

YiKingSoft

Software Manual

Doc.No. RP-KZ-01-001-C

Copyright and Disclaimer

Content in this manual subjects to change without prior notification and neither can it be viewed as commitment of Robotphoenix Industrial Robotics Co., Ltd (hereinafter referred as Robotphoenix). Unless otherwise stated in this manual, any content in this manual cannot be deemed as Robotphoenix' s guarantee or promise for users' physical and property loss, compatibility towards special usage or things alike.

Without written permission of Robotphoenix, it is prohibited to copy or print any part of this manual, leak the information to third party, nor other unauthorized application. The company reserve the lawsuit right for any violation of this term.

Catalogue

Chapter 1 Basic Concept of Robots	3
1.1 Introduction to Robotic Systems	3
1.2 Classification	3
1.3 Coordinates System	5
1.3.1 Joint Coordinates System	5
1.3.2 Base Coordinates System	5
1.3.3 Tool Coordinate System	6
1.3.4 User Coordinates System	7
1.4 Gravitational Singularity Positions	8
1.5 Offset	9
1.5.1 Offset of Tool Coordinate System	10
1.5.2 Offset of Base Coordinate System	10
Chapter 2 YiKingSoft Software Introduction	11
2.1 Introduction	11
2.2 Partial Functions of Software Products	11
Chapter 3 Entry level guidance of YiKingSoft	12
3.1 Operation Procedures	12
3.2 Power-on and Turn-on	12
3.3 Erroneous Alarming Notes	13
3.4 Major Screen Menu	14
3.5 Shortcut Icon	15
3.5.1 Log-in	15

3.5.2 Information Search	16
3.5.3 Shortcut Icon of Help	18
3.5.4 Shortcut Icon of Turn-off	18
3.6 Status Column	18
3.7 Speed Bar and Robot Switching	19
Chapter 4 Set robot	21
4.1 System Settings	21
4.1.1 System Settings	22
4.1.2 Language Settings	23
4.1.3 Server Setup	23
4.1.4 Log Management	23
4.1.5 Teach pendant Settings	27
4.1.6 Kithara Settings	28
4.1.7 Button Settings	30
4.1.8 Plug- in Settings	30
4.1.9 Modbus Settings	33
4.2 Robotic Settings	36
4.2.1 Robot Parameters Setting	37
4.2.2 Robot axis Parameter Setting	40
4.3 IO Configuration	41
4.4 Conveyor Belt Parameter Settings	41
4.5 UI Settings	43

4.6 Visual Settings	43
4.7 AD Configuration	44
Chapter 5 Function Operation	45
5.1 Conveyor Belt Calibration	45
5.2 Visual Calibration	46
5.3 Operation	46
5.3.1 Return to Zero	47
5.3.2 Record Zero Point	47
5.3.3 Opening the Holding Brake	48
5.3.4 Closing the Holding Brake	49
5.3.5 Calibrate Arm Length	49
5.4 TCP Configuration	49
5.5 Log	52
5.5.1 Log View	53
5.5.2 Log Screening	53
5.5.3 Log Management	54
5.6 Kithara Monitoring	55
5.7 I/O Configuration	55
5.8 Input Register Configuration	58
5.8.1 Function Status	58
5.8.2 Function Configuration	60
5.9 TCP Sockets	61

5.9.1 Client Settings	62
5.9.2 Server Settings	64
5.10 Simulation Quantity Setting	65
5.11 External Axis Setting	65
5.11.1 basic Screen	66
5.11.2 Calibration Screen	67
5.12 BASE Configuration	67
5.13 LOAD Configuration	70
5.14 Jog	72
5.14.1 Teach Pendant Touch Screen	72
5.14.2 Instruction on Keys	73
5.14.3 Robot PTP Function	74
Chapter 6 Robot Programming	75
6.1 Modules - Commands	75
6.2 Module—Command Set	76
6.3 Modules - Debugging	77
6.3.1 Running Variable Screen	77
6.3.2 Operation Point Screen	79
6.3.3 Input and Output Screen	79
6.3.4 Simulation Screen	81
6.3.5 Object Screen	81
6.3.6 Status Screen	83

6.3.7 Compile Screen	83
6.4 Modules - Variables	84
6.4.1 Local Variables	84
6.4.2 Global Variables	85
6.4.3 Modbus Variables	85
6.4.4 Add Variables	85
6.4.5 Variable Editing	86
6.5 Modules—Arrays	87
6.5.1 Array Variable Addition	88
6.5.2 Array Variable Editing	89
6.6 Modules - Points	91
6.6.1 Point Variable Addition	92
6.6.2 Point Variable Editing	93
6.6.4 Add to PTP	94
6.7 Module—Run Log	94
6.8 Plug-in—YiExVision	95
6.8.1 Object Screen	95
6.8.2 TcpSocket Screen	95
6.8.3 Advanced Screen	97
6.9 plug-ins—YiVision	100
6.9.1 Visual Screen	100
6.9.2 Object Screen	104

6.9.3 Template Screen	104
6.9.4 Basic Screen	105
6.9.5 Hikvision camera Screen	106
6.10 Plug-in—YiFly	108
6.10.1 Basic Parameters	108
6.10.2 Configuration Screen	108
6.10.3 Calibration Settings	110
6.10.4 Calibration Settings	112
6.10.5 Global Parameters	114
6.10.6 Function Keys	114
6.11 Plug-in—YiModel	115
6.11.1 Edit Screen	115
6.11.2 Template screen	115
6.10.3 Graphical Screen	116
6.12 Plug-ins—YiPath	116
Chapter 7 Running the Program	117
7.1 Select Program to Run	117
7.2 Program Running Screen	118
7.3 Image Table	119
7.4 Running Variables	119
7.5 Running Log	120

Introduction

About the Manual

This manual shares the basic operation, program editing, debugging and running, parameter setting, and etc. of YikingSoft software. (The current version of the software user handbook is for official 2.8.2(R)-2.8.8(R))

YiKingSoft software is used for programming, debugging, status monitoring, parameter setting and other functions of Robotphoenix robots, applied with Robotphoenix Bat series robots, Python series robots, Mantis series robots and Lobster series robots.

Usage

Users can refer to this manual to judge whether a certain option can resolve a certain problem, or follow the commands in this manual. This manual doesn't include commands and cases of KingKong commands.

For details, please refer to the relevant manuals of each series.

Prerequisites

Users of this manual should meet following requests:

- Being familiar to industrial robots and professional terms
- Being familiar to programming language of KingKong command
- Being familiar to systemic parameters and usages

Reference

References	Number
Kingkong command handbook-Bat series	RP-KZ-01-010-C
Kingkong command operation-Pythons series	RP-KZ-01-003-C
King Kong command handbook-Mantis series	RP-KZ-01-011-C
Kingkong command handbook-Lobster series	RP-KZ-01-012-C
YiKingSoft visual application manual	RP-KZ-01-007-C
Modbus communication manual	RP-KZ-01-004-C
Socket communications manual	RP-KZ-01-005-C
Configure configuration manual	RP-KZ-01-006-C

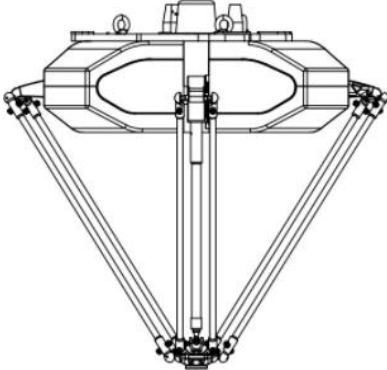
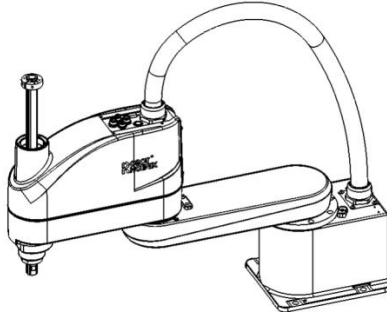
Chapter 1 Basic Concept of Robots

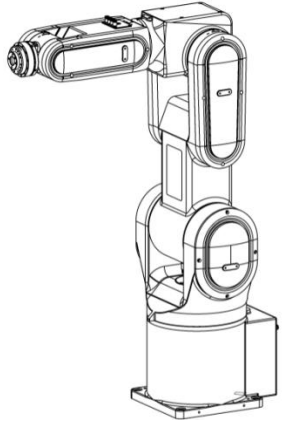
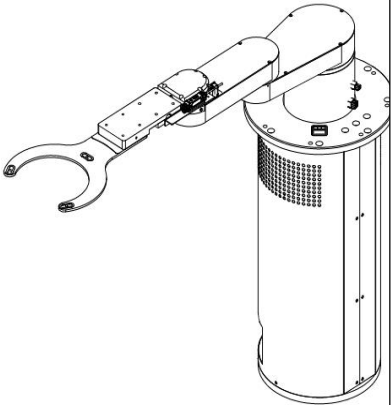
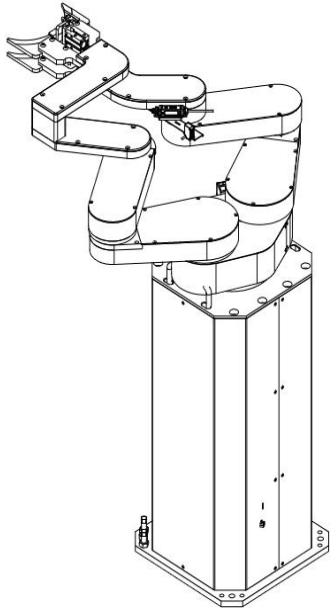
1.1 Introduction to Robotic Systems

The robot system is mainly composed of teach pendant, electrical controlling cabinet, robot body and secondary development software.

1.2 Classification

In light of the characteristics of robots, including the number of axes, serial and parallel connection, robots can be classified as the following:

Name	Type	Number of axes	serial or parallel connection	Features	Applications
Delta robot		4	Parallel	High precision and fast speed.	Medicine , food, 3C daily handling, sorting, etc.
SCARA robot		4	Serial	Light structure , swift response and high precision	Mechanical assembly, material distribution, distribution, 3C industrial robot assembly, etc.

Six-axes robot		6	Serial	Great flexibility and applied to work of any track or angle	Loading , painting, measuring, arc welding, spot welding, packaging, etc.
Wafer Single Arm robot		3 axes	Single arm serial	High precision and flexibility	Precision transportation, transfer processing components , wafer handling, LED handling, panel handling
Wafer Dual Arms robot		4 axes	Dual arms serial	High precision and flexibility	Precision transportation, transfer processing components , wafer handling, LED handling, panel handling

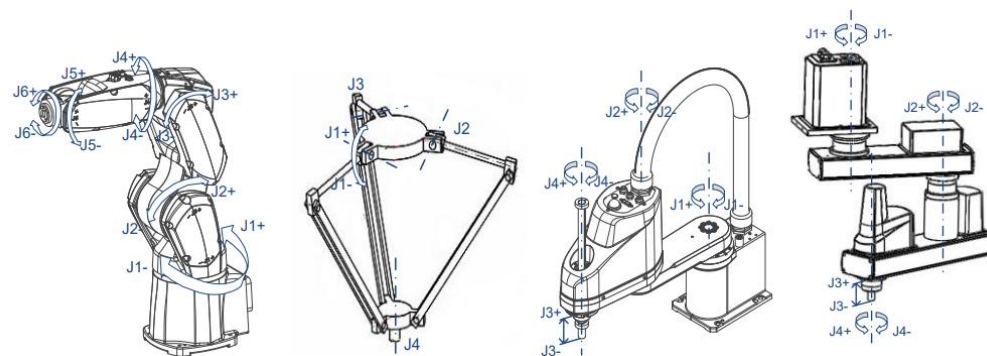
1.3 Coordinates System

In Robotphoenix YiKingSoft robots systems , there are five types of coordinates.

Number	name	Definition
1	Base coordinate system	Defined with the base of the robot, it's often used as a reference for motion
2	Tool coordinate system	Defined with the tool, it can be customized
3	User coordinate system	Usually defined with the work-piece, it can be customized
4	Joint coordinate system	Defined at each joint of the robot, it is a direct description of the joint motion of the robot

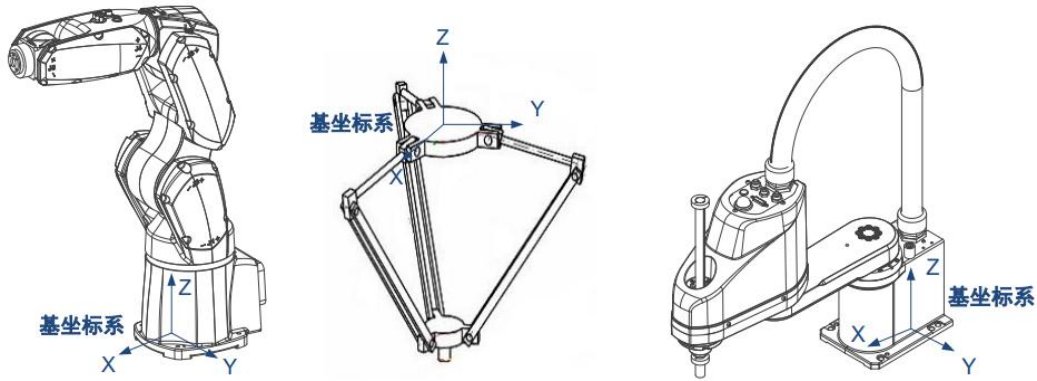
1.3.1 Joint Coordinates System

In robots joint position (those photos just for reference and needs confirmation)

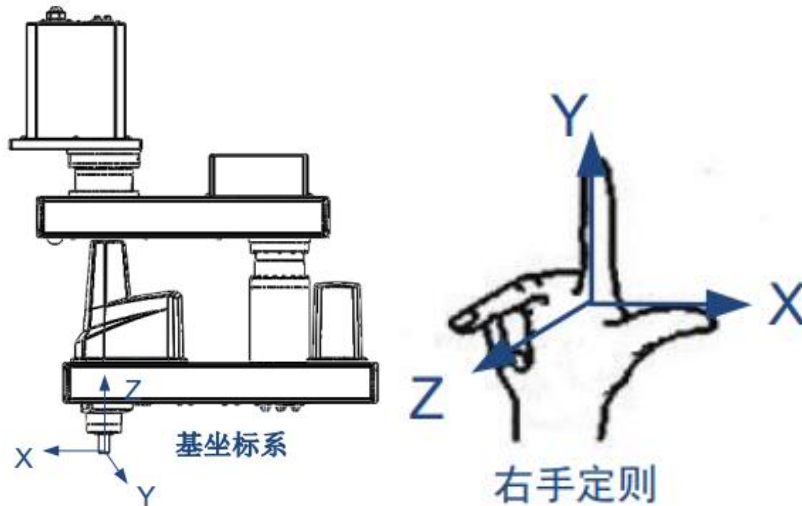


1.3.2 Base Coordinates System

Base robots coordinates system, usually sits at the bottom of robots (just for reference)



The base coordinate system of ceiling mounting Scara sits at the flange of the zero point. (just for reference)

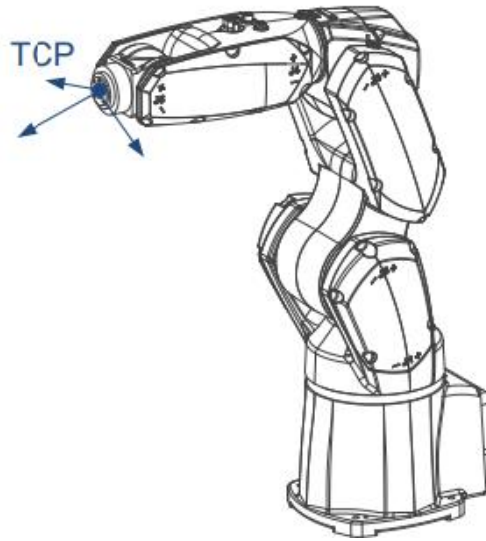


The base coordinate system is a Cartesian type coordinate system (as well as the tool coordinate system, user coordinate system). The X and Z directions can be determined first, and then the Y direction can be determined by the "right-hand rule". (Just for reference)

1.3.3 Tool Coordinate System

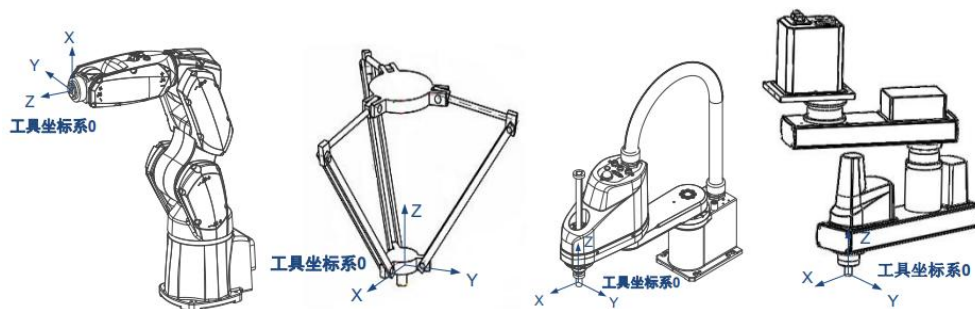
The tool coordinate system is attached to the tool, with TCP (Tool Center Point, the origin of tool coordinate), as a reference point for measuring the robot's arrival position. The end point of the tool is taken as the TCP, and the direction

can be freely defined. (Just for reference)



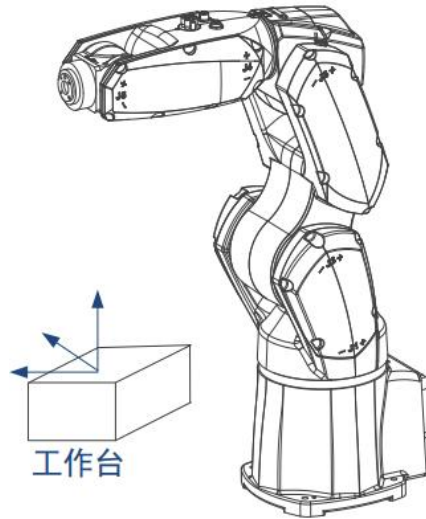
In the robot YiKingSoft control system, up to 23 tool coordinate systems can be defined. The tool tool0 means non-use of tools (tool0 can't be changed), and the tool coordinate system sits at the end flange of the robot body.

Tool tool1-23 can be freely defined by users.



1.3.4 User Coordinates System

The user coordinate system is a user-defined coordinate system. The user coordinate system can be set freely, and it is generally on a specific object, such as: workbench, conveyor belt...



In the robot YiKingSoft control system, up to 23 user coordinate systems can be defined. Among them, user 0 means that no additional user coordinate system is used, and the user coordinate system coincides with the base coordinate system at this time; user 1- 23 can be freely defined by the user.

1.4 Gravitational Singularity Positions

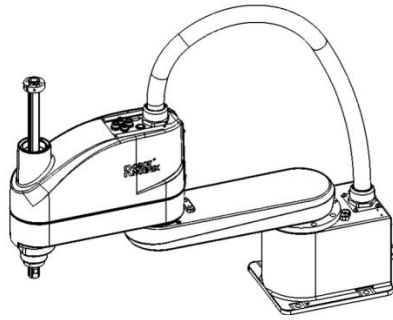
Motioning under the non-joint coordination axis system, robot may move to some specific positions, at which time, it will report an error. Those special positions are named gravitational singularity points.

In the articulation movement of MoveJ, the position of gravitational singularity will not meddle with the general movement of robots, but in the linear travel of MoveL and circular motion of MoveC, the position of gravitational singularity will hamper the motion of robots.

Warning: Encountering the alarming of gravitational singularity positions, articulation movement mode can be used to exit the alarming in a inching way.

Scara robot has only one gravitational singularity position, and when it sits at

$J2=0^\circ$, the first and second arm form a straight line.



(Scara gravitational singularity position $J2=0^\circ$)

Delta robot's gravitational singularity position is not in working range and has no gravitational singularity position

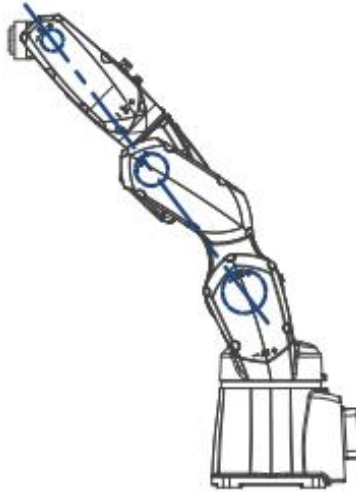
Six-axe robot has three types of gravitational singularity positions, as following:



Upper gravitational singularity

the intersection of 4-axis, 5-axis

and 6-axis being above J1

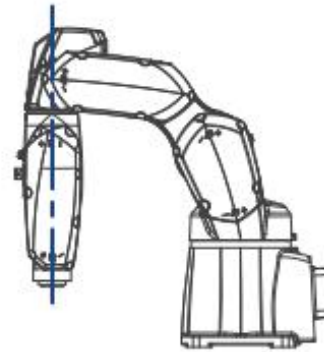


Extension gravitational

singularity

2-axis, 3-axis, 5-axis joint

center collinear



Wrist gravitational

singularity

4-axis and 6-axis collinear

1.5 Offset

It is used to accurately get the center of the object by adjusting the Tool offset or

the base coordinate Base skewing when there is an offset during robot picking the object. Before using ToolOffset and BaseOffset, please judge whether the center of the object can be got at present, if not, please use BaseOffset to adjust. After being able to pick the center of the object, if you want to pick one side of the object, please adjust the ToolOffset.

1.5.1 Offset of Tool Coordinate System

Function: By setting the ToolOffset, the TCP point can be offset along the tool coordinate system of the motion reference. If we write $X = 20$ in ToolOffset, then the robot tool center will traverse 20 with reference to the X direction of the tool coordinate system. Command name: ToolOffset, refer to KingKong command handbook for details

1.5.2 Offset of Base Coordinate System

Function: By setting the BaseOffset, the offset of the TCP point with the user coordinate system of the motion reference can be realized. If we write $X = 20$ in BaseOffset, then the robot tool center will traverse 20 with reference to the X direction of the user coordinate system. Command name: BaseOffset, please refer to the KingKong command handbook for details.

Chapter 2 YiKingSoft Software Introduction

2.1 Introduction

YiKingSoft software is a product independently developed by Hangzhou Robotphoenix Robot, which is used to improve production efficiency and reduce the difficulty of using robot operation software, making it easier for operators to operate robots. Hangzhou Robotphoenix focuses on the research and development of new products since its establishment. After years of development and study, the company owns to its name various series of robot products.

2.2 Partial Functions of Software Products

The internal screen of YiKingSoft software displays assorted functions, which are divided into several parts for details in light of the needs of customers, which is more expedient for customers and programmers to understand:

Options	Description
Run Program	Run the normal operating program that is debugged
Program robots	Write and debug robot operation programs: including command resources, variable resources, IO resources, command running debugging, debugging status viewing, etc.
Options	Work -related auxiliary operations: including visual calibration, configuring TCP, BASE, setting external axes, TCP communication, etc.
Set robot	View and set robot configuration parameters: including robot plug-in settings, ModBus communication settings, IO settings, etc.
Shutdown robot	Exit YiKingSoft software

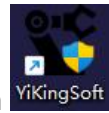
Chapter 3 Entry level guidance of YiKingSoft

3.1 Operation Procedures

Following content is a demonstration on how to use the software. After the steps of power-on, opening the software, user login, status check, and reset, the robot is operated to move and configure other parameters of the robot.

3.2 Power-on and Turn-on

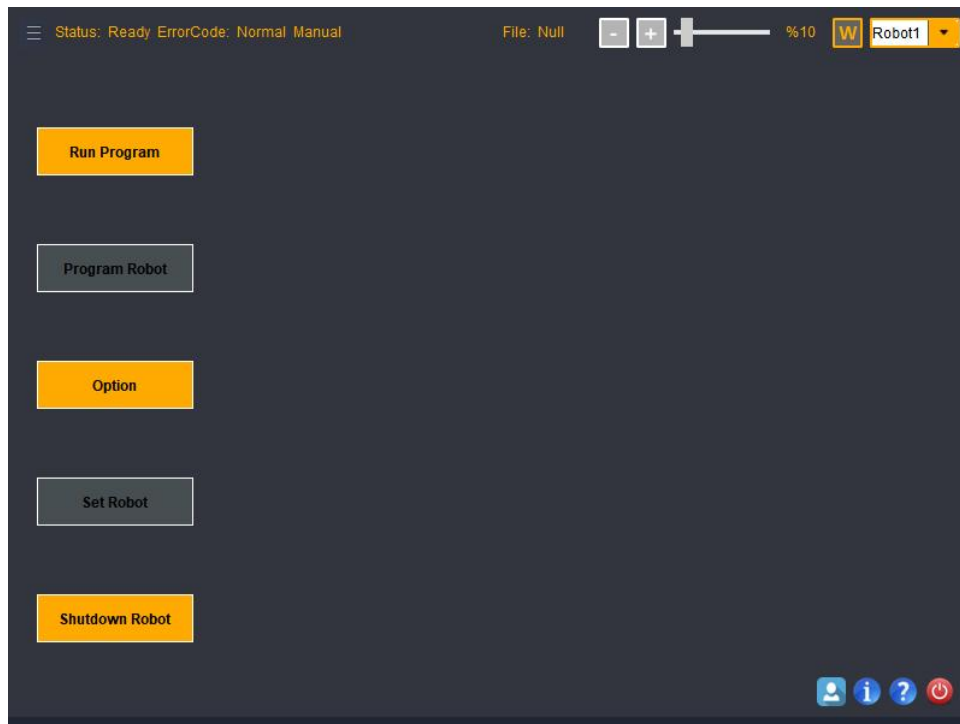
Power on the electrical controlling cabinet and teach pendant will automatically



shows the software. Double pressing the icon, and a loading progress bar appears, and when it's 100%, it will automatically enter the software.



Successful entry is as follows:



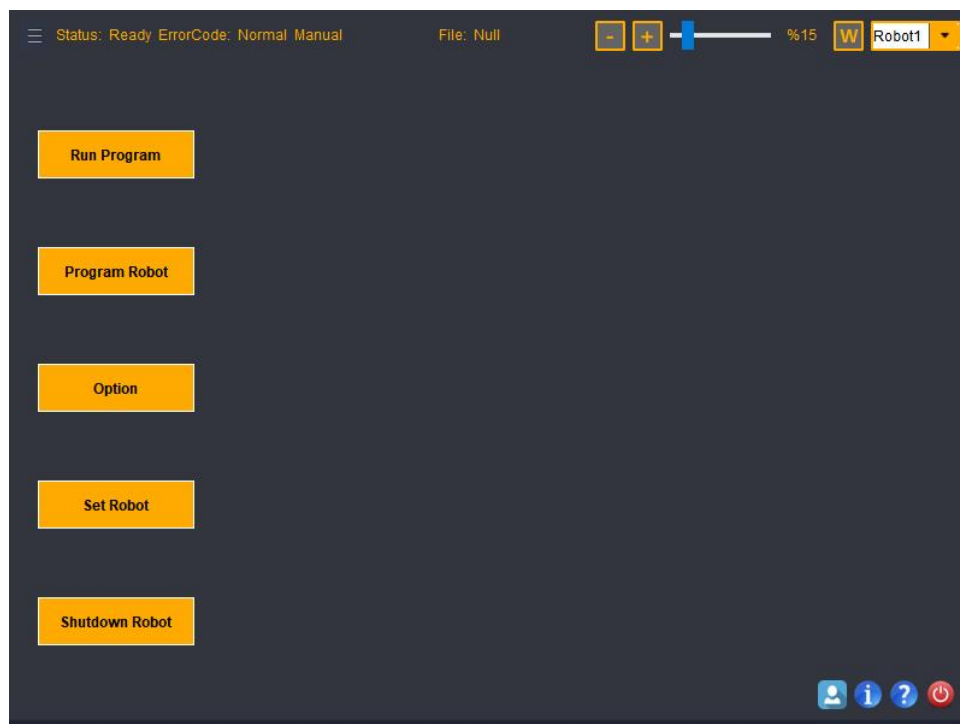
3.3 Erroneous Alarming Notes

Status and information	Issue description	Settlement
	Start the software for the first time or hardware of the electrical control cabinet is replaced.	<pre> graph TD Error[Error] --> Decision{Whether Kithara is activated, whether topology is right} Decision -- Yes --> Normal[Normal launch] Decision -- NO --> Steps[1. Search Kithara activation mandate
2. Search its settings] </pre>
Warning : Kithara : can't use drivers	Enter the software screen and status information	Kithara authorization is not activated.

	prompts Kithara: can not open the driver	
Kithara network interface error	Status showing: slave is not connect	The network interface selection being wrong, enter the software, click the "set robot" button, and go to System Settings→Kithara Settings→ Network Card. (The network card default is 0, if the network interface error is reported, try to switch to 1,2,3)
Topology debug assertion failed	Topology Empty	Add or set correct topology
	Topology ERROR	

3.4 Major Screen Menu

YiKingSoft's major screen menu is composed of the followings:



Run Program

(RUN Program): run program, can be selected from programs list

Program Robot

(Program Robot): newly-add, edit, modify, and run programs,

monitor variables, I/O, robot positions, camera, peripherals, etc., and add virtual objects, etc.

Option

(Option) : Conveyor belt calibration, visual calibration, robot zero return, robot electric motor brake control, TCP setting, and log file viewing.

Set Robot

(Set Robot): setting the information of robots

Shutdown Robot

(Shutdown Robot) turn off the YiKingSoft

Whether the above mentioned buttons can be chosen depends on the login identity of the user.

3.5 Shortcut Icon

Shortcut icon sits at the lower right of the major screen



3.5.1 Log-in

Users are divided into 4 levels: **basic**/operator/engineer/manufacturer, each of which has different permissions. Click the icon  in the lower right corner to log in to different user levels.

Active User: Base

User: (dropdown menu open showing: Base, Operator, Engineer, Manufacturer)

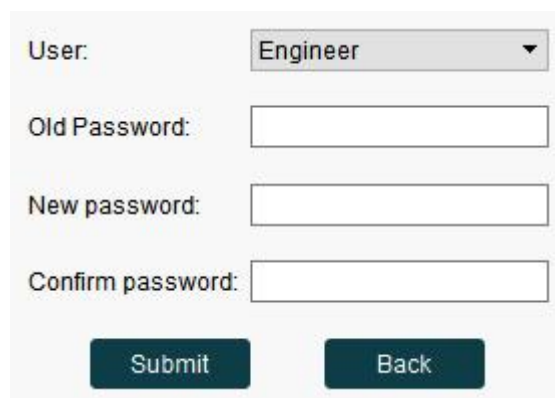
Password:

Login Advanced Cancel

In light of the roles of users, select different modes and log in :

Mode	Target users	Password
base	Applied towards operators in production line. It can control the robot to run the program, allowing controlling the IO status	No
Operator	For programming operators, it has more program editing functions than user mode, allowing operators to teach and programming.	yifei
Engineer	Apart from non-altered partial core settings, most of which are operable.	yifei
Manufacturer	For manufacturer, reserved with highest rights.	Reserved

Press the "Advanced" button to modify the login password



User: Engineer

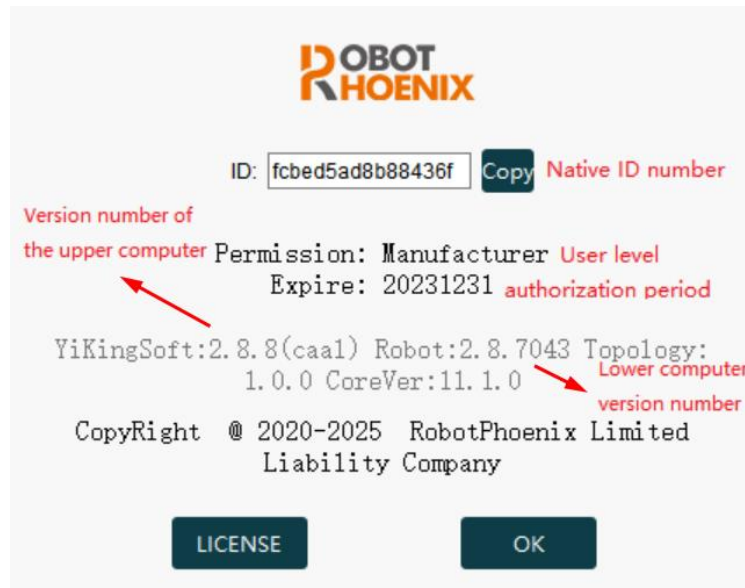
Old Password:

New password:

Confirm password:

3.5.2 Information Search

Press  icon to search this machine's ID, The highest user level, authorization period, software version number, lower computer version number, copyright information, etc.

**Note:**

If all the buttons in the main screen are in gray color except the showdown robot button, it means that the authorization is expired. Please apply for a new authorization file (Yikingsoft.sn file) from the manufacturer, and paste it into the Yikingsoft installation folder to replace the original Yikingsoft.sn file.

Application of authorization file:**Method 1:**

The ID button in the above shows the ID number of the machine. Click the button  on the right to copy the ID number. And contact the manufacturer after sales and provide the ID of the machine to obtain a new authorization file.

Method 2:

If the machine ID isn't shown on the information search screen, please go to the Yikingsoft installation folder and double-click GetDiskID.exe. Record the machine ID, contact the manufacturer after-sales with the machine ID to obtain a new authorization file.

3.5.3 Shortcut Icon of Help

Press  to obtain help files and it would automatically exit the screen



3.5.4 Shortcut Icon of Turn-off

Press  to turn off the machine and automatically exit YiKingsoft and stop the system

3.6 Status Column

After successfully login, please check the status and error code in the upper left corner. If the state shows emergency stop, please press the "reset" in the electrical controlling cabinet to make the robot exit the emergency stop state. If the error code shows normal and unexpected information, please contact with the manufacturer's personnel.



Status column	Description
Status	Exhibit the current robot status, such as ready/pause/run/emergency stop, etc.
Error code	Exhibit an error when an error emerges, shows "Normal" when there is no error
Manually/Automatically	Manual mode shows “Manually” and the automatic mode shows “Automatically”
Files	Show the selected program name, and "empty" if no program is selected
Speed bar	The speed percentage when running the program automatically, click on the white part  to increase or decrease the speed, and you can also adjust the speed by "plus and minus signs" (every time you press the "plus and minus signs", the speed changes by $\pm 1\%$)
Reference coordinate system when for inching movement	W: World coordinate B:Base coordinate T: Tool coordinate J: Joint coordinate
Robots number	Currently show the information of robot No.

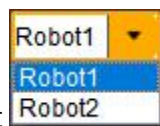
3.7 Speed Bar and Robot Switching

Without log-in authorization, the speed bar is gray and can't be adjusted. After the authorization login, the speed bar can be adjusted. The "-" and "+" buttons are used to control the speed by minus or plus 1% speed for each click. Click the blank area of the speed bar to control the speed of the speed bar to add or

subtract 10%. You can also directly drag the blue control block in the speed bar to control the speed.



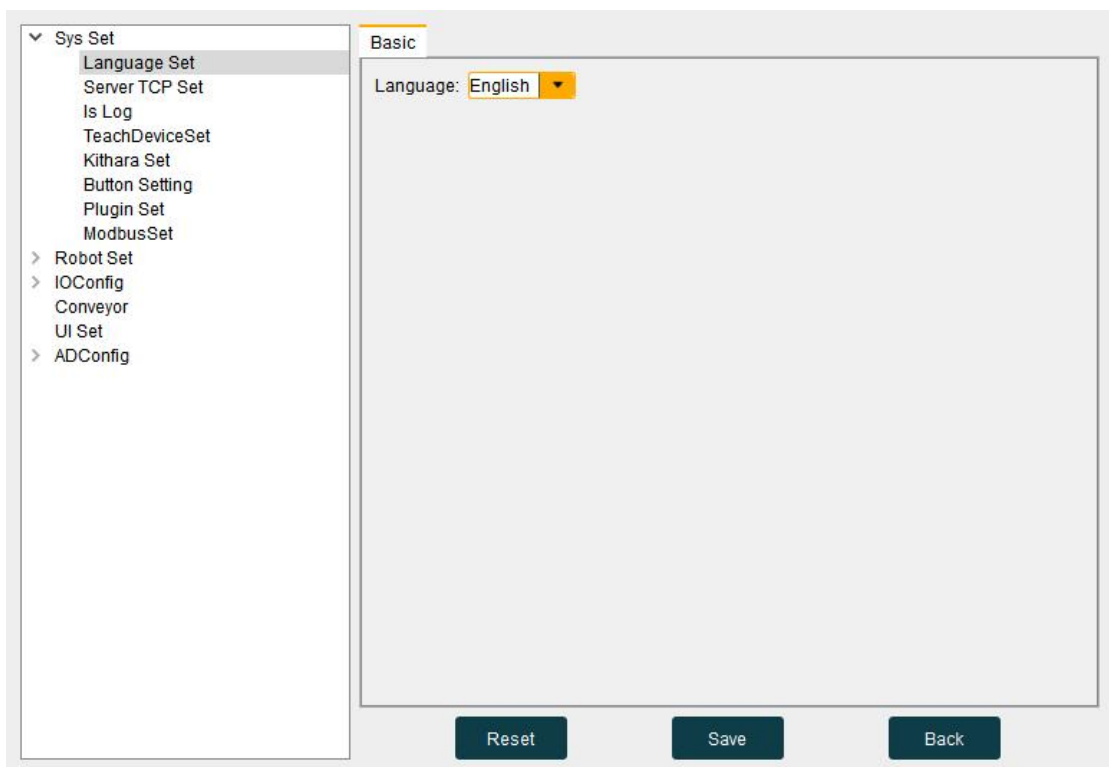
When there are multi-machines, you can choose to switch robots through the



robot alternation drop-down list.

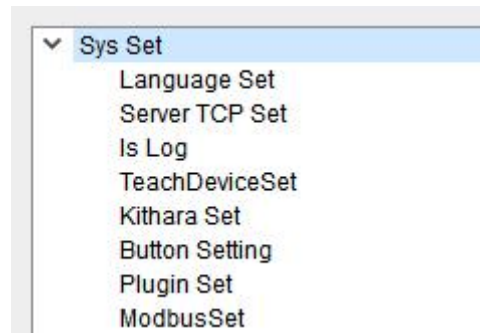
Chapter 4 Set robot

Click the "Set robot"  button on the main screen to enter the robot core parameter settings. Enter the robot setting screen, which is mainly used to set the robot and check the information of the workstation. For example, the number of robots, electric motor and mechanical parameters, the number of conveyor belts included, plug-in settings, etc.

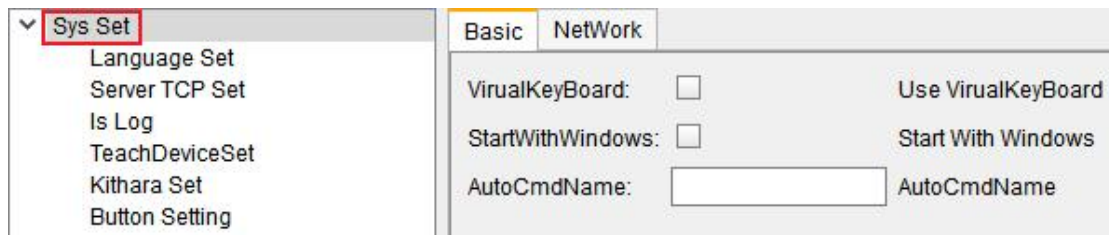


4.1 System Settings

Under the setting surface, system settings include the following :



4.1.1 System Settings



① Basis

virtual keyboard	Use the soft keyboard in the software. After selecting, when the cursor appears in the input field, the soft keyboard will pop-up automatically.
automatic launch when power-on	Automatically launch YiKingSoft software when power-on.
Name of automatic launching programs	If it needs to automatically run a certain program after the robot is powered on, please write the name of the program to be run, and there is no need to select the program when the robot is turned on.

② Network



4.1.2 Language Settings

Two languages can be chosen

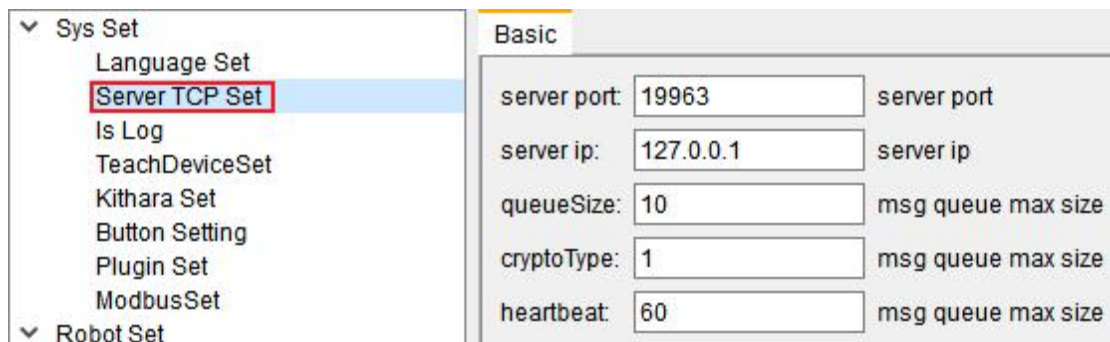


① Basis

Language	Chinese	Chinese
	English	English

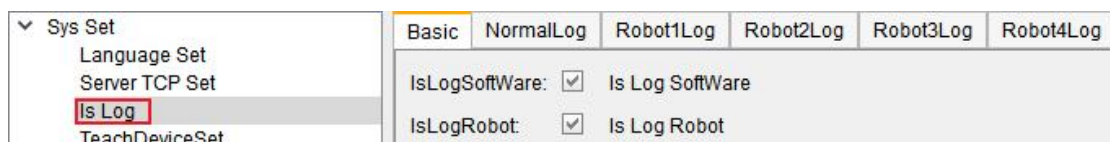
4.1.3 Server Setup

Parameters in this page are basic, no need to change.



4.1.4 Log Management

The system only set up logs for selected programs, it can record the running time of the software and the alarming. It's expedient for problem tracing.



① Basis

Software log	whether log outputs control switch
Robots log	Whether log outputs control switch

②Normal log

Basic	NormalLog	Robot1Log
1002:	☒	StartCmd
1003:	☒	EndCmd
1004:	☐	Kithara
1005:	☒	CmdFile
1006:	☒	RobotHeart
1007:	☒	EmergencyStop
1008:	☒	ParameterSet
1009:	☒	ButtonOperate

Regular logs are used in the same manner as robot logs, to check robot error, etc.

1002	Command program launch log	Default output
1003	Command program ending log	Default output
1004	Kithara operation log	Default non-output
1005	Command files operation log	Default output
1006	Robot heartbeat log	Default output
1007	Emergency stop log	Default output
1008	Parameters settings log	Default output
1009	Key -pressing operation log	Default output

③Robot log

4001: ☐ AddObject	7018: ☐ SetOutputPulse
4002: ☐ AddBox	7019: ☐ SetOutputPWM
4003: ☐ DropObject	7020: ☐ TriggerCamera
4004: ☐ DropBox	7021: ☒ FlyTriggerCamera
4005: ☐ ObjPixelPos	7022: ☐ SetMobileBox
4006: ☐ ObjRobotPos	7023: ☐ SetFixedBox
7000: ☐ PickSuccess	7024: ☐ ResetFixedBox
7001: ☐ PickFaild	7025: ☐ Break
7002: ☐ Move	7026: ☐ Continue
7003: ☐ PlaceSuccess	7027: ☐ CallSubprogram
7004: ☐ PlaceFaild	7029: ☒ ErrorCode
7005: ☐ SyncCmd	7030: ☐ GetIntObjInfo
7006: ☐ Expression	7031: ☐ GetParameters
7007: ☐ SetPickObjFilter	7032: ☐ Loop
7008: ☐ SetBoxTransfer	7033: ☐ SocketSend
7009: ☐ SetObjectTransfer	7034: ☐ SocketReceiveSuccess
7011: ☐ OffSet	7035: ☐ SocketReceiveFailed
7013: ☐ SetCuboidSpace	7036: ☐ GetIntObjNum
7014: ☒ Pause	7037: ☐ UpdateEncoder
7015: ☐ Wait	7038: ☐ SearchL
7016: ☐ SetOutput	7039: ☒ AddLog
7017: ☐ SetOutputBeforEnd	9000: ☐ MoveSpeed

4001	Add object log	Default non-output
4002	Add box log	Default non-output
4003	Desert object log	Default non-output
4004	Desert box log	Default non-output
4005	Object visual coordinate log	Default non-output
4006	Object robots coordinate log	Default non-output
7000	Pick success log	Default non-output
7001	Pick failure log	Default non-output

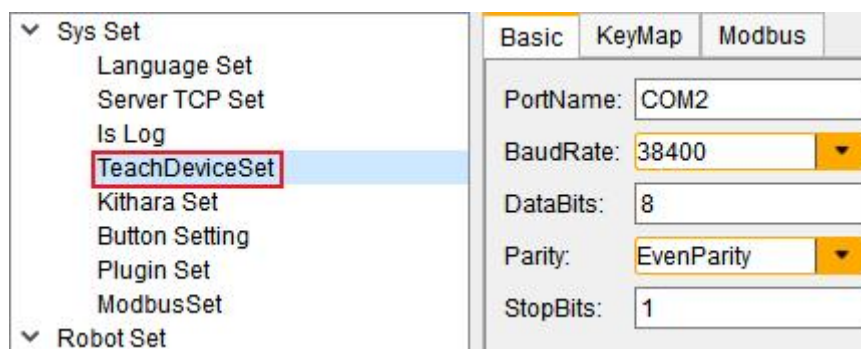
7002	Motion command log	Default non-output
7003	Placing success log	Default non-output
7004	Placing failure log	Default non-output
7005	Synchronous command log	Default non-output
7006	Expression command log	Default non-output
7007	Set pick object filtering command log	Default non-output
7008	Set box shift command log	Default non-output
7009	Set object shift command log	Default non-output
7011	Offset command log	Default non-output
7013	Set cube space command log	Default non-output
7014	Suspend command log	Default output
7015	Waiting command log	Default non-output
7016	IO output command log	Default non-output
7017	IO ante-output command log	Default non-output
7018	IO pulse output command log	Default non-output
7019	IO PWM output command log	Default non-output
7020	Trigger camera command log	Default non-output
7021	Flying shooting triggers camera command log	Default output
7022	Set moving box command log	Default non-output
7023	Set fixed box command log	Default non-output
7024	Re-set fixed box command log	Default non-output
7025	Interrupt command log	Default non-output
7026	Continue command log	Default non-output
7027	Call sub-program command log	Default non-output
7029	Erroneous code	Default output
7030	Obtain integer information of object command log	Default non-output
7031	Obtain object parameters command log	Default non-output
7032	Recycle command log	Default non-output
7033	Socket sending log	Default non-output

7034	Socket receiving success log	Default non-output
7035	Socket receiving failure log	Default non-output
7036	Obtain object number command log	Default non-output
7037	Renew triggering encoder value log	Default non-output
7038	Linear search command log	Default non-output
7039	Messages print command log	Default output
9000	Motion speed command log	Default non-output

Robot 2, 3, and 4 log tables are used to build logs for robots 2, 3, and 4, which are the same as those for robot 1 log table

4.1.5 Teach pendant Settings

The buttons of the teach pendant can be used with serial port communication and Modbus communication, with default being serial port communication. If it is equipped with a new version of the teach pendant, it needs to be configured as Modbus communication, and the following is for details.



① Basis

The serial port communication parameters of the teach pendant buttons can be kept as default.

② Button Mapping

The mapping relation between button and button functions can be set here, keep

it as default.

③Modbus

If using a new version of the teach pendant, select the "Use Modbus" parameter and save the change, re-start the software, and the Modbus can communicate with the buttons of the teach pendant.

Modbus IP and port configuration parameters maintain default commonly.

4.1.6 Kithara Settings

① Basis

Imitation	Software imitation function. If use the robot in practical situation, cancel “√”
Adjustment	Adjustment function just for developers and after selection of "√", robots operation is not possible.
Activating SM	Restore the servo drive parameters to factory settings. (primarily no need to change)
Network	EtherCat major station (Beckhoff EK1100) occupied number of network

adapter card	adapter card
Error detection level	For controlling the error detection strictness, no need to change.

② EGM Function

EGM is an external extension function, the activation and use of the function. If the user needs real-time path change, such as visual track, obstacle evasion, or cooperation with various sensors, the user can send the real-time speed through a certain frequency to control the robotic arm.

This function is used in conjunction with the robot's external extension commands and special guidance software, which is not enabled by default.

Basic	EGM	IntimeVision
EGM Enable: ☐		
EGM NetCard: 0		
EGM IP: 192.168.0.181		
EGM Mask: 255.255.255.0		
EGM Gate: 192.168.0.1		
EGM Port: 2120		

③ Real-time Visual

The function can only be employed in the occasion with higher real-time requirement of visual computing settlement and currently only for functional support in flying shooting. It's defaulted as turn-off.

Basic	EGM	IntimeVision
Enable: ☐		
RobotNo.:	1	1-4
CameraID:	0	0-1
VisionName:	YiVision	same to vision plugin name
NetCard:	0	
NetCard IP:	192.168.0.100	
NetCard Mask:	255.255.255.0	
NetCard Gate:	192.168.0.0	
Image FlipX:	☐	
Image FlipY:	☐	
Image Rotate:	0	

4.1.7 Button Settings

It is used to configure whether the screen function buttons are banned. If it is configured as banned, the function buttons can't be operated on the screen.

① Operation Screen

Used for functional buttons configuring in the Run Program page, At external IO or Modbus applications to start or stop controlling robots, if just permitting effective in external operation, not allow screen functional button effective, just configure to ban corresponding functional buttons.

Sys Set	Robot Run UI
Language Set	Run Button Disable: ☐
Server TCP Set	Pause Button Disable: ☐
Is Log	Resume Button Disable: ☐
TeachDeviceSet	Stop Button Disable: ☐
Kithara Set	
Button Setting	
Plugin Set	
ModbusSet	

4.1.8 Plug- in Settings

① Basis

It is used to configure the loading and unloading of generally used function plug-ins.

The first column is the name, and the second column is the number of plug-ins loaded (0 means no loading, 1 means loading one); with the third column is the description of the plug-in.

Different plug-ins have different configuration permissions and configuration quantity restrictions. Only with configuration permissions and a legal quantity can the configuration be successful.

Basic	CommonCustom	VisionCustom
YiModel:	0	Model Tool
YiPath:	0	Path Tool
YiFly:	0	Fly Shoot Tool
YiWafer:	0	Wafer Function Package
YiWheel:	0	
YiSmoke:	0	
YiVision:	0	Internal Vision
YiExVision:	0	External Vision

② Common customization setting

It's used to configure functional plug-ins that do not exist in the basic plug-ins and need to work in the Common plug-in directory, mainly for the configuration loading of project custom plug-ins, which can only be configured with manufacturer permissions.

Complete configuration loading by the followings:

- copy the plug-in into the “plug-in/Source” menu ;
- input the name of plug-in into the column and maintain both identical ;
- input the number into corresponding column ;

d) save the parameters and re-start the software, seeing whether it's in normal loading

Basic	CommonCustom	VisionCustom
CustomPluginName1:	CommonName1	
CustomPluginCount1:	0	
CustomPluginName2:	CommonName2	
CustomPluginCount2:	0	
CustomPluginName3:	CommonName3	
CustomPluginCount3:	0	
CustomPluginName4:	CommonName4	
CustomPluginCount4:	0	

③ Vision Customization Settings

It is used to configure the functional plug-ins that do not exist in the basic plug-ins and need to work in the Vision plug-in directory. It is mainly for the configuration loading of project custom plug-ins, which can only be configured with manufacturer permissions.

Complete configuration loading by the followings:

- copy the plug-in into the “plug-in/Source” menu ;
- input the name of plug-in into the column and maintain both identical ;
- input the number into corresponding column ;
- save the parameters and re-start the software, seeing whether it's in normal loading

Basic	CommonCustom	VisionCustom
CustomPluginName1:	VisionName1	
CustomPluginCount1:	0	
CustomPluginName2:	VisionName2	
CustomPluginCount2:	0	
CustomPluginName3:	VisionName3	
CustomPluginCount3:	0	
CustomPluginName4:	VisionName4	
CustomPluginCount4:	0	

4.1.9 Modbus Settings

When the robot uses Modbus to communicate with external devices through a network cable or serial port, complete the Modbus parameter configuration on this page.

The Modbus function currently supports a slave station function, and the slave station ID allows configuration within the range of 1-4; supports 4 main station functions; the main station and slave station functions can be supported at the same time.

For details about Modbus settings, refer to Modbus manual.

	ModbusSlave	ModbusMaster1	ModbusMaster2	ModbusMaster3	ModbusMaster4	Other
▼ Sys Set Language Set Server TCP Set Is Log TeachDeviceSet Kithara Set Button Setting Plugin Set ModbusSet Robot Set	Enabled: ☐ Connection: ModbusTcp IP: 127.0.0.1 Port: 502					

①Modbus Slave Station

The configuration of slave station function are as followings:

- Select the enable button ;
- Set the slave station ID;

- c) Select the connection method (network cable/serial port) ;
- d) Being the network cable link, directly keep the parameters ;
- e) Being a serial port connection, select the connection mode as "SerialPort", save and restart the software ;
- f) After the software restarts, configure the serial port communication parameters in light of the specific situation and save it.

ModbusSlave	ModbusMaster1	Modi
Enabled: ☐		
Connection: ModbusTcp		
Port: 502		
SlaveID: 1 1~4		

②Modbus Main Station

Four Modbus main stations can be set, each main station can be configured separately, and one or several main stations can also be enabled, configuration as follows :

- a) Select the enable button ;
- b) Select the connection method (network cable/serial port) ;
- c) Being the network cable link, directly keep the parameters ;
- d) Being a serial port connection, select the connection mode as "SerialPort", save and restart the software ;
- e) After the software restarts, configure the serial port communication parameters in light of the specific situation and save it.

ModbusSlave	ModbusMaster1
Enabled:	☐
Connection:	ModbusTcp
IP:	127.0.0.1
Port:	502

③ Others

In configuring the parameters related to Modbus main station and slave station communication, generally use the default parameters.

SlaveIntFormat:	AB CD	A:Low Byte D:High Byte AB:One Word
SlaveRealFormat:	AB CD EF GH	A:Low Byte D:High Byte AB:One Word
SlaveStringFormat:	Intel	Intel low bytes ahead,Motorola high byte in front
MasterIntFormat:	AB CD	A:Low Byte D:High Byte AB:One Word
MasterRealFormat:	AB CD EF GH	A:Low Byte D:High Byte AB:One Word
MasterStringFormat:	Intel	Intel low bytes ahead,Motorola high byte in front
SlaveHoldRegPowerSave:	☐	Register hold during power off
SlaveRegistersNb:	400	
MasterRegistersNb:	400	
RegisterBlock1Enable:	☐	
RegisterInBlock1Start:	0	
RegisterInBlock1End:	99	
RegisterOutBlock1Start:	100	
RegisterOutBlock1End:	199	
RegisterBlock2Enable:	☐	
RegisterInBlock2Start:	200	
RegisterInBlock2End:	299	
RegisterOutBlock2Start:	300	
RegisterOutBlock2End:	399	

Parameters	Description
Integer-form format	Supports 4 byte sequence combinations, which can meet all integer formats
Float-type format	Supports 4 byte sequence combinations, which can satisfy all floating-point type formats. Motorola is a kind of version-compatible type and is not suggest.
Character-type format	Support two byte sequence combination.

Slave station conducting register save	Whether the value of the slave station conducting register is saved under power-off, it is not saved by default.
Register quantity	Decide the registers quantity allocated
Register block	Main station register address, used to determine the range of input and output register addresses.

4.2 Robotic Settings

The robot settings mainly reflect the selected robot model and the settings of the robot axis joints. Up to 4 robots can be connected through the software, that is, 1 main and 3 slaves.

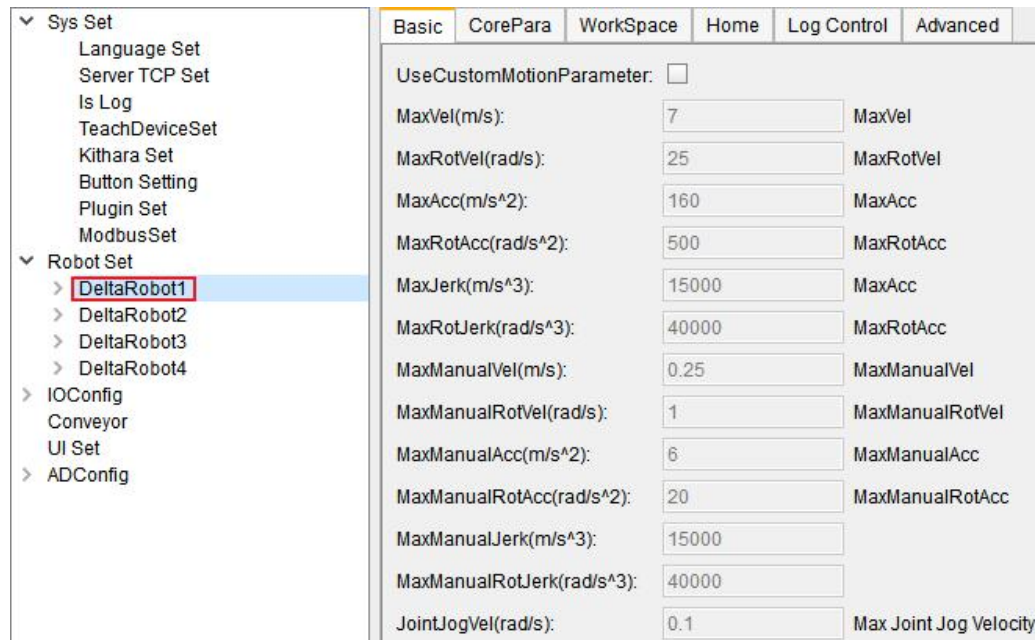
The parameters have been set before leaving the factory, and generally the user does not need to change them.

The screenshot displays the 'Robot Set' configuration window in the YiKingSoft software. The left sidebar shows a tree view of settings categories: Sys Set, Language Set, Server TCP Set, Is Log, TeachDeviceSet, Kithara Set, Button Setting, Plugin Set, ModbusSet, Robot Set (highlighted), DeltaRobot1, DeltaRobot2, DeltaRobot3, DeltaRobot4, IOConfig, Conveyor, UI Set, and ADConfig. The main panel is divided into two tabs: 'Basic' and 'Speed'. The 'Basic' tab is active, showing configuration for four robots. The 'Robot Number' is set to 4. The 'Robot1 Type' through 'Robot4 Type' are all set to 'ERT_Delta'. The 'Robot1 Model' through 'Robot4 Model' are all set to 'BAT1100-A3'. The 'Pause Control Set' through 'End Control Set' are all set to 'CurrentRobot'. The 'ExStartCmdFormat' is set to 'Cmd_XXX', and the 'Heartbeat Log Time' is set to 180.

Parameter	Value	Description
Robot Number	4	Set robot number
Robot1 Type	ERT_Delta	Set the first robot type
Robot2 Type	ERT_Delta	Set the second robot type
Robot3 Type	ERT_Delta	Set the third robot type
Robot4 Type	ERT_Delta	Set the fourth robot type
Robot1 Model	BAT1100-A3	Set the first robot model
Robot2 Model	BAT1100-A3	Set the second robot model
Robot3 Model	BAT1100-A3	Set the third robot model
Robot4 Model	BAT1100-A3	Set the fourth robot model
Pause Control Set	CurrentRobot	Set Pasue Control Action
Resume Control Set	CurrentRobot	Set Resume Control Action
Jump Control Set	CurrentRobot	Set Jump Control Action
NextStep Control Set	CurrentRobot	Set NextStep Control Action
Step Control Set	CurrentRobot	Set Step Control Action
Continue Control Set	CurrentRobot	Set Continue Control Action
End Control Set	CurrentRobot	Set End Control Action
ExStartCmdFormat	Cmd_XXX	Set the ExStart Cmd Format
Heartbeat Log Time	180	

Parameters	Description
Type	Delta = Bat parallel robot (3+1 axis) Diamond = Bat Parallel Robot (2 Axis) Scara = Python scara robot Lobster = wafer handling robot PumaArm = Mantis six-axis robot
Model	Specific model
Multi-machine debugging button control	CurrentRobot: The current robot is valid AllRobot: Valid for all robots
Externally launch command format	Externally launch command file naming format, such as Cmd_001, Cmd_0001, etc.
Heartbeat log time	Heartbeat log output interval, keep the default value

4.2.1 Robot Parameters Setting



① Basic

For the motion parameters of the robot, the default values can be used commonly, and the “manufacturer” authority is required to modify them.

② Core Parameters

Viewing the core parameters of the robot, some parameters can be modified with engineer authority.

Basic	CorePara	WorkSpace	Home	Log Control	Advanced
Robot Model: BAT1100-A3					
UpLeg(mm): 385					
LowLeg(mm): 900					
StaticPlate(mm): 200					
MovingPlate(mm): 60					
SPlateFai1(°): 0 Rotation angle of coordinate system					
SPlateFai2(°): 120 Rotation angle of coordinate system					
SPlateFai3(°): 240 Rotation angle of coordinate system					
Fei1(°): 0 Rotation angle of coordinate system					
Fei2(°): -90 Rotation angle of coordinate system					
Fei3(°): -180 Rotation angle of coordinate system					
Fei4(°): -270 Rotation angle of coordinate system					
RotGearRatio4(°): 30 Rotation angle of coordinate system					

Parameter	Description
Static platform angle	Modify the angle between the robot's base coordinate system and the initial coordinate system, which is generally used to ensure the incoming material direction being consistent with the robot's Y-axis direction when for tracked pick place. static platform angle can only increase positively

③ Working Space

Viewing the workspace of the robot, modification is not allowed.

Basic	CorePara	WorkSpace	Home
R1Lim(mm): 550			
R2Lim(mm): 425			
ZMaxLim(mm): -700			
Z1MinLim(mm): -950			
Z2MinLim(mm): -1000			

④ Home

Check the Home point position of the robot and the corresponding zero point encoder value of each axis, and modification is not allowed.

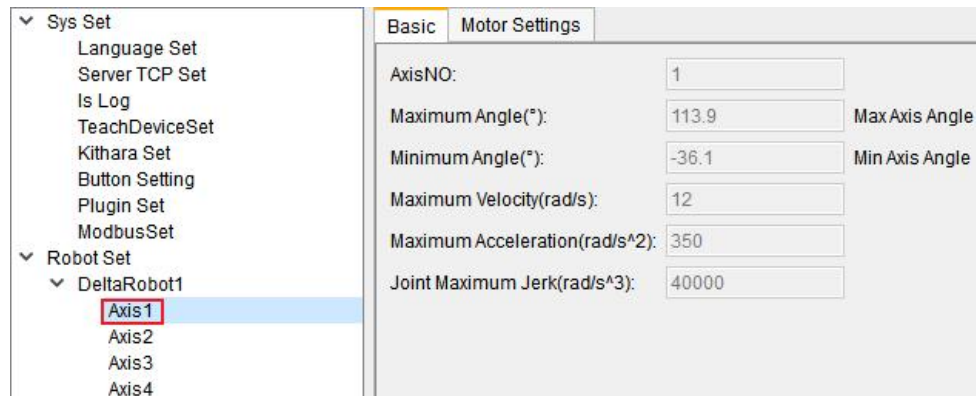
Basic	CorePara	WorkSpace	Home
HomeX(mm): 0			
HomeY(mm): 0			
HomeZ(mm): -710			
HomeRot(*): 0			
ZeroEncoder1: 0			
ZeroEncoder2: 0			
ZeroEncoder3: 0			
ZeroEncoder4: 0			

⑤ Log Management

Robot operation log output control, used when debugging and tracing robot exceptions, under normal circumstances, use the default parameters.

Basic	CorePara	WorkSpace	Home	Log Control	Advanced
Debug Axis: ☒					
Debug AxisGroup: ☒					
Debug Motor: ☒					
Debug Traj: ☒					
Debug Kin: ☒					
Debug IO: ☒					
Debug Level: Debug_No_Level ▼					

4.2.2 Robot axis Parameter Setting



①Basics

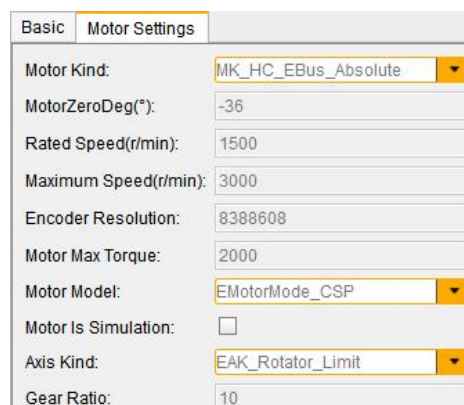
View or modify the joint limit and motion speed parameters of the motion axis.

Some parameter modifications require the engineer login authority.

②Electric Motor Settings

View or modify the servo electric motor parameters of the motion axis, and some parameter modifications require the support of engineer login authority.

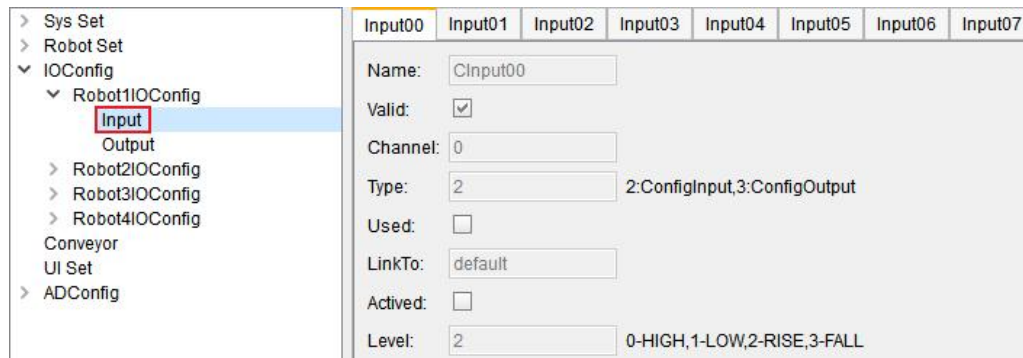
The Bat delta robot model needs to fill in the zero angle of the electric motor, and copy the angle value marked on the upper arm of each axis to the red box area (write 0° for 4th axes). This angle has been filled in before leaving the factory, and the user does not need to change it. Check it if necessary, if it is wrong and not the value marked on the upper arm, fill in again.



4.3 IO Configuration

IO configuration needs to be done in "IO Configuration "of" Functional Operation".

This screen is only used for viewing configuration parameters.



Parameter	Description
Name	Configure the IO path name
Validness	Whether the channel is valid, it is valid by default
Activated	Whether IO configuration is enabled, if not enabled, the configuration is invalid
Connected to	IO-related functions
Activity	The IO function is determined during the operation of the program, such as the camera triggering channel
Electric level	When the associated function is valid, the corresponding IO state, 0-high, 1-low, 2-rising edge, 3-falling edge

4.4 Conveyor Belt Parameter Settings

The value in the red box is generated after conveyor calibration. These values are by default when the belt is not calibrated.

Describing the positional relationship of the conveyor belt, we kept two kinds of data to describe the positional relationship of the conveyor belt in consideration

of intuition and ease of use.

1. Conveyordeg_X, Conveyordeg_Z: horizontal angle, vertical angle, this expression relationship is more intuitive after the calibration, in line with the actual object
2. dXUnit, dYUnit, dZUnit: unit vectors in the three directions of x, y, and z
3. Conveyor belt proportional coefficient: conveyor traveling distance that corresponds to one pulse (unit: mm)
4. Resolution: the resolution of the encoder
5. Conveyor belt speed check interval (m): specify what distance the conveyor belt moves when the system refreshes the current speed once
6. Conveyor belt inspection time interval (s): Specify how long the interval is and the system renews the current speed
7. Speed filter: Specify how many previous speed calculation results are added to the current result to calculate the average one.
8. Tracking coefficient: During the tracking process, due to the fluctuation of the conveyor belt, the target position of the robot calculated by the system departs from the actual position of the object on the conveyor belt. From the moment the tracking action starts, a few percent of the deviation is taken per cycle to compensate the calculated robot target position. (Tracking coefficient * deviation + robot target position)
9. Simulation Velocity (SimulationVel) (m/s): Conveyor belt simulation velocity
10. Simulation: Simulation function, if checked, the system will load a virtual

moving conveyor belt, even if the real conveyor belt does not move, the robot will think that the conveyor belt is moving.

Conveyor0	Conveyor1	Conveyor2	Conveyor3
Conveyordeg_X: -1.23782			
Conveyordeg_Z: -1.74786			
dXUnit: -0.0215923			
dYUnit: 0.999301			
dZUnit: 0.305011			
ConveyorRatio: 0.0505493			
EncoderResolution: 1000			
dCheckDisVer: 0.1			
dCheckTime: 0.1			
dVertFilt: 9			
IrTrack: 0.1			
simulationVel(m/s): 0.2			
Simulation: ☐			

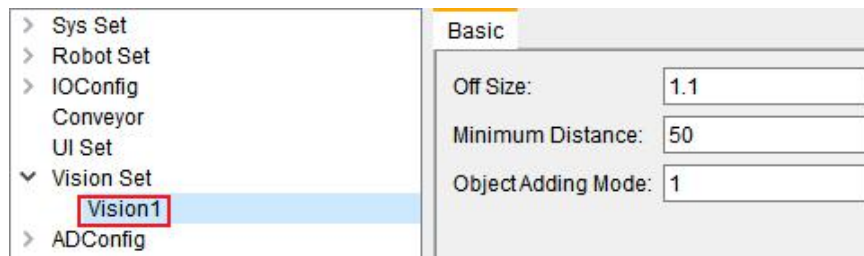
4.5 UI Settings

Basic
Resolution: 1 Resolution0 640*480 1 800*600 2 1024*768 3 1280*800 CanMove: ☒ Can be dragged

Parameter	Description
resolution	Set the display size of the software screen, select an appropriate resolution in light of the screen size of the teach pendant actually equipped with the robot, and restart the software to activate the new setting
window position change	Whether the software screen is running when is dragged. After checking, the parameters will put into use immediately

4.6 Visual Settings

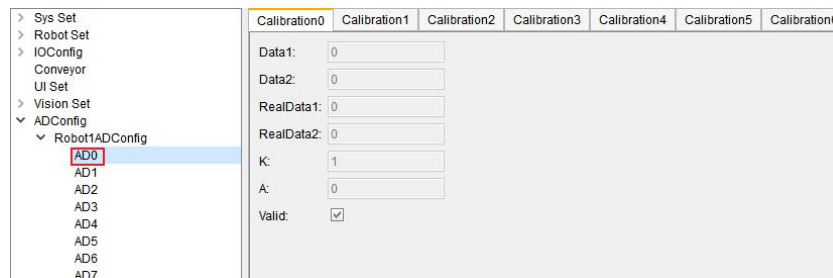
This configuration item will appear only after the vision plug-in (YiVision/YiExVision) is successfully loaded.



Parameter	Description
Size tolerance	The fault tolerance for object size when it is added, the default is 1.1, the larger the fault tolerance value means the greater the fault tolerance, generally not more than 2
Shortest distance	The minimum distance between picked objects, used for screening, the unit is mm
Object adding mode	0-no sorting, 1-automatic sorting

4.7 AD Configuration

Refer to the AD function manual for details



Chapter 5 Function Operation

Function operation menu is a collection of robot-related functions, including conveyor belt calibration, visual calibration, operation, TCP configuration, log, Kithara monitoring, IO configuration, input register configuration, TCP socket, simulation, external axis, BASE configuration, LOAD configuration, Jog and etc..

5.1 Conveyor Belt Calibration

Conveyor belt calibration requires a calibration cone and a plate. Within the working area of the robot, inch the robot to make the calibration cone at the end of the robot to contact the plate.

After the conveyor belt is calibrated, verification is required to check whether the calibration is within the allowable range of error. The verification method is the same as the calibration one.

Refer to YiKingSoft Vision Manual for details.

ConveyorCalibrate VisionCalibrate Operation TCP Config Log KitharaMonitor IOConfig InputRegConfig TcpSocket End

Conveyor Select: Conveyor0

Tip: 1.Click button "Start" to Calibrate the Conveyor.

	X	Y	Z	Encoder
Point1				
Point2				

Result

ConveyerRate: 0.001

HorizontalAngle: 0

VerticalAngle: 0

Horizontal angle:a

Vertical angle:b

Calibrate Verification

LoadPoint Start End Save

5.2 Visual Calibration

The only difference between vision calibration and conveyor belt calibration is vision calibration requires setting of a camera environment. When the vision can be showed in the software, calibration can be used.

Refer to YiKingSoft Vision Application Manual for details.

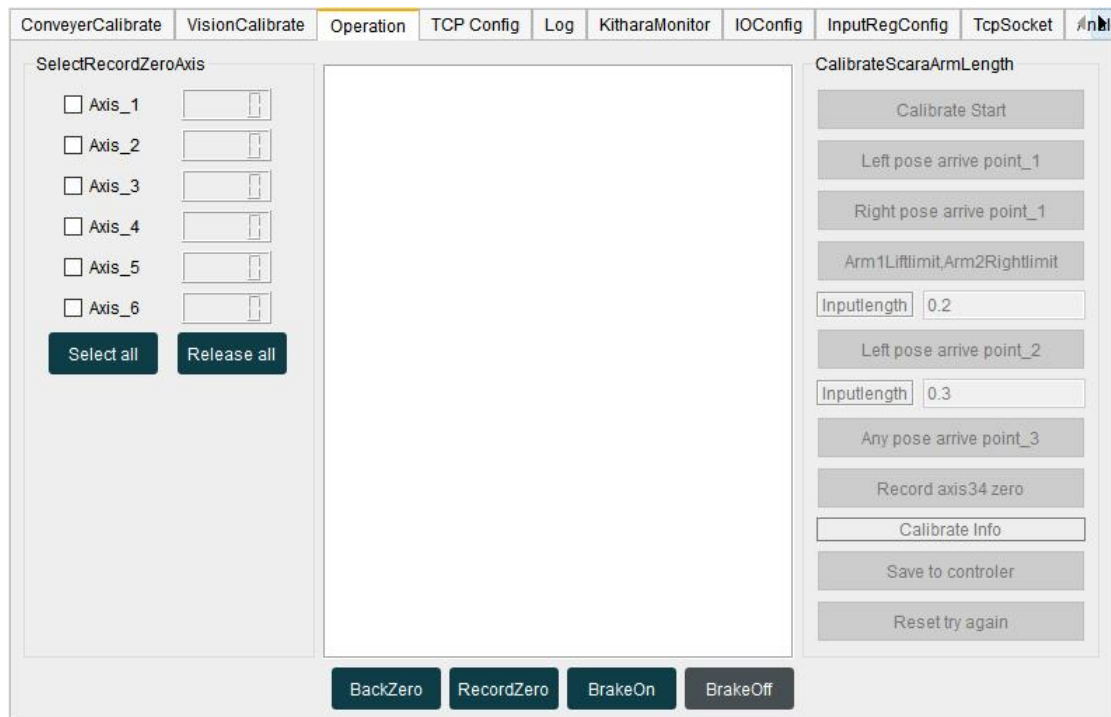
The screenshot shows the 'VisionCalibrate' window of the YiKingSoft software. The window has a menu bar at the top with options: ConveyorCalibrate, VisionCalibrate (selected), Operation, TCP Config, Log, KitharaMonitor, IOConfig, InputRegConfig, TcpSocket, and a help icon. Below the menu bar, there are two dropdown menus: 'Vision Select' set to 'YiExVision' and 'Calibration Mode' set to 'VNRN'. A tip message reads: 'Tip: 1.Select one vision and calibration mode,Click the button "Start" to Calibrate the vision.' The main area contains a table for inputting coordinates:

	X	Y	Z	Conveyor
Vision Point1:				
Vision Point2:				
Robot Point1:				
Robot Point2:				

Below the table is a 'Result' section with four input fields: 'OffSetX:', 'Scale:', 'OffSetY:', and 'RotationAngle:'. At the bottom of the window are four buttons: 'LoadPoint', 'Start' (highlighted in dark blue), 'End', and 'Save'. On the right side of the window, there is a vertical toolbar with three buttons: 'Calibrate' (highlighted in yellow), 'Verification', and 'Image'.

5.3 Operation

The operation page can realize the functions of "record axis zero point", "brake operation", "return to zero" and "arm length calibration".



5.3.1 Return to Zero

Specially for Delta robot models, after pressing the button, the 1st, 2nd, and 3rd axes of the robot will automatically be at the zero point, and the 4th will record the current position as zero. The Delta robot must be under the automatic mode when using the "Return to Zero" button to return to zero, and the Delta robot with 4th axes must remove the lower arm before returning to zero. Delta models can utilize the "Return to Zero" button, or utilize "Record Zero" to record the axis zero.

5.3.2 Record Zero Point

Any robot model can use this function, which features that multiple axes or a single axis zero point can be recorded at the same time by ticking the option. Just move each axis that you want to record zero point to the zero point position, check these axes on the left, and press "Record Zero Point".

- ① The zero point of the 1st, 2nd, and 3rd axes of the Delta model is at the hardware limit of the upper arm. Just open the brake and push the upper arm to the hardware limit. There is no regulation for the 4th axis zero point. You can choose an expedient position in light of the actual situation.
- ② The 1st and 2nd axes of the Scara robot model have zero scale grooves (each axis has a groove at the fixed position and rotation position), aligning the grooves to get the zero position. The 3-axis zero point is above the 3-axis motion area. Without specific regulation, it should be as high as possible but not hit the load mounting block. There is no regulation on the 4-axis zero point, and a expedient position can be selected in light of the actual situation. If you do not use the teach pendant to move the 3rd and 4th axes, you must press and hold the brake release button on the fore-arm at the same time when moving. The 3rd and 4th axes are mutually coupled, and their zero points must be recorded at the same time.
- ③ The 1-5 axis of the Mantis robot model has a zero scale groove (each axis has a groove at the fixed place and rotation position), aligning the grooves to get the zero position. There is a groove at the fixed position of the 6-axis. It is suggested to make a groove when designing the gripper or make a mark when it is installed. The position where the gripper groove or mark is aligning with the 6-axis groove is set as the 6-axis zero point.

5.3.3 Opening the Holding Brake

Release the brake button.

5.3.4 Closing the Holding Brake

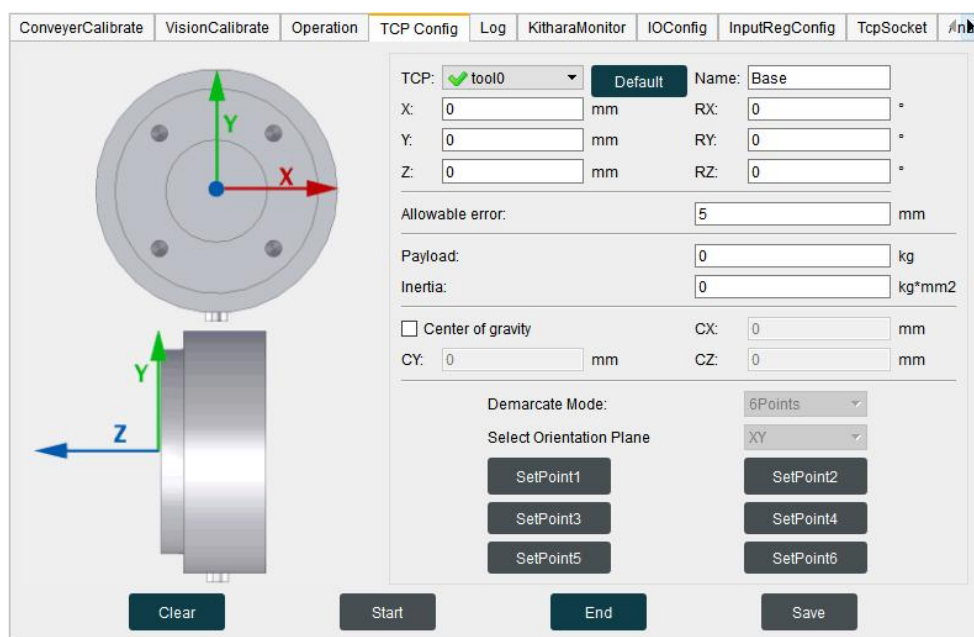
Re-close the brake button, which is only valid when the brake is released.

5.3.5 Calibrate Arm Length

Only for Scara robot series, it is used for robot calibration during manufacturing, and it's prohibited for user.

5.4 TCP Configuration

The TCP configuration screen mainly is for the calibration and parameter settings of the tool coordinate system at the end of the robot arm.



1. Tool0 (Name: Base) is the default TCP, located at flange center at the end of the robot which can't be modified by users and is the basis of other TCP.
2. Other options rather than tool0 in the TCP drop-down list, the Tool

corresponding to the red circular exclamation mark is permitted to make a new tool coordinate system for the currently selected robot, such as Tool04; the Tool corresponding to the yellow triangle exclamation mark is already taken up by other robots. The tool coordinate system, such as Tool01, it is not suggested to directly build a new tool coordinate system on base of the existing one.

3. Select Input from the calibration mode drop-down list. In the X/Y/Z column, fill in the offset between the tool tip point and tool0 on the X-axis, Y-axis, and Z-axis. In the RX/RZ column, fill in the Euler angle value between the pointing direction of the tool tip and tool0 on the X-axis, Y-axis, and Z-axis, with unit being the angle unit, following the right-hand rule.

4. Allowable error: In light of the project requirements, the user can input the allowable error value of the TCP calibration. When Input is selected in the calibration mode drop-down list, Allowable error can be left empty.

5. Payload: tool load

6. Inertia: tool inertia

7. Center of gravity: Check this option to set position of gravity center, CX, CY, and CZ are the gravity center offsets along the XYZ axis.

8. Save (Save): Save the current settings

9. Default: Click this button, the current TCP in the TCP list is set as the default, and all point coordinates are captured by the default TCP. Before selecting, you should write the "name", then change the coordinate value, and click the "save" button, and then clicking "Default" to set it as the default.

Note: The interjection mark means not saved with name. The mark before tool06 will be removed after saving.

The screenshot shows the TCP configuration interface. The 'TCP' dropdown menu is set to 'tool06' with a red warning icon. The 'Default' button is highlighted. The 'Name' field contains 'tool6'. The coordinate fields are: X: 1 mm, Y: 2 mm, Z: 3 mm, RX: 0, RY: 0, and RZ: 10. The RY field has a red highlight.

After clicking the "Default", "√" will appear before tool06, and tool06 is set as the current default tool coordinate system.

The screenshot shows the TCP configuration interface after clicking the 'Default' button. The 'TCP' dropdown menu is now set to 'tool06' with a green checkmark icon. The 'Default' button remains highlighted. The 'Name' field contains 'tool6'. The coordinate fields are: X: 1 mm, Y: 2 mm, Z: 3 mm, RX: 0, RY: 0, and RZ: 10.

10. Clear: Clear the Current Settings

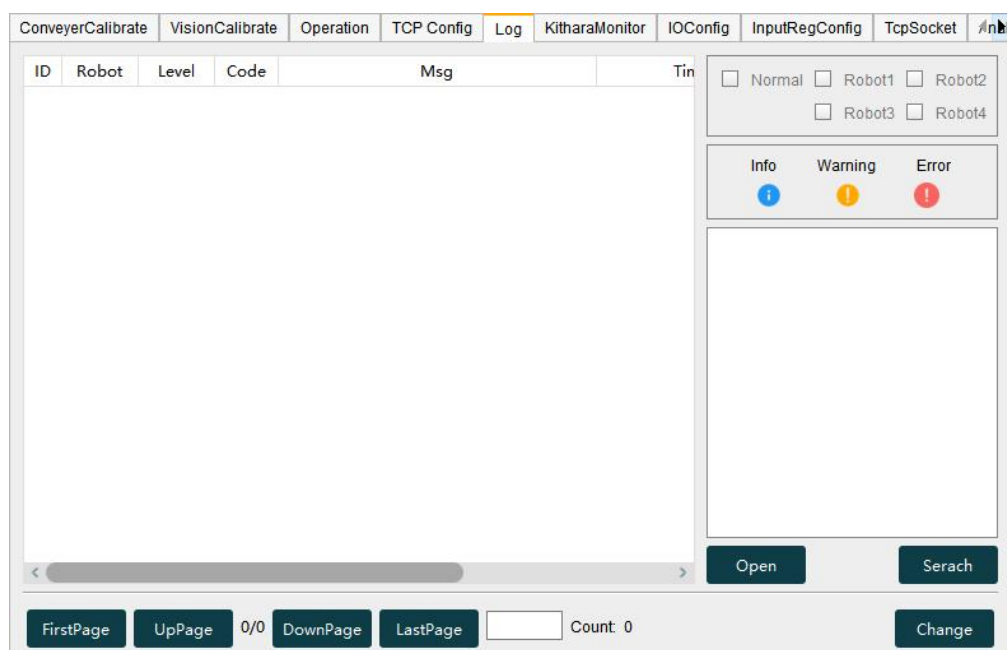
Note: For Mantis models, when the gripping structure being complex, the 3-point /4-point /6-point method can also be utilized to calibrate TCP.

Select 3 points/4 points/6 points in the "Calibration Mode" drop-down list, select the XY axis/YZ axis/ZX axis first in the "Select Coordinate System Plane", and press "Start" to manually calibrate. After the calibration is done, please press "Save" to save the calibration results. If you want to give up the calibration during the process, press "End" to exit.



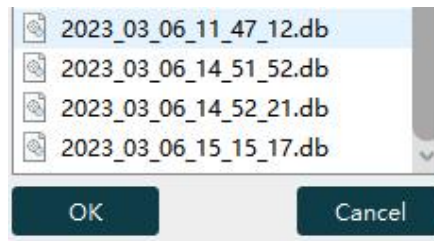
5.5 Log

The log screen is shown below, whose function is to view the status and issues that may happen when the robot is running. Every time the robot software starts up, a log file will be automatically created to record the events during the software running. The log file name is created in light of the system time since the software starts.



5.5.1 Log View

Click "Open", the screen below will appear, select the log to view, click "Finish", and the log screen will load the corresponding information; click "Cancel", and return to the screen.



Note: The log list is sequenced in accordance with creation dates by default, and the last one is the current one.

5.5.2 Log Screening

After the log is loaded, you can screen out the logs that meet the requirements by selecting "robot number", "log level" and "function code", which is expedient for log view.

As shown below, check "Robot 1", click "Error", check "7029", and then click "Search", logs with corresponding function code of "Robot 1" as "7029" and the information level as "Error" can be screened out. If it is not checked, logs will display without screening.

☐ Normal	☒ Robot1	☐ Robot2
	☐ Robot3	☐ Robot4
Info 	Warning 	Error 
☐ 1006	RobotHeart	
☒ 7029	ErrorCode	

Open

Serach

5.5.3 Log Management

The software will automatically clear earlier logs to avoid excessive log data.

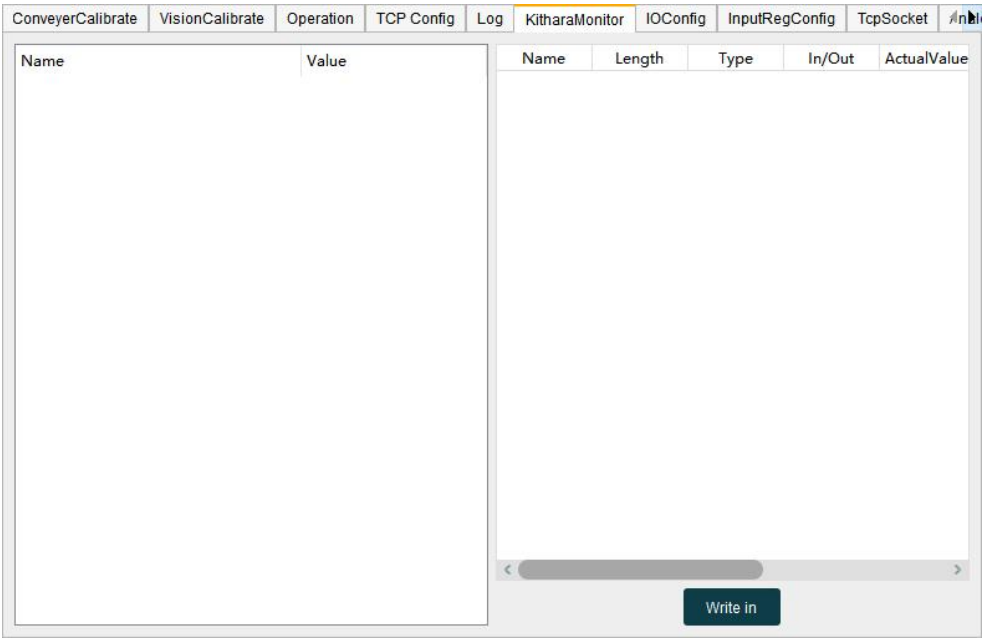
When tracing problems, in order to avoid important logs from being deleted, the following strategy is adopted.

Log protection policy: save at least the last 10 log files, and at least save the last 3 days log files.

To judge whether the logs will be kept, you can sort the logs by date. The last 10 log files will be kept. If the logs with an earlier date are within 3 days, they will be kept, otherwise, the it will be deleted.

5.6 Kithara Monitoring

Through this screen, you can check what modules the hardware contains and its status, no need to change.

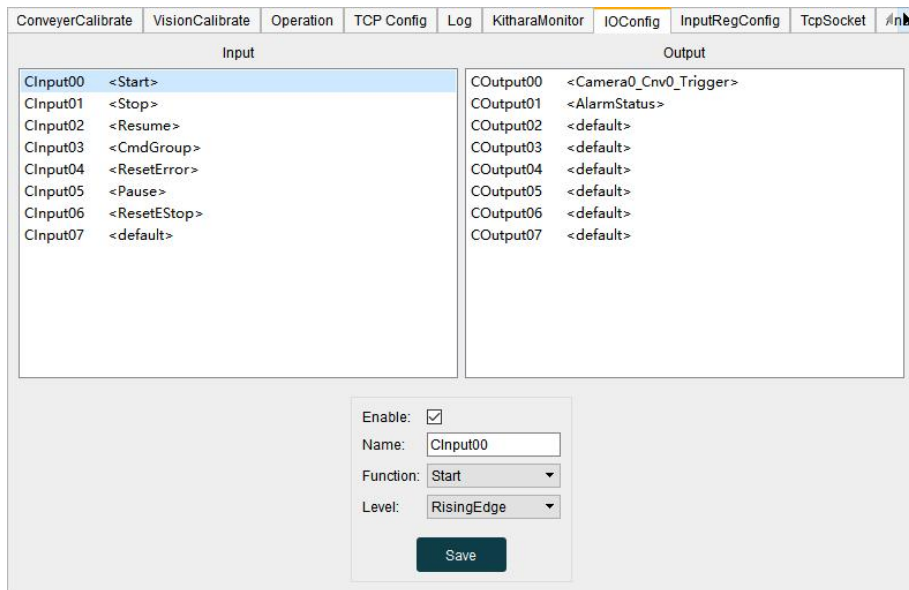


5.7 I/O Configuration

IO configuration realize the binding of IO channels with specific functions or status, and the binding function will be activated by triggering IO signals.

The left column is the input IO configuration, with right one being the output IO configuration.

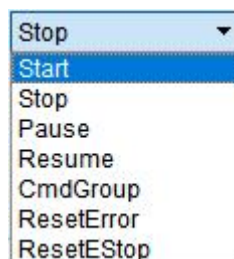
By default, the non-configured IO exhibits default, and "enable" is unchecked. If it has been configured, it shows the specific function or status name.



Configuration steps:

- a) Select the IO to be configured
- b) Check the "Enable" option to enable the configuration of the selected IO
- c) Click the "Function" drop-down list and select the corresponding function (status)
- d) Click the "Level" drop-down list to select the corresponding level state when the function (state) triggered

Enter the IO function drop-down list



Start: start the robot, the level can only be in the rising edge or falling edge

Stop: Stop the robot; the level can only be in rising edge or falling edge

Pause: Pause the robot, the level can only be in rising edge or falling edge

Resume: Resume the robot; the level can only be in rising edge or falling edge

CmdGroup: command calculation group, utilized to determine starting up of the corresponding command, any group of inputs can be configured

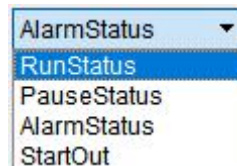
CmdGroup, can be set at intervals, after the configuration complete, the command will be produced in light of the current level of all configured groups

For the serial number, the level can only be in high level

ResetError: error reset, the level can only be in rising edge or falling edge

ResetEStop: emergency stop reset, the level can only be in rising edge or falling edge

Output IO status drop-down list



RunStatus: The running state of the robot. When it is in the running state, it outputs the set level; otherwise, the vice versa

PauseStatus: The robot is in the pause state. When it is in the pause state, it outputs the set level; otherwise, the vice versa

AlarmStatus: robot error state, when in error state, output setting level; otherwise, the vice versa

StartOut: The heartbeat state of the robot, the automatic mode is valid, when the heartbeat is normal; the pulse cycle is 2S, with the duty cycle being 50%

5.8 Input Register Configuration

Configure or check the corresponding relationship between the Modbus slave input register address and the function status. The default value can meet requirements of most of the usage occasions. If it is not necessary, it is not suggested to modify it.

The byte order of the Modbus type is consistent with the format of the slave station data type in the [Robot Settings]-[System Settings]-[Modbus Settings]-[Others] page.

In a multi-machine system, the register address corresponding to the functional state of the slave = configuration address + 50*(robot number-1).

ID	Name	Type	Reg Address	Reg Length
0	RobotState	Int	0	1
1	RunStatus	Int	1	1
2	ErrorCode	Int	2	1
3	PosX	Int	3	1
4	PosY	Int	4	1
5	PosZ	Int	5	1
6	PosRX	Int	6	1
7	PosRY	Int	7	1
8	PosRZ	Int	8	1
9	RunMode	Int	9	1

Name: Type:
 Reg Addr: Reg Len:

5.8.1 Function Status

Name	Register length	Status
robot state	1	Robot axis group status, 0: Uninitialized

		1: Initialize 2: Self-test state 3: Ready state 4: Running state 5: Jog running state 6: Back to zero state 9: Brake state 10: Emergency stop 11: Error state 12: Safe running state
Operating status	1	The specific small state when the robot state is in the running state, 0:Running 1:Pause 2: Steady state, then you can jog 3: Run to the pause point
error code	1	The error code value when the robot is in an error state, 0 means no error
position X	1	X coordinate value of robot Tool0 relative to Base0, unit 0.1mm
	2	X coordinate value of robot Tool0 relative to Base0, unit 0.001mm
position Y	1	Y coordinate value of robot Tool0 relative to Base0, unit 0.1mm
	2	Y coordinate value of robot Tool0 relative to Base0, unit 0.001mm
Position Z	1	Z coordinate value of robot Tool0 relative to Base0, unit 0.1mm
	2	Z coordinate value of robot Tool0 relative to Base0, unit 0.001mm
position RX	1	The RX coordinate value of robot Tool0 relative to Base0, the unit is 0.1 degree
	2	RX coordinate value of robot Tool0 relative to Base0, unit 0.001

		degree
position RY	1	RY coordinate value of robot Tool0 relative to Base0, unit 0.1 degree
	2	RY coordinate value of robot Tool0 relative to Base0, unit 0.001 degree
position RZ	1	The RZ coordinate value of robot Tool0 relative to Base0, the unit is 0.1 degree
	2	RZ coordinate value of robot Tool0 relative to Base0, unit 0.001 degree
manual mode	1	0: Manual mode 1: Automatic mode

5.8.2 Function Configuration

"Engineer" or higher authorization is required, otherwise, only allows viewing but not changing.

	ID	Name	Type	Reg Address	Reg Length
1	0	RobotState	Int	0	1
2	1	RunStatus	Int	1	1
3	2	ErrorCode	Int	2	1
4	3	PosX	Int	3	1
5	4	PosY	Int	4	1
6	5	PosZ	Int	5	1
7	6	PosRX	Int	6	1
8	7	PosRY	Int	7	1
9	8	PosRZ	Int	8	1
10	9	RunMode	Int	9	1

Name:

Type:

Reg Addr:

Reg Len:

1
2

Save

a) In the configuration graph, select the row corresponding to the configuration function item

b) If the length can be configurable, the register length drop-down list can be

selected, as well as the corresponding length item

c) Click the "Save" button, and the register addresses corresponding to all functional states are renewed.

d) Repeat the steps above mentioned to complete the configuration of all function items, restart the software, and the configuration parameters will put into use.

5.9 TCP Sockets

It is used for TCP Socket communication configuration, and supports up to 4 Socket communication at the same time: TCP Socket 0, TCP Socket 1, TCP Socket 2, and TCP Socket 3.

After the socket configuration is finished, the data can be sent through the robot SocketSend/SocketReceive command.

Each Socket communication can be used as "Tcp server" or "Tcp client", and the default is "unused".

Taking TCP socket 0 as an example, if you use a robot as a server, select Tcp Server (TcpServer), and if you use a robot as a client, select Tcp Client (TcpClient), or if you do not use Socket, select Unused, and click The "Save" button.

5.9.1 Client Settings

The socket can be used as a client. Firstly, select "Tcp client" and set the parameters. After saving, you can complete the "client setting" without re-start.

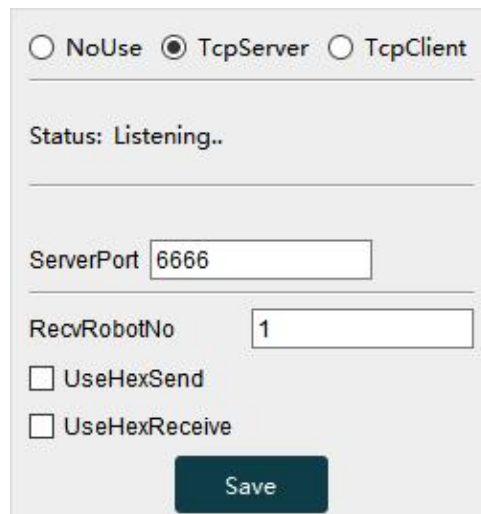
Once the association is established, you should modify the IP or port parameters.

After modifying, you will switch to "Unused" to disconnect, and then select "Tcp

Client" to make the parameters change valid. In the case of multi-Socket communication, special attention is needed.

Parameter	Description
IP	IP address of the server, if the server and the client sits on the same industrial computer, the IP can be set to 127.0.0.1
The port number	The port number that the server monitors.
Receive robot serial number	The socket is bound to the robot, and the reception can only be conducted by the bound robot
client port number	The default value is -1, and the port number is automatically allocated after the connection is established
	Specified value, if the specified port is empty, the connection will be established in light of the specified port number
Hexadecimal delivery	The delivered string is processed as a hexadecimal character, and written to the memory for sending. For instance, the delivered "FF" string actually sends one byte, and the memory value is 0xFF, which is 255.
Hexadecimal reception	The received character string is processed in hexadecimal, for example, the received "FF" character string corresponds to a byte of memory, and the value is 0xFF, which is 255
status	Exhibit the communication status between client and server end

5.9.2 Server Settings



The socket can be used at the server side, firstly select "Tcp server" and set the parameters, after that, the "server setting" can be completed without re-start.

Once the association is established or the monitoring is established, the port parameters needs modification. After that, it is necessary to switch to "unused" to release server resources, and then select "Tcp server", the parameters will put into use. In the case of multi-Socket communication, special attention is needed.

Parameter	Description
Port serial number	The port number that the server monitors, the port number is suggested to be greater than 1024, otherwise it is easy to be at conflict with the system port number
Receive robot serial number	The socket is bound to the robot, and the reception can only be conducted by the bounded robot
Hexadecimal delivery	The delivered string is processed as a hexadecimal character, and written to the memory for sending. For instance, the sent "FF" string actually sends one byte, and the memory value is 0xFF, which is 255.
Hexadecimal	The received character string is processed in hexadecimal, for

reception	example, the received "FF" character string corresponds to a byte of memory, and the value is 0xFF, which is 255
status	Exhibit server-side communication status

5.10 Simulation Quantity Setting

The simulation quantity is an AD configuration function, a special demand function. If necessary, it can be completed with the technical support of the manufacturer.

The screenshot shows the 'AnalogIO' configuration window. It includes a table for AD channels (AD0 to AD7) with 'Value' and 'RealValue' columns. Below this is a table for 'Arr: K', 'A', 'data1', 'data2', 'realdatal', and 'realdatal'. To the right, there are settings for 'Method' (2DataGroup), 'Mode' (Auto), and a grid for 'Data1' to 'Data8' with 'Analog Data', 'Real Data', 'K', and 'A' columns. At the bottom are buttons for 'Load Data', 'Calibration', 'Save', 'Cancel', and 'Clear'.

5.11 External Axis Setting

When configuring external axes, the rules for external axis functions are: each robot can have two external axis groups, and each external axis group can have up to three external axes.

Refer to the "Independent External Axis Function Manual" for details.

5.11.1 basic Screen

Config Log KitharaMonitor IOConfig InputRegConfig TcpSocket AnalogIO **ExAxis** BASE Config LoadConfig Jog

Axis Group: ExAxisGroup1

Base Calibration

Valid: False Axis Number: 1 Axis1: EAxis1 Axis2: EAxis2 Axis3: EAxis3

Position Switch

Axis1 To Robot:
 X(mm): 0.0000 Y(mm): 0.0000 Z(mm): 0.0000 RX(°): 0.0000 RY(°): 0.0000 RZ(°): 0.0000

Axis2 To Axis1:
 X(mm): 0.0000 Y(mm): 0.0000 Z(mm): 0.0000 RX(°): 0.0000 RY(°): 0.0000 RZ(°): 0.0000

Axis3 To Axis2:
 X(mm): 0.0000 Y(mm): 0.0000 Z(mm): 0.0000 RX(°): 0.0000 RY(°): 0.0000 RZ(°): 0.0000

Flange To Axis3:
 X(mm): 0.0000 Y(mm): 0.0000 Z(mm): 0.0000 RX(°): 0.0000 RY(°): 0.0000 RZ(°): 0.0000

Point To Flange:
 X(mm): 0.0000 Y(mm): 0.0000 Z(mm): 0.0000 RX(°): 0.0000 RY(°): 0.0000 RZ(°): 0.0000

Zero Encoder

Parameter	Description
axis group	Configured axis group number
Validness	The current axis group configuration parameters are valid or not.
number of axes	The number of external axes configured by the current axis group should be consistent with the actual number.
Axis 1/Axis 2/Axis 3	The correspondence between the external axis and its ID, to keep the external axis number and ID number consistent
Position switch	The relative positional relationship between the external axis and the robot can be calibrated or input directly
zero point coding	Record the encoder value of each external axis of the current axis group as the zero point value.

5.11.2 Calibration Screen

The Calibration Screen is part of the 'ExAxis' configuration window. It features a 'Base' tab and a 'Calibration' sub-tab. The interface includes four sets of input fields for 'RobotPoint' and 'EAxisPoint' (1 through 4), each with X, Y, and Z coordinates in mm, and RX, RY, and RZ angles in degrees. To the right of each set is a 'Set Point' button. Below these are four horizontal sliders for 'EAxis1' through 'EAxis4', each with a numerical value field set to 0.000.

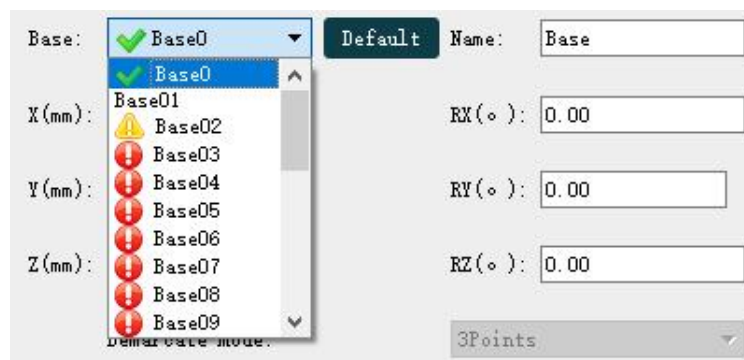
The calibration screen can be used to complete the position shift calculation between the external axis and the robot, and this screen can be used directly to conduct jog control on the external axis.

5.12 BASE Configuration

The Base configuration screen are mainly used for calibration of robot users coordinate system (also known as the work-piece coordinates system) and parameter setting.

The BASE Configuration screen is part of the 'BASE Config' window. It features a 'Base' dropdown menu set to 'Base0' and a 'Default' button. The 'Name' field is set to 'Base'. Below are input fields for X, Y, and Z coordinates in mm, and RX, RY, and RZ angles in degrees, all set to 0.00. A 'Demarcate Mode' dropdown is set to '3Points'. The screen includes three tips for setting the origin of the workpiece coordinate system: Tip1 (Origin), Tip2 (Point in the direction of the X-axis), and Tip3 (Point on the X-Y plane or X-Z plane). Each tip has corresponding input fields and a 'SetPoint' button. At the bottom are 'Save', 'Start', 'Clear', and 'Cancel' buttons.

1. Base0 (name: Base) is the base coordinate system of the robot, generally located at the center of the robot installation base which can't be modified and is the basis of other Bases.
2. Other options rather than Base0 in the BASE drop-down list, the Base corresponding to the red circular exclamation mark allows for a creations of new user coordinate system for the current robot, such as Base04; the Base corresponding to the yellow triangle exclamation mark is already occupied by other robots, such as Base02. It is not suggested to directly create a new user coordinate system based on existing one. The one without any symbols, such as Base01, it means that the current robot has created a user coordinate system.



3. Select Input from the calibration mode drop-down list. In the X/Y/Z column, fill in the offset between the origin of the user coordinate system and Base0 on the X-axis, Y-axis, and Z-axis. In the RX/RX/RZ column, fill in the Euler angle value between the user coordinate system pointing and Base0 on the X-axis, Y-axis, and Z-axis. The measurement unit is the angle unit, following the right-hand rule.

X (mm):	0.00	RX (°):	0.00
Y (mm):	0.00	RY (°):	0.00
Z (mm):	0.00	RZ (°):	0.00
Demarcate Mode:		Input 3Points 4Points Input	

4. Save (Save): Save the current settings

5. Default: Click this button, the current Base in the Base list is set as the default value, and all point coordinates are captured with the default Tcp relative to the default Base. Before selecting, you need to write the "name", then change the coordinate value, and click the save button, you can click the "default" button to set it as default.

Note: The interjection mark means it is not saved. The mark before Base04 will be removed after saving.

Base:	! Base04	Default	Name:	Base4
-------	---	--	-------	------------------------------------

Clicking the "Default" button, "√" will appear before Base04, and Base04 is set as the current default tool coordinate system.

Base:	√ Base04	Default	Name:	Base4
-------	---	--	-------	------------------------------------

6. Clear: Clear the Current Settings

Note: For Mantis models, when the gripper structure is complex, the 3-point /4-point method can also be used to calibrate the Base.

Select 3-point/4-point in the "Calibration Mode" drop-down list, and press "Start" to start manual calibration. In light of the calibration prompt, teach and load the

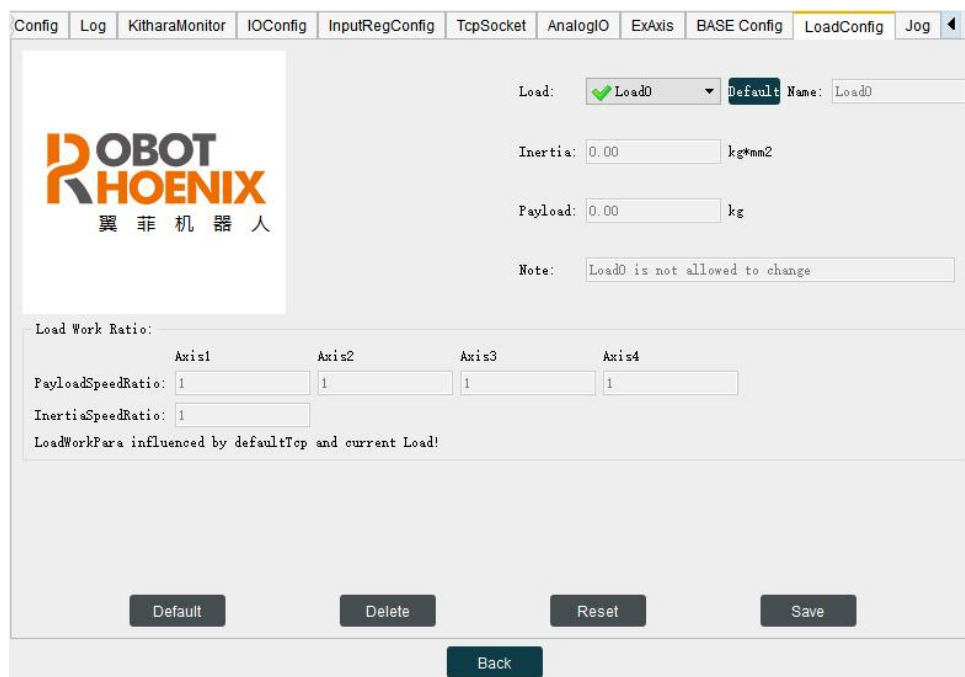
coordinate points. After the calibration is completed, please press "Save" to save the calibration results. If you want to abandon the calibration during the calibration process, press "End" to exit.



5.13 LOAD Configuration

LOAD is the load configuration, which configures the load parameters of the work-piece picked by the tool. Currently, only the "inertia" and "load" parameters are supported.

After the load setting is completed, in accordance with the load command, the robot will automatically adjust the speed in light of the load during the movement to avoid overload and over-velocity alarming.



Config Log KitharaMonitor IOConfig InputRegConfig TcpSocket AnalogIO ExAxis BASE Config LoadConfig Jog

Load: ☒ Load0 Name:

Inertia: kg*mm2

Payload: kg

Note:

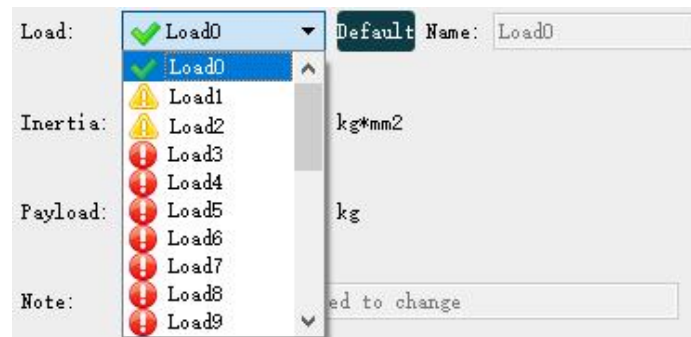
Load Work Ratio:

	Axis1	Axis2	Axis3	Axis4
PayloadSpeedRatio:	1	1	1	1
InertiaSpeedRatio:	1			

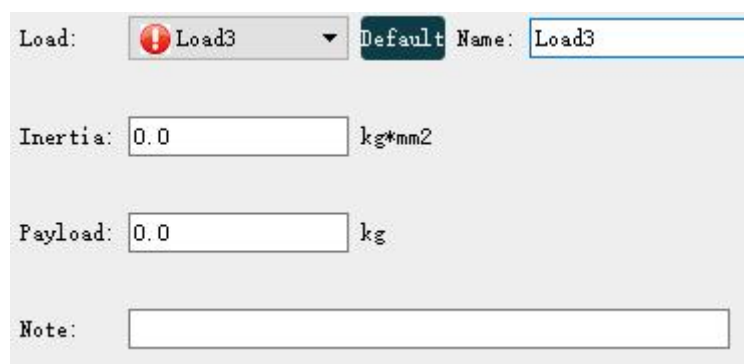
LoadWorkPara influenced by defaultTop and current Load!

Default Delete Reset Save Back

1. Load0 is the base load. The inertia and load parameters are both 0, and Load0 can't be modified.
2. Other options rather than Load0 in the LOAD drop-down list, the Load corresponding to the red circular exclamation mark allows for creating load parameters for the current robot, such as Load3; the Load corresponding to the yellow triangle exclamation mark is already occupied by other robots, such as Load1. It is not suggested to create a new LOAD directly on top of it.



3. Write the inertia value, load value and load description into the corresponding edit box, click the "Save" button, and the load parameters are created.



5. Default: Click this button, and the current Load in the LOAD list is set as the default. If the set load is not showed during motion, it will run in light of the default load.

Note: The exclamation mark is shown when it is not saved. The mark will disappear after saving.

Clicking the "Default" button, a "√" will appear before Load, and Load is set.



6. Delete: Clear the current settings

7. After the load setting is completed, the load value can be set through SetLoad when writing the command, and the subsequent movement will adjust the velocity in light of the load value set by the command.

5.14 Jog

5.14.1 Teach Pendant Touch Screen

Turn the key of the teaching pendant mode to the manual mode position, lightly press the enable switch to the middle gear, and after sounding the robot electric motor being powered on and the brake,

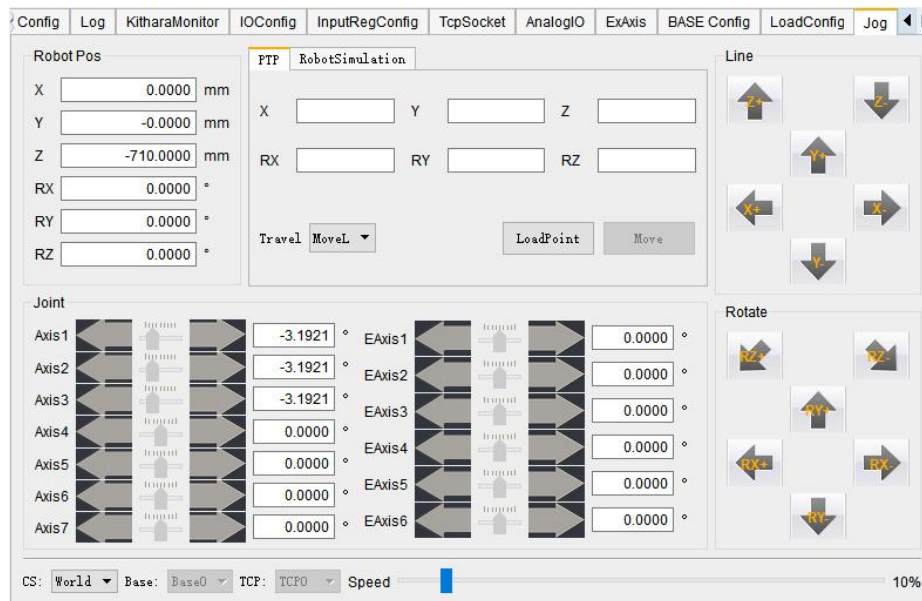
Click the arrow in the straight line area, the robot will move in a straight line in the direction indicated by the arrow, and release the arrow to stop it.

Click the arrow in the rotation area, the robot will rotate in the direction indicated by the arrow, and the robot will stop it when the arrow is released.

Click the arrows of axis 1-axis 6 in the joint area, and the corresponding single axis will rotate in the direction from the arrow.

If the system is equipped with external axes, click the arrows of EAxis1 — EAxis6 in the joint area, and the corresponding external axes will rotate in the direction

indicated by the arrows.



The TCP area shows the coordinates of the default TCP point of the current robot.

CS at the bottom left of the screen shows which coordinate the manual movement is moving along (World/Base/TCP), Base shows the current default Base coordinate name, and TCP shows the current name.

The speed drag bar at the bottom right of the screen can control the robot jog speed. When the robot jogs to the expected position, release the enable switch, and the electric motor will be powered off. Press "Back" to exit, and in automatic mode, this screen is invalid.

5.14.2 Instruction on Keys

The button on the left side of the teach pendant can control the robot for jogging.



It is the coordinate system selection button; press this button to switch CS among World, Base, Tool and Joint alternatively.

延X轴负方向移动/1轴反转	$\ominus \frac{1}{X}$	$\oplus$	延X轴正方向移动/1轴正转
延Y轴负方向移动/2轴反转	$\ominus \frac{2}{Y}$	$\oplus$	延Y轴正方向移动/2轴正转
延Z轴负方向移动/3轴反转	$\ominus \frac{3}{Z}$	$\oplus$	延Z轴正方向移动/3轴正转
延X轴反转/4轴反转	$\ominus \frac{4}{RX}$	$\oplus$	延X轴正转/4轴正转
延Y轴反转/5轴反转	$\ominus \frac{5}{RY}$	$\oplus$	延Y轴正转/5轴正转
延Z轴反转/6轴反转	$\ominus \frac{6}{RZ}$	$\oplus$	延Z轴正转/6轴正转
7轴反转	$\ominus 7$	$\oplus$	7轴正转

5.14.3 Robot PTP Function

Robot PTP function, click the "LoadPoint" button to load the point information of the robot's current position or manually input point information in X, Y, Z, RX, RY, RZ, press the enable button, and hold onto "Move" button, the robot can run to the set position. Currently, the PTP function has two postures, MoveL and MoveJ.

PTP

RobotSimulation

X

Y

Z

RX

RY

RZ

Travel

MoveL

LoadPoint

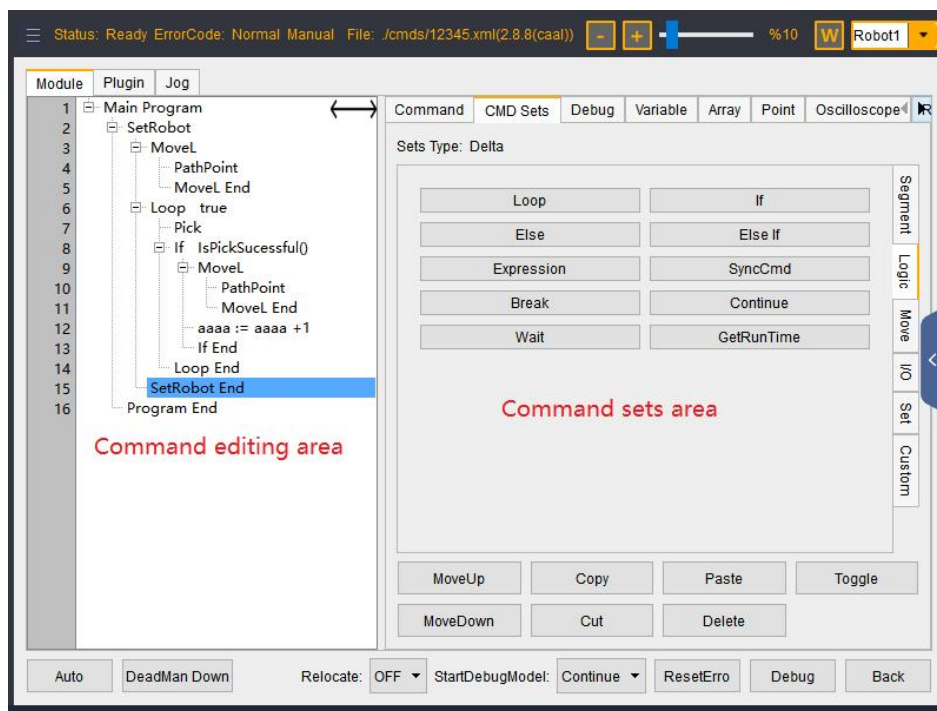
Move

Chapter 6 Robot Programming

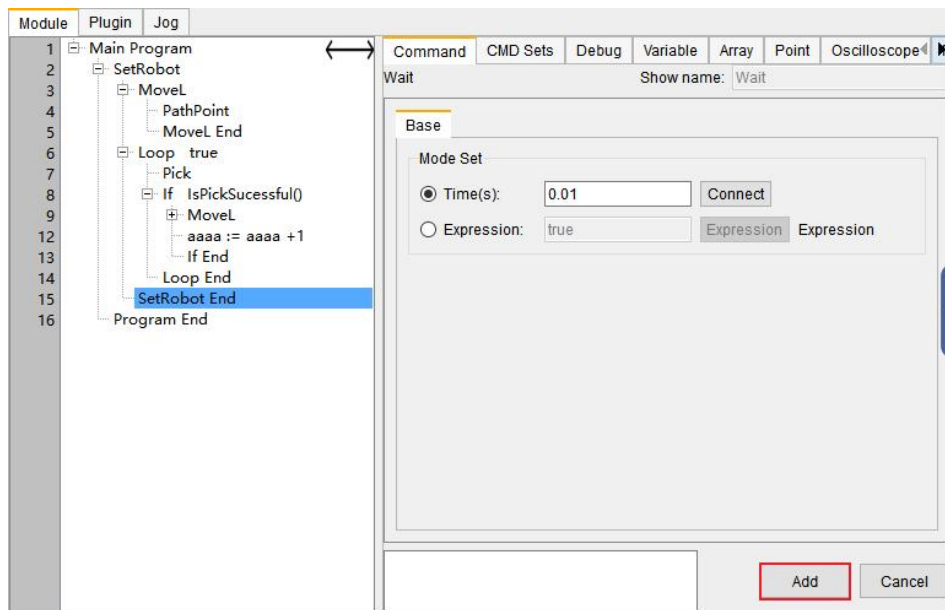
6.1 Modules - Commands

The command set area contains all program commands.

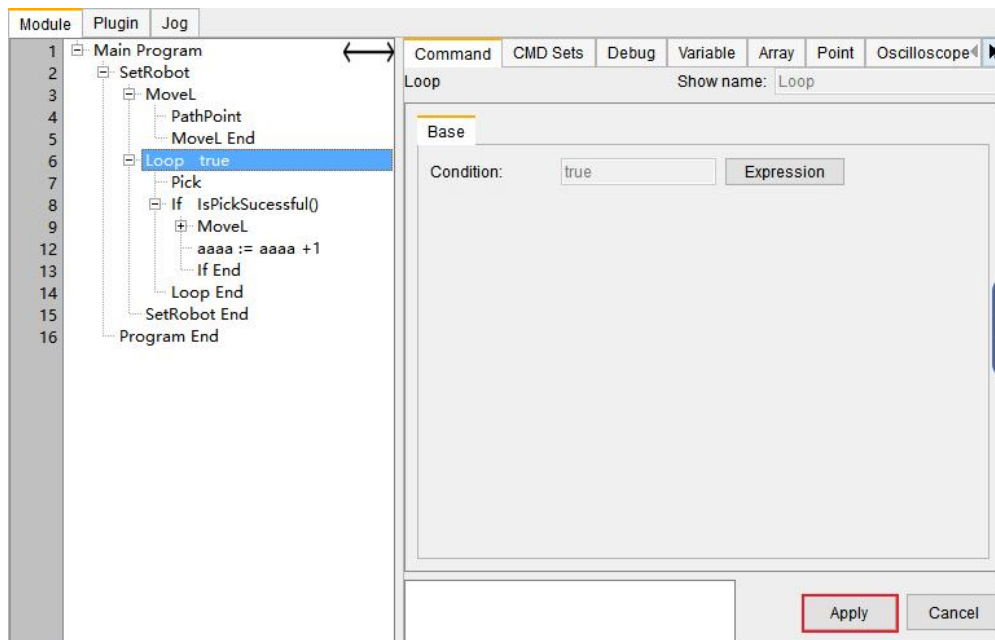
(1) Click to select an existing command in the editing area, and press "Move Up/Move Down/Copy/Cut/Paste/Delete/Comment" on the right to exercise corresponding editing operations.



(2) Click the command you want to add from the command set, set the parameters, and press the "Add" command to add it to the program on the left.



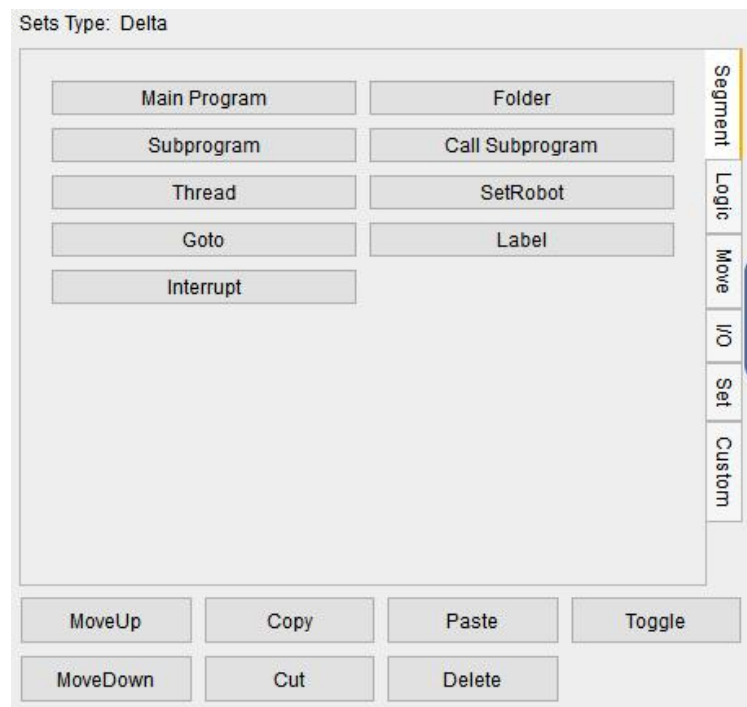
(3) Click the command whose parameters you want to modify in the program, and the parameters of this command will be displayed in the command table on the right. After modifying the parameters, press "Apply" to make the changed value put into use.



6.2 Module—Command Set

The command set includes all the commands of the software, and the command

set is divided into 6 parts, such as code segment logic, motion, input and output, setting, and customization.



6.3 Modules - Debugging

Debug (Debug) screen is mainly used for monitoring and control, which is divided into seven parts: operating variable, operating point, input/output, simulation quantity, object, status, and compilation.

6.3.1 Running Variable Screen

The local variables, global variables, and Modbus variables in the running variable screen are all created based on the "variable screen". The running variable screen will show the current status of the variables in real time.

RunVariable	RunPoint	I/O	AnalogIO	Object	State	Compile
Name		Value				
▼ Local						
Bool						
▼ Int						
total		80				
Double						
String						
▼ Global						
▼ Bool						
a		false				
b		false				
bb		false				
CC		false				
c		false				
Pt		false				
▼ Int						
idx		0				
abcd_1		0				

Varname:	idx
Type:	int
MinValue:	
MaxValue:	
InitValue:	
S/C:	
S/CID:	
I/O:	
Reg:	
Bit:	
RegNumber:	
CurValue:	0
ForceValue:	
Force	

In the running variable table, all categories are showed in the list. When for running, click the variable name, and the current value will be showed on the right. If you want to change the "variable value", you can input a new value in the mandatory value edit box, and then press the "mandatory" button. For example, in the figure below, in light of checking, the current value of the integer variable total is 80. Enter 0 in the mandatory value, and then press "mandatory" to force the value in total to 0.

RunVariable	RunPoint	I/O	AnalogIO	Object	State	Compile
Name		Value				
▼ Local						
Bool						
▼ Int						
total		80				
Double						
String						
▼ Global						
▼ Bool						
a		false				
b		false				
bb		false				
CC		false				
c		false				
Pt		false				
▼ Int						
idx		0				
abcd_1		0				

Varname:	total
Type:	int
MinValue:	
MaxValue:	
InitValue:	80
S/C:	
S/CID:	
I/O:	
Reg:	
Bit:	
RegNumber:	
CurValue:	80
ForceValue:	0
Force	

6.3.2 Operation Point Screen

View the current value of a point variable. Point variables can also be modified online without stopping the running program. Click the point variable expected to modify, select the component in Parameter, input a new value in ForceValue, press the "mandatory" button, and the new value will put into use immediately. Modifying point variables online will change the path position and track of the robot without a trial run, so be cautious.

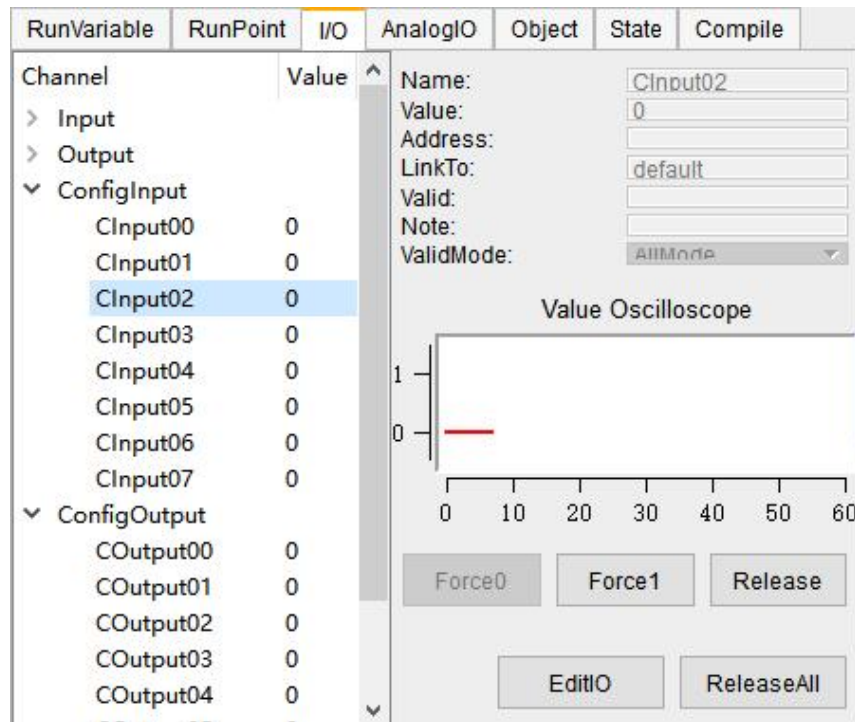
The screenshot shows the 'RunPoint' tab selected in the software interface. The main area displays a table of point variables. The table has columns for Name, ID, X, Y, Z, RX, RY, and RZ. A point named 'pt1' is selected, showing its coordinates. Below the table, there is a 'Parameter' dropdown menu set to 'X', a 'ForceValue' input field, and a 'Force' button.

Name	ID	X	Y	Z	RX	RY	RZ
pt1	0	1.00	2.00	-753.00	0.00	0.00	10.00

Parameter: X
ForceValue:
Force

6.3.3 Input and Output Screen

Open the Input/Output table to view or add the current value of the I/O channel. Press "Cancel mandatory" to cancel the mandate on the current one. Press "Cancel All Mandate" to remove mandate of all channels except ConfigI/O. Configure the channels for input and output services.



Oscilloscope: Exhibit the real-time value fluctuation of the selected IO channel, the resumption period is 100ms, and the IO value changes at small intervals may not be captured.

Mandate: For the selected IO channel, the "mandate 0" or "mandate 1" button can force the status of this I/O port to change to 0 or 1 state value.

Cancel mandate: Cancel the mandate for the selected channel.

Cancel mandate of all: Release all channels from mandate.

Description: Comments for the IO channel.

Valid mode: The channel allows only for being mandatory in the pre-set mode, and currently only the "output" and "configuration output" channels are allowed. For example, if the mode is set to "Auto", the channel will be forced to be invalid, which is a kind of safety design.

Edit IO/Save IO: realize the addition of IO description and configuration saving of effective mode.

6.3.4 Simulation Screen

In general, there is no need for this screen function.

RunVariable	RunPoint	I/O	AnalogIO	Object	State	Compile
Channel	Value	RealValue		Name:		
AD(Input)				ADValue:		
AD0	0	0.00		RealValue:		
AD1	0	0.00		Address:		
AD2	0	0.00		LinkTo:		
AD3	0	0.00		Forced:		
AD4	0	0.00				
AD5	0	0.00				
AD6	0	0.00				
AD7	0	0.00				
DA(Output)						
DA0	0	0.00				
DA1	0	0.00				
DA2	0	0.00				
DA3	0	0.00				
DA4	0	0.00				
DA5	0	0.00				
DA6	0	0.00				
DA7	0	0.00				

☒ Value ☐ RealValue ForceValue:

6.3.5 Object Screen

The object table is used to add virtual objects, which can be used for Pick command simulation, and there are 3 sub-tables on the right (simulation 1, simulation 2, simulation 3). Each table has the followings that can be set:

The screenshot shows a software window with a tabbed interface. The 'Object' tab is selected, showing configuration for 'Simulation1'. The fields are as follows:

Field	Value
RobotID:	Robot1
ConveyorID:	Conveyor0
Size:	10000
X(mm):	0
Y(mm):	1000
Z(mm):	-800
RX(*):	0
RY(*):	0
RZ(*):	0
Length(mm):	10
Width(mm):	10
Height(mm):	10
Color(mm):	0
Time(s):	0

At the bottom of the form are two buttons: 'Add' and 'Stop'.

Robot ID: Add the simulation to the robot

Conveyor Belt ID: Add the virtual object on conveyor belt

Size: Virtual object area

X/Y/Z: The position where the virtual object appears

RX/RY/RZ: The angle at which the virtual object appears

Length/width/height: The length, width and height of the virtual object

Color: Color code, For example, 1 is red, 2 is yellow, 3 is green, etc. The specific color that each code represents in light of the specific project.

Time (s): The time to automatically add virtual objects. For example, when times is 1, after pressing the "Add" button, a virtual object will be automatically added to the conveyor belt every 1 second, and pressing the "Stop" button will stop adding the virtual object. When the time(s) is 0, click the "Add" button to add one virtual object to the conveyor belt per one click, no object will be added without clicking.

6.3.6 Status Screen

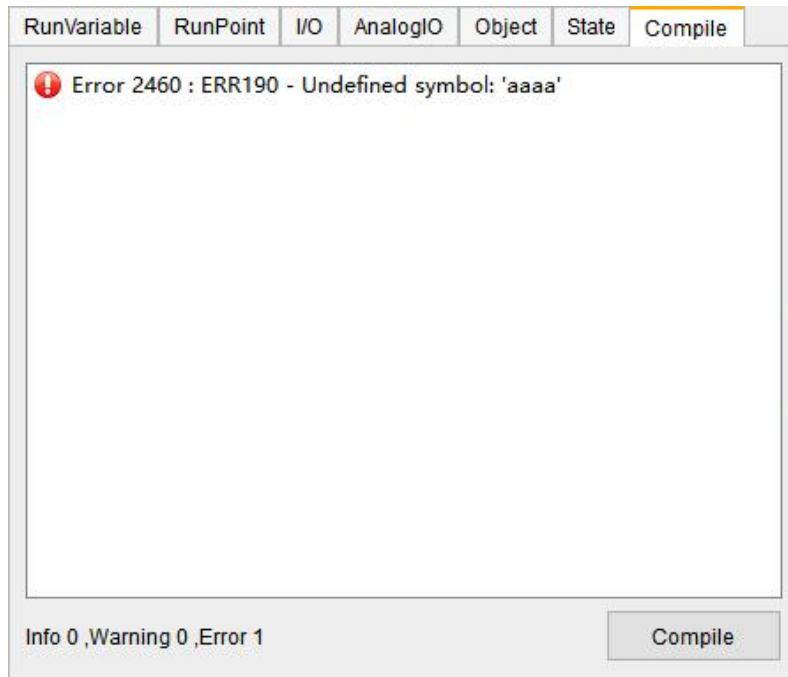
On the Status table you can see:

- (1) The moving distance and speed of conveyor belt 1-conveyor belt 4 (in mm/s);
- (2) The current position of the robot: including the body X/Y/Z/RX/RY/RZ, and the angles of up to 6 external independent axes 1 to 6;
- (3) The current position X/Y/Z/RX/RY/RZ of external axis group 1 and external axis group 2;
- (4) Modbus communication status

RunVariable	RunPoint	I/O	AnalogIO	Object	State	Compile
Conveyor						
Conveyor0Distance:	0.00			Vel:	0.00	
Conveyor1Distance:	0.00			Vel:	0.00	
Conveyor2Distance:	0.00			Vel:	0.00	
Conveyor3Distance:	0.00			Vel:	0.00	
Robot Pos						
X	Y	Z	RX	RY	RZ	
0.00	-0.00	-710.00	0.00	0.00	0.00	
EA1	EA2	EA3	EA4	EA5	EA6	
0.00	0.00	0.00	0.00	0.00	0.00	
ExAxisGroup Pos						
	X	Y	Z	RX	RY	RZ
Group1:	0.00	0.00	0.00	0.00	0.00	0.00
Group2:	0.00	0.00	0.00	0.00	0.00	0.00
Modbus						
Slave: Disable						
Master1: Disable			Master2: Disable			
Master3: Disable			Master4: Disable			

6.3.7 Compile Screen

When compiling the user program, click the "compile button" on this screen to compile and check the issue.



Double-click the error line to locate the problem command string.

6.4 Modules - Variables

Variables table can add or delete variables. Variables are divided into local variables (Local), global variables (Global), and Modbus communication variables.

Each variable has int type, bool type, double type and String type.

Note: This module only conducts variable management, that is, adding, deleting, and editing. During the operation of the command program, the actual value of the variable can't be updated and displayed in real time on this screen.

6.4.1 Local Variables

Local variables are only allowed to be used by the current program. After the program ends, it will be opened again, and its value will start to run from the

initial value.

6.4.2 Global Variables

Global variables can be used by any user programs, and when the program is closed, the software is closed, and the power is turned on again, its value remain the same as the value before the program ends or the software is closed.

6.4.3 Modbus Variables

Modbus communication variable, used for Modbus communication, has the same properties as the global variable.

Local	Global	Modbus				
Index	Name	Type	Value	MinValue		
0	a	bool	false			
1	b	bool	false			
2	bb	bool	false			
3	CC	bool	false			
4	c	bool	false			
5	Pt	bool	false			
6	idx	int	0			
7	abcd_1	int	0			
8	L001_X	double	400.0000			
9	L001_Y	double	340.0000			
10	L001_Z	double	1200.0000			

6.4.4 Add Variables

Name: variable name (only alpha characters, numbers and underscores can be used in the name and the first character can't be a number)

Type: int/bool/double/string

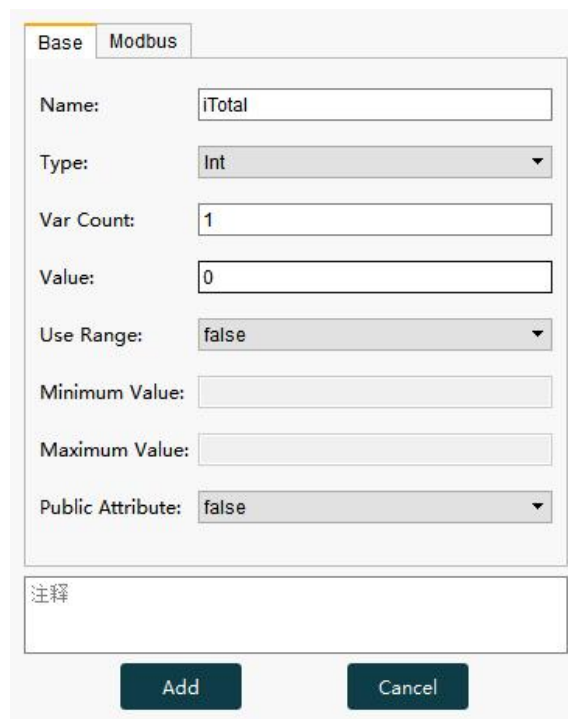
Number of variables: the number of variables to add, multiple variables can be added at one time

Value: allocate initial value to the variables

Use range limit: If true, the following maximum and minimum input values are valid, and no matter how the variable value increases or decreases, it will never exceed the maximum and minimum values. If false, the maximum and minimum values are invalid, and the input area is gray and can't be modified.

Add to the parameter list: Select true, it can be seen in the running program screen, that is, users who do not have authority can also view this variable. If false, this variable can only be viewed in the robot programming screen.

Note: Variable related commands



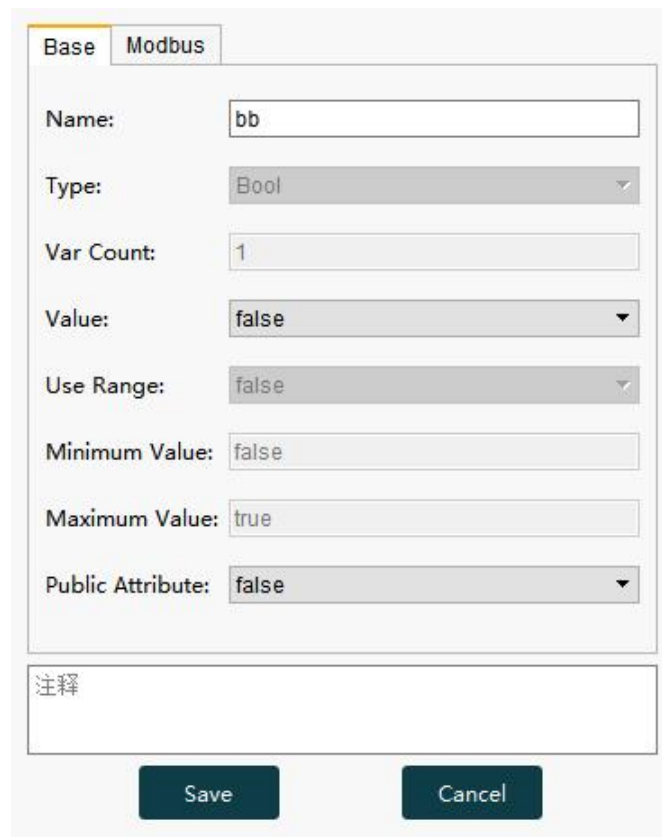
The image shows a variable modification dialog box with two tabs: 'Base' and 'Modbus'. The 'Base' tab is selected. The dialog contains the following fields and controls:

- Name:** A text input field containing 'iTotal'.
- Type:** A dropdown menu set to 'Int'.
- Var Count:** A text input field containing '1'.
- Value:** A text input field containing '0'.
- Use Range:** A dropdown menu set to 'false'.
- Minimum Value:** A disabled (grayed out) text input field.
- Maximum Value:** A disabled (grayed out) text input field.
- Public Attribute:** A dropdown menu set to 'false'.
- 注释 (Comment):** A large text area at the bottom for additional notes.
- Buttons:** 'Add' and 'Cancel' buttons at the bottom right.

6.4.5 Variable Editing

Double-click the variable in list, and the variable modification screen will pop-up

in the figure below. After completing, click "Save" to finish the modification, and click "Cancel" to abandon the modification.



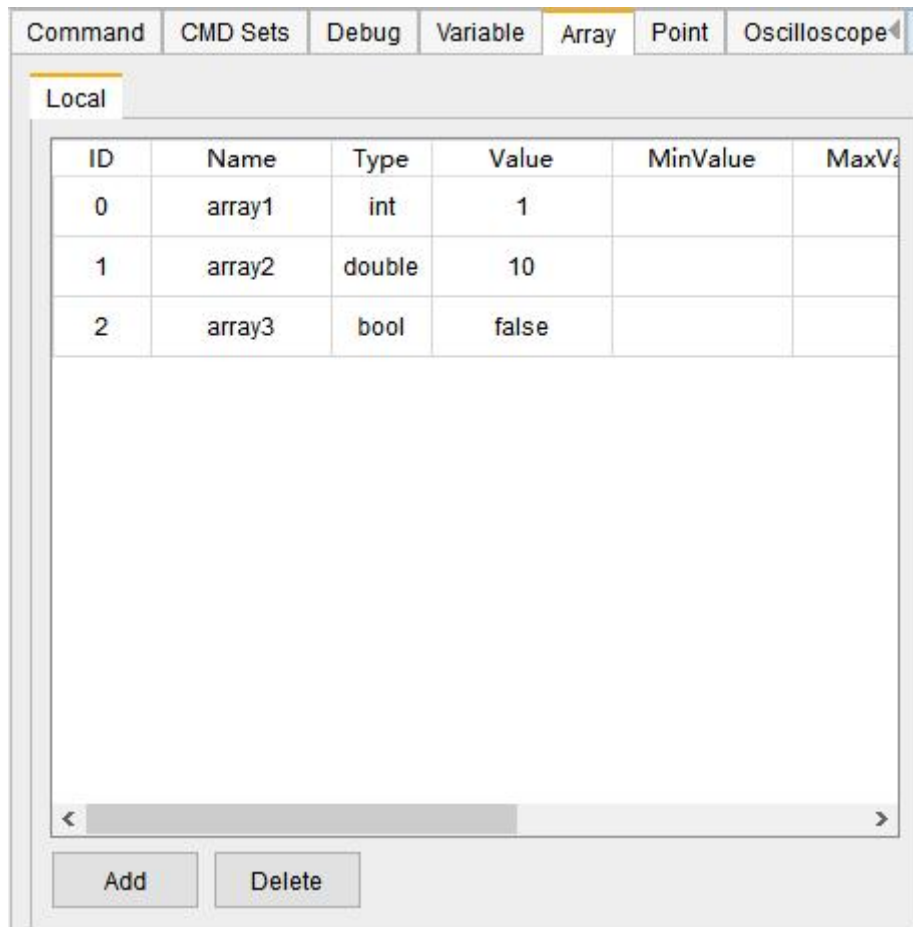
The image shows a configuration window with two tabs: "Base" and "Modbus". The "Modbus" tab is selected. The window contains the following fields and controls:

- Name:** A text input field containing "bb".
- Type:** A dropdown menu with "Bool" selected.
- Var Count:** A text input field containing "1".
- Value:** A dropdown menu with "false" selected.
- Use Range:** A dropdown menu with "false" selected.
- Minimum Value:** A text input field containing "false".
- Maximum Value:** A text input field containing "true".
- Public Attribute:** A dropdown menu with "false" selected.
- 注释 (Comment):** A large text area at the bottom, currently empty.
- Buttons:** "Save" and "Cancel" buttons at the bottom right.

6.5 Modules—Arrays

The array table can be used to add or delete array variables which can be accessed or decided by index. Currently only local variables (Local) are used. The format supports int type, bool type, and double type.

Note: This module only exercises array management, including adding, deleting, and editing. During the operation, the actual value of the variable can't be updated and displayed in real time.



6.5.1 Array Variable Addition

Name: variable name (only alpha characters, numbers and underscores can be used in the name and the first character can't be a number)

Type: int/bool/double

Quantity: the number of elements in the array variable

Value: the initial value

Use range limit: If true, the following maximum and minimum input values are valid, and however the variable value increases or decreases, it will never exceed the maximum and minimum. If false, the maximum and minimum values are invalid, and the input area is gray and can't be modified

Add to the parameter list: Select true, it can be seen in the running program screen, that is, users who do not have authority can also view this variable. If false, this variable can only be viewed in the robot programming screen, and this parameter has no functional support.

Note: Variable related commands

The image shows a 'Base' configuration dialog box with the following fields and values:

- Name: array
- Type: Bool
- Size: 1
- Value: false
- Use Range: false
- Minimum Value: (empty)
- Maximum Value: (empty)
- Public Attribute: false

Below the fields is a text area labeled '注释' (Remarks) and two buttons at the bottom: 'Add' and 'Cancel'.

6.5.2 Array Variable Editing

Double-click the variable in the array variable list, and the array modification screen will pop-up as shown in the figure below. After complete, click "Save" to finish the modification, and click "Cancel" to abandon the modification.

The screenshot shows a configuration window titled 'Base' for an array variable. It contains the following fields and controls:

- Name:** array2
- Type:** Double (dropdown menu)
- Size:** 120
- Value:** 10
- Use Range:** false (dropdown menu)
- Minimum Value:** 0
- Maximum Value:** 0
- Public Attribute:** false (dropdown menu)
- 注释 (Comment):** A text area for additional notes.
- Buttons:** 'Save' and 'Cancel' buttons at the bottom.

6.5.3 Array Variable Expression Operation

Array variables can access and determine elements values through the element index. The element index starts at 0 with the maximum being the number of elements minus one.

The element index can be an integer. During the running, if the value of the variable is over the range of the index, the command program will stop, and in the "Real-time Log" screen there will be a log alarm that indicates index is over limitation.

Example 1: Element access in an array variable, set a double type variable dValue, array variable array1, run `dValue := array1[0]`, and this assigns the value of the first element of the array variable array1 to the dValue variable.

Example 2: Assignment of elements in the array variable, set a double type variable dValue, array variable array1, run `array1[0]:= dValue`, then it assigns the dValue variable value to the first element of the array variable array1.

6.6 Modules - Points

The Points table allows adding or removing point variables. The properties of the point variable are the same as the local variables, only the current program can be used, and the program will be re-opened after ending, whose value will run with the initial value.

Note: This module only conducts point variable management, including adding, deleting, and editing. During the operation of the command program, the actual value of the point variable can't be updated and showed in real time.

Command	CMD Sets	Debug	Variable	Array	Point	Oscilloscope		
ID	Name	X	Y	Z	RX	RY	RZ	EA1
0	pt1	550.00	0.00	0.00	0.00	0.00	0.00	0.000

<

>

Add

Delete

AddToPTP

6.6.1 Point Variable Addition

Name:	pt
RobotType:	Scara
SubType:	Sub_Base
PointCount:	1
X:	550
Y:	0.00
Z:	0.00
RX:	0.00
RY:	0.00
RZ:	0.00
E1:	0.000
E2:	0.000
E3:	0.000
E4:	0.000
E5:	0.000
E6:	1 000
Toggle:	
Load Add Cancel	

Name: Variable name (only alpha characters, numbers and underscores can be used in the name and the first character can't be a number)

Robot type: Set in light of the actual situation

Number of points: Number of added points, multiple points can be added at a time

Note: Notes on point variables

Load: Load the point coordinates of the current robot (default Tcp relative to default Base)

Add: Add the variable to list

Cancel: Abandon the current operation and return to the point variable list

6.6.2 Point Variable Editing

Double-click the point in the point variable list, and the point modification screen will pop-up as shown in the figure below. After complete, click "Save" to finish the modification, and click "Cancel" to abandon the modification.

Name:	pt2
RobotType:	Scara
SubType:	Sub_Base
PointCount:	1
X:	550
Y:	0
Z:	-100
RX:	0
RY:	0
RZ:	10
E1:	0.000
E2:	0.000
E3:	0.000
E4:	0.000
E5:	0.000
E6:	1.000
Toggle:	

Load Save Cancel

6.6.3 Point Variable Expression Operation

Each element in the point variable can be accessed or decided by an expression, and the corresponding relationship is as follows:

X, Y, Z, RX, RY, RZ values, corresponding to the 0-5 index of the point variable;

Joint angles of J1, J2, J3, J4, J5, J6, corresponding to 6-11 index;

E1, E2, E3, E4, E5, E6, the external axis angle of the point, corresponding to the 12-

17 index;

Example 1: The x value of point p1 can be expressed as p1[0], the y value of point p1 can be expressed as p1[1], and the z value of point p1 can be expressed as p1[2]. By simulation, the angle value of joint 1 at point p1 is p1[6], and the angle value of external axis 1 can be expressed as p1[12].

Example 2: Element access in the point variable, set a double variable X_p1, run X_p1:=p1[0], and assign the x coordinate value of p1 to the X_p1 variable.

Example 3: Assignment of elements in the point variable, set a double variable X_p1, run p1[0]:=X_p1, and set the x coordinate value of p1 to the value of the X_p1 variable.

6.6.4 Add to PTP

Add the selected point coordinates to the PTP screen of the jog, and the PTP screen can be used to finish the PTP movement.

6.7 Module—Run Log

In the running log screen, the in-job information of the robot will be exhibited in real time, such as normal, alarm, error, time, etc. When abnormality appears, check the log information at first time to see if the cause of the problem can be located.

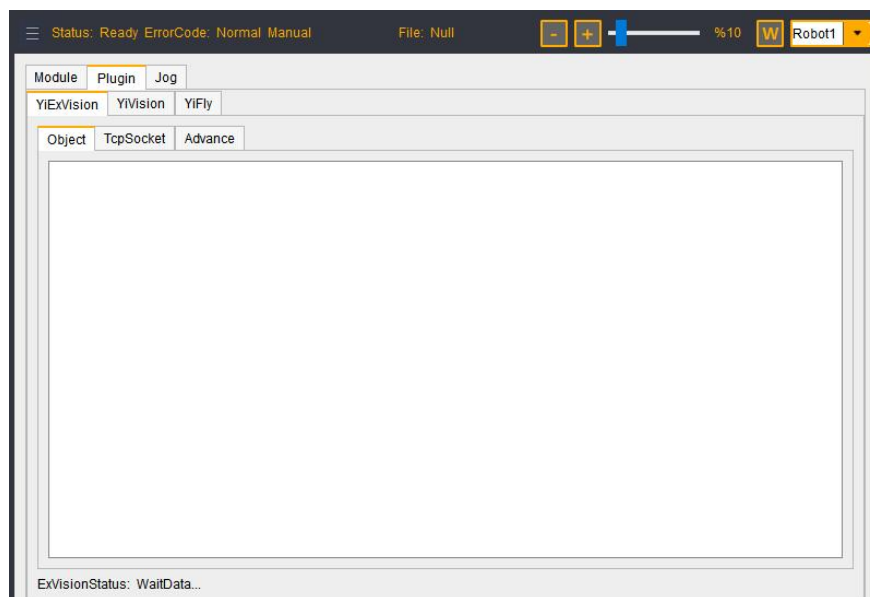
CMD Sets	Debug	Variable	Array	Point	Oscilloscope	RunLog
ID	Robot	Code	Msg			
0	4	7029	Module:0 Normal Channel0			

6.8 Plug-in—YiExVision

Collaboration of external vision plug-in and YiKingSoft, and refer to the "YiKingSoft Vision Application Manual" for details.

6.8.1 Object Screen

The Object screen shows information such as the serial number, coordinate value, length, width, height, size, and color of the visually recognized object.



6.8.2 TcpSocket Screen

The external vision plug-in communicates with the vision platform software through TcpSocket protocol.

A robot can be used as a client or a server. Select TcpServer as a server. Select TcpClient as the client, and fill in the IP address of the other party's server. Modify the PORT (port) to make the port numbers of both communication parties consistent (the port of the server and the client should be consistent). Press the "save" button to save, and re-start the software to make the settings put into use.

Status shows the status of socket communication.

ExVisionStatus displays the external vision status.

Check whether to display the data received to exhibit the received data.

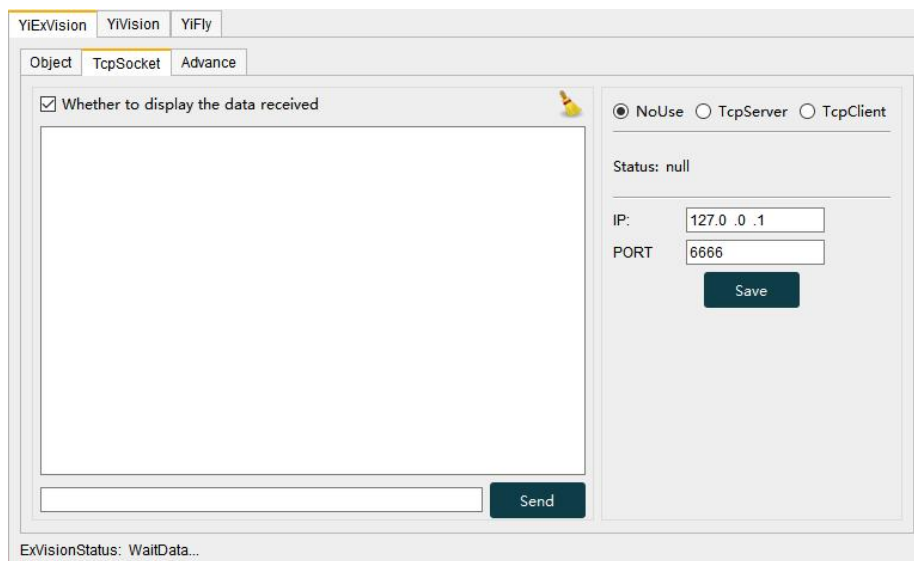
Press  the button to clear the display.

After entering the content in the Send input field, press the "Send" button to send the data.

Note: The object data transmitted externally must be of string type. And must use the following protocol format

Data;frameNum;triggerTime;objNumber;|id,x,y,z,angle,angley,anglez,lenth,width,height,size,color,msg

Data in other formats will be received, but the robot will not perceive them as objects. For protocol details, refer to Appendix: External Visual communication screen Description



6.8.3 Advanced Screen

This is used to modify parameters for triggering and using camera.

(1) Vision ID: the ID number of the external vision plug-in, in the above picture, there is only one YiExVision plug-in and ID number is 0; if being another YiExVision plug-in, ID number is 1, this parameter can't be changed.

(2) Robot serial number: to decide which robot is used to trigger the camera and assigns the object acquired by YiExVision to which robot.

(3) Trigger mode: only valid when the acquisition mode is set as trigger. 0—auto trigger, the robot automatically sends a signal to make the camera take a picture. 1—Triggered by command, user programming is required to send a signal through the command to make the camera take pictures.

(4) Trigger delay: After receiving the visual information returned by the previous triggering, the delayed milliseconds for the next trigger.

(5) Maximum interval time: the maximum number of milliseconds to wait for the next trigger. If the visual information returned by the previous trigger hasn't been

received, the next trigger will be triggered after reaching this parameter.

(6) Auto start: Only valid when the acquisition mode is set to 0—trigger, and the trigger mode is set to 0—auto trigger. After checking, the robot will automatically send a signal to the camera to take pictures, and after receiving, it will add the object to the object list, which can be viewed in the object screen.

(7) Using height: Use the object height information (coordinate Z value) transmitted from the outside. Only valid when PixelCoordinates is selected for the coordinate system.

(8) Acquisition mode: 0 — trigger, the robot sends a trigger signal to make the camera take pictures. 1 — Continuous, the robot does not signal. Note: Different from the internal vision, when the external vision collection mode is consistent, only the trigger signal is not sent, the camera can't automatically shift to the selfie mode, and additional camera settings are required.

(9) Exposure time: Fill in the real camera exposure time, same as the exposure time set in the camera. Pay attention that the exposure time set is only for internal calculation and can't change the real exposure time setting value of the camera, different from internal vision.

(10) Trigger channel: By which channel sending the signal to make the camera take pictures. Depending on the hardware wiring, only the configuration output interface can be selected.

(11) Conveyor belt channel: By which conveyor belt the camera is installed on, and with which conveyor belt the object is transported to the robot workspace

after picture taking.

(12) Working mode: LOOP+Time——Loop + time synchronization, which is applied in the case of externally provided image time stamps. LOOP——Loop only, used when image timestamps are not provided externally. It is suggested to use timestamps whenever possible.

(13) Coordinate system: The external transmission by PixelCoordinates is the image coordinate value. RobotCoordinates externally transmits robot coordinate values. Note that if the image coordinates are transmitted externally, then YiExVision calibration is required. If the coordinate value of the robot is transmitted from the outside, it is required that the external value should be calibrated ready, and the image coordinates have been converted into robot coordinates before the data is transmitted.

(14) Trigger control: When triggering automatically, whether stopping for the Tcp communication to be established before triggering the camera.

The screenshot shows the 'YiExVision' software interface with the 'Advance' tab selected. The settings are as follows:

Parameter	Value	Range/Options
VisionID	0	
RobotNo	1	RobotNum 1~4
Trigger Mode	0	0- Auto trigger 1- Command trigger
DelayTriggerTime(ms)	100	
MaxIntervalTime(ms)	500	
Autostart	☐	
UseHeight	☐	
Collection Mode	0	0-Trigger, 1-Continuous
Exposure Time(us)	1000	34~1000000
Trigger Channel	0	0~7
Trigger Edge	RisingEdge	
Conveyor Channel	0	0~3
WorkModel	Loop+Time	
Coordinates	PixelCoordinates	
Trigger Control	0	0:No Wait Socket Connect State 1:Trigger Wait Socket Connect Establish

A 'Save' button is located at the bottom right of the configuration area.

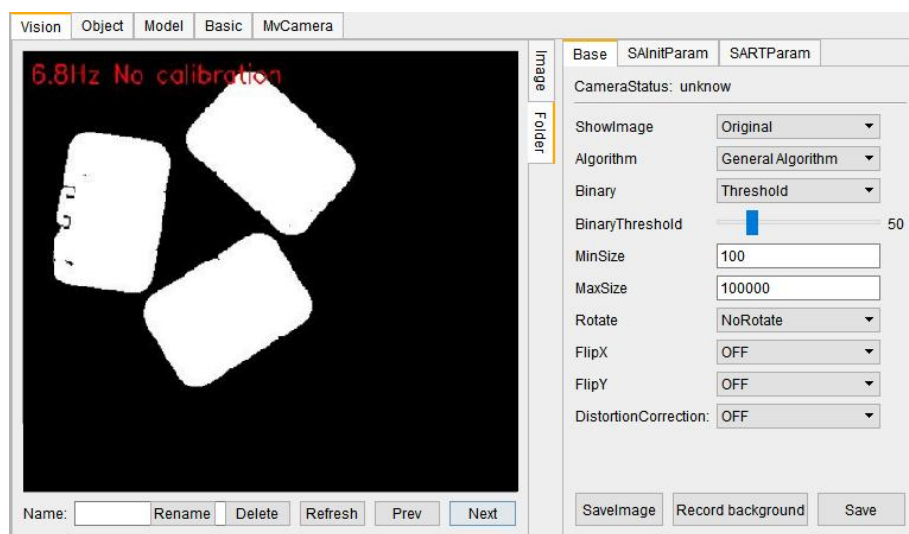
6.9 plug-ins—YiVision

The internal visual plug-in can be invoked and debugged through YiKingSoft software.

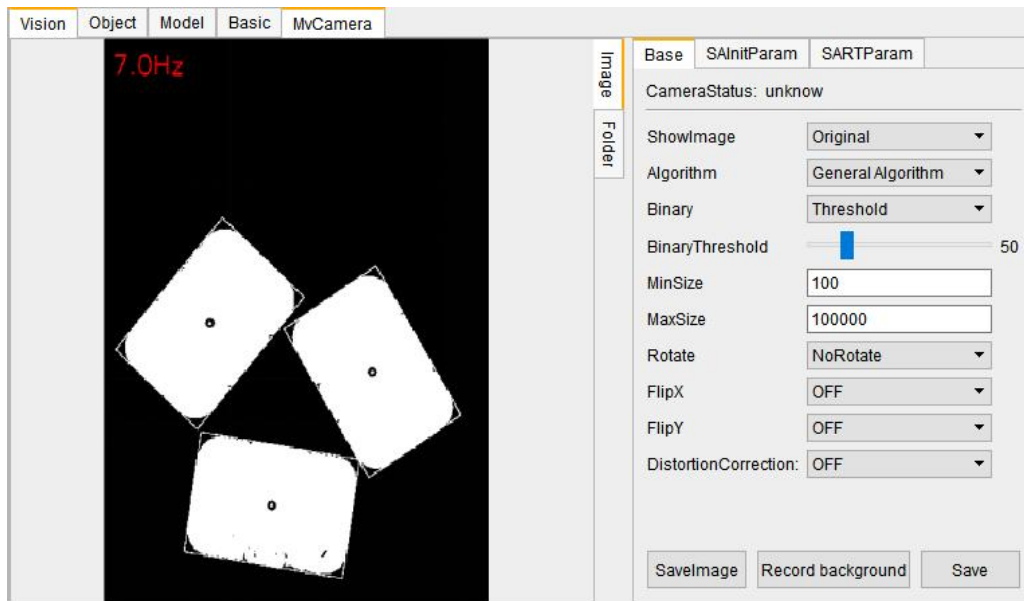
6.9.1 Visual Screen

Visual table → Image table

Show the current camera viewing field and the objects recognition status within the field. As shown above, the red flashes: No calibration above the viewing field, indicating that the camera hasn't been calibrated. Under the normal algorithm, there is no object outline and no center position, indicating that the object hasn't been recognized.

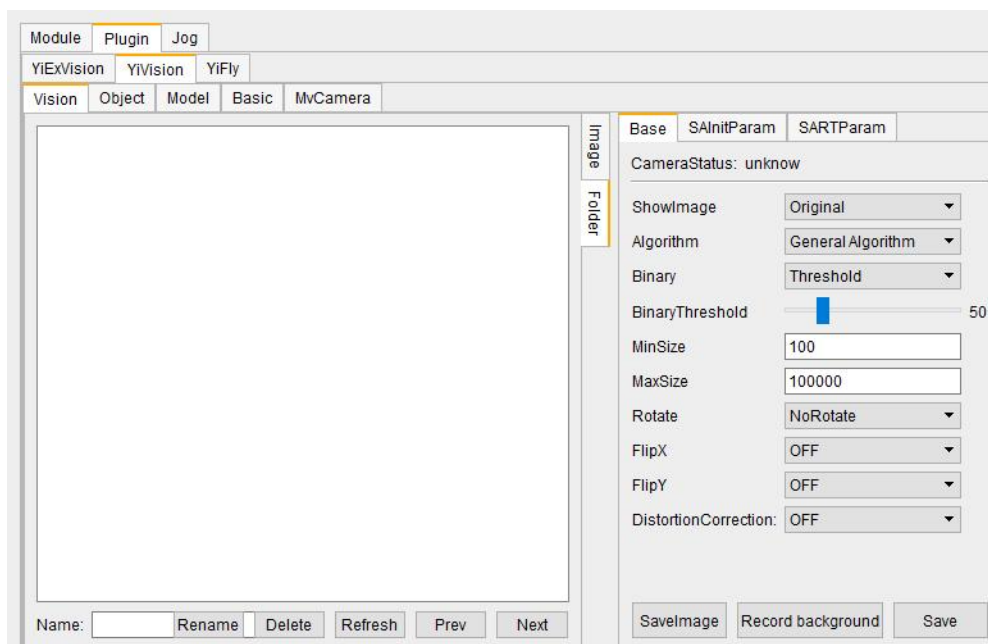


As shown in the figure below, if the camera has been calibrated, only the camera trigger frequency is exhibited above the field. Under the normal algorithm, the outline and center position of the object are displayed, showing that the object has been recognized.



Folder table

Show photos kept by the user, using for making templates.



(1) Base table on the right → picture display:

1. The original picture (with light and dark changes, color changes, as shown in the left picture below)
2. Processing diagram (binary processing, either black or white, as shown on the right)

Generally, the original image is used when the camera hardware is newly connected or when the focus and aperture are adjusted; and other applications can be based on actual needs.

Base	SAInitParam	SARTParam
CameraStatus: unknow		
ShowImage	Original	
Algorithm	General Algorithm	
Binary	Threshold	
BinaryThreshold	<input type="range"/> 50	
MinSize	100	
MaxSize	100000	
Rotate	NoRotate	
FlipX	OFF	
FlipY	OFF	
DistortionCorrection	OFF	



(2) Base table on the right → Algorithm:

1. Ordinary algorithm: Only realize the outline and center of the object
2. Special algorithm: In addition to outline and center, there are recognition of other features.

(3) Base table on the right → binarization:

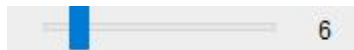
1. Binarization: binarize the image
2. Data table: Use the background data, and the highlights collected in the

data table are all processed as the backdrop.

Example: When there are some bright spots on the black belt, the system will consider these spots as objects without using the background data table. After using the background data table, these spots are also considered as the background. Only the bright areas with different spots are considered as objects by the system.

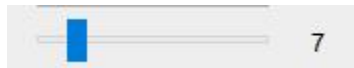
(4) Basic table on the right → Binarization Threshold/Background Binarization Threshold:

1. Binarization threshold



It is showed when binarization is selected above.

2. Background binarization threshold



Background binarization threshold adjustment bar, displayed when the data table is selected.

(5) Basic table on the right → minimum/maximum size:

1. Minimum size: Objects smaller than this size can't be recognized

2. Maximum size: Objects larger than this size can't be recognized

(6) Basic table on the right → Rotation:

1. No turning

2. Turn clockwise 90°

3. Turn 90°counterclockwise

(7) Basic table on the right → X-axis/Y-axis flip:

1. X-axis flip: flip in light of the X-axis
2. Y-axis flip: flip in light of the Y-axis

(8) Basic table on the right → Distortion Correction: Distortion correction (for special algorithms)

6.9.2 Object Screen

The object screen displays the serial number, coordinate value, length, width, height, size, color and other information of the visually recognized object.



6.9.3 Template Screen

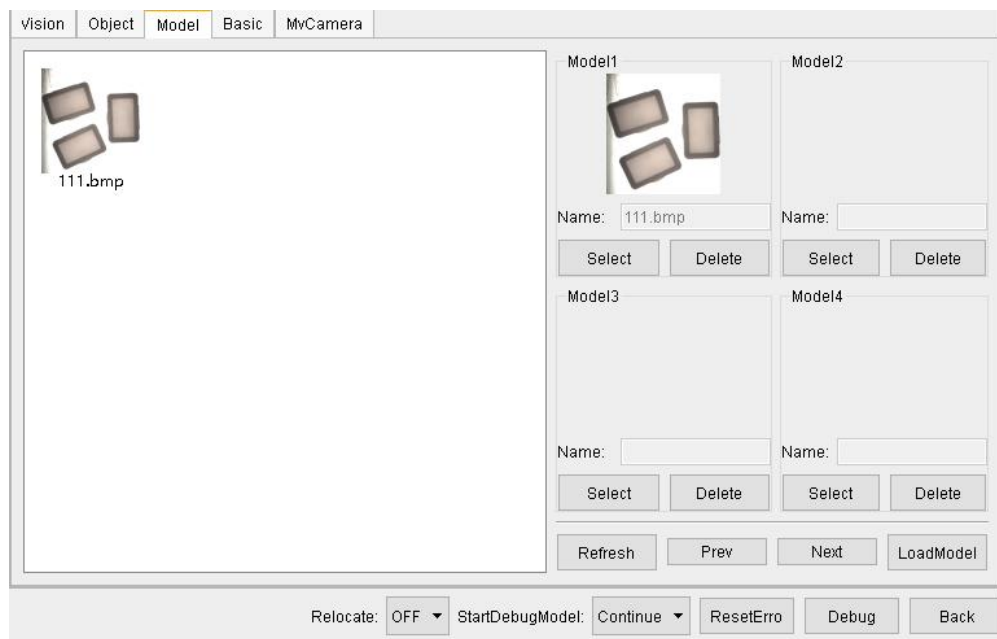
The templates set by the user will be displayed, if not all templates are displayed; please press the "Refresh" button. Up to 12 templates can be showed on the left. If there are more than 12 templates displayed in pages, press "prev" or "next" to select the next page.

1. Template allocation

Select the wanted template on the left, and press the "Select" button on the Model1/Model2/Model3/Model4 on the right to finish the assignment. For example: click the left to select 123.bmp, and press "Select" on Model1 on the right to assign the template 123.bmp to Model1.

2. Load the template

Press "Load Template" to make the template put into use. If more than one of Model1/Model2/Model3/Model4 is assigned a template, then press "Load Template" to put into use is the first.



6.9.4 Basic Screen

camera Type (Camera Type): No choice, because now there is only Hikvision camera to choose, and it will be gradually enriched in the future

camera Kind (Camera Kind): camera Kind: Color/Black & White

Vision ID (Vision ID): Vision ID,

Robot number (Robot No): Robot number. To which robot this camera shall be assigned to.

Trigger mode (Trigger mode): 0 — automatic trigger (trigger continuously when Yikingsoft starts), 1—user trigger (user programming or manually trigger)

Delayed trigger time (Dalay Trigger Time): Delayed trigger time, the next trigger

will be triggered 100ms after the image is triggered and received.

Max interval time: when no image is received after triggering, next trigger would be activated after 500ms the most.

Autostart: to launch the camera automatically when yikingsoft starts running, and automatically trigger to take pictures (in the automatic trigger mode). After configuring the above parameters, please press save, and restart YiKingsoft to put the configuration into use.

YiExVision		YiVision	YiFly
Vision	Object	Model	Basic
MvCamera			
CameraType:	MvCamera		
CameraKind:	ColorCamera		
VisionID:	1		
RobotNo:	-1		
Trigger Mode:	0 0-Auto trigger 1- Command trigger		
DelayTriggerTime(ms):	100		
MaxIntervalTime(ms):	500		
Autostart:	☐		
Save			

6.9.5 Hikvision camera Screen

Press the "Robot Programming" button on the main menu and select the plug-ins table. Select the camera table to be configured (for example, YiVision), and then select the MvCamera table

camera: Select the appropriate camera model from the drop-down list.

Frequency (Frequency): camera frequency, generally 1.3 million pixels Hikvision camera, the frequency is 45000000.

Collection Mode: photoing mode, 0—trigger photoing, 1—continuous photoing.

Exposure Time: The longer the exposure time, the brighter the image, but if the exposure time is too long, it is easy to produce smear, which affects the pick accuracy. Exposure time is generally 200-500us, not greater than 5ms, the shorter the better on the premise of clear pictures.

Gain: The larger the gain, the brighter the image will be, but increasing the gain will also inflate the noise. Generally, the default value can be used.

Trigger channel (Trigger channel): trigger channel, select a ConfigOut output signal to trigger the camera to take pictures.

Trigger Edge: trigger on rising or falling edge.

Conveyor Channel: Conveyor ID.

After configuring the above parameters, please press save and restart YiKingsoft to put the configuration into use.

The screenshot shows the 'MvCamera' configuration window in YiKingSoft. The window has a tabbed interface with 'YiExVision', 'YiVision', and 'YiFly' at the top, and 'Vision', 'Object', 'Model', 'Basic', and 'MvCamera' below. The 'MvCamera' tab is active. The configuration parameters are as follows:

Parameter	Value	Range/Options
Camera:	NOCamera	
Frequency:	45000000	Crystal frequency
Collection Mode:	0	0-Trigger, 1-Continuous
Exposure Time(us):	500	34~1000000
Gain:	10	0~20
Trigger Channel:	0	0~7
Trigger Edge:	RisingEdge	
Conveyor Channel:	0	0~2

A 'Save' button is located at the bottom of the configuration area.

6.10 Plug-in—YiFly

Visual plug-in for flying shooting, see "flying shooting function manual" for details.

6.10.1 Basic Parameters

Name: the name of the flying shooting plug-in

Command plug-in ID: the ID number used in the flying shooting command

Fixed speed: the speed from the start point to the end point of the flying shooting, to ensure that the shooting is not affected by the global speed

RZ correction: the correction rule of RZ angle during placing

Flying shooting index: a flying shooting plug-in supports a maximum of 10 shooting jobs which can be switched through index.

6.10.2 Configuration Screen

(1) Basic options

Configure robot and vision serial number through this table.

1. Trigger type: select the trigger type, there are two ways to trigger: proportional and distance triggering.

2. Trigger value: Modify the trigger value parameter. When the proportional trigger is selected, this value represents the proportion (percentage) of the flying shooting trigger point between the starting point and the ending point of the shooting.

When distance trigger is selected, this value represents the distance from the start point to the trigger point.

3. $\triangle$ MX- $\triangle$ MRZ: This parameter represents the error value tolerance allowed to be corrected by the flying shooting. When it exceeds the set value, the flying shooting will only be corrected by the maximum value set.

(2) Starting point

Set the starting point position and speed of flying shooting

1. Loading point: obtain the current point coordinates (default Tcp relative to default Base)

2. Move to: MoveL move to the starting point

(3) Ending point

Set the ending point position and speed of flying shooting.

1. Loading point: obtain the current point coordinates (default Tcp relative to default Base)

2. Move to: MoveL to move to the starting point

6.10.3 Calibration Settings

The screenshot shows the 'Calibration Set' tab in the software. It contains three main sections for setting calibration points: 'WaitPoint', 'PickPoint', and 'PlacePoint'. Each section has a grid of input fields for position (X, Y, Z) and orientation (RX, RY, RZ), as well as motion parameters (Velocity, Acceleration, Rot Velocity, Rot Acceleration, DifferPose). 'LoadPoint' and 'MoveTo' buttons are provided for each section. Below these sections is a 'Time' section with input fields for 'Static Stop Time', 'Static Max Wait Time', and 'Dynamic Max Wait Time'. A 'Save' button is at the bottom right. The sidebar on the right includes 'GlobalParameter', 'Config', 'Calibration Set' (which is selected and highlighted in orange), and 'Calibrate'.

(1) Waiting point

Set the waiting point position and speed of calibration for flying shooting.

1. Loading point: obtain the current point coordinates (default Tcp relative to default Base)

2. Move to: MoveL to move to the starting point

(2) Picking point

Set the waiting point position and speed of calibration for flying shooting.

1. Loading point: obtain the current point coordinates (default Tcp relative to default Base)

2. Move to: MoveL to move to the starting point

(3) Placement point

Set the placement point position and speed of calibration for flying shooting.

1. Loading point: obtain the current point coordinates (default Tcp relative to default Base)

2. Move to: MoveL to move to the starting point

(4) Time

Static stop time: Set the waiting time after the static flying shooting reaches the trigger point. After the waiting time, the camera will be triggered, to avoid the robot shooting when it reaches the trigger point.

maximum static waiting time: set the maximum waiting time for static flying shooting, the maximum time from when the camera is triggered to the image is acquired, this time can be set in light of the cycle, or to be determined in light of the image processing time. The general time for taking photos and image processing is very short, so setting the waiting time to 0.1~0.2s is enough. When the image receiving time is within the maximum waiting time, the flying shooting function can correct the error commonly. When the image is not received after the maximum waiting time, the robot will not wait any longer, and the flying shooting can not correct the error.

Maximum dynamic waiting time: Set the maximum waiting time for dynamic flying shooting, and start counting when the flying shooting reaches the starting point. If no image is received within the waiting time, the shooting fails.

Note: ① flying shooting calibration movement process, waiting point-pick point-start point-end point-placement point

②The static stop time, static maximum waiting time, and dynamic maximum

waiting time set in this screen are used for calibration and correction of flying shooting, and does not affect specific flying shooting commands.

6.10.4 Calibration Settings

ViFly-2.0.1

	X:	Y:	Z	RX:	RY:	RZ:
RobotPos:						
VisionPos:						
RelativePos:						

	X:	Y:	Z	RX:	RY:	RZ:
RobotPos:						
VisionPos:						
RelativePos:						

TCP: Output:

Static calibration Dynamic calibration Static verify Dynamic verify

Global Parameter Config Calibration Set Calibrate

After setting the parameters in the Configuration and Calibration Settings table, set the current robot TCP in this table, and an output IO channel is required when triggering photo taking.

Static Calibration: Press this button to exercise static flying shooting calibration, and record the current object position as the static flying shooting base.

Dynamic Calibration: Press this button to exercise dynamic flying shooting calibration, and record the current flying shooting object position as the dynamic flying shooting base.

Static verification: exercise static flying shooting verification. After static calibration, place the object at the picking position a little bit off course. Press this button, and the robot will exercise picking, flying shooting, correction, and placement. If the object is placed in an accurate position, it means verification is

successful.

Dynamic verification: exercise dynamic flying shooting verification. After dynamic calibration, place the object at the picking position a little bit off course. Press this button, and the robot will exercise picking, flying shooting, correction, and placement. If the object is placed in an accurate position, it means the verification is successful.

(1) Static

Static calibration data, no need to change, automatically generated after calibration is done. (robot point, vision point, deviation).

(2) Dynamics

Dynamic calibration data, no need to change, automatically generated after calibration is done. (robot point, visual point, deviation).

(3) Error

The difference between the statically calibrated error value and the dynamically calibrated error value.

It is suggested to do static calibration again without placing down the object after dynamic calibration. The comparison results of the two calibrations will appear here. In theory, the results of the two calibrations should be the same, and the less the difference, the better. If being a significant difference in the flying shooting direction, adjust the trigger synchronization time (see below for details) to reduce the calibration difference.

6.10.5 Global Parameters

Robot1
SynTriggerTime: 3

Robot2
SynTriggerTime: 3

Robot3
SynTriggerTime: 3

Robot4
SynTriggerTime: 3

Save

Trigger synchronization time: This parameter adjustment ensures that the moment when the camera takes pictures and the recording time of the robot position is as close as possible to the same time, thereby improving the accuracy of flying shooting. This parameter only affects dynamic flying shooting.

The parameters on this page are global parameters and are valid for all flying shooting.

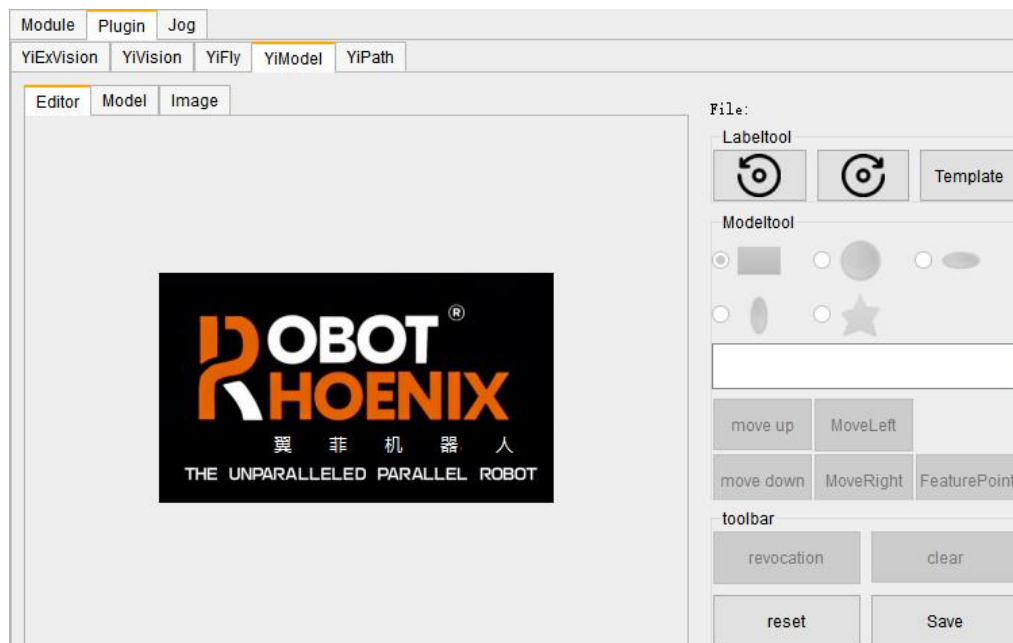
6.10.6 Function Keys

Save as: save the current flying shooting parameters to other flying shooting action, expedient for quickly creating new flying shooting

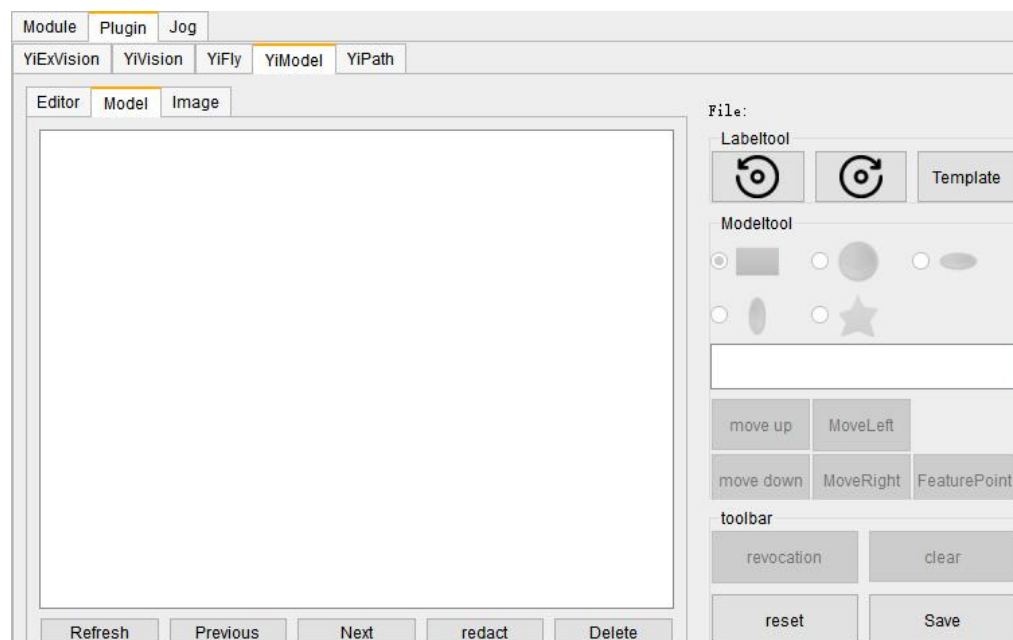
Save: save the parameters of the current page

6.11 Plug-in—YiModel

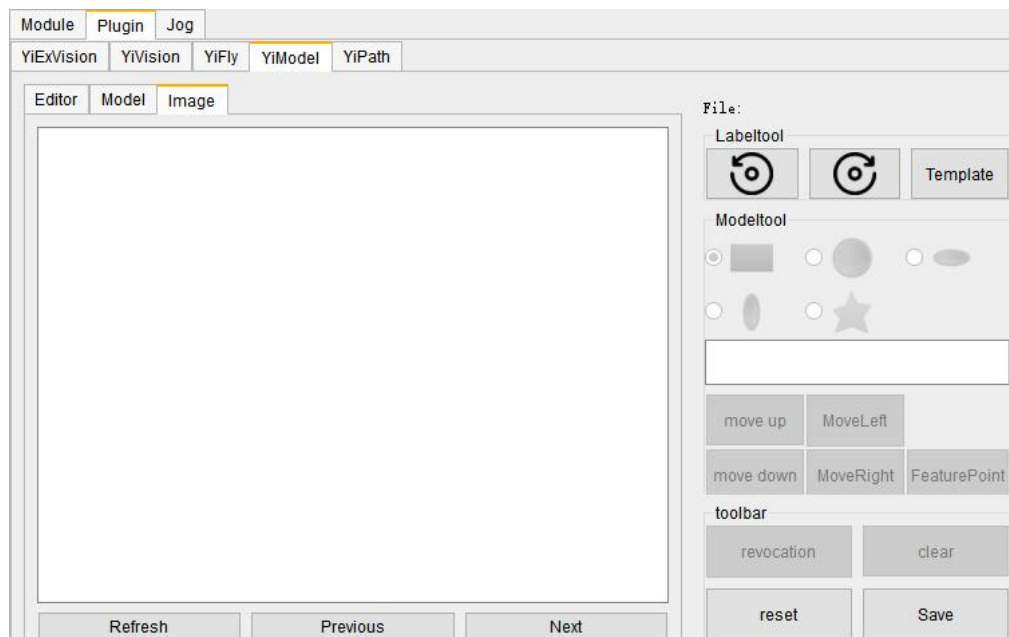
6.11.1 Edit Screen



6.11.2 Template screen

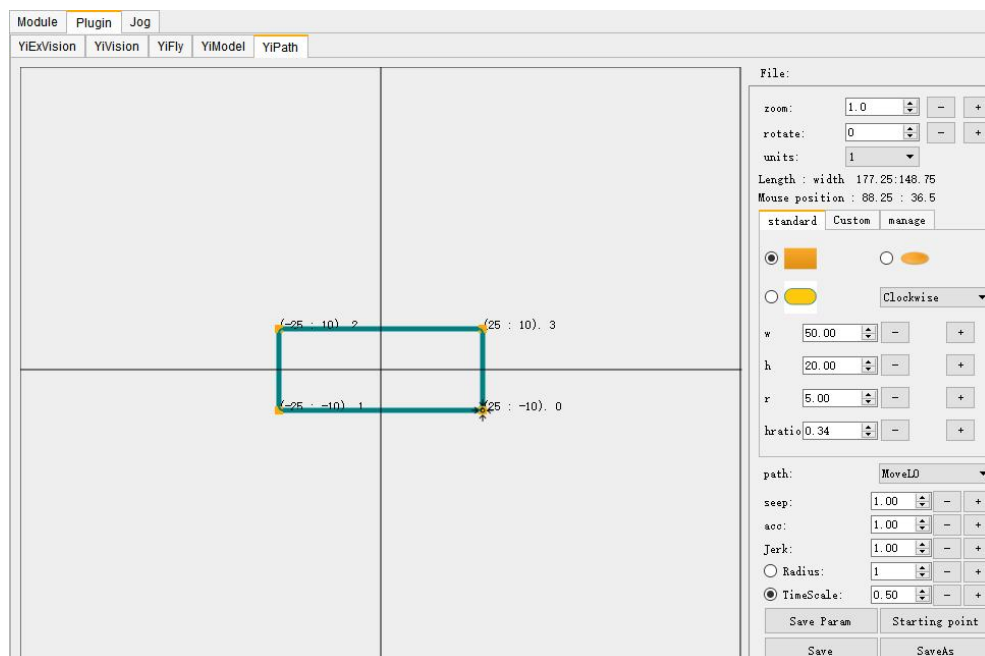


6.10.3 Graphical Screen



6.12 Plug-ins—YiPath

Create a motion path in collaboration with the MoveTrack command.



Chapter 7 Running the Program

7.1 Select Program to Run

Enter the main menu to run the program with the "Basic" identity authority, the "Delete" button will not be showed, select a program:

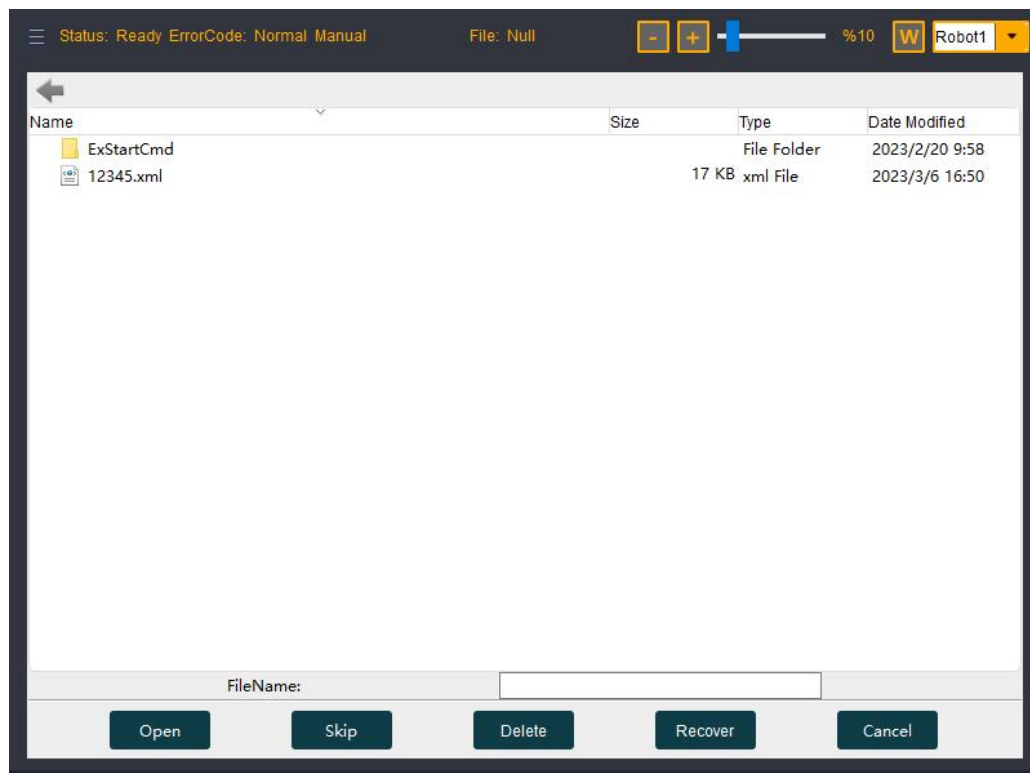
Press "Open" to use the selected program as the running program;

Press "Delete" to delete this program;

Press "Restore" to restore the deleted program, and the restored programs are all restored to the cmds folder;

Press "Cancel" to return to the main menu;

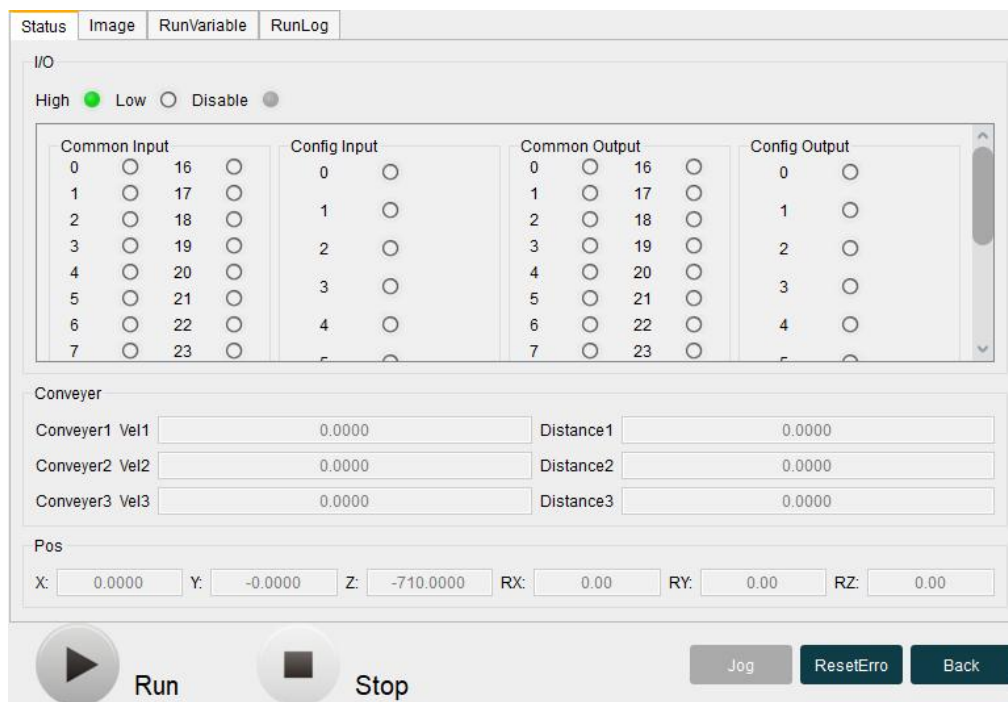
Press "Skip" to skip the selected program and go directly to status monitoring, vision, I/O, etc.



7.2 Program Running Screen

Select the program to enter the running screen, press "Run", the program will run immediately, the upper left status displays running, and the key will become a pause one.

If not choose the program, press "Skip" to jump to the running page, but the status on the upper left shows ready. At this time, press the run key, it will indicate that the file doesn't exist.



On the status page, I/O signals, conveyor belt movement, and robot position can also be monitored (if there are multiple robots, Robot1) select the robot to be viewed through the drop-down menu in the upper right corner)

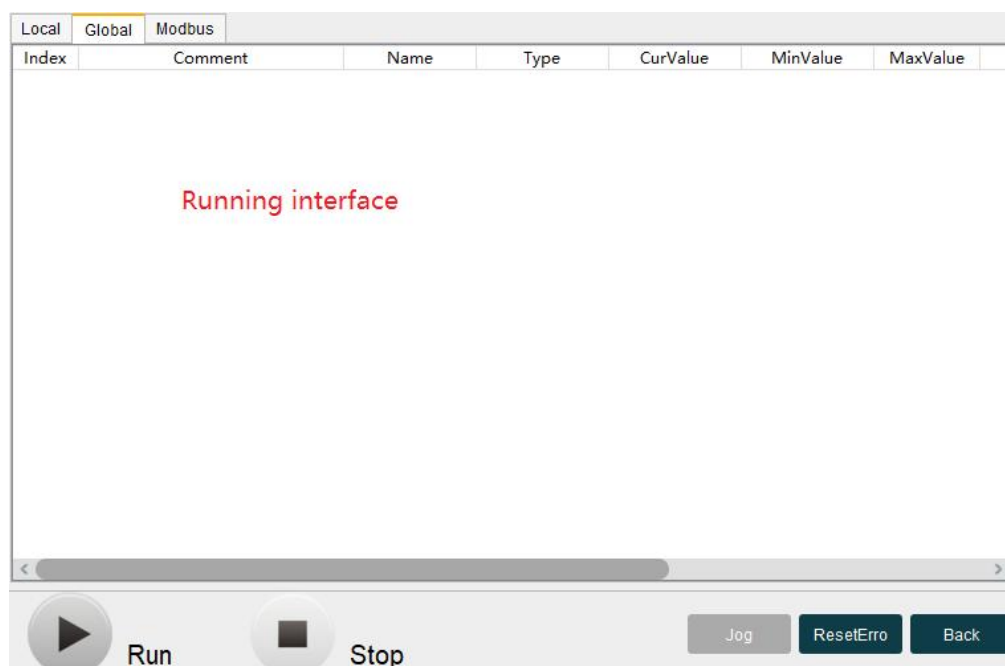
7.3 Image Table

Monitor visual images



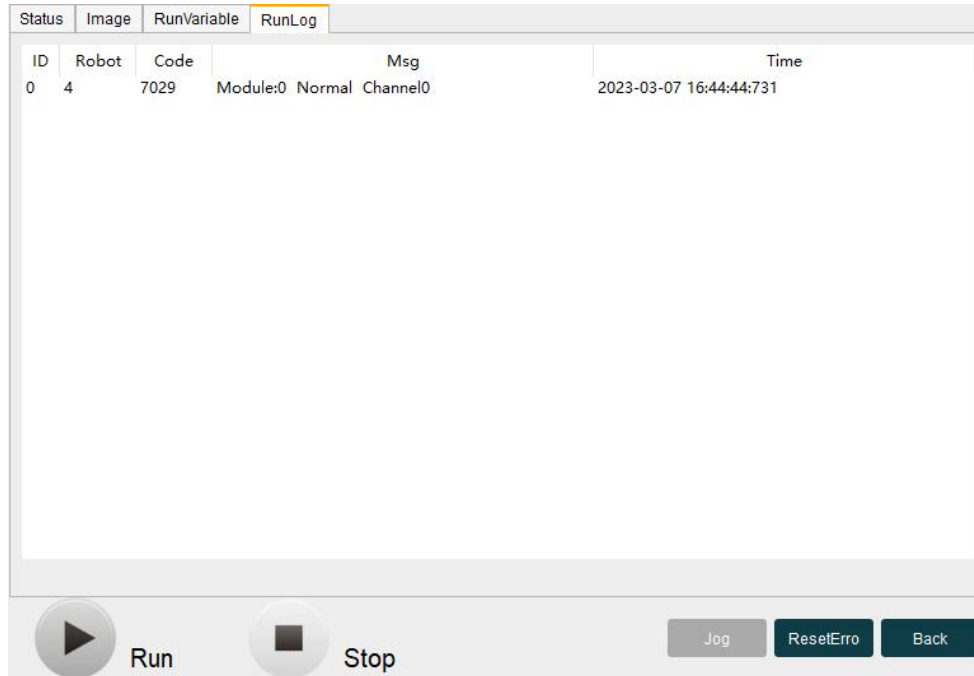
7.4 Running Variables

View variable values. Only when the "Add to variable list" parameter of a variable being True, this parameter will appear in the running variable page.



7.5 Running Log

Check the log for the log information during the robot running.



◎ Technical Services

Hangzhou Robotphoenix Industrial Robotics Co., Ltd is willing to provide information about the performance and relevant details of the robot, and help you for troubleshooting with detailed consultation. If your robot fails during running, you can immediately contact our service center and provide detailed robot information as much as possible, so as to help you to solve problems in time.

◎ Contact Information

Hangzhou Robotphoenix Industrial Robotics Co., Ltd

Jinan Headquarters: GLP--Robotphoenix Robotic Industrial Park, Hangtian Ave.,

Hi-tech Dist., Jinan Shandong Province

Hangzhou Subsidiary: Building 4, Phase II, Xiaoshan Robot Town, No. 477,

Hongxing Road, Xiaoshan Economic Development Zone, Xiaoshan District,

Hangzhou City

Tel: 400-116-0626

Email: Services@robotphoenix.com

Postcode: 250000

Website: <http://www.robotphoenix.com>