

3.5.2 All IO reset

External communication provides a one-click all IO reset function code support, facilitating user operations in

certain scenarios. This function requires the "All IO Reset" feature to be enabled.

All IO Reset Function Configuration: This function is disabled by default and requires manual activation by the user to take effect. A screenshot of the configuration is shown below:

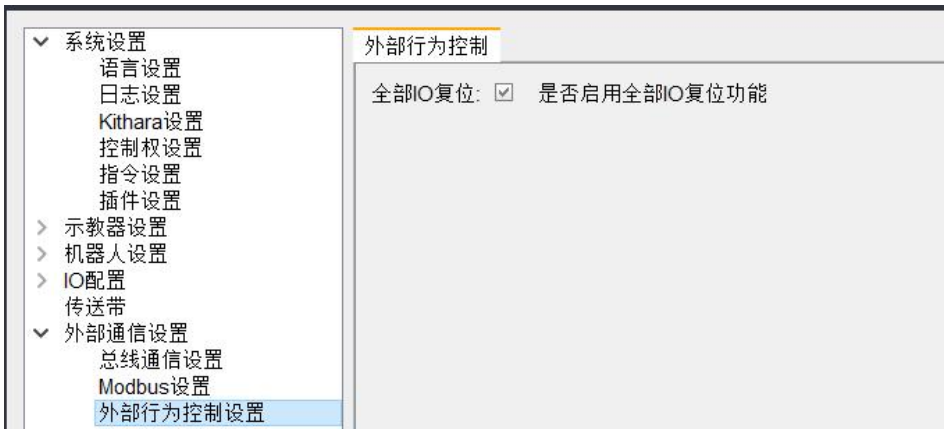


Figure 3-8 [System Settings/External Communication Settings]

Opcode

Set all general output I/Os to 0: Robot number * 1000 + 500;

Chapter Four Robot Slave Address Allocation Table

4.1 Input Addresses

Addresses 0-24: System addresses, robot function control area, used for external input of start, stop, pause, and resume functions.

Addresses 25-49: System addresses, common area for robots, used to control all robots dragged on the same controller.

Addresses 50-126: Assigned to users for custom communication via commands.

4.1.1 Robot function control address

Address	Functions and parameters
0: Startup command number	The default value is 0, and the valid value range is [0-100000]. This corresponds to starting the Cmds\ExStartCmd\Cmd_XXX command program.
1: Start/Stop Commands	0 - Default value 1 - Start all robots 2 - Stop all robots 3 - Enable all robots 101 - Start Robot 1 102 - Stop Robot 1 103 - Command to turn on Robot 1 201 - Start Robot 2 202 - Stop Robot 2 203 - Command to Open Robot 2 301 - Start Robot 3

	302 - Stop Robot 3 303 - Open Robot 3 command 401 - Start Robot 4 402 - Stop Robot 4 403 - Open Robot 4 command
2: Pause/Restore	0 - Default value 1 - Pause all bots 2 - Resume all bots 101 - Pause Robot 1 102 - Resume Robot 1 201 - Pause Robot 2 202 - Resume Robot 2 301 - Pause Robot 3 302 - Resume Robot 3 401 - Pause Robot 4 402 - Resume Robot 4
3: Error Reset	0 - Default value 1 - Reset and stop all robots Command to restore all robots is not supported 101 - Reset and stop robot 1 102 - Command to resume robot 1 201 - Reset and stop robot 2 202 - Command recovery robot 2

	301 - Reset and stop robot 3 302 - Command to resume robot 3 401 - Reset and stop robot 4 402 - Command to resume robot 4
4: Emergency stop and reset	0 - Default value 1 - Reset and stop all robots Command to restore all robots is not supported 101 - Reset and stop robot 1 102 - Command to resume robot 1 201 - Reset and stop robot 2 202 - Command recovery robot 2 301 - Reset and stop robot 3 302 - Command to resume robot 3 401 - Reset and stop robot 4 402 - Command to resume robot 4
5: Automatic speed value	Default value is 0, valid value range is [1-100]
6: Automatic speed control	The default value is 0, and only valid operations are effective. Receiving 1 sets the global speed of all robots to the speed value at the "Auto Speed Value" address. Receiving 2 reduces the global speed of all robots based on the current speed level. Receiving 3 increases the global speed of all robots based on the current speed

	level. Receiving 12 decreases the global speed of all robots by 1 based on the current speed. Receiving 13 increases the global speed of all robots by 1 based on the current speed. Robot 1 is based on all robot function codes plus 100. Robot 2 is based on Robot 1 function codes plus 100. Robot 3 is based on Robot 2 function codes plus 100. Robot 4 is based on Robot 3 function codes plus 100.
7	0 - Default value 1 - Power off
……	

4.1.2 Teaching control address

Address	Functions and parameters
25: Demonstration Operation	Default value 0
	Public function code range (0001-0999) **Mode switch** 101 **Modify** posx 211 **Modify** posy 212 **Modify** posz 213 **Modify** posrx 214 **Modify** posry 215 **Modify** posrz 216

	Robot 1 Operation Function Codes (1001-1999) Enable/Can Enable (Startup Confirmation) 1001 Speed Level Deceleration 1002 Speed Level Acceleration 1012 Speed Fine-Tuning Deceleration 1003 Speed Fine-Tuning Deceleration 1013 The robot's current position is written to the system position variable 1020 MoveL moves the robot to the system position variable p 1030 **Jog Control** X/J1- 1101 X/J1+ 1111 Y/J2- 1102 Y/J2+ 1112 Z/J3- 1103 Z/J3+ 1113 RX/J4- 1104 RX/J4+ 1114 RY/J5- 1105 RY/J5+ 1115 RZ/J6- 1106 RZ/J6+ 1116 **Incremental Motion Control** X/J1- 1201 X/J1+ 1211 Y/J2- 1202 Y/J2+ 1212 Z/J3- 1203 Z/J3+ 1213

	RX/J4- 1204 RX/J4+ 1214 RY/J5- 1205 RY/J5+ 1215 RZ/J6- 1206 RZ/J6+ 1216 **IO Control** Output IO 00-31 to 0 1300 - 1331 Output IO 00-31 to 1 1400 - 1431 Reset All I/O 1500
	机器人 2 操作功能码 (2001-2999) Enable / Disable (Start Confirmation) 2001 Speed Level Decrease 2002 Speed Level Increase 2012 Speed Fine-Tune Decrease 2003 Speed Fine-Tune Increase 2013 Write Robot Current Position to System Position Point Variable 2020 MoveL to System Position Point Variable 2030 **Jog Control** X/J1- 2101 X/J1+ 2111 Y/J2- 2102 Y/J2+ 2112 Z/J3- 2103 Z/J3+ 2113 RX/J4- 2104 RX/J4+ 2114 RY/J5- 2105 RY/J5+ 2115

	RZ/J6- 2106 RZ/J6+ 2116 **Incremental Control** X/J1- 2201 X/J1+ 2211 Y/J2- 2202 Y/J2+ 2212 Z/J3- 2203 Z/J3+ 2213 RX/J4- 2204 RX/J4+ 2214 RY/J5- 2205 RY/J5+ 2215 RZ/J6- 2206 RZ/J6+ 2216 **I/O Control** Set Output I/O 00–31 to 0: Command 2300–2331 Set Output I/O 00–31 to 1: Command 2400–2431 Reset All I/O: Command 2500
	The jog command codes of Robot 3 are based on those of Robot 2 plus 1000.
	The jog command codes of Robot 4 are based on those of Robot 3 plus 1000.
26 Set PTP Motion Step Size	Unit: 0.01 mm, Maximum: 10000
27 Set Current Coordinate System	0: World Coordinate System 1: Base Coordinate System 2: TCP Coordinate System 3: Joint Coordinate System

28 System Position Variable ID	Valid Range: 10000–10199
29 System Position Variable Write Value	Together with Address 95 to form the full system position variable write value Unit: 0.001 mm
30	
……	

4.2 Output Address Mapping

Address 0–24: Robot Status Area

Address 25–49: Common Output Area

Note:

The addresses listed in the table below are default addresses.

In actual applications, the address mapping can be customized according to user requirements.

For detailed configuration methods, please refer to Section “3.1” of this document.

4.2.1 Robot 1 Status Area

Address	Function & Parameters
0：Robot Status Code	0：Not Initialized 1：Initializing 2：Self-check 3：Ready 4：Running (sub-status code see Address 1) 6：Homing 10：Emergency Stop 11：Error
1：Operation Status Code	This status is valid only when Address 0 is in the Running state. 0：Running 1：Pause Initiated 2：Paused (in this state, the robot can switch to jog mode or resume motion)

<table>
<tr><td></td><td colspan="2">3：Moving to Pause Point
5：Resume Operation Initiated</td></tr>
<tr><td>2：Error Code</td><td colspan="2">0：Normal
Non-zero: Robot fault.: Refer to Robot Alarm Information for detailed fault descriptions according to the error code.</td></tr>
<tr><td>3：PosX</td><td>Unit: 0.1 mm</td><td rowspan="6">If the position resolution does not meet application requirements, high-precision position display can be configured.
For detailed configuration, please refer to Section “3.1” of this document.</td></tr>
<tr><td>4：PosY</td><td>Unit: 0.1 mm</td></tr>
<tr><td>5：PosZ</td><td>Unit: 0.1 mm</td></tr>
<tr><td>6：RX</td><td>Unit: 0.1 mm</td></tr>
<tr><td>7：RY</td><td>Unit: 0.1 mm</td></tr>
<tr><td>8：RZ</td><td>Unit: 0.1 mm</td></tr>
<tr><td>9：Operation Mode Query</td><td colspan="2">0: Manual Mode
1: Automatic Mode
2: Remote Teaching
3: Remote Debugging
4: Remote Operation</td></tr>
<tr><td>10：Current Speed Query</td><td colspan="2">Unit: %</td></tr>
<tr><td>11：General-purpose Input I/O (Lower 16 bits)</td><td colspan="2">Binary representation: 0 = Low level, 1 = High level</td></tr>
<tr><td>12：General-purpose Input I/O (Upper 16 bits)</td><td colspan="2">Binary representation: 0 = Low level, 1 = High level</td></tr>
</table>

13：General-purpose Output I/O (Lower 16 bits)	Binary representation: 0 = Low level, 1 = High level
14：General-purpose Output I/O (Upper 16 bits)	Binary representation: 0 = Low level, 1 = High level
15：Heartbeat	Robot heartbeat signal / status indication.
16：Control Authority	1: HMI Control Authority 2: I/O Module Control Authority 3: Modbus Module Control Authority 4: SDK Control Authority
17：Current Start Command ID	-1: Invalid command ID ≥ 0: Indicates the currently active command ID, or the value of the “Start Command ID” address currently in use.
……	

4.2.2 Common Status Area

The address mapping of the Common Status Area is not configurable.

The default addresses and corresponding functions are listed in the table below:

Address	Function & Parameters
25	PTP Step Size for Remote Teaching
26	Coordinate System for Remote Teaching (Jog / Incremental Motion)
27	Operation Status Code (Reserved)
28	System Position Variable ID

29	System Position Variable X (High word)
30	
31	Together with the next address to form the Y value Unit: 0.001 mm
32	
33	Together with the next address to form the Z value Unit: 0.001 mm
34	
35	Together with the next address to form the RX value Unit: 0.001°
36	
37	Together with the next address to form the RY value Unit: 0.001°
38	
39	Together with the next address to form the RZ value Unit: 0.001°
40	
41	Arm Parameter 1
42	Arm Parameter 2

43	Arm Parameter 3
44	Arm Parameter 4
45	Solution Index for System Position Variable
46	stem Position Variable TCPNo
47	System Position Variable BaseNo
48	Coordinate System of System Position Variable
……	

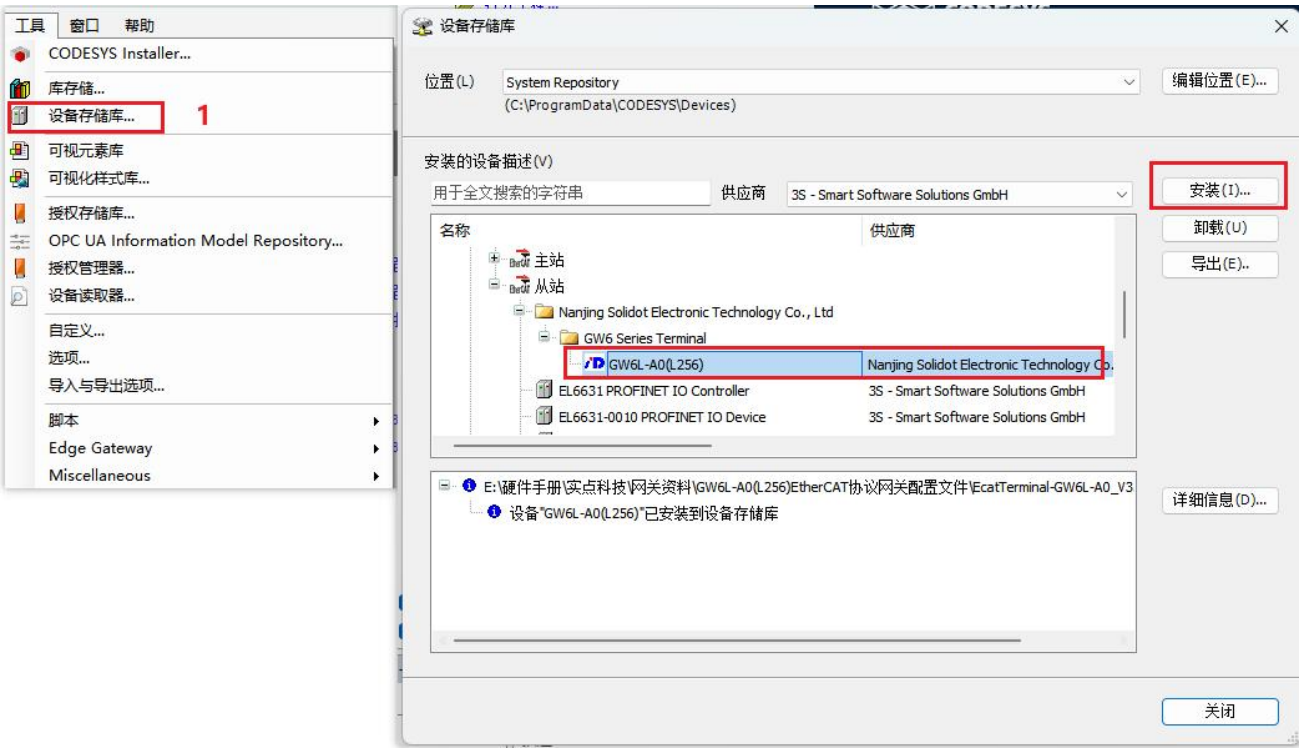
Chapter Five Connection and Configuration between CODESYS and Kingkong Controller

5.1 Project Creation and Device Configuration

1.Open CODESYS V3.5.

From the menu bar, select “Tools” > “Device Repository”, then import the ESI file provided by the manufacturer (EcatTerminal-GW6L-A0_V3.00_BOOL).

After successful installation, the interface is shown as in the figure below.



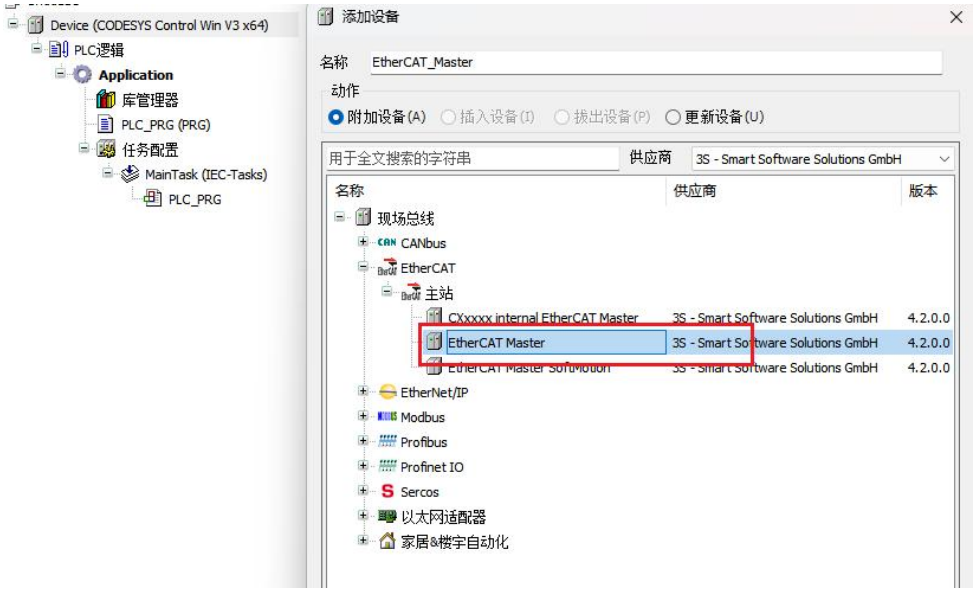
2.Create a New Project and Configure Devices

Open CODESYS V3.5, select “New Project” > “Project” > “Standard Project”, and click OK to create a new project.

In the Standard Project dialog, select “CODESYS SoftMotion Win V3” as the device, and choose Structured Text (ST) as the PLC_PRG programming language.

3.In the Device Tree, right-click “Device (CODESYS SoftMotion Win V3)” and select “Add Device”.

In the Add Device dialog, choose “Fieldbus” > “EtherCAT” > “EtherCAT Master”.



4.Assign a network interface to the EtherCAT Master:

In the Device Tree, double-click “EtherCAT_Master”, go to “EtherCAT NIC Settings”, and click “Browse” to select the network adapter.

5.Right-click **“EtherCAT_Master”** and select **“Scan Devices”** to import the actual hardware configuration into the project. The detected physical devices will be displayed in the scan window.

6.Download, Run, and Monitor the Program