User Manual

Teaching Pendant Operation

Date: Jun, 2017

Vision: V2.0 (English)

SHINI

Contents

1. Safety

1.1 Transport and storage

- Product package iteration of no more than six
- It is not available in the product box on the climb, stand or place heavy objects
- Cannot use drag the cables attached to the product or handling products
- Avoid collision, scratching to the Panel and display screen
- Product box contact should be avoid from moisture , dust and the rain

1.2 Opening inspection

- After opening the packaging please confirm your purchased product
- Check whether the products in transit damage
- Control list identifies whether the part is complete, there is no damage
- Product model, lackof accessory or transport damage, please contact with me

1.3 Wiring

- To participate in connections and inspection person must have the appropriate skills for profession
- Products must have reliableearthing , grounding resistance should be less than 4 (Ω)ohms;
  you cannot use the neutral (zero line) instead of ground
- Wiring must be properly and firmly, so as not to lead to product failure or unexpected consequences
- And surge voltage absorption diode must be connected in accordance with the the circuit connections , otherwise you will damage
- Before plugs in or opens the front of the chassis; you must cut off the power supply
- They are also used in conditional judgments of process control commands. Example 3 is shown below:

1.4 Maintenance

- Must cut off the power suppy before repair or replacement of components
- Should check the fault when a short circuit or overload occurs, overcome the faults before they can restart
- Don’t restart frequently, if required to restart after a power failure, time interval of at least 1 minute

1.4.1 Others

- Do not open the Cabinet without permission,
- Long period when not in use, please cut off the power.
- To pay special attention controller not in contact withdust, iron powder etc..
- Output relay if the use of solid state relays shall be freewheeling diode in parallel in the relay coil. Check if the power supply meets the requirements, put an end to the

controller is burnt out.

- Controller temperature has much to do with the environment, if the processing temperature is too high, please install the cooling fan. Controller working ambient temperature range in between 0 ℃-60 ℃.
- Avoid high temperatures, humidity, dust or corrosive gas environments.
- Shake strongly to add buffer rubber Rails.

1.5 Maintenance

Under normal conditions of use (environment conditions: average 30 ℃, load 80%, running 12 hours a day), please press the following items for routine checks and regular checks.

Daily Check	Daily	●Recognition of environmental temperature, humidity, dust and foreign bodies ●There are no abnormal vibrations,
Period Check	1 year	● Substantial part is loose or not ● Terminal block damage

Since the robot system is more complex, dangerous keep manual records and security-related precautions. Please strictly observe variations as recorded.

1.5.1 Safety Precautions and mark

Mark		**The meaning of mark**
◇	Danger	Use wrongly, it will lead to a dangerous situation, causing serious injury or death.
⚠	Caution	Use wrongly, It will lead to a dangerous situation that may cause personal injury or damage to equipment which caused material damage.
🛇	Ban	Absolutely unenforceable
❗	Force	Must be implemented

1.5.2 Danger

Please do not use this system in the flammable and explosive environment.	
◇	Likely to cause injuries or fire.

Please follow the instructions drawings or wiring.	
◇	Prone to electrical shock and damage the motor.

In an energized state, do not arbitrarily pull the plug, in the operating state; do not touch the robot operation site.

⚠	Easy electric shock, causing personal injury.

Energized state, not for wiring, maintenance and other operations, be sure to power at least 5 minutes before proceeding.	
⚠	Easy electric shock.

Please place robot controller and robot body firmly stand on the ground.	
⚠	When the fault occurs easily lead to electric shock, fire incident, easy to mistake

Non professional person, please do not open the robot controller case, please do not use hand to touch the drive and control of internal components	
⚠	Easy electric shock

The energized state, do not touch the power plug of the robot controller	
⚠	Easy electric shock

Please do not damage, press of cable heavily or cable suspended heavy load.	
⚠	Easy electric shock

The energized state, do not plug off the port of robot controller.	
⚠	Easy electric shock and short circuit

The running state, do not pull out the terminal of robot controller.	
⚠	Easy electric shock and short circuit

1.5.3 Caution

Please caution the radiation of the motor of robot controller、robot body and accessories.	
⚠	Vulnerable to burn

When a fault occurs, the power supply must be cut off, the cause is identified and removed, and the low speed running equipment should be removed.	
⚠	If there is residual adverse factors, easy cause malfunction.

When using the controller and the robot body, do not exceed the specifications.	
⚠	Easily cause damage to the product.
When the robot is handling, it needs to be fixed with the attached fixed tool.	
⚠	To prevent the lifting arm, due to robot arm moving cause accidents.

Before installation, operation, maintenance check, be sure to read the instructions carefully, according to the instructions in the steps	
⚠	Easy electric shock, catch fire

Power supply voltage, power capacity must be specified by the company's specifications.	
⚠	Improper use of equipment failure, easy to catch fire.

Please correct use of the correct control of each other to robot controller and robot.	
⚠	Failure-prone

Regularly maintenance and inspection work for robot controller	
⚠	Ignoring maintenance and inspection, are important causes of equipment failure and accidents

Do not place heavy objects on the product	
⚠	Easy to damage

Please correct wiring according to the instructions in the wiring	
⚠	Wrong wiring easily lead to incorrect wiring or robot drive control machine damage or cause a fire

When an exception occurs, please stop immediately	
⚠	Prone to electric shock, injury, fire

When in need of repair, please contact our company, do not attempt to disassemble	
⚠	Could easily lead to malfunction

Do not strike	
⚠	Could easily lead to malfunction

1.5.4 Ban

Robot operation, the officer is not allowed to stand in the area of robot motion.	
🛇	Major disability incidents occur.

Banned in the workplace stacked hinder the operation of the robot equipment.	
🛇	When the device is abnormal, likely to cause injury.

Prohibit the handheld emergency stop switch on the teaching pendant short.	
🛇	Robot accident or not functioning properly, you need an emergency stop switch, stop operation of the equipment.

Prohibited without recording the instructions, incorrect operation.	
🛇	Improper operation will bring a malfunction of the device.

Other than operator personnel is prohibited near the equipment	
🛇	Touch dangerous parts can cause injury or serious accidents

When the accident occurred, to cut off the power, clear bad reasons	
🛇	Adverse residual reason, the robot may malfunction, causing adverse consequences.

Prevent users from unauthorized replacement parts and carry out reconstruction	
🛇	It will reduce system performance and may malfunction

Do not remove the cleanup by yourself	
🛇	Easily lead to fire, electric shock

Do not place the product stored in leaking, water, gas and other hazardous environments	
🛇	Failure-prone

1.5.5 Mandatory

Avoid direct sunlight when you save	
❗	Could easily lead to malfunction

Use within a predetermined range	
❗	Easily lead to burnout, failure

During operation of the device must be switched off guard	
❗	If open the protective cover could cause melectric shock, the risk of disability

Operators should adequately trained	
❗	Improper operation can cause equipment to malfunction, resulting in injury or major disaster

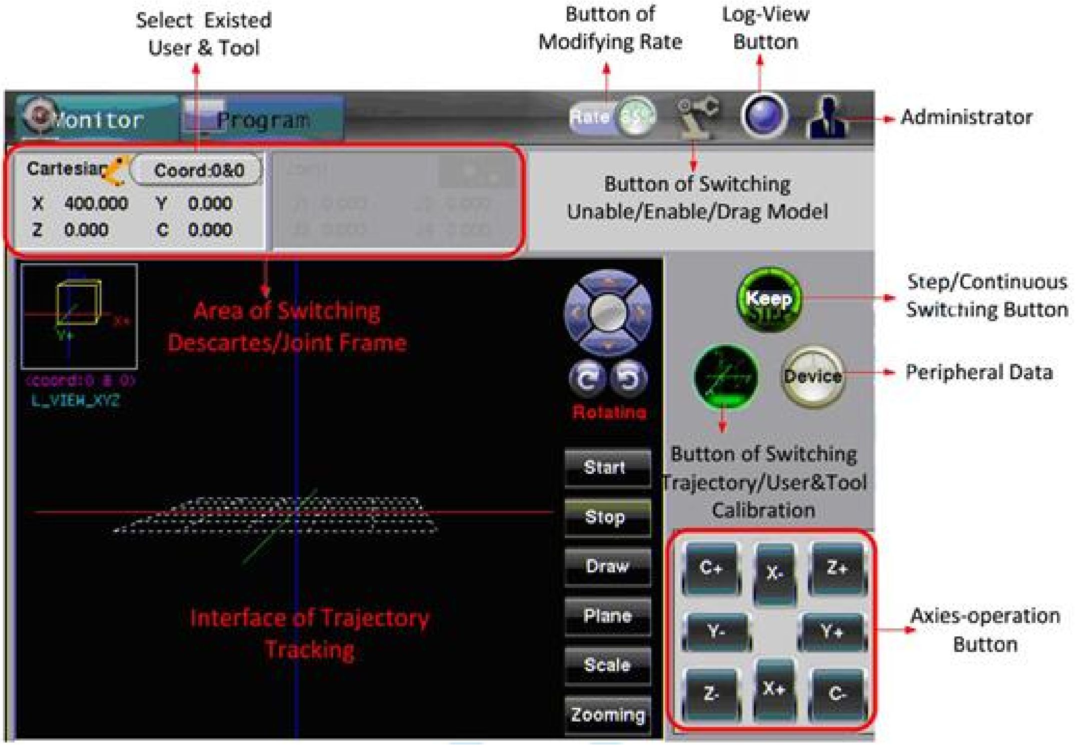
Manual teaching, if the robot is not in the specified direction of movement, immediately press the emergency stop, stop equipment operation	
❗	Prone to accidents and failures

Be sure to use the specified power line wires	
❗	Prone to fire and malfunction

1.6 Safety Rules

1) Before starting the operation, you should know that all the tasks in accordance with the robot programmed to be executed;
2) When running in automatic mode, any movement of its person will not allowed to enter reach areas;
3) When the need for programming, testing and maintenance work, shall robot shall be under manual mode;
4) When debugging, person enters into the robot work area, he must carry the teach pendant, to prevent others from malfunction;
5) When the robot for a long time does not work, the fixture should not attached shall be empty machine;
6) After a power outage must close the main power switch on the robot, and remove the tool holder.

2. List of Monitoring Interface


Select Existed User & Tool
Button of Modifying Rate
Log-View Button
Administrator
Monitor
Program
Rate
35
Cartesian
Coord:0&0
Button of Switching Unable/Enable/Drag Model
Area of Switching Descartes/Joint Frame
<coord:0 & 0>
L_VIEW_XYZ
Rotating
Keep Step
Step/Continuous Switching Button
Device
Peripheral Data
Button of Switching Trajectory/User&Tool Calibration
Start
Stop
Draw
Plane
Scale
Zooming
Interface of Trajectory Tracking
C+
X-
Z+
Y-
Y+
Z-
X+
C-
Axies-operation Button

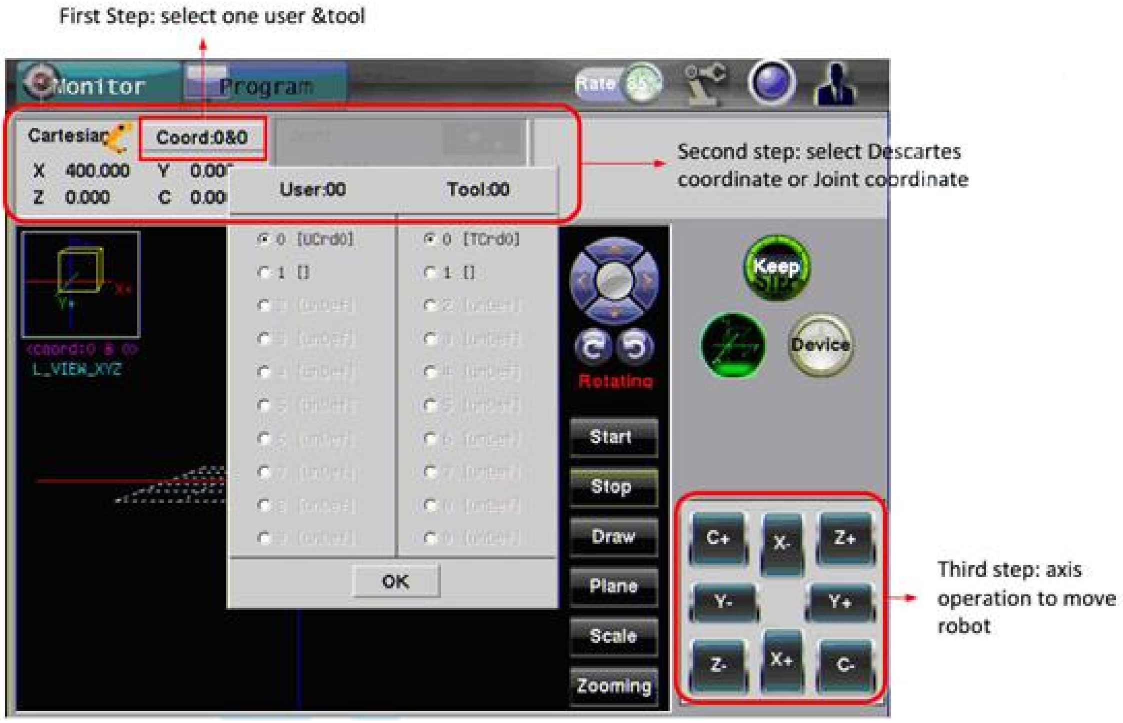
X 400.000 Y 0.000
Z 0.000 C 0.000

Some of the icons in this specification are named:

Icon	Name	Function
	Little orange-people	Be used to call the Project management Interface
	Robot	Be used to view state of the robot (Enable / Disable / Light-drag)
	Administrator	Be used to call interfaces of Parameters, SysInfo, Debug and Setup.
●	Log ball	Be used to enter the interface of recording alarms

Descartes position refers to the actual X/Y/Z/C coordinates which are relative to the robot origin under the current coordinate system; Joint position refers to actual coordinates of J1/J2/J3/J4 axies under the current coordinate system. Cartesian or joint coordinate system can be switched by clicking on the Cartesian or Joint area. Three basic steps can be followed to operate in descartes/joint coordinate system:

1) Select a corresponding user and tool coordinate in existing users and tools.
2) Select coordinate or joint coordinate;
3) Corresponding axis operations to move robot.



First Step: select one user &tool

Second step: select Descartes coordinate or Joint coordinate

Third step: axis operation to move robot

Cartesian Coord:0&0

X 400.000 Y 0.000

Z 0.000 C 0.000

User:00 Tool:00

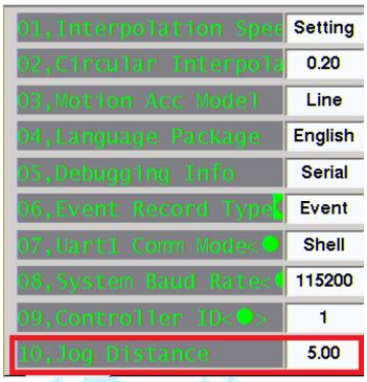
For Descartes/joints manually, there are some details, such as continuous/step switching, rate changes, and so on.

Tip: manual continuous or step movement determines the manual positioning accuracy; Rate is a global variable which affects the manual and auto speed.

2.1 Manual Continuous/Single Step Movement

Click button to switch continuous and single step movement. Here are some instructions:

1) the single step involves parameters setting of jog (default to 5.00), singlestep model by distance divided into three: 0.10 1.00 and 5.00 (custom);
2) under the Cartesian coordinate system, for X/Y/Z axes, length units are millimeter (mm); for C-axis, length unit is °;
3) Under joint coordinates system, for X/Y/C axes, length unit are ° ; for Z-axis, length unit is millimeter (mm).



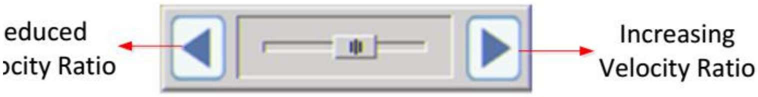
01,Interpolation Spee	Setting
02,Circular Interpola	0.20
03,Motion Acc Model	Line
04,Language Package	English
05,Debugging Info	Serial
06,Event Record Type	Event
07,Uart1 Comm Mode<●	Shell
08,System Baud Rate<	115200
09,Controller ID<●>	1
10,Jog Distance	5.00

Tip:
- Continuous function is used to quickly locate an pointed position in manualmode.
- Switch to single step and select an appropriate feed for high precision positioning.

2.2 Rate Modification

Rate: auto or manual rate are speed percentage referred to the current speed set in parameters. Rate description:

1) Rate variable is global, manual and automatic operation is called 1 time rate;
2) Rate affect the actual speed of manual and automated runs;
3) Manual actual speed is speed multiplied by the ratio, for example:
   If J2 axis speed is set as 200, , such as the current rate is 50%, so the manual actual speed of current J2: 200*50%=100;
4) Automatic grinding speed is the current speed multiplied by the ratio of the program;
5) Manual and automatic speed needed in the parameter interface settings, including interpolation speed and velocity.



Tips:

- Call this tool in any interface by clicking “F7” button to increase the speed, “F8” button to decrease the speed.
- Automatic grinding speed is the current speed multiplied by the ratio of the program.

2.3 Real/Virtual Position Switch

The gear has two states: open and close, which are respectively corresponding to the virtual and actual position of robot.

Click “⚙⚙” button to switch robot’s virtual and actual position. It records the virtual position when two gears separates（⚙⚙）; and it records actual position when two gears closes（⚙⚙）.

Tip:
In some demonstration operation, you can switch gears closed (red) to obtain the actual position of robot.

2.4 Robot State Switch

Robot has three states: non-enabled, enable and light-drag state. Both non-enabled and enable state can be used to move robot in automatic and manual mode; light-drag only applies to manual mode. Robot “”chart， which is used to switch robot three states:

1) Non-enabled state is the default (the chart colors for gray);
2) Light-touch the robot chart to switch the enable state (chart from grey to green);
3) Press and hold robot figure to switch light-drag mode (figure from gray to yellow);
3) Three states switches are in relation to non-enabled mode.

Disable: Robot is in a state of off-line simulation

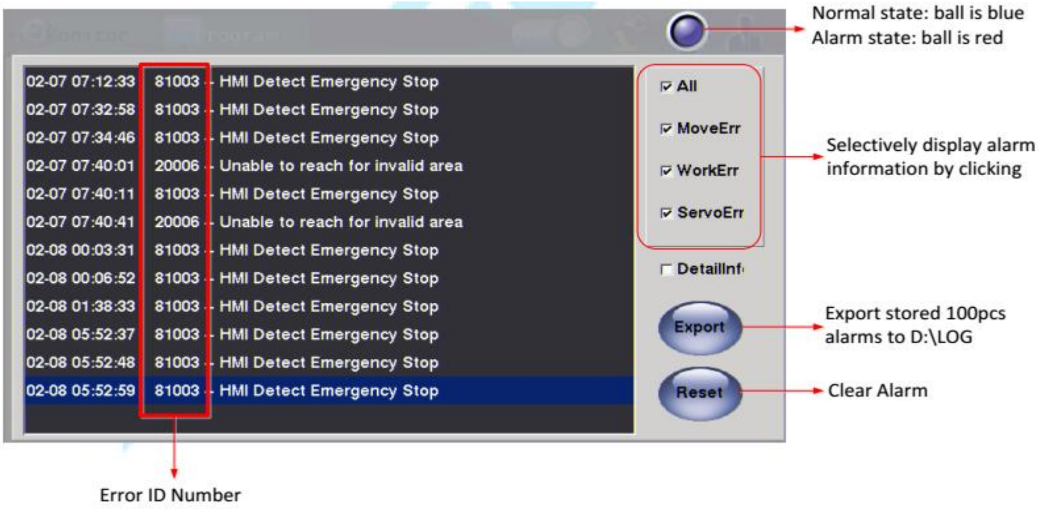
Enable: Robot is in a state of on-line simulation

Light-drag: Easy to move robot manually

Tip:
If you want to push the robot to a teaching position in manual mode, robot can be switched to light-drag mode; Enable robot manually to relieve light-drag mode.

2.5 Log View

Click log“ ”diagram,switch to log interface which can show the recording alarms and latest 12pcs messages:



```
02-07 07:12:33  81003 - HMI Detect Emergency Stop
02-07 07:32:58  81003 - HMI Detect Emergency Stop
02-07 07:34:46  81003 - HMI Detect Emergency Stop
02-07 07:40:01  20006 - Unable to reach for invalid area
02-07 07:40:11  81003 - HMI Detect Emergency Stop
02-07 07:40:41  20006 - Unable to reach for invalid area
02-08 00:03:31  81003 - HMI Detect Emergency Stop
02-08 00:06:52  81003 - HMI Detect Emergency Stop
02-08 01:38:33  81003 - HMI Detect Emergency Stop
02-08 05:52:37  81003 - HMI Detect Emergency Stop
02-08 05:52:48  81003 - HMI Detect Emergency Stop
02-08 05:52:59  81003 - HMI Detect Emergency Stop
```

☑ All
☑ MoveErr
☑ WorkErr
☑ ServoErr
□ DetailInf

Normal state: ball is blue
Alarm state: ball is red

Selectively display alarm information by clicking

Export stored 100pcs alarms to D:\LOG

Clear Alarm

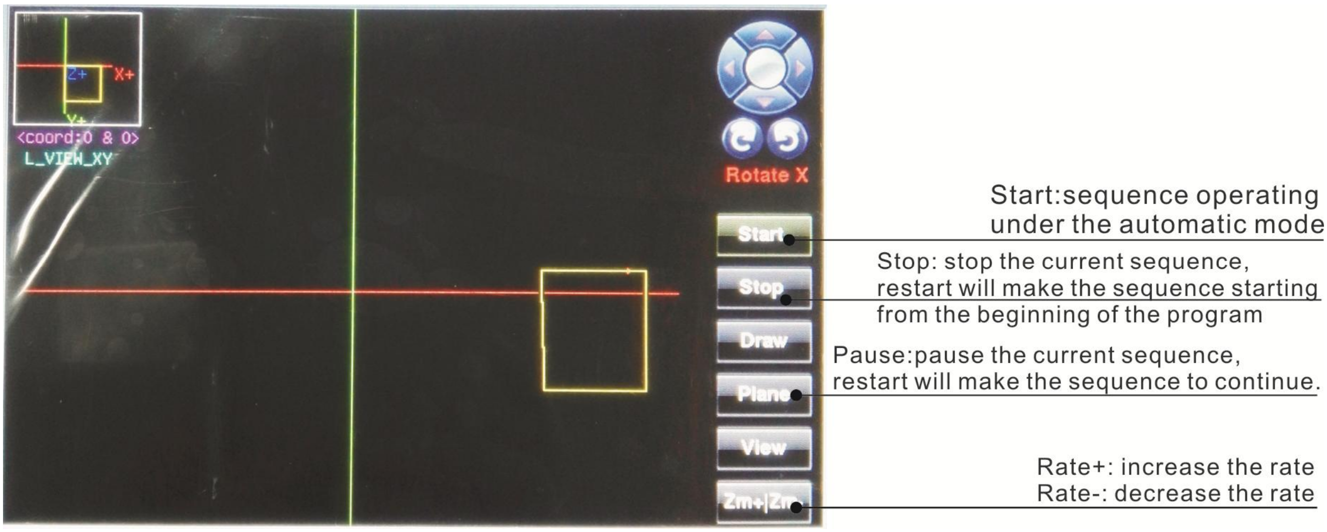
Error ID Number

Tip:

According to alarm records, it is easy to analyze causes of alarm, then to solve it. List of monitoring interface.

2.6 Trajectory Tracking

Tracking interface is primarily for trajectory simulation of running programs. In the running process, it is intuitive to see trajectory of end of the robot.



Start:sequence operating under the automatic mode

Stop: stop the current sequence, restart will make the sequence starting from the beginning of the program

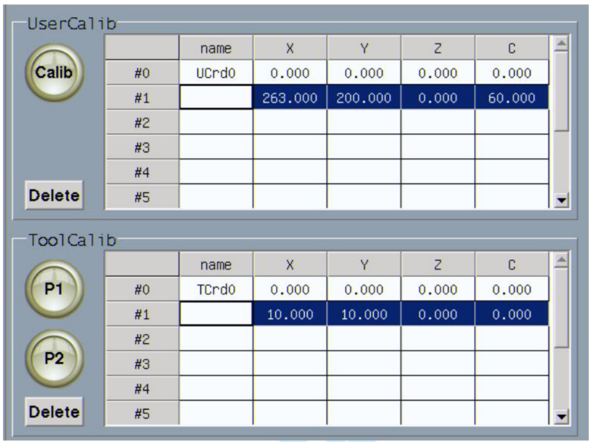
Pause:pause the current sequence, restart will make the sequence to continue.

Rate+: increase the rate
Rate-: decrease the rate

Tip:
Tracking is often used in automatic mode to preview whether the trajectory is correct.

2.7 User/Tool Coordinate Calibration

When the base frame is not at reference zero, user coordinate systems can be easily measured the points' location when operates teaching position and calculations. RC400 controller can contain 10 user coordinate systems, in which user 0 is default as base frame of robot. User 1-9 can be set manually or generated directly by three -point method.
When a fixture is added at the end of a robot, trajectory of the movement will not be referred to the center of flange, but to end of the fixture. Tool frame will make teaching and programming more flexible. 10 tool frames can be contained in RC400 controller, in which tools 0 is default. Tool 1~9 can be set manually or generated directly through twopoints teaching method. In the monitor interface, click on “ ”chart to enter user/tool calibration interface.



UserCalib

	name	X	Y	Z	C
#0	UCrd0	0.000	0.000	0.000	0.000
#1		263.000	200.000	0.000	60.000
#2					
#3					
#4					
#5					

ToolCalib

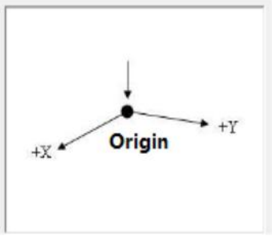
	name	X	Y	Z	C
#0	TCrd0	0.000	0.000	0.000	0.000
#1		10.000	10.000	0.000	0.000
#2					
#3					
#4					
#5					

2.7.1 User Coordinates Calibration

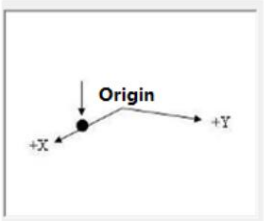
User coordinates calibration steps:

1) Select a user number from 1 to 9 (the line color of selected user number will mark as blue);

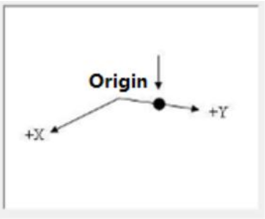
2) Click “Calib” diagram to enter user coordinate calibration interface;



3) Select “Org” in “UserCalib” interface; Manually adjust the end of the robot to overlap the origin of the user coordinate system under Descartes coordinate system; Then click on “Teach” to assign the current robot’s position to “Org”.

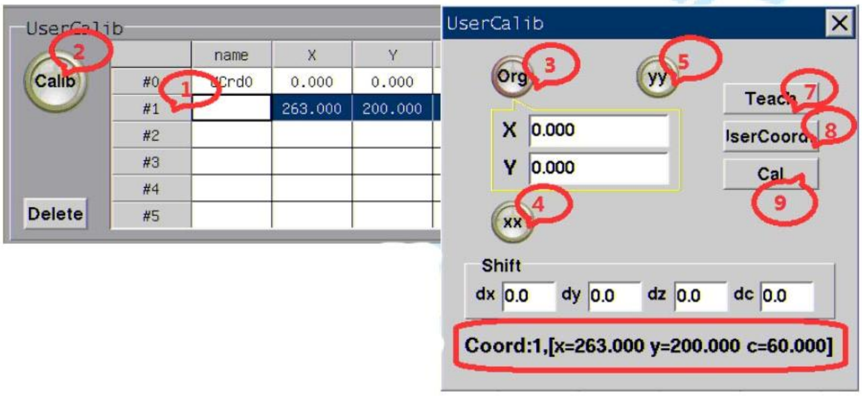


4) Select “xx” in “UserCalib” interface; Then move along the x-direction of the workpiece to reach a appropriate position; Then click on “Teach” to assign the current robot’s position to “xx”. Notice that C-axis is forbidde to be rotated, or it will lead errors during calculation.



5) Select “yy” in “UserCalib” interface; Then move along the x-direction of the workpiece to reach a appropriate position; Then click on “Teach” to assign the current robot’s position to “yy”. Notice that C-axis is forbidde to be rotated, or it will lead errors during calculation.

6) After org/xx/yy is teached completely, then click “Cal” to generate user coordinates. View results of generated user coordnates (X,Y,Z,C). Specific orders are as follow:1,2, 3,7,4,7,5,7,8,9.



	name	X	Y
#0	UCrd0	0.000	0.000
#1		263.000	200.000
#2			
#3			
#4			
#5			

X 0.000

Y 0.000

Shift

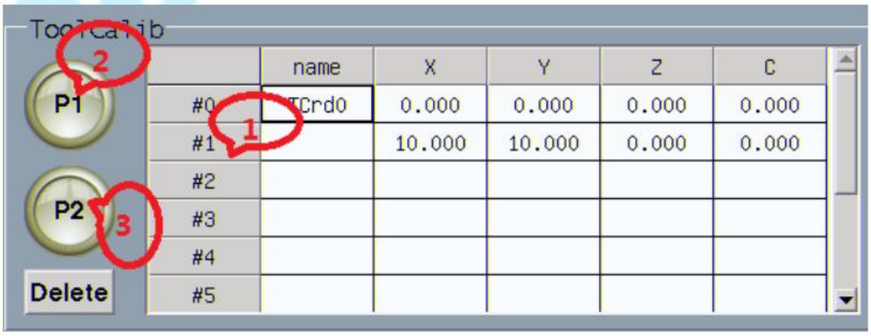
dx 0.0 dy 0.0 dz 0.0 dc 0.0

Coord:1,[x=263.000 y=200.000 c=60.000]

Tip:

The coordinate shifting represents the original point shifting on X,Y,C axis; If there is coordinate shifting, clicking “Cal” to re-calculate;

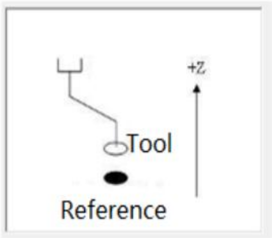
2.7.2 Tool Coordinates Calibration



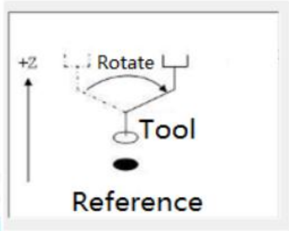
	name	X	Y	Z	C
#0	TCrd0	0.000	0.000	0.000	0.000
#1		10.000	10.000	0.000	0.000
#2					
#3					
#4					
#5					

Calibration steps of tool coordinates are as follows:

1) Select a tool from 1~9 which are not used(the line color of selected tool number will mark as blue);
2) Two-points teaching method, in left-hand coordinate, adjusting the robot tool tip coinciding with the reference point, click on the “P1” (the current position assignment value for P1 points);



3. Under Right-hand frame, adjust the tool tip again coinciding with the reference point, click on the “P2”(location assigned to the P2). Calculates the tool parameters (X/Y/Z/C) records in the selected row.

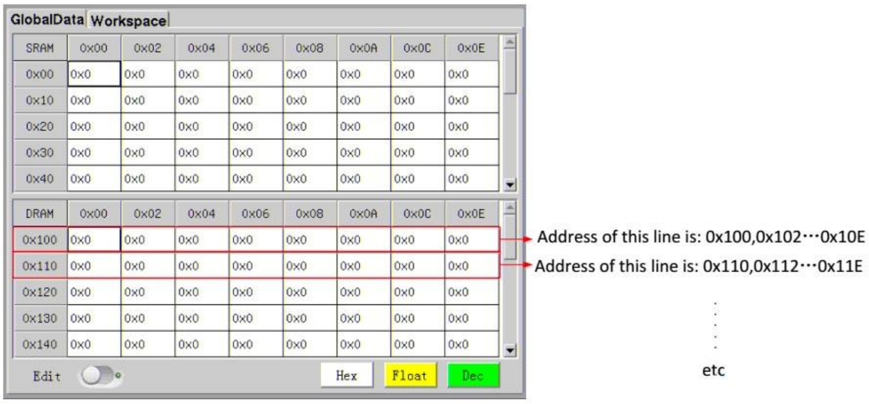


2.8 Peripheral

2.8.1 GlobalData

Click “Device” to enter external setting interface which process external PLC and RC400 controller for Modbus communication data memory. There are two kinds of storage methods: SRAM (Static RAM) and DRAM (Dynamic RAM). SRAM is nonvolatile memory, which means that the stored data will not be lost in case of power off; Contrary, DRAM is a volatile memory, which means that the stored data will be lost in the case of power off.

RC400 controller communicates with external devices through Modbus, RC400 controller is used as a slave station and the external device is used as a master station. Length of stored data is 32 bits, and each data is occupied with 2-length address. Both SRAM and DRAM can store 128 data, in which memory address of SRAM is from address 0 to 254 and memory address of DRAM is from 256 to 510.



SRAM	0x00	0x02	0x04	0x06	0x08	0x0A	0x0C	0x0E
0x00	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
0x10	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
0x20	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
0x30	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
0x40	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

DRAM	0x00	0x02	0x04	0x06	0x08	0x0A	0x0C	0x0E
0x100	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
0x110	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
0x120	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
0x130	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
0x140	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

Address of this line is: 0x100,0x102···0x10E

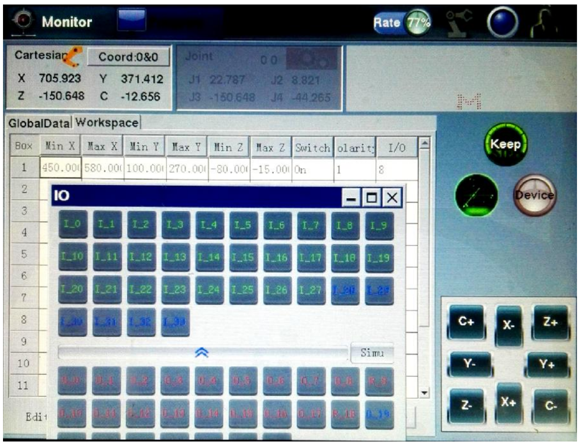
Address of this line is: 0x110,0x112···0x11E

etc

Tip:
The type of read data should be consistent with the type of PLC data written to the controller. For example, if the type of data written to controller is float, then they must be read in the form of float.

2.8.2 Workspace

Workspace: the maximum moving space of tool when the machine is normally function, also known as “safespace”.



X 705.923 Y 371.412
Z -150.648 C -12.656

J1 22.787 J2 8.821
J3 -150.648 J4 -44.265

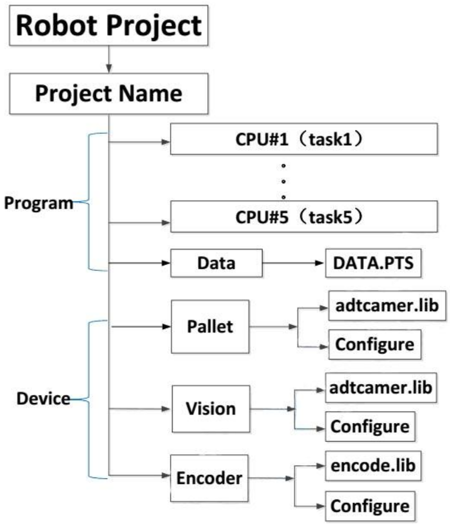
Box	Min X	Max X	Min Y	Max Y	Min Z	Max Z	Switch	olarit	I/O
1	450.00	580.00	100.00	270.00	-80.00	-15.00	On	1	8

Tip:

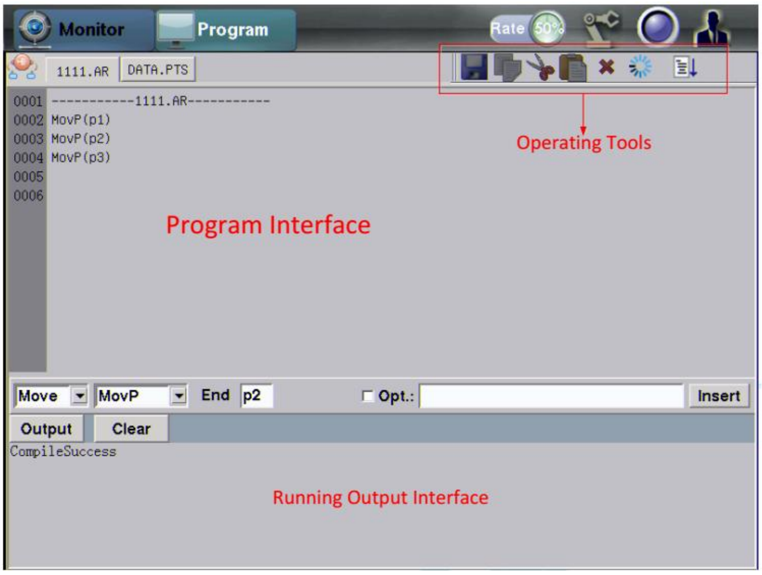
- Based in the desired working process to set up the proper range of XYZ;
- “Switch” includes “on” and ”off”. “On” will turn on the Workspace function, “Off” will turn off the Workspace function;
- Polarity could be classified as 0 and 1. 0 represents that the tools cross over the setting workspace, the corresponded port would open, 1 represents that the tools is in the setting workspace, the corresponded port would open;
- I/O is based on the required setting of connection to set the output port;

3. Programming

RC400 controller programming is mainly around the project tree:



Programming interface:



```
-----------1111.AR-----------
MovP(p1)
MovP(p2)
MovP(p3)
```

```
CompileSuccess
```

3.1 Project

1) Robot project is managed in a form of project, which contains configurations of alldevices (visual communication, external encoder and pallet) and programming (eachCPU task program);

. 2) It is convenient to copy one project from one controller to another controller with sametype.

3.1.1 Build Minimum Project

Here, we establish a minimum project to demonstrate its operation and application; a minimum project includes a CPU and a point table (DATA.PTS).

3.1.1.1 New-built

Steps to establish a new minimum project are as below:

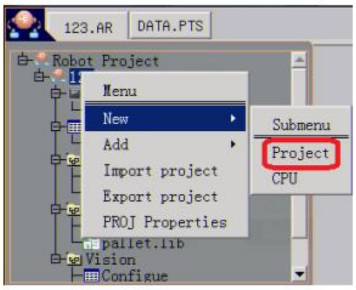
1) Click the small orange ball“ ” figure to pop up a “Robot Project” menu;

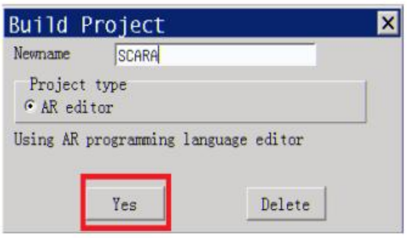
2) In “Robot Project” menu , long press an existing project name (Assuming 123) to pop up a “Menu” list;

3) Select “New” to pop-up “submenu” list;

4) Select “Project” in the “sub-menu” list to pop-up “Build Project” dialog box;

5) In “New Project” dialog box, type a new name (Assuming SACRA), then “Yes”. So Programming.of the framework called SCARA and the minimum project has generated, then you will need to configure CPU #1 and teach some points.

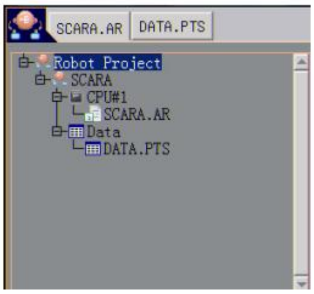




Newname SCARA

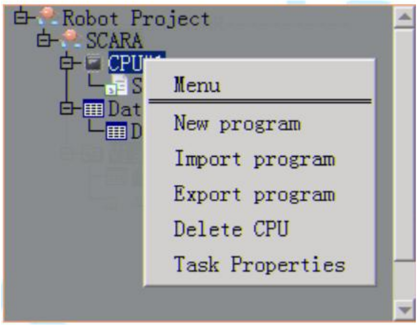
Project type
● AR editor

Using AR programming language editor



3.1.1.2 CPU#1Setting

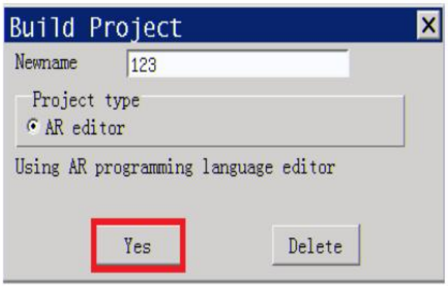
CPU#1’s mission is to perform some motion commands, delay command, IO, and user &&tool coordinate system settings. CPU#1 setting is including new/import/export programs, delete CPU and task properties. Press “CPU # 1” to pop up “menu” list:



➢ New Program

It is suitable for some simple testing programs, such as point, line, arc, arch and some simple motion commands. Specific steps are as follows:

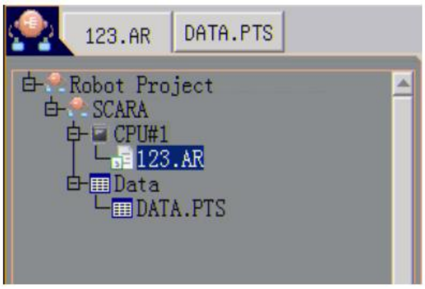
1. Select the “New Program” in “Menu” list to pop-up “Build Project” dialog box, then type a new name (Assuming 123), then “OK”. For example, to achieving to run a square in 123.AR, then you need to teach some points and write AR programming.



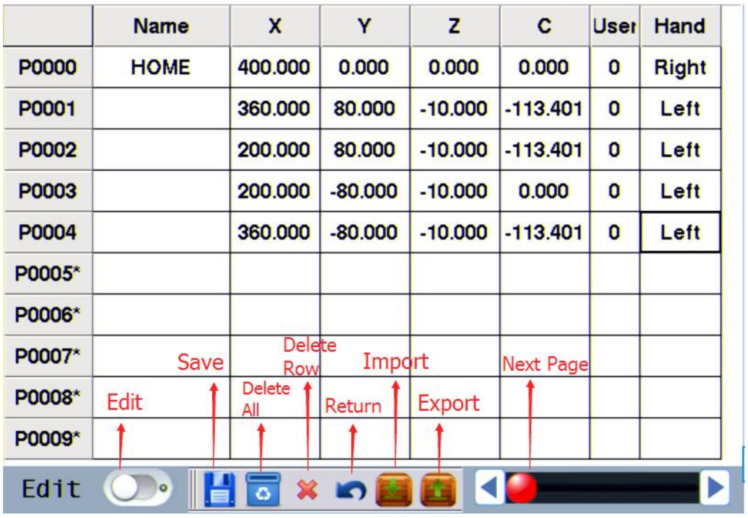
Newname 123

Project type
● AR editor

Using AR programming language editor

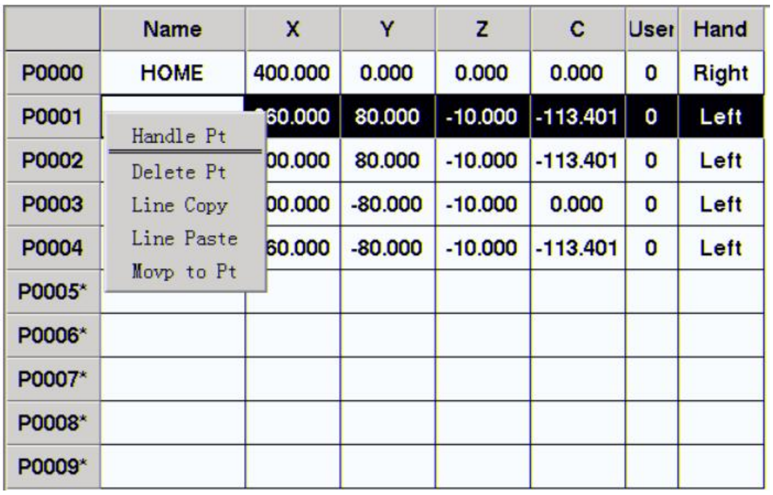


2. Teach point. Open the “DATA.PTS” file and select P001*1 (if selected, this line becomes black), then move the robot to (368, 80, -10, -113.401) position; Next, click “Teach” to assign this point to P001. And then teach P002/P003/P004 with the same ways, thus four points are recorded in “DATA.PTS” list, and then click the save“💾” button.



	Name	X	Y	Z	C	User	Hand
P0000	HOME	400.000	0.000	0.000	0.000	0	Right
P0001		360.000	80.000	-10.000	-113.401	0	Left
P0002		200.000	80.000	-10.000	-113.401	0	Left
P0003		200.000	-80.000	-10.000	0.000	0	Left
P0004		360.000	-80.000	-10.000	-113.401	0	Left
P0005*							
P0006*							
P0007*							
P0008*							
P0009*							

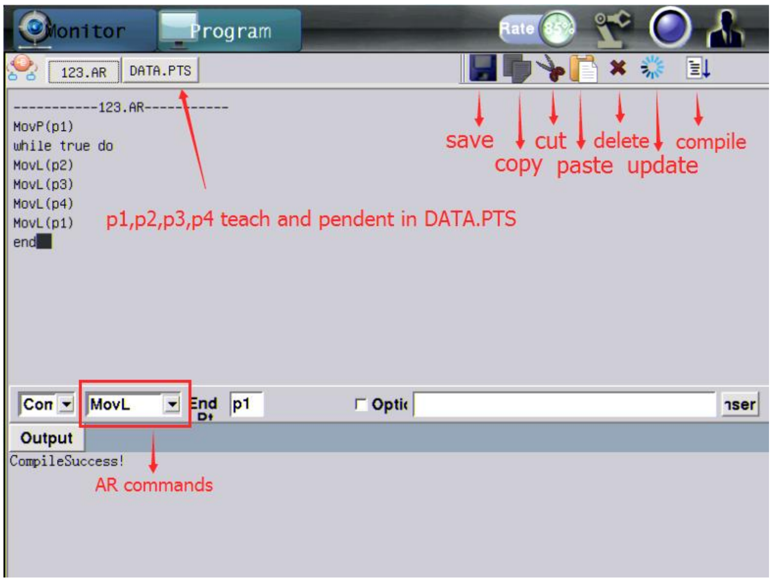
Tips:
P000 is fixed robot’s zero point which cannot be modified. Usually, tracking this point can quickly find zero point.Taught points can be processed. For example, long press “P0001” to the pop-up “Handle Pt” (handle point) menu list, including Delete Pt, Line Copy, Line Paste, and Movp to Pt (track this point with MovP command).



	Name	X	Y	Z	C	User	Hand
P0000	HOME	400.000	0.000	0.000	0.000	0	Right
P0001		60.000	80.000	-10.000	-113.401	0	Left
P0002		00.000	80.000	-10.000	-113.401	0	Left
P0003		00.000	-80.000	-10.000	0.000	0	Left
P0004		60.000	-80.000	-10.000	-113.401	0	Left
P0005*							
P0006*							
P0007*							
P0008*							
P0009*							

Delete Pt: Delete the current coordinate data;
Line Copy: Copy the current coordinate data;
Line Paste: Replace the current coordinate data by copied coordinate data;
Movp to Pt:Move the current coordinate;

3. AR programming; unfold “123.AR”, Then insert the square movement commands, as below:



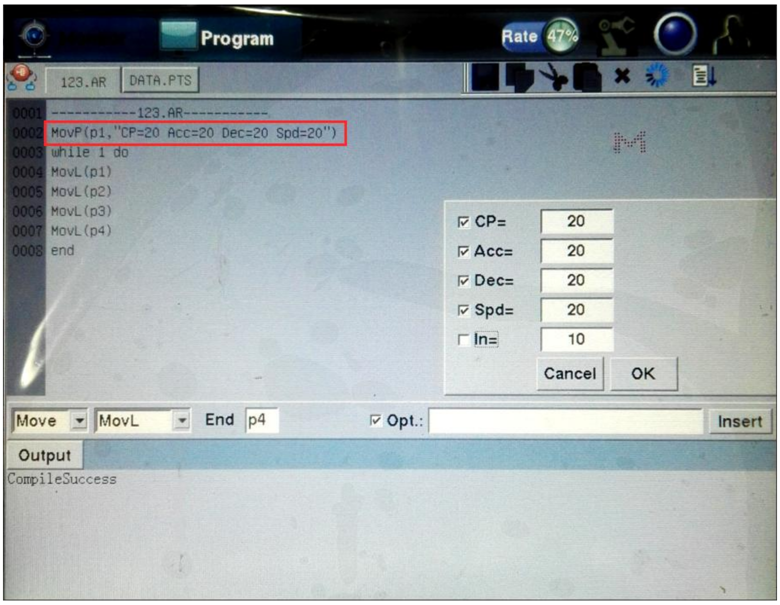
```
-----------123.AR-----------
MovP(p1)
while true do
MovL(p2)
MovL(p3)
MovL(p4)
MovL(p1)
end
```

p1,p2,p3,p4 teach and pendent in DATA.PTS

```
CompileSuccess!
```

AR commands

For some commands, such as MovP/MovL/MArchP/MArc et al., they are related to some optional parameters. Take MovP as an example, which includes CP/Acc/Dec/Spd.



```
-----------123.AR-----------
MovP(p1,"CP=20 Acc=20 Dec=20 Spd=20")
while 1 do
MovL(p1)
MovL(p2)
MovL(p3)
MovL(p4)
end
```

☑ CP= 20
☑ Acc= 20
☑ Dec= 20
☑ Spd= 20
□ In= 10

Move, MovL, End p4, ☑ Opt.:

```
CompileSuccess
```

CP	Optional parameter (0~100), which specifies whether smoothly move to target.
Acc	Optional parameter (1~100), which specifies percentage of acceleration to move to target.
Dec	Optional parameter (1~100)， which specifies percentage of deceleration to move to target.
Spd	Optional parameter (1~100), which specifies percentage of speed to move to target.

Note:Optional parameters set for other movement commands can refer to AR language manual.

- Export Program

You want to back up one AR language to a U-disk though export program operation, for example back up 123.AR to U-disk, specific steps are as follows:

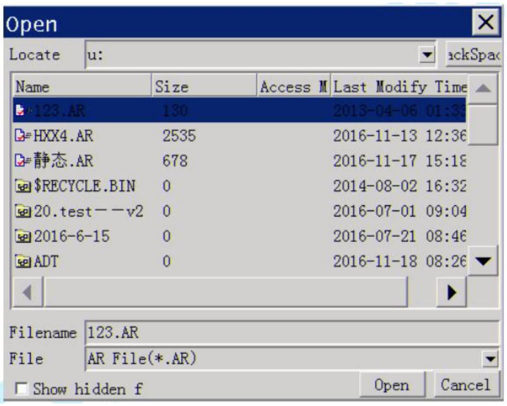
1) Insert one U-disk to the bottom of the teach pendant or MEN port;

2) Press “CPU#1” to pop up a “menu” list, then select “Export program” to enter “Save” interface;
3) Find “u:” in “Locate” drop-down list, then click “Save”.

- Import program

If one project is very complex (maybe up to 100 lines or more), it is inconvenient to insert these commands on teach pendent. In this case, it is better to edit AR language using LuaEditor, and then export it (assuming 123.AR) to controller. Specific steps are as follows:

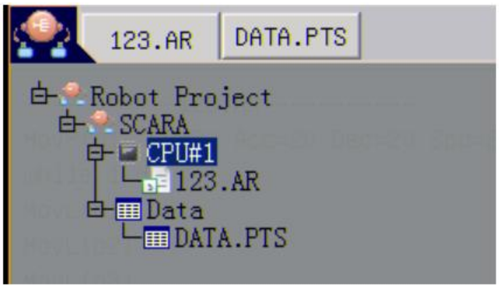
1) Import 123.AR to a U-disk;
2) U drive into the bottom of the teach pendant or USB interface controller side MEM port;
3) Press “CPU#1” to pop-up “menu” screen, then select “Import program” to enter “Open” interface;
4) Find “123.AR” from “u:” in “Locate” drop-down list, then click “Open”.



Name	Size	Access M	Last Modify Time
123.AR	130		2013-04-06 01:33
HXX4.AR	2535		2016-11-13 12:36
静态.AR	678		2016-11-17 15:18
$RECYCLE.BIN	0		2014-08-02 16:32
20.test——v2	0		2016-07-01 09:04
2016-6-15	0		2016-07-21 08:46
ADT	0		2016-11-18 08:26

Filename 123.AR
File AR File(*.AR)

If the following screen shot, the program is successfully imported.



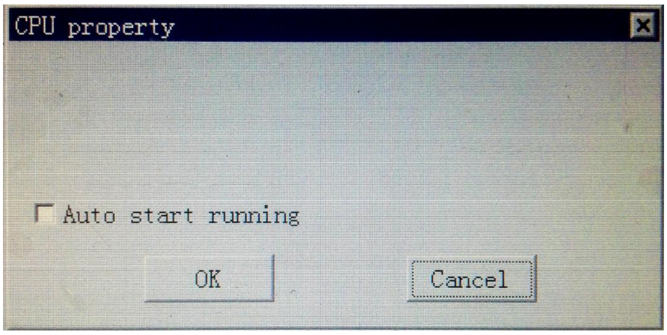
5. Teach some points in DATA.PTS, which are used in 123.AR

- Delete CPU

This operation is only used for several CPU in one projct.

- Task Properties

1) CPU1Task Properties

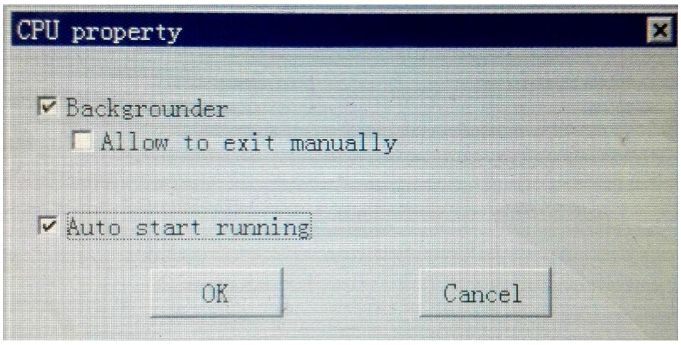


CPU property

□ Auto start running

Auto start running: Switch the key on the control panel into gear “A” then auto run the CPU after reboot the system. Normally apply in the situation of off the plug. Auto running function only works after rebooting the system, it will stop after stop running.

2) CPU2~CPU5 task properties



CPU property

☑ Backgrounder

□ Allow to exit manually

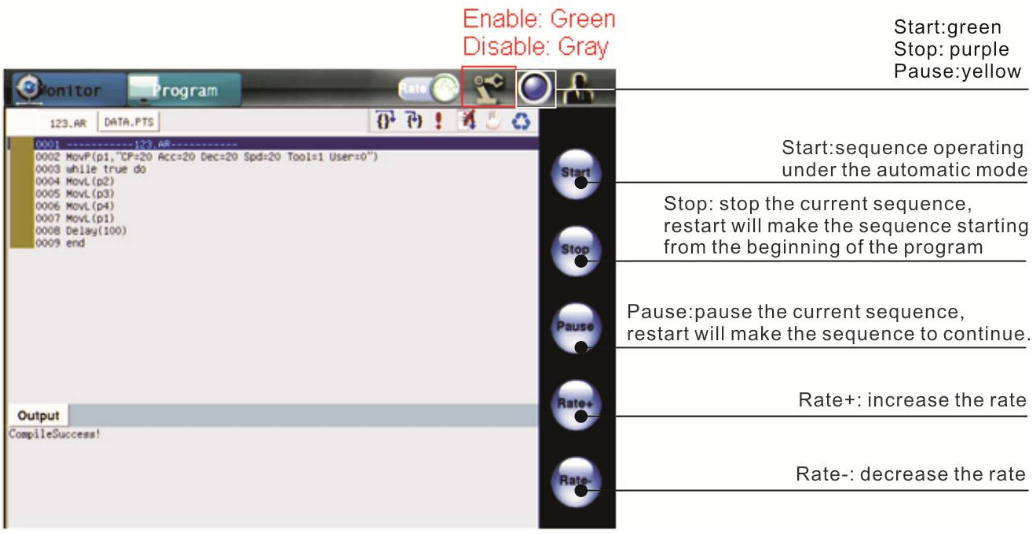
☑ Auto start running

Backgrounder: CPU is always running at background, not affected by system working condition. Mostly using in the project with multiple CPU, such as: CPU1 for controlling movement, CPU2 for monitoring IO. If “allow to exit manually” is picked, when the program sequence alarm，stop，reset，CPU2 will stop running. If “allow to exit manually” is not being picked and the program sequence alarm，stop，reset，or switch to manual control，CPU2 will continue to run at background，CPU2 will only stop at background by turn off the machine. Auto start running: Switch the key on the control panel into gear “A” then auto run the CPU after reboot the system. Normally apply in the situation of off the plug. Auto running function only works after rebooting the system, it will stop after stop running.

3.1.2 Test Running

If program is compiled correctly, you can testing running. For safety, you should operate robot with off-line simulation firstly, then which means that program will run but the robot does not move; then view movement trajectory to judge whether program’s logic is right and points is within robot’s working range. The speed ratio 50% is suitable when off-line simulation.

Off-line simulation: Key stays at automatic (A) model, Robot icon “ ” is gray. For off-line simulation, click “Start” button to run 123.AR and view trajectory from monitor interface.



```
------------123.AR------------
MovP(p1,"CP=20 Acc=20 Dec=20 Spd=20 Tool=1 User=0")
while true do
MovL(p2)
MovL(p3)
MovL(p4)
MovL(p1)
Delay(100)
end
```

```
CompileSuccess!
```

Enable: Green
Disable: Gray

Start:green
Stop: purple
Pause:yellow

Start:sequence operating under the automatic mode

Stop: stop the current sequence, restart will make the sequence starting from the beginning of the program

Pause:pause the current sequence, restart will make the sequence to continue.

Rate+: increase the rate

Rate-: decrease the rate

Descriptions of some buttons are in following table:

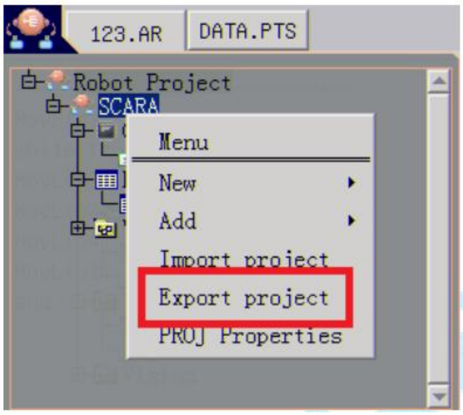
{}↓	Single segment debugging
{↑}	Step through each line of code as it runs
!	Start running the program (as with start button function)
✗	Stop running the program (and stop button function)
✋	Add breakpoints
♻	Delete breakpoints

Online Simulation: Key stays at automatic (A) model, Enable the robot “ ” and click “Start” to run AR.123, then robot will move to the corresponding positions. Press “Rate+” or “Rate-” to increase or decrease the speed ratio;

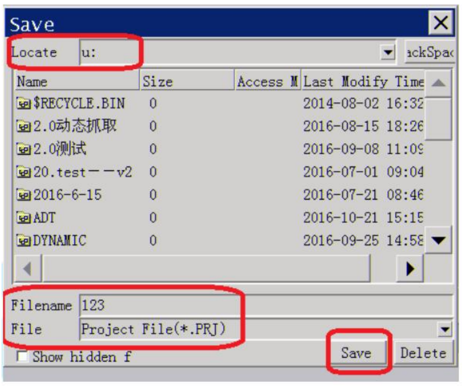
3.1.3 Export Project

Operation of exporting project is used for saving a built project. Then import the saved project to other robots to do the same process in order to save time and improve efficiency. For example, export the SCARA project of section 3.1.1 to a U-disk by following. steps:

1) Insert a U-disk into the bottom of the teach pendant or MEM port of RC400 controller;
2) Press project name “SCARA” to pop up a “Menu” list, and select “Export project” to enter “Save” dialog;



3) Find “u:” in “Locate” drop-down list, then click “Save” to export project completely.



Locate u:

Name	Size	Access M	Last Modify Time
$RECYCLE.BIN	0		2014-08-02 16:32
2.0动态抓取	0		2016-08-15 18:26
2.0测试	0		2016-09-08 11:09
20.test——v2	0		2016-07-01 09:04
2016-6-15	0		2016-07-21 08:46
ADT	0		2016-10-21 15:15
DYNAMIC	0		2016-09-25 14:58

Filename 123

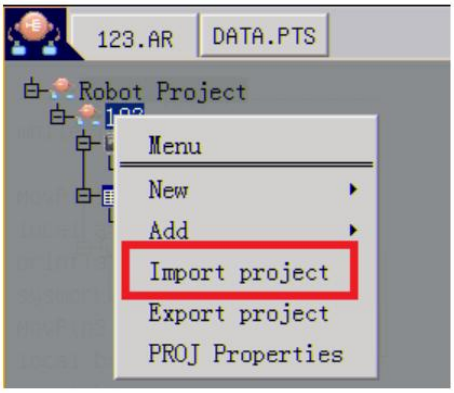
File Project File(*.PRJ)

3.1.4 Import Project

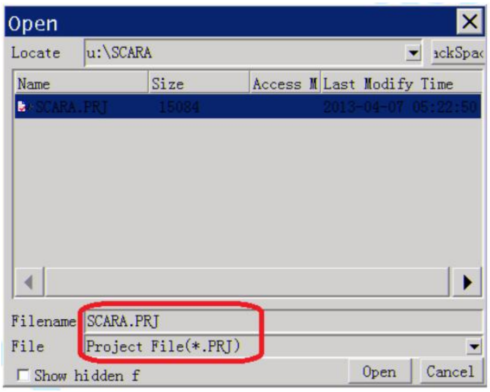
Operation of import project is used for a having been testing project which can be directly imported to controller to run. Specific steps are as follows:

1) Insert a U-disk into the bottom of the teach pendant or MEM port of RC400 controller;

2) Press the current project name (assumed to be 123) to pop-up a “Menu” list, then select “Import project” to enter “Open” dialog box:



3) Find the imported project in “u:” which is located in “Locate” drop-down list; then click “Open” to finish this operation. Notice that the type of imported file must be end up with .PRJ



Open

Locate u:\SCARA

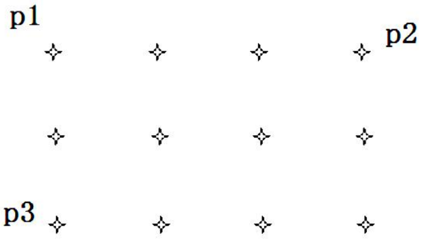
Name	Size	Access M	Last Modify Time
SCARA.PRJ	15084		2013-04-07 05:22:50

Filename SCARA.PRJ

File Project File(*.PRJ)

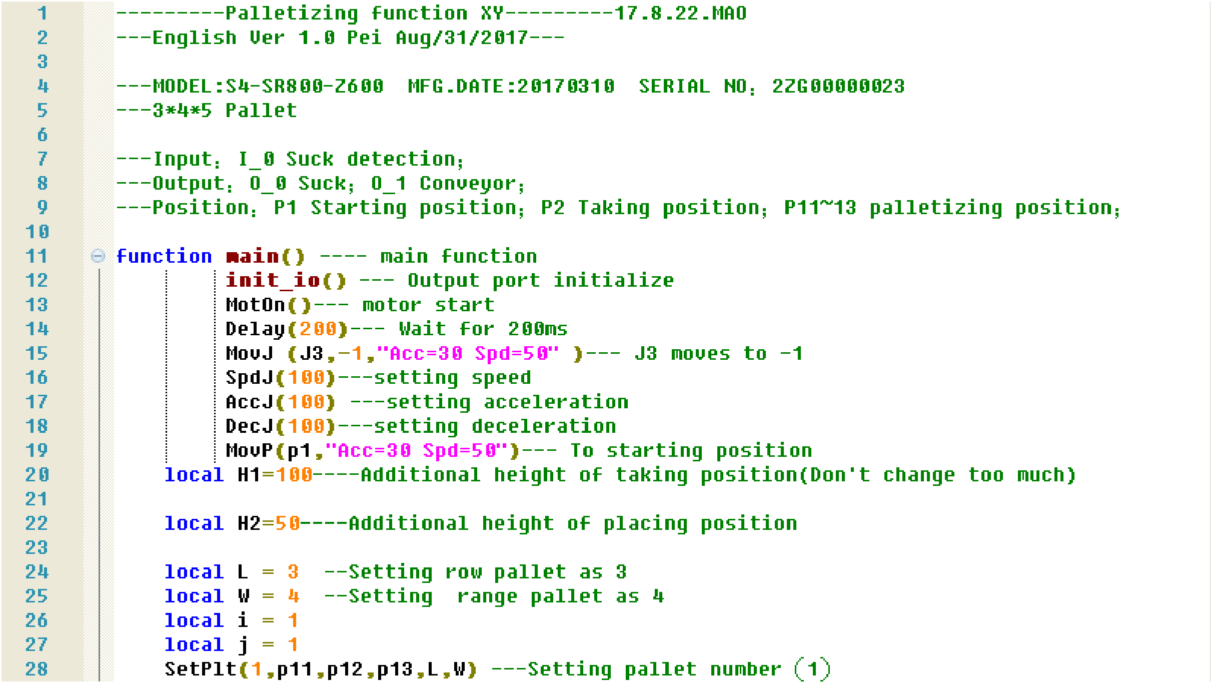
3.2 Palletizing

During the moving application sequence, some of the sequence would place the product in regulation, equal distance. Setting and teaching the points of product one by one would cause error and wasting time. The palletizing function would solve this problem. In the case of changing pallet and product the efficiency would be significant increasing. The graph is the example of pallet.

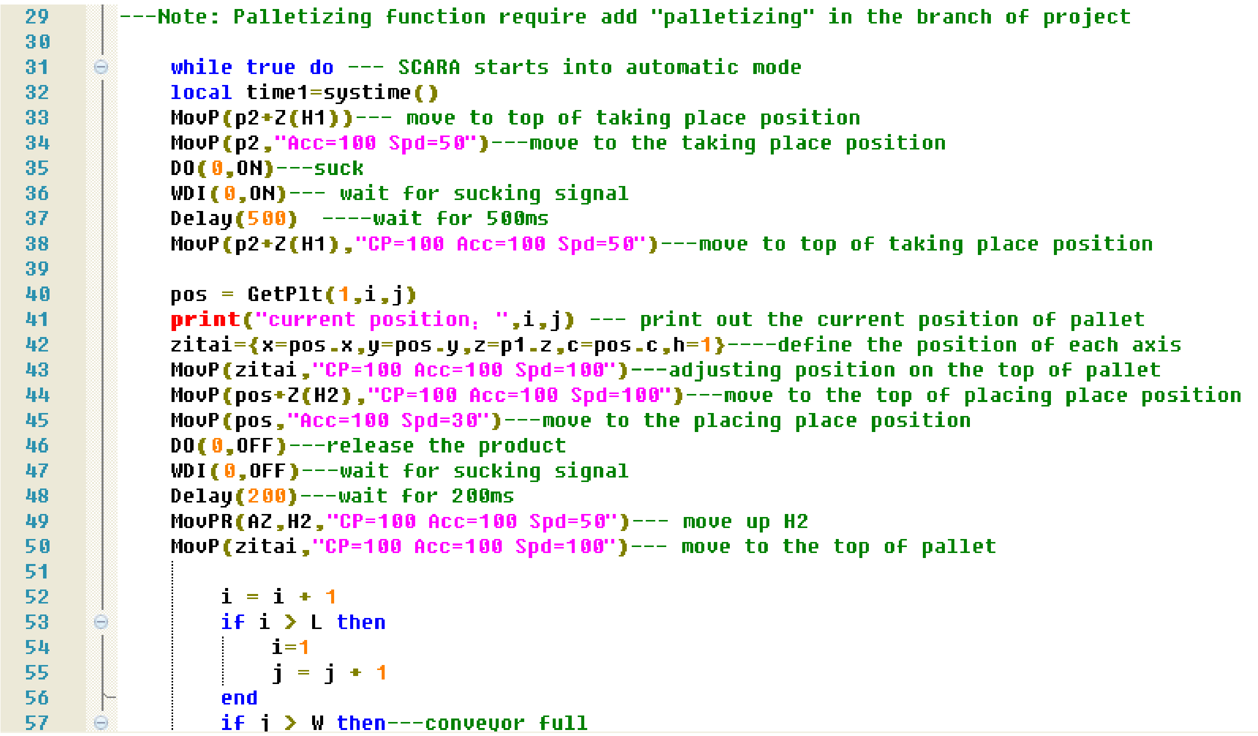


3 * 4 matrix pallet

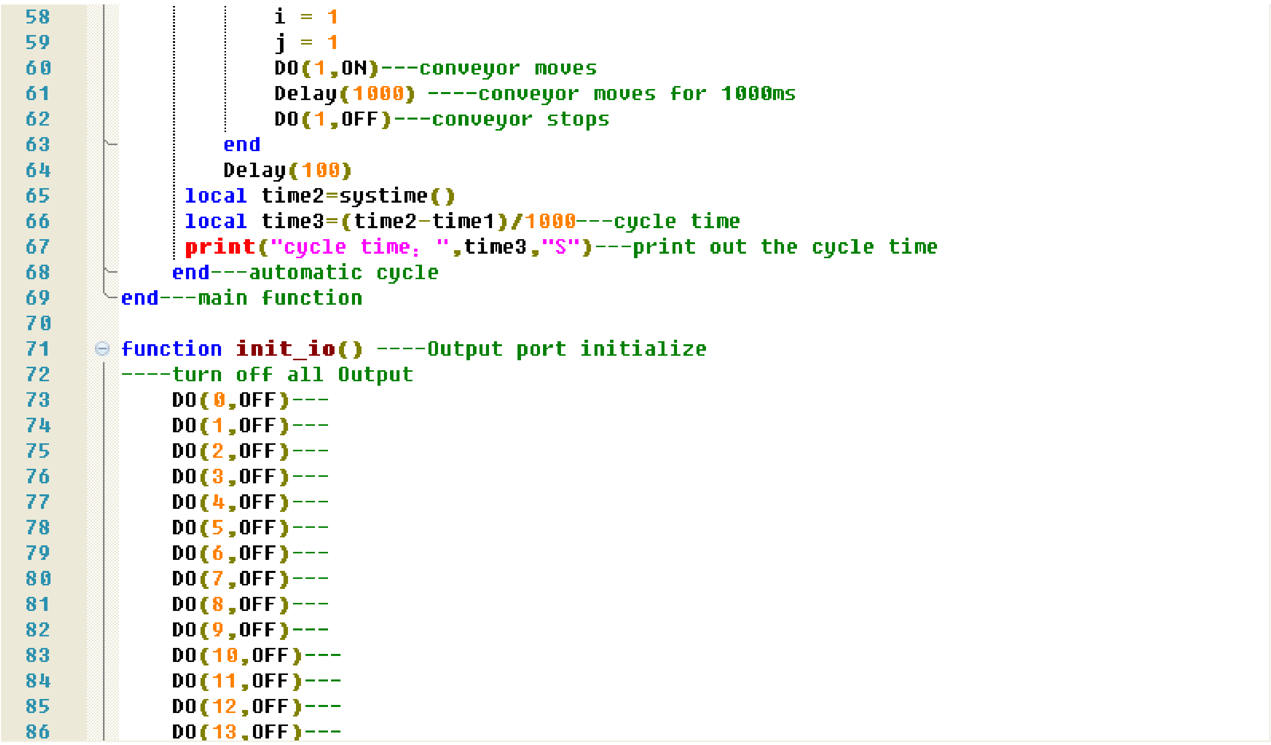
Palletizing function XY:



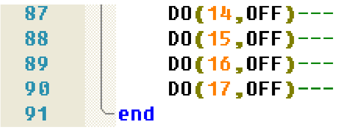
```
---------Palletizing function XY---------17.8.22.MAO
---English Ver 1.0 Pei Aug/31/2017---

---MODEL:S4-SR800-Z600  MFG.DATE:20170310  SERIAL NO： 2ZG00000023
---3*4*5 Pallet

---Input： I_0 Suck detection；
---Output： O_0 Suck； O_1 Conveyor；
---Position： P1 Starting position； P2 Taking position； P11~13 palletizing position；

function main() ---- main function
         init_io() --- Output port initialize
         MotOn()--- motor start
         Delay(200)--- Wait for 200ms
         MovJ (J3,-1,"Acc=30 Spd=50" )--- J3 moves to -1
         SpdJ(100)---setting speed
         AccJ(100) ---setting acceleration
         DecJ(100)---setting deceleration
         MovP(p1,"Acc=30 Spd=50")--- To starting position
    local H1=100----Additional height of taking position(Don't change too much)

    local H2=50----Additional height of placing position

    local L = 3  --Setting row pallet as 3
    local W = 4  --Setting  range pallet as 4
    local i = 1
    local j = 1
    SetPlt(1,p11,p12,p13,L,W) ---Setting pallet number（1）
```



```
---Note: Palletizing function require add "palletizing" in the branch of project

    while true do --- SCARA starts into automatic mode
    local time1=systime()
    MovP(p2+Z(H1))--- move to top of taking place position
    MovP(p2,"Acc=100 Spd=50")---move to the taking place position
    DO(0,ON)---suck
    WDI(0,ON)--- wait for sucking signal
    Delay(500)  ----wait for 500ms
    MovP(p2+Z(H1),"CP=100 Acc=100 Spd=50")---move to top of taking place position

    pos = GetPlt(1,i,j)
    print("current position： ",i,j) --- print out the current position of pallet
    zitai={x=pos.x,y=pos.y,z=p1.z,c=pos.c,h=1}----define the position of each axis
    MovP(zitai,"CP=100 Acc=100 Spd=100")---adjusting position on the top of pallet
    MovP(pos+Z(H2),"CP=100 Acc=100 Spd=100")---move to the top of placing place position
    MovP(pos,"Acc=100 Spd=30")---move to the placing place position
    DO(0,OFF)---release the product
    WDI(0,OFF)---wait for sucking signal
    Delay(200)---wait for 200ms
    MovPR(AZ,H2,"CP=100 Acc=100 Spd=50")--- move up H2
    MovP(zitai,"CP=100 Acc=100 Spd=100")--- move to the top of pallet

        i = i + 1
        if i > L then
            i=1
            j = j + 1
        end
        if j > W then---conveyor full
```

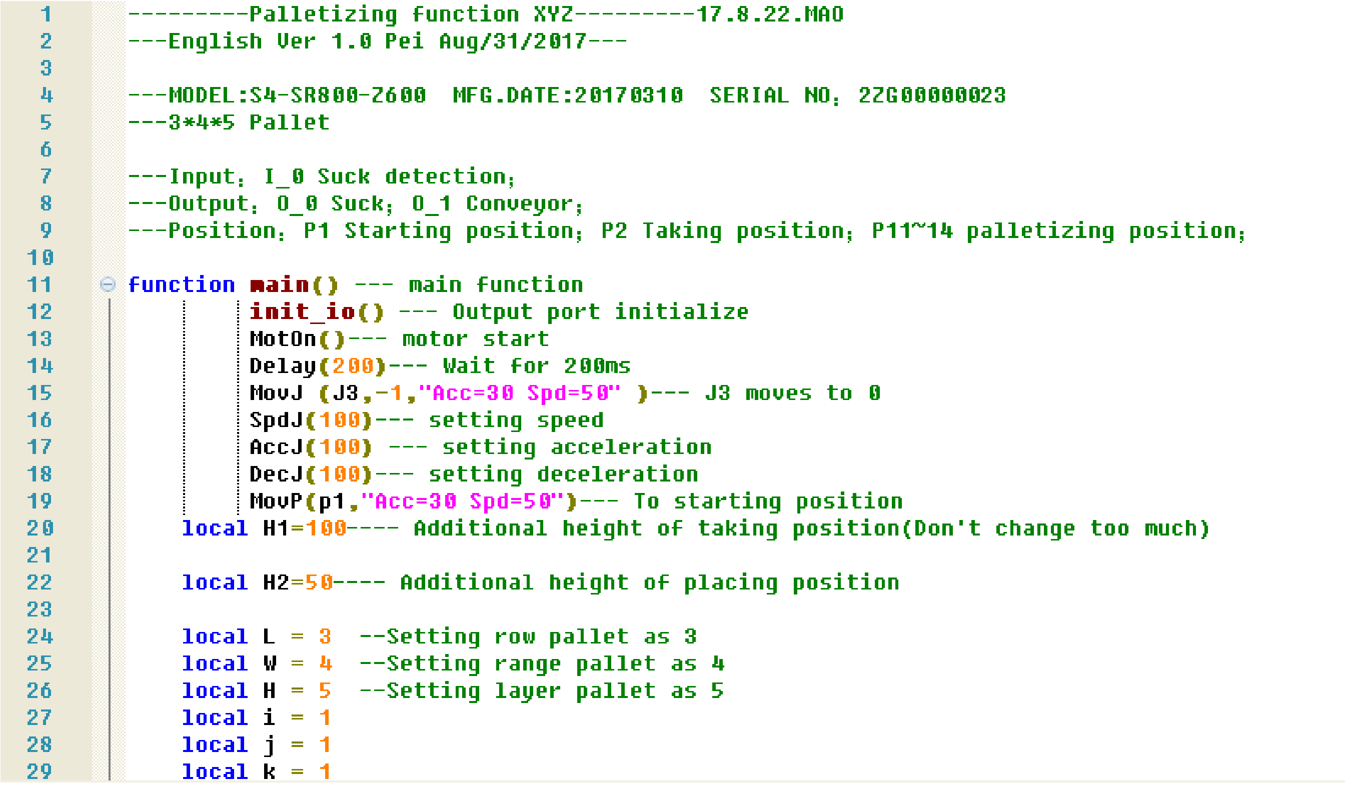


```
            i = 1
            j = 1
            DO(1,ON)---conveyor moves
            Delay(1000) ----conveyor moves for 1000ms
            DO(1,OFF)---conveyor stops
        end
        Delay(100)
      local time2=systime()
      local time3=(time2-time1)/1000---cycle time
      print("cycle time： ",time3,"S")---print out the cycle time
    end---automatic cycle
end---main function

function init_io() ----Output port initialize
----turn off all Output
    DO(0,OFF)---
    DO(1,OFF)---
    DO(2,OFF)---
    DO(3,OFF)---
    DO(4,OFF)---
    DO(5,OFF)---
    DO(6,OFF)---
    DO(7,OFF)---
    DO(8,OFF)---
    DO(9,OFF)---
    DO(10,OFF)---
    DO(11,OFF)---
    DO(12,OFF)---
    DO(13,OFF)---
```

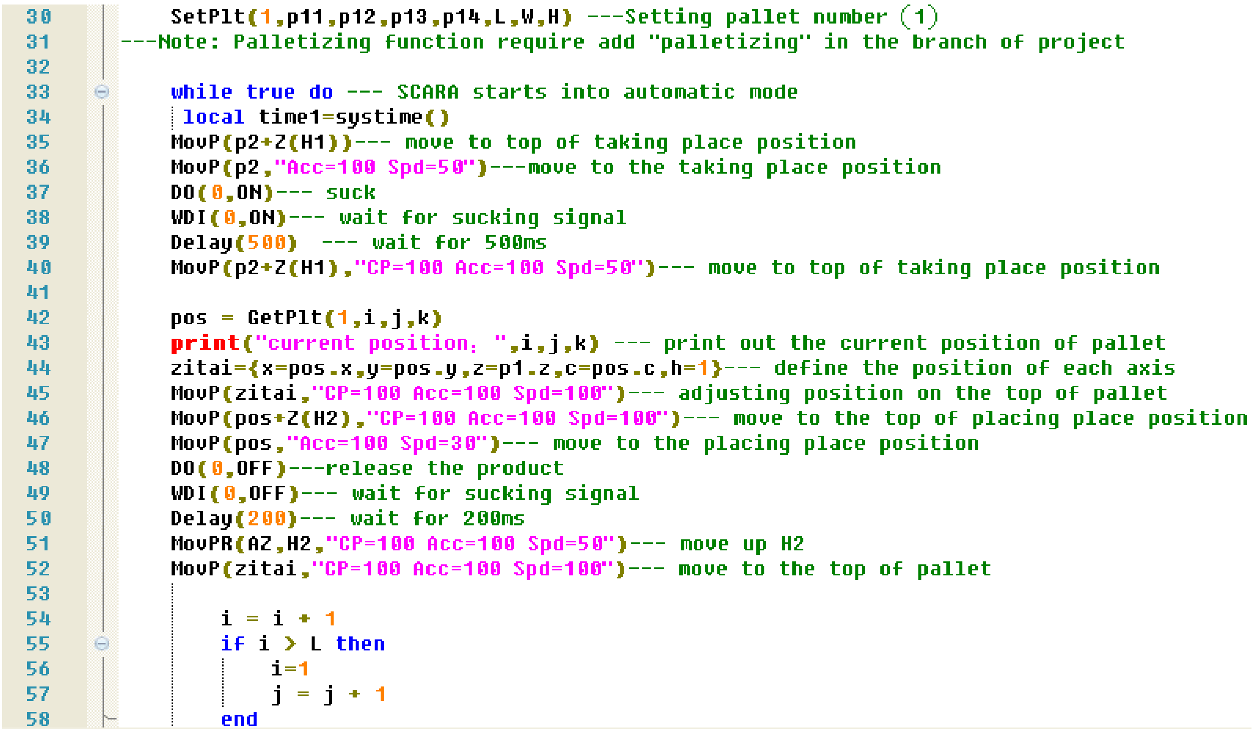


```
    DO(14,OFF)---
    DO(15,OFF)---
    DO(16,OFF)---
    DO(17,OFF)---
end
```

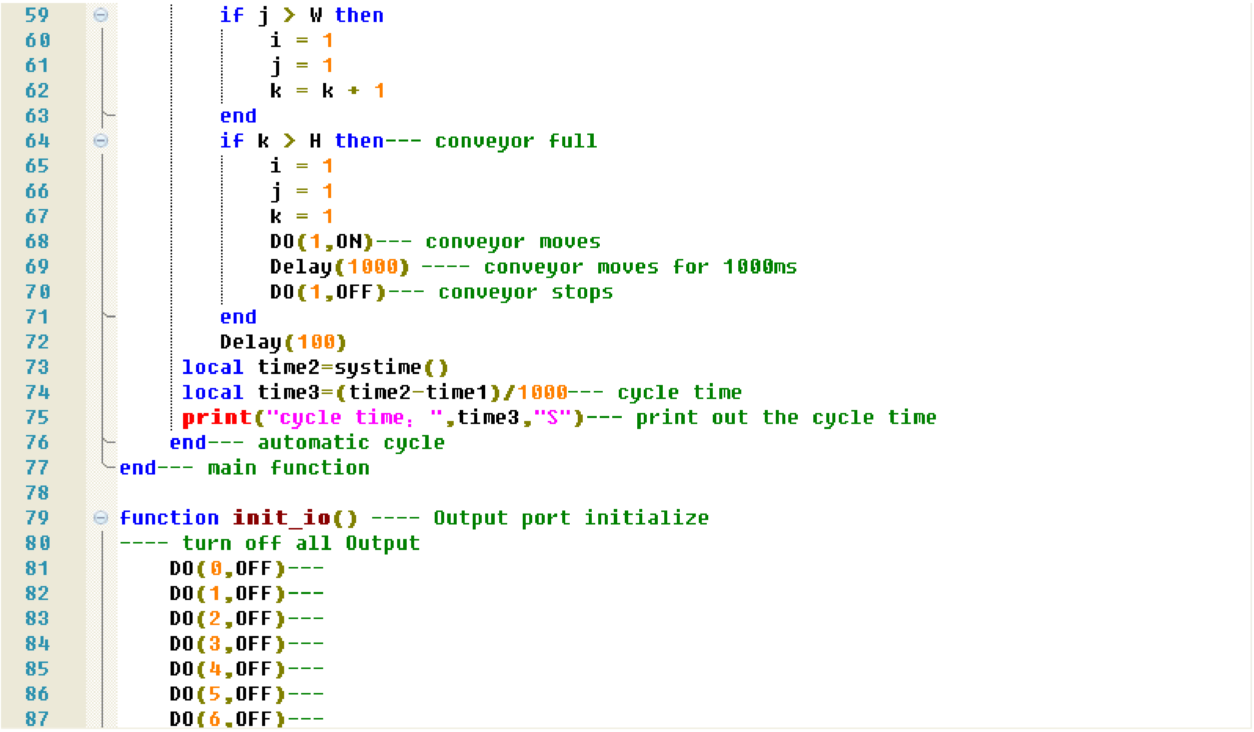
Palletizing function XYZ:



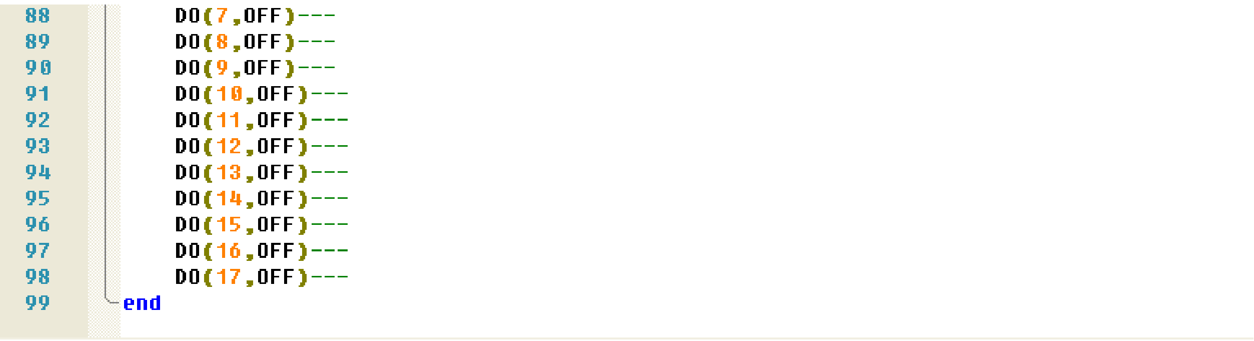
```
---------Palletizing function XYZ---------17.8.22.MAO
---English Ver 1.0 Pei Aug/31/2017---

---MODEL:S4-SR800-Z600  MFG.DATE:20170310  SERIAL NO：22G00000023
---3*4*5 Pallet

---Input： I_0 Suck detection；
---Output： O_0 Suck； O_1 Conveyor；
---Position： P1 Starting position； P2 Taking position； P11~14 palletizing position；

function main() --- main function
        init_io() --- Output port initialize
        MotOn()--- motor start
        Delay(200)--- Wait for 200ms
        MovJ (J3,-1,"Acc=30 Spd=50" )--- J3 moves to 0
        SpdJ(100)--- setting speed
        AccJ(100) --- setting acceleration
        DecJ(100)--- setting deceleration
        MovP(p1,"Acc=30 Spd=50")--- To starting position
    local H1=100---- Additional height of taking position(Don't change too much)

    local H2=50---- Additional height of placing position

    local L = 3  --Setting row pallet as 3
    local W = 4  --Setting range pallet as 4
    local H = 5  --Setting layer pallet as 5
    local i = 1
    local j = 1
    local k = 1
```



```
    SetPlt(1,p11,p12,p13,p14,L,W,H) ---Setting pallet number （1）
---Note: Palletizing function require add "palletizing" in the branch of project

    while true do --- SCARA starts into automatic mode
     local time1=systime()
    MovP(p2+Z(H1))--- move to top of taking place position
    MovP(p2,"Acc=100 Spd=50")---move to the taking place position
    DO(0,ON)--- suck
    WDI(0,ON)--- wait for sucking signal
    Delay(500)  --- wait for 500ms
    MovP(p2+Z(H1),"CP=100 Acc=100 Spd=50")--- move to top of taking place position

    pos = GetPlt(1,i,j,k)
    print("current position: ",i,j,k) --- print out the current position of pallet
    zitai={x=pos.x,y=pos.y,z=p1.z,c=pos.c,h=1}--- define the position of each axis
    MovP(zitai,"CP=100 Acc=100 Spd=100")--- adjusting position on the top of pallet
    MovP(pos+Z(H2),"CP=100 Acc=100 Spd=100")--- move to the top of placing place position
    MovP(pos,"Acc=100 Spd=30")--- move to the placing place position
    DO(0,OFF)---release the product
    WDI(0,OFF)--- wait for sucking signal
    Delay(200)--- wait for 200ms
    MovPR(AZ,H2,"CP=100 Acc=100 Spd=50")--- move up H2
    MovP(zitai,"CP=100 Acc=100 Spd=100")--- move to the top of pallet

        i = i + 1
        if i > L then
            i=1
            j = j + 1
        end
```



```
        if j > W then
            i = 1
            j = 1
            k = k + 1
        end
        if k > H then--- conveyor full
            i = 1
            j = 1
            k = 1
            DO(1,ON)--- conveyor moves
            Delay(1000) ---- conveyor moves for 1000ms
            DO(1,OFF)--- conveyor stops
        end
        Delay(100)
     local time2=systime()
     local time3=(time2-time1)/1000--- cycle time
     print("cycle time: ",time3,"S")--- print out the cycle time
    end--- automatic cycle
end--- main function

function init_io() ---- Output port initialize
---- turn off all Output
    DO(0,OFF)---
    DO(1,OFF)---
    DO(2,OFF)---
    DO(3,OFF)---
    DO(4,OFF)---
    DO(5,OFF)---
    DO(6,OFF)---
```



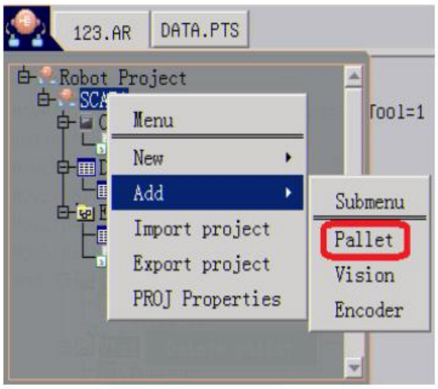
```
    DO(7,OFF)---
    DO(8,OFF)---
    DO(9,OFF)---
    DO(10,OFF)---
    DO(11,OFF)---
    DO(12,OFF)---
    DO(13,OFF)---
    DO(14,OFF)---
    DO(15,OFF)---
    DO(16,OFF)---
    DO(17,OFF)---
end
```

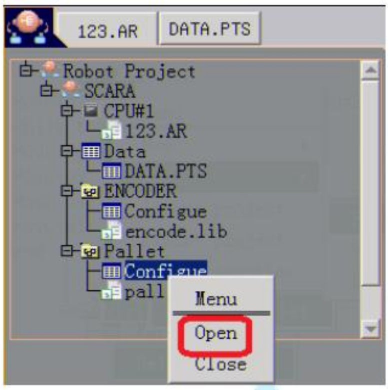
Tip:

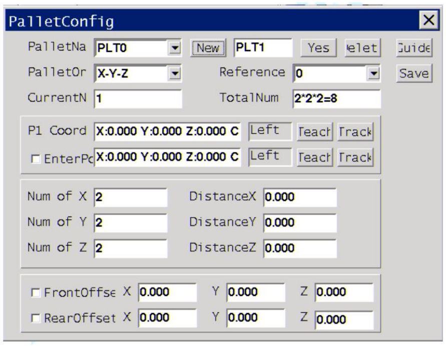
- palletizing function is proper to be used in plane XY, and it is also can be used in plane XYZ;
- The main command of palletizing are SetPlt and GetPlt; SetPlt is for setting a pallet, including pallet number, The palletizing origin point (p1), The row palletizing final point (p2),The range palletizing final point (p3), number of row, number of range;GetPlt is to know the position of each points, including pallet number (must have to match with SetPlt), postion of row, position of range.
- using palletizing command must have to add the database of pallet.

Demonstration of adding the database of pallet:

1) Press the existing project name (SCARA) to pop up a "Menu" list, and select "Add" to pop up a "Submenu" list, then select "Pallet";
2) Open palletizing configuration interface. Press "Configue" to pop up a "Menu", then select "Open" to enter "PalletConfig" interface;and then only click the "save" button.






PalletConfig

PalletNa PLT0 New PLT1 Yes elet Guide

PalletOr X-Y-Z Reference 0 Save

CurrentN 1 TotalNum 2*2*2=8

P1 Coord X:0.000 Y:0.000 Z:0.000 C Left Teach Track

□ EnterPc X:0.000 Y:0.000 Z:0.000 C Left Teach Track

Num of X 2 DistanceX 0.000

Num of Y 2 DistanceY 0.000

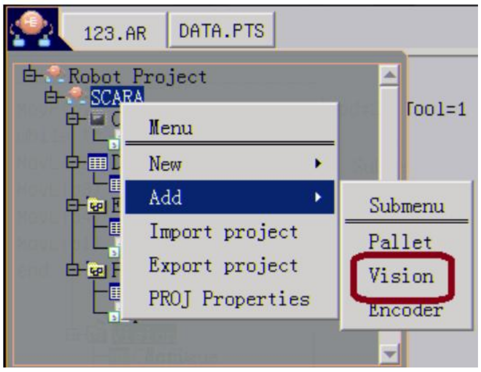
Num of Z 2 DistanceZ 0.000

□ FrontOffse X 0.000 Y 0.000 Z 0.000

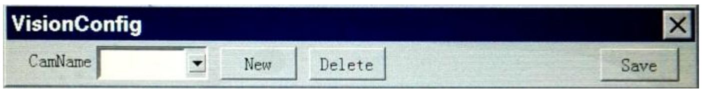
□ RearOffset X 0.000 Y 0.000 Z 0.000

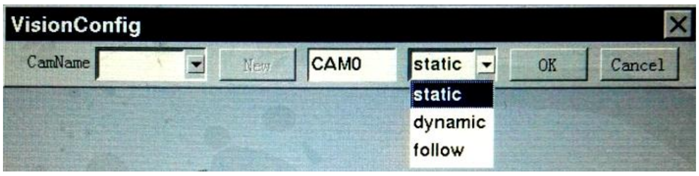
3.3 Vision

For Many occasions, robot is needed to communicate with visual device (cameras) to receive visual data sent from vision, and then operates the corresponding motion to complete the process requirements. ADT-400C controller can be used in three types of visual applications: static vision, dynamic visual and follow vision. Firstly, add the vision to the directory of robot project, and then open the visual configuration interface to configure some visual parameters required.

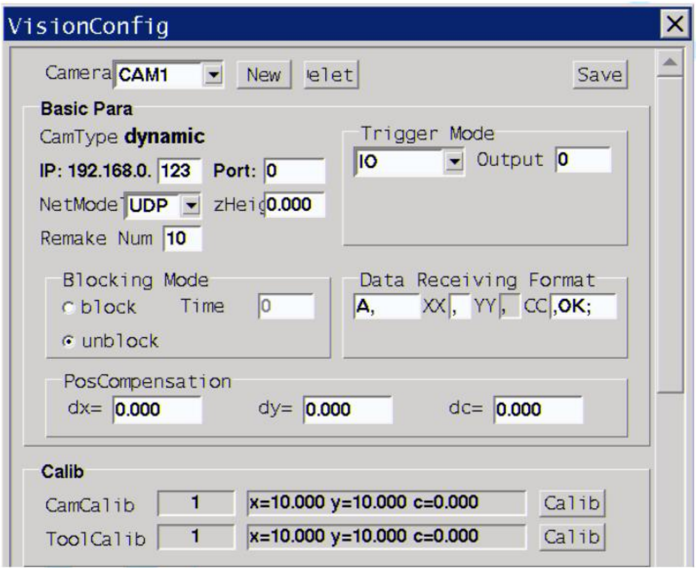


The parameter setting includes visual IP and Port, Trigger Mode (IO: hard trigger or Network: soft trigger), Blocking mode (block or unblock), Data Receiving Format and PosCompensation.





CAM0
static



Camera CAM1

Basic Para

CamType dynamic

IP: 192.168.0. 123 Port: 0

NetMode UDP zHei 0.000

Remake Num 10

Trigger Mode: IO Output 0

Blocking Mode: ○ block Time 0 ● unblock

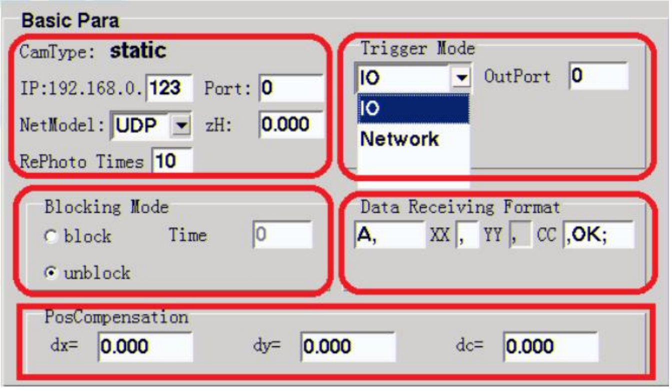
Data Receiving Format: A, XX , YY , CC ,OK;

PosCompensation: dx= 0.000 dy= 0.000 dc= 0.000

Calib

CamCalib 1 x=10.000 y=10.000 c=0.000

ToolCalib 1 x=10.000 y=10.000 c=0.000



Basic Para

CamType: static

IP:192.168.0. 123 Port: 0

NetModel: UDP zH: 0.000

RePhoto Times 10

Trigger Mode: IO OutPort 0

Blocking Mode: ○ block Time 0 ● unblock

Data Receiving Format: A, XX , YY , CC ,OK;

PosCompensation: dx= 0.000 dy= 0.000 dc= 0.000

NetModel: including UDP、 TCP_Client、 TCP_Server

- **UDP protocol suit:** must with the IP of visual system and port. Note: the IP of visual system have to be the same as the control panel 192.168.0.123; then the data could be monitored in communication station;
- **TCP_Client protocol suit:** the control panel is Client, visual system server, the IP of visual system also have to be the same as the control panel 192.168.0.123;But the data could not be monitored in communication station ;
- **TCP_Server protocol suit:** the control panel is Client, visual system server, the IP of visual system also have to be the same as the control panel 192.168.0.123; and the data can be monitored in communication station

RePhoto Times: When CCDrecv runs one time, the maximum number of photo re-taking (original setting:10) if any failure of receiving data or wrong format
Trigger Mode: Dynamic vision can be triggered by two ways, trigger by IO or by network

- **Trigger by IO:** based on connection port corresponds with the output port
- **Trigger by ethernet:** based on network transmitting format .(based on receiving format to match)
- **Trigger by distance:** used for application of followed vision, output port and area of taking photo are needed.
- **Trigger by input:** used for application of followed vision, input port is needed.
- **Trigger by distance and input:** used for application of followed vision output port, input port and area of taking photo are needed.
- **Blocking Mode:**Ethernet receiving data can be classified into two types: block and unblock. Normally, most of the cases are used in unblock mode; the block mode is used in a single task (unblock mode requires continuously scanning which occupy more resources and decrease the using rate of CPU.)
- **Block mechanism:** after sending the trigger signal, sequence during the period of blocking will remain in the function of receiving data, the AR sequence will continuously run until receiving the data from vision system.
- **Unblock mechanism:** after sending the trigger signal, no matter receiving the data from vision system or not, AR sequence will still run.
- **Data Receiving Format:** the date format sent by vision system can be classified into 4 types:
  Without starting and ending position: XX,YY,CC
  With starting but without ending position NO,XX,YY,CC
  Without starting but with ending position:XX,YY,CC,STA
  With starting and ending position

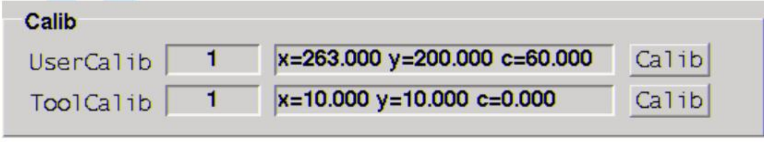
NO - starting position, STA - starting posotion XX/YY/CC means the axis of X/Y/C in vision system

Tips:

After setting the parameter must press the “save” button on the right up corner on the vision interface. For how to trigger could see the example on instruction as reference.

3.3.1 Static Vision

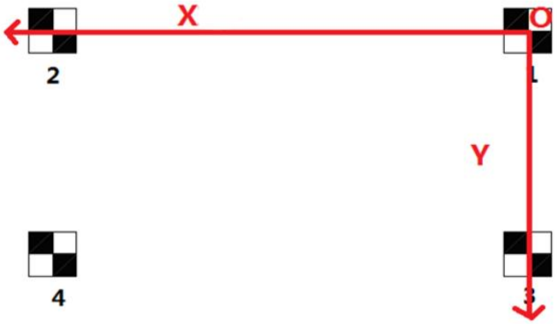
Static vision refers to a camera fixed at one place, which is triggered by IO or a soft command to take pictures and send data over the network to the controller. In the interface of visual configuration, click “New” to write the name of the camera (CAM0 ~ CAM9) and select the camera type “static”, then press “OK”. Next, you need to calibrate the user coordinate system tool coordinate system and set basic parameters of static visual.



Calib

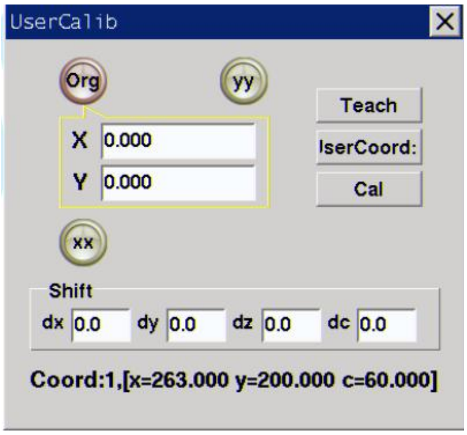
UserCalib	1	x=263.000 y=200.000 c=60.000	Calib
ToolCalib	1	x=10.000 y=10.000 c=0.000	Calib

1. User coordinate system calibration. Calibration purpose is to establish a relationship between the vision coordinate system and the robot coordinate system. Provided that the visual system has its own calibration a visual coordinates (XOY), as shown below, that is pixel coordinate conversion to metric units(mm)



Click “calibration” to enter the “UserCalib” interface:

1) Select the “Org”, then move the robot to the origin O of visual coordinates, then click “Teach”;
2) Select “xx”, then move the robot to a point in the direction of the X axis of the visual coordinates, then click “Teach”;
3) Select “yy”, then move the robot to a point in the direction of the Y axis of the visual coordinates, then click “Teach”;
4) Click “UserCrd:1” button to select a user number from 1 to 9, then press “OK”;
5) Click “Calcu”. Thus one user calibration has been completed.



X 0.000
Y 0.000

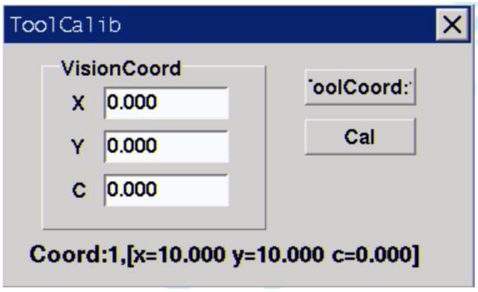
Shift
dx 0.0 dy 0.0 dz 0.0 dc 0.0

Coord:1,[x=263.000 y=200.000 c=60.000]

2. Tool coordinate calibration

Tool coordinates calibration is calibrated by means of visual coordinate. Notice that Descartes coordinate system must be switched to the same user which is set in user coordinate calibration.

1) Click “calibration” to enter the “ToolCalib” interface;
2) Create a visual recognition template, then manually write the visual coordinates X/Y/C of template to the corresponding “VisionCoord”;
3) Move the robot to ensure that the end of the tool to grab this template with appropriate position and attitude;
4) Click “ToolCrd:” to select a tool number from 1 to 9, then press “OK”;
5) Click “Cal”, then the tool calibration is completed.



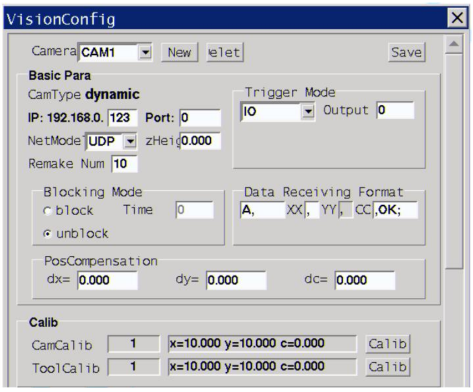
X 0.000
Y 0.000
C 0.000
Coord:1,[x=10.000 y=10.000 c=0.000]

Tips:

If a camera can return the coordinates which is referred to base coordinate of robot, the calibration of user coordinate can be ignored.

3.3.2 Dynamic Vision

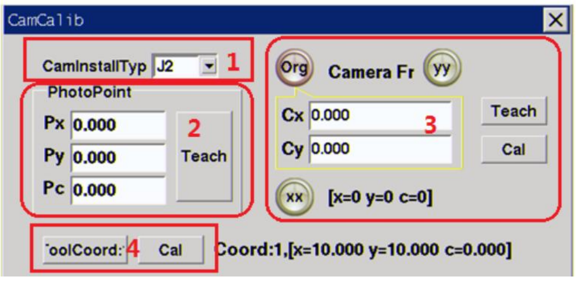
Dynamic camera is fixed on the robot arm (J2 or J4 axis), and which will move along with robot moving. So it is called as dynamic vision. In the visual configuration interface, click “new”, then write a camera name (CAM0~CAM9) and select the camera type “dynamic”, and then press “OK”. The configuration of dynamic vision includes camera-tool calibration (CamCalib), fixture-tool calibration (ToolCalib) and basic parameter configuration (Basic Para).



Camera CAM1
CamType dynamic
IP: 192.168.0. 123 Port: 0
NetMode UDP zHei 0.000
Remake Num 10
IO Output 0
○ block Time 0
● unblock
A, XX , YY , CC ,OK;
dx= 0.000 dy= 0.000 dc= 0.000
CamCalib 1 x=10.000 y=10.000 c=0.000
ToolCalib 1 x=10.000 y=10.000 c=0.000

1. Camera-tool calibration

Corresponding to the end of the robot, the dynamic camera acted as a tool, so a key step is to build a relationship between robot and the camera. Click “Calib” to open “CamCalib” interface.



CamInstallTyp J2

PhotoPoint

Px 0.000

Py 0.000

Pc 0.000

Cx 0.000

Cy 0.000

[x=0 y=0 c=0]

'oolCoord:

Coord:1,[x=10.000 y=10.000 c=0.000]

Step1: Select the type of camera installed(CamInstallType): J2 or J4 axis.

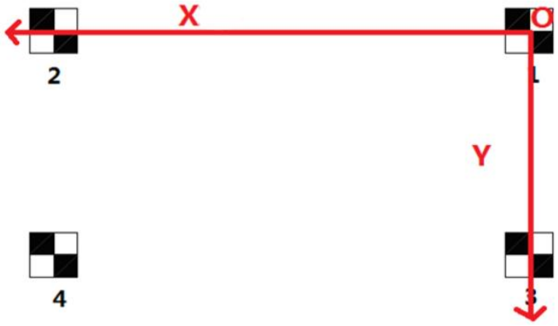
Step2: Fix the calibration paper, and move robot to obtain the photo point (PhotoPoint).

Please pay attentions:

1) Note that the position should not exceed the scope of the robot.

2) Click “teach” to assign current robot’s position to PhotoPoint.

Step3: Calibrate the relationship between the camera and the paper. Provided that the camera has set up a visual coordinate system XOY:



1) Select the “Org”, then move the robot to the origin O of visual coordinates, then click “Teach”;
2) Select “xx”, then move the robot to a point in the direction of the X axis of the visual coordinates, then click “Teach”;
3) Select “yy”, then move the robot to a point in the direction of the Y axis of the visual coordinates, then click “Teach”;

4) Calculate the camera tool

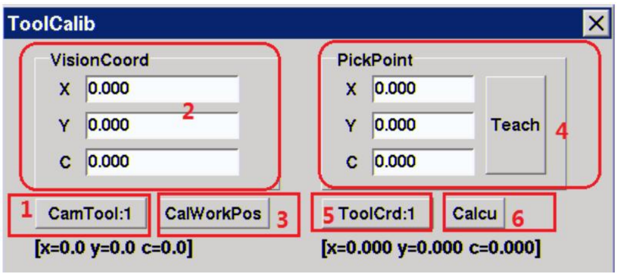
Step4: Click “ToolCoord: ” to select a tool number;

Step5: Press “Cal” to obtain the camera tool.

2. Fixture-tool calibration(ToolCalib)

1) Press “CalWorkPos” to calculate the piecework’s coordinate corresponding to robot’s base coordinate system;
2) Move robot to pick the piecework using the end of fixture which is installed at the end of the robot;

3) Click “Teach” to assign current robot’s coordinate to “PickPoint” point;
4) Press “ToolCrd: ” to select another tool which is different from “CamTool” tool;
5) Click “Calcu” to calculate the fixture tool.



ToolCalib

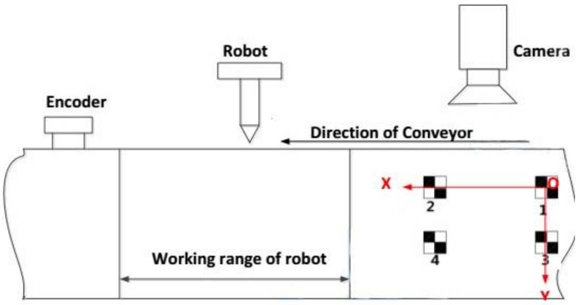
VisionCoord
X 0.000
Y 0.000
C 0.000

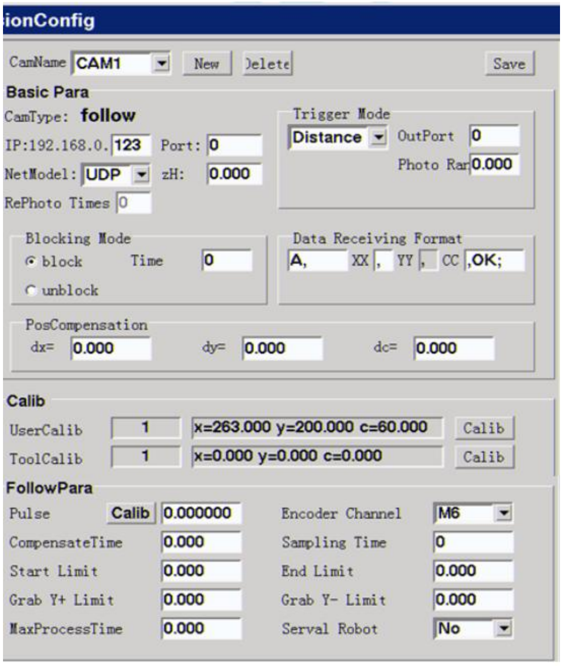
PickPoint
X 0.000
Y 0.000
C 0.000

[x=0.0 y=0.0 c=0.0]
[x=0.000 y=0.000 c=0.000]

3.3.3 Follow Vision

With the help of vision and encoder, follow vision is used for grasping piecework put on a moving belt conveyor. This application needs some external equipment: belt conveyor, encoder (fixed at conveyor), a camera, and needle. In the interface of visual configuration, click “New” to write the name of the camera (CAM0 ~ CAM9) and select the camera type “follow”, then press “OK”. Then, you need to calibrate the user coordinate system with the tool coordinate system, and set basic parameters of static visual and parameters of follow vision (FollowPara).





ionConfig

CamName: CAM1

Basic Para
CamType: follow
IP:192.168.0.123 Port: 0
NetModel: UDP zH: 0.000
RePhoto Times 0

Trigger Mode: Distance
OutPort 0
Photo Ran 0.000

Blocking Mode
● block Time 0
○ unblock

Data Receiving Format
A, XX , YY , CC ,OK;

PosCompensation
dx= 0.000 dy= 0.000 dc= 0.000

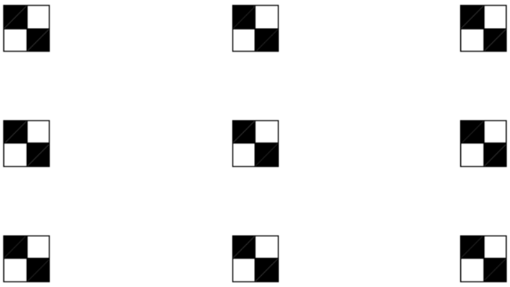
Calib
UserCalib 1 x=263.000 y=200.000 c=60.000 Calib
ToolCalib 1 x=0.000 y=0.000 c=0.000 Calib

FollowPara
Pulse Calib 0.000000
Encoder Channel M6
CompensateTime 0.000
Sampling Time 0
Start Limit 0.000
End Limit 0.000
Grab Y+ Limit 0.000
Grab Y- Limit 0.000
MaxProcessTime 0.000
Serval Robot No

3.3.4 Manual 9 points calibration

9 points calibration related to 3 points is more complicated but more precise. 9 points calibration is used in 2 circumstances: 1. The vision system can do 9 points calibration 2. The vision system cannot do 9 points calibration, in other words, the vision system can only provide pixel coordinate. The principle of 9 points calibration: apply 9 sets of pixel coordinate and 9 sets robot basic coordinate (or user coordinate) into a math model (based on the vision feature), obtain one on one relation between pixel coordinate and robot basic coordinate. 9 points calibration requires a 3*3 calibration plate (board) printed in 1:1.

Tips: 9 points in calibration board should cover the vision area as much as it can.



Situation 1: vision system accomplished 9 points calibration

- After 9 points calibration, vision data directly generate the coordinate (robot basic coordinate) and mark each steps:

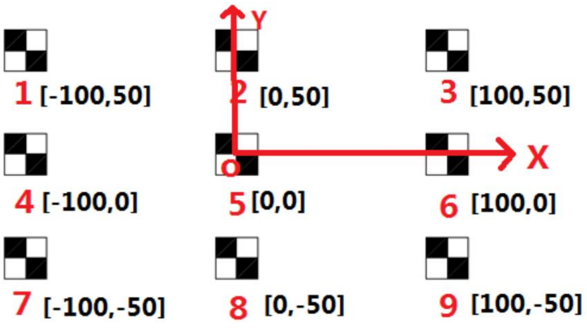
Step1: Fixing the calibration plate in the visible area and placing a calibration pin on the end of the robot.

Step2: Triggering the camera to taking photo, recording the 9 points pixel coordinate in vision system.

Step3: Moving the robot, using the end of the calibration pin to touch the 9 points on calibration plate and recording these measured 9 points as machine points.

Step4: Transferring the 9 measure machine points from step 3 to vision system to complete the 9 point calibration, the pixel coordinate will automatically turn into robot basic coordinate.

- 9 points calibration, a given coordinate under the specified coordinate system (OXY) from vision system and steps are shown in the following figure



Step1: Fixing the calibration plate in the visible area and placing a calibration pin on the end of the robot.

Step2: Triggering the camera to take photo, recording the 9 points pixel coordinate in vision system.

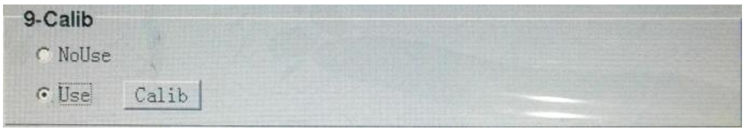
Step3: Specifying a vertical coordinate system, EX: specifying point 5 on calibration plate as the original point, point 6 as a point on positive direction of X axis, point 2 as a point on positive direction of Yaxis, as a result the position of 9 points on the OXY coordinate system is confirmed. (because the calibration plate is printed 1:1, the distance from each point are known.)

Step4: Enter the user coordinate values correspond to 9 points pixel coordinate in vision system to accomplish the 9 points calibration. But this 9 points calibration is built on connection between pixel coordinate system and OXY coordinate system, the connection between OXY coordinate and machine is still needed (only need to built the user coordinate)

Step5: Building an user coordinate in user interface. The 3 points in user coordinate are point 5, point 6, point 2. The method of coordinate calibration could be found in chapter 2.8.1.

Situation 2: the vision system only output pixel coordinate.

Similar to situation 1, algorithm of 9 points calibration completed in RC400 control panel system, the 9 points pixel coordinate on calibration plate needed to be enter in robot control panel, the calibration method should complete in vision interface.

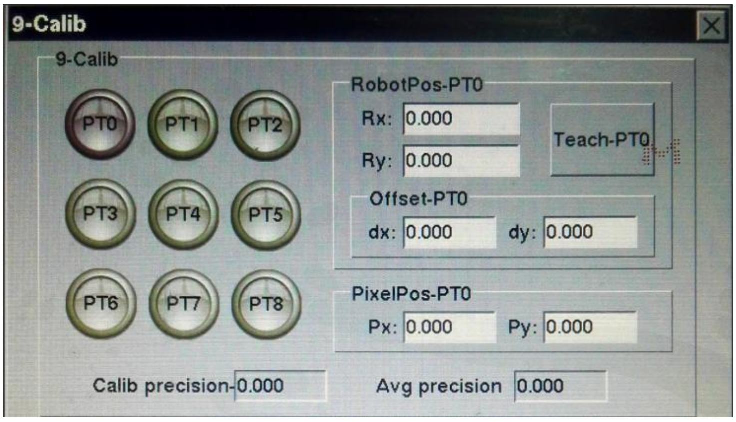


9-Calib

○ NoUse

● Use

In vision interface, pick “use” then “calibration” to enter 9 points calibration interface.



9-Calib

9-Calib

RobotPos-PT0

Rx: 0.000

Ry: 0.000

Offset-PT0

dx: 0.000 dy: 0.000

PixelPos-PT0

Px: 0.000 Py: 0.000

Calib precision-0.000 Avg precision 0.000

Calibration step:

Step1: Gixing the calibration plate in the visible area and placing a calibration pin on the end of the robot.

Step2: Triggering the camera to take photo, manually enter the 9 points pixel coordinate on calibration plate to the point 1 - point 9 on the 9 points calibration interface.

Step3: Moving the robot and using the end of calibration pin to touch the 9 points on calibration plate, then click :”teach” so the 9 points (on calibration plate) of robot coordinate will be recorded on the specified area point 1 - 9 which marked by robot.

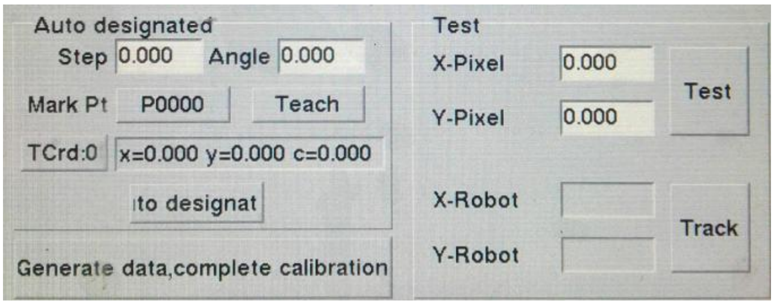
Step4: Click “generate date, complete calibration”

Step5: Close the 9 points calibration interface, click “save” the button the visual configuration interface on the right up corner.

Tips:

- The point 1 - 9 in robot marking area and the point 1 - 9 in camera specifying area must match.
- 16 points calibration could be taken reference from 9 points calibration method(same method)

3.3.5 Automatic 9/16 points calibration



Step 0.000 Angle 0.000

Mark Pt P0000

TCrd:0 x=0.000 y=0.000 c=0.000

X-Pixel 0.000

Y-Pixel 0.000

X-Robot

Y-Robot

Step1: Place the calibration plate (or object) to the proper position in the visible area and make it fixed.

Step2: Creating the calibration plate to verify the object of vision interface and switch to the automatic move.

Step3: Enter length in “step” (mm) and “angle” in “auto designated”

- Step: the robot moving length each time. In other word, the direct trace distance of generating 9 or 16 points.
- Angle: the rotation angle of robot.

Note: setting the length of step must make sense to insure the taking picture of mark point in the area of 9 or 16 points after rotation, or the error will be very big.

Step4: make sure the robot starting position of taking picture, pick a proper position (p1 -p999) and click “teach” as mark point.

Step5: Click “TCrd” and pick tool 1 - 9 to save the camera tool coordinate.

Step6: Switch the key on control panel to automatic gear (A) and click auto designed until “9 calibration points complete” pops up and click “ok” to close it.

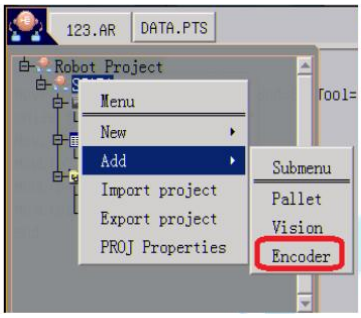
Step7: Click “generate data, complete calibration” to complete the automatic 9 points calibration. The “test” button can be used for checking the accuracy of calibration

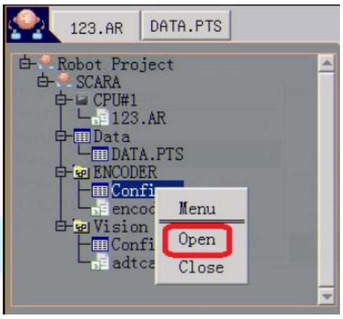
Step8: Close the 9 points calibration interface, click “save” the button the visual configuration interface on the right up corner.

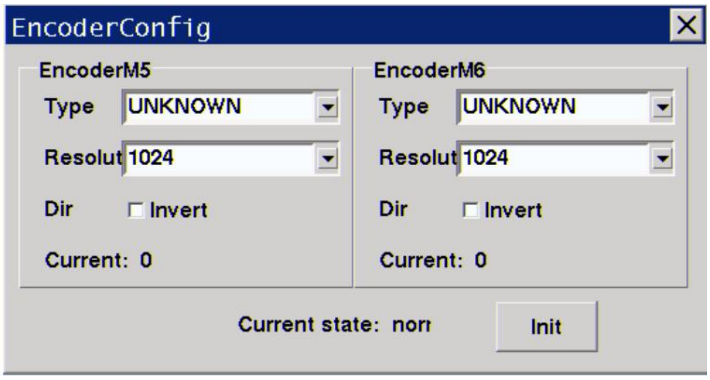
3.4 Encoder

An encoder is an indispensable external device for follow grasping process, which can give real-time feedback of the distance of the object on the conveyor belt:

1. Long press the current project name, pop-up menu interface, click “add”;
2. Select “Encoder” in the pop-up “Submenu” interface, so an encoder device has added to the current project;
3. Long press “Configue” in “ENCODER” list, then click “Open” to enter “EncoderConfig” interface;
4. In the “EncoderConfig” interface, set the type and resolution of the encoder connected to controller (M5 or M6 port of encoder).






EncoderConfig

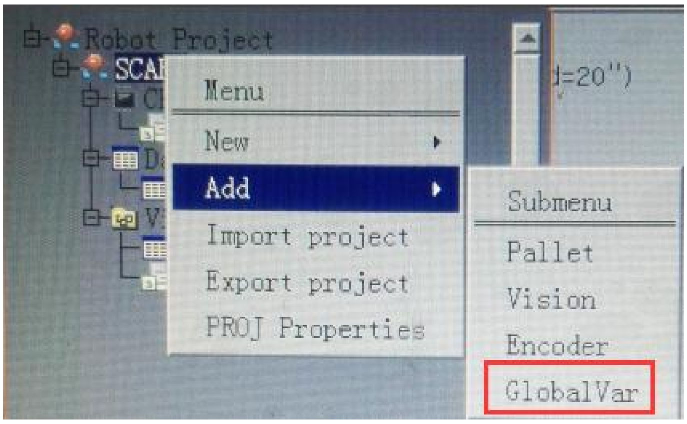
EncoderM5
Type UNKNOWN
Resolut 1024
Dir □ Invert
Current: 0

EncoderM6
Type UNKNOWN
Resolut 1024
Dir □ Invert
Current: 0

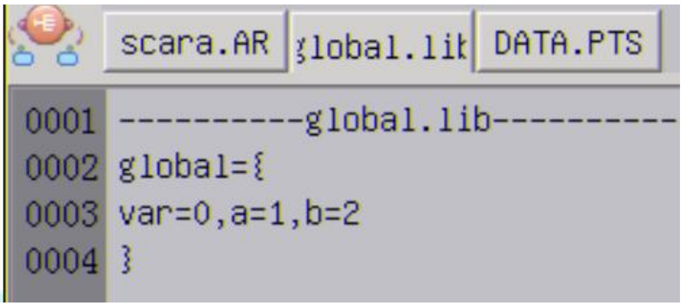
Current state: norr

3.5 GlobalVar

Based on the realistic application, 1 project can include many CPU. Global.lib can solve the problem of multiple CPU sharing the same variety. Follow the demonstration and create “global.lib”.



Add the shared variety in global.lib



```
----------global.lib----------
global={
var=0,a=1,b=2
}
```

Tips:

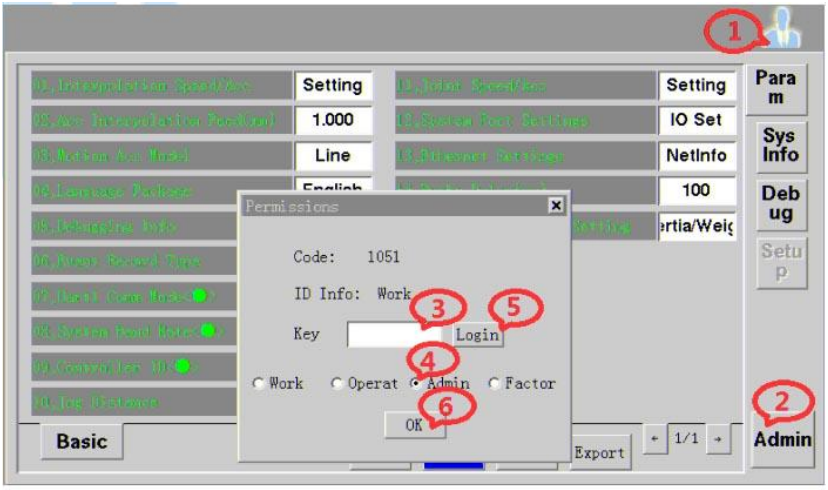
- The changing method of shared variety in CPU
- Shared variety cannot be inserted array

4. Installation Wizard

Light administrator icon “👤 ” to enter the 【 Param】/【 SysInfo】/【 Debug】/【 Setup】 interface. At this moment, the color of “Setup” icon is gray Setup ; then it is required to obtain the login authority. Four levels are included: Worker/Operator/Admin/Factor, in which worker has lowest authority and factor has highest authority. For different levels, it has different tasks:

- Worker can only operate some icons on the flexpedant, and cannot modify any parameters;
- Operator has authority to modify some parameters in【 Param】 list. However, they have no authority to upgrade program and modify parameters in 【 Setup】;
- Admistrator(Admin) have authority to do any operation if it is possible.

From lower authority to higher authority, three cases are included:
Case1: Work to Factory/Operator to Factory/Admin to Factory
[Key]: 1101(Hex) is 4353(Decimal), then do [or operation] with current code (Decimal); operation result is key;
Case2: Work to Admin/Operator to Admin
[Key]: ********（Contact SHINI in case of immergency）
Case3: Work to Operator
[Key]: ********（Contact SHINI in case of immergency）



01.Interpolation Speed/Acc	Setting	11.Joint Speed/Acc	Setting
02.Acc Interpolation Pass(mm)	1.000	12.System Port Settings	IO Set
03.Motion Acc Model	Line	13.Ethernet Settings	NetInfo
04.Language Package	English		100
05.Debugging Info		Setting	ertia/Weig
06.Alarm Record Type			
07.Hand Comm Mode ●			
08.System Baud Rate ●			
09.Controller ID ●			
10.Jog Distance			

Permissions

Code: 1051

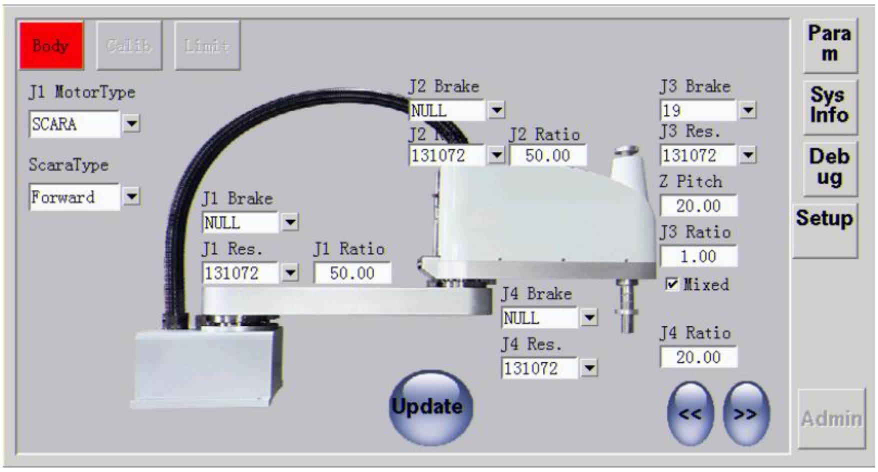
ID Info: Work

Key

○ Work ○ Operat ● Admin ○ Factor

4.1 Robot Body

The robot installation involves four axis brake output port, encoder resolution, reduction ratio, third axis' pitch, and whether third axis and four axis pitches being compound. Click "Update" button to complete the configuration.



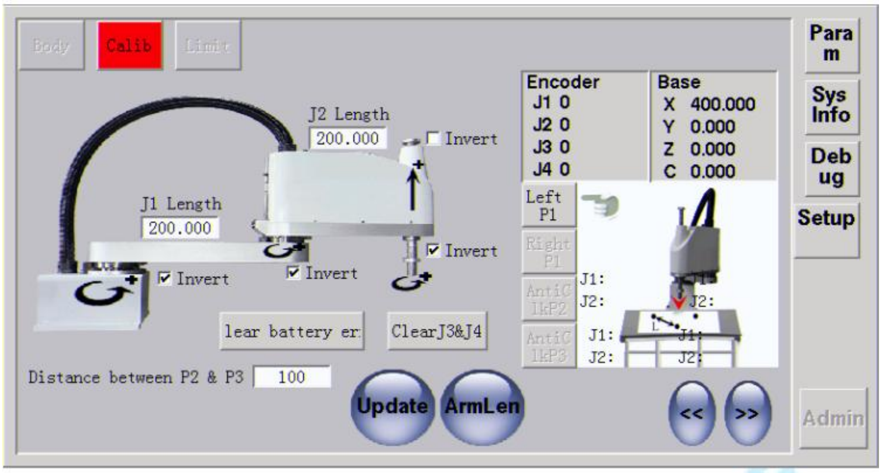
J1 MotorType: SCARA
ScaraType: Forward
J1 Brake: NULL
J1 Res.: 131072
J1 Ratio: 50.00
J2 Brake: NULL
J2 R: 131072
J2 Ratio: 50.00
J3 Brake: 19
J3 Res.: 131072
Z Pitch: 20.00
J3 Ratio: 1.00
☑ Mixed
J4 Brake: NULL
J4 Res.: 131072
J4 Ratio: 20.00

Tips:

- The resolution of the encoder, the reduction ratio of the reducer and the pitch of Z-axis should be written according to the actual motor situation;
  The brake is connected to the Relay1~Relay4 which are respectively corresponding to
- output ports OUT23~OUT26;
- Click the "Update" button to complete update of above parameters;
- Through the left and right buttons "<< >>" to switch the installation of the three interface.

4.2 Calibration

In calibration interface, it contains to calibrate arms' length of J1&J2 axes, direction configuration of the four axes and clearing encoder.



J2 Length: 200.000
□ Invert
J1 Length: 200.000
☑ Invert
☑ Invert
☑ Invert
Distance between P2 & P3: 100

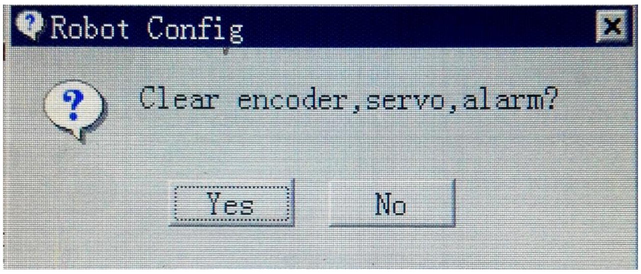
Encoder	Base
J1 0	X 400.000
J2 0	Y 0.000
J3 0	Z 0.000
J4 0	C 0.000

J1:
J2:
J1:
J2:

1) The direction setting principle is: for the rotating shaft, anti-clockwise movement is positive direction and clockwise movement is negative direction; For the upper and lower axes, upward movement is positive, downward movement is negative;

2) Encoder cleared in two cases:

a) Four axes: J1 and J2 at the same time cleared in a straight case, according to the “encoder” pop-up "Robot Config" dialog box, click the “Yes”. If the current encoder coordinates all 0, clear success. Encoder application: if the origin is lost, can be used for rough calibration the origin.



Clear encoder,servo,alarm?

b) Only J3&J4: if the J3 or J4 soft limit exceeded the alarm, you simply need to clear the J3&J4 encoder. By clicking “Clear J3&J4” button to complete the operation.

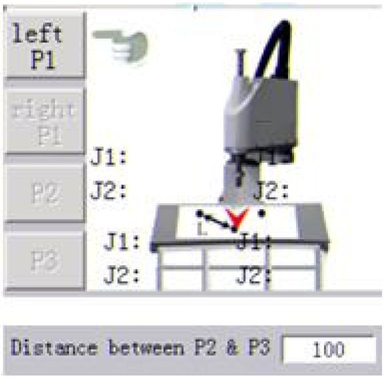
3) Upper/lower arm length and origin calibration:

Calibration Steps:

1. Prepare an paper with equilateral triangle(P1/P2/P3), which installed within range of the robot;
2. Robot changed into light drag model;
3. Move robot to P1 with left hand, then press “ left P1” ;
4. Move robot to P1 with right hand, then press “ right P1” ;
5. Manually write the distance between P2&P3;
6. Move the robot to P2, then press “ P2” ;

7.Move the robot to P3,then press “ P3” ;

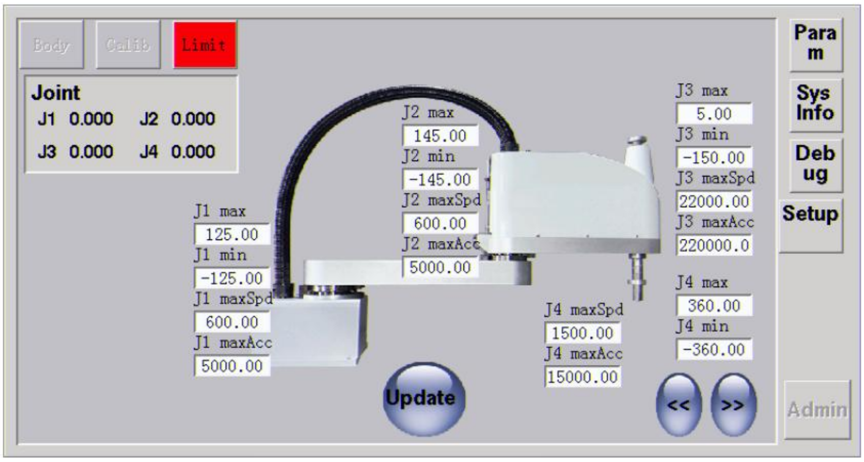
8. click “ ArmLen” button to calculate arm length;
9. At last, click “ Update” to finish calibration



Distance between P2 & P3 100

4.3 Limit Setting

In this interface, positive and negative of S J1&J2&J3&J4 axes, maximal speed and acceleration of each axis are set. After setting, click on “Update” to finish configuration.



Joint

J1 0.000 J2 0.000

J3 0.000 J4 0.000

J1 max 125.00

J1 min -125.00

J1 maxSpd 600.00

J1 maxAcc 5000.00

J2 max 145.00

J2 min -145.00

J2 maxSpd 600.00

J2 maxAcc 5000.00

J3 max 5.00

J3 min -150.00

J3 maxSpd 22000.00

J3 maxAcc 220000.0

J4 max 360.00

J4 min -360.00

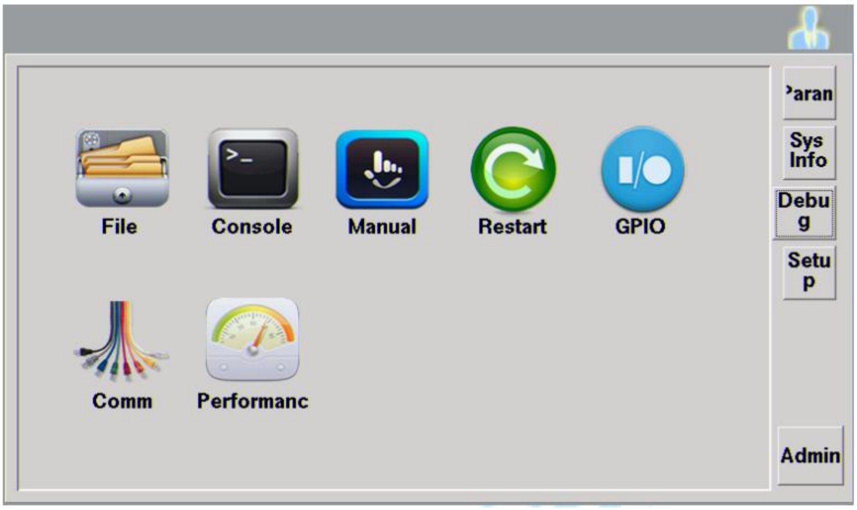
J4 maxSpd 1500.00

J4 maxAcc 15000.00

Tips:

- The maximum speed of each joint could take the reference: Maximum speed *360 / (60 * decreasing rate)
- Set the maximum accelerating speed of each joint as 8 -10 times of maximum speed.
- The original setting of axis and state is on. If the robot is only with X/Y/Z axis, the J4 axis could be turnoff. If the robot is only with X/Y/C axis, the J3 axis could be turn off

5. Debugging Tools



Debugging tools are some auxiliary tools in the robot debugging process, including file management console, manual debugging, GPIO, communication station, and the performance of the system. Then some tools and tips application will be introduced which can be help in debugging process.

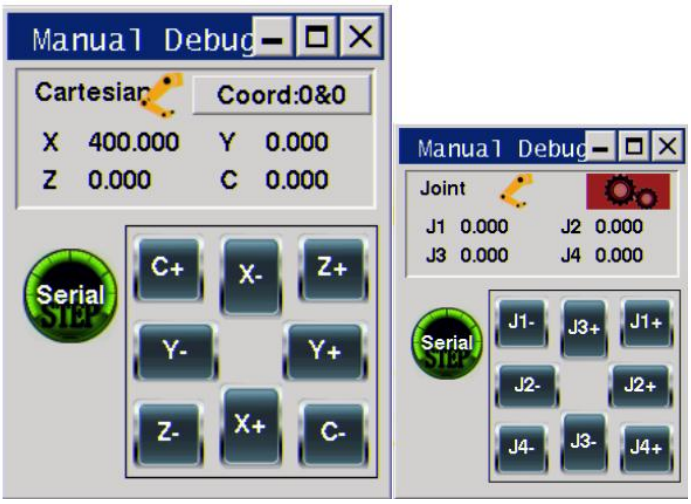
5.1 File Management Tool

The task of document management is responsible for the import and export procedures. If you need to insert the U disk operation, click "Load U disk" to load U disk.

Tips:

- The file management task is responsible for the import and export procedures;
- File management can only be used in manual mode;

5.2 Manual Debugging Tool



X 400.000 Y 0.000
Z 0.000 C 0.000

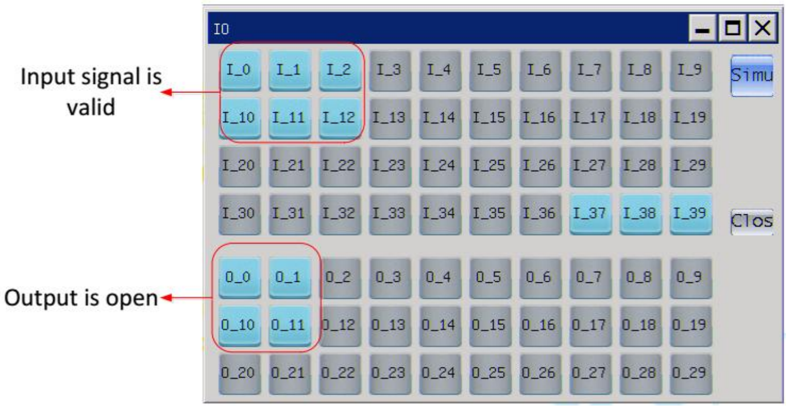
J1 0.000 J2 0.000
J3 0.000 J4 0.000

Tips:

- Call this tool in any interface by clicking "F2" button to view the current joint coordinates and Descartes coordinates, such as, you want to see whether the actual position is consistent with the position taught in AR program when debug in single step.

- By clicking the coordinates of the displayed area to switch the “Cartesian and Joint” coordinates, or you can also choose the corresponding coordinate system, or small gear,
- to switch to the actual and virtual location of the display.
- In manual mode, the tool will be pop up with operational button to move robot.

5.3 GPIO Tool


Input signal is valid
Output is open
IO
I_0 I_1 I_2 I_3 I_4 I_5 I_6 I_7 I_8 I_9
I_10 I_11 I_12 I_13 I_14 I_15 I_16 I_17 I_18 I_19
I_20 I_21 I_22 I_23 I_24 I_25 I_26 I_27 I_28 I_29
I_30 I_31 I_32 I_33 I_34 I_35 I_36 I_37 I_38 I_39
O_0 O_1 O_2 O_3 O_4 O_5 O_6 O_7 O_8 O_9
O_10 O_11 O_12 O_13 O_14 O_15 O_16 O_17 O_18 O_19
O_20 O_21 O_22 O_23 O_24 O_25 O_26 O_27 O_28 O_29
Simu
Clos

1) I_0~I_27 match the input signal 0-27 on input board;

2) I_28~I_33 represent correspond to input signal 28-66 of 16 wires IO in Heavy load connector.

3) O_0~O_17 match the output signal 0-17 on output board;

4) R9, R18 represent correspond relay SVST_A,SVST_B and EMSST_A,EMSST_B.

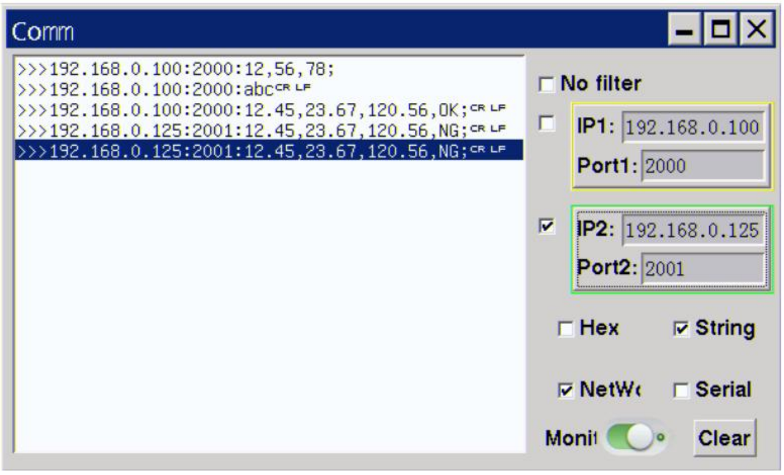
5) O_19~O_22 represent correspond output signal 19-22 of 16 wires in Heavy load connector.

6) O_23~O_26 represent correspond relay (23-26) of 16 wires in Heavy load connector

Operating tips:

- ”F6” button can be use for monitoring the I/O situation in any circumstance.the blue button represents the input or output port in on, the gray represents off;
- GPIO tool ball has three functions/output/monitoring/simulation;
- Output: click “Open”, you can see the status of each output point, and in the manual mode, you can also manually output;
- Monitor: you can see the input / output of the real state, in the automatic mode of security, can only see cannot be modified;
- Simulation: click the "Simu" button, you can interface to the input port of different states, so that you cannot take the real IO signal can complete the program debugging.

5.4 Communication Station Tool



```
>>>192.168.0.100:2000:12,56,78;
>>>192.168.0.100:2000:abcCR LF
>>>192.168.0.100:2000:12.45,23.67,120.56,OK;CR LF
>>>192.168.0.125:2001:12.45,23.67,120.56,NG;CR LF
>>>192.168.0.125:2001:12.45,23.67,120.56,NG;CR LF
```

□ No filter

□ IP1: 192.168.0.100 Port1: 2000

☑ IP2: 192.168.0.125 Port2: 2001

□ Hex ☑ String

☑ NetWo □ Serial

Operating tips:

- ”F5” button can be use for monitoring the connection situation in any circumstance.
- “Comm” is used to monitor whether network/serial communication is connected successfully. And It can be used for judge whether the received data is normal or not;
- For the head of the monitoring data, “<<<” represents output, ">>>"means input;
- You can configure the display format for the string or Hex by ticking the appropriate selection;
- Can choose the “No filter" / "IP1, IP2” the way to capture the data after the screening.
- This application is mainly used in the background when many communication equipment, we need to observe a certain IP data, this time need to use the IP filter function. The operation method is also very simple, select the corresponding IP data, and then click the check IP, screening conditions will be automatically set up.

6. System Information

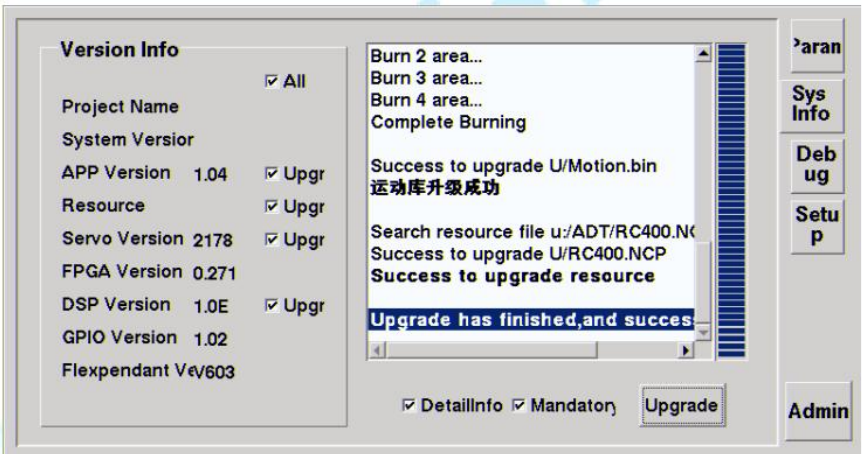
The system information is the software version information of each function module of the display system. Click on “👤” figure, then press “SysInfo” button to enter upgrading interface of system. Specific upgrading steps are as follows:

1) Upgrading operations must be carried out in manual mode;

2) Put application program (ADTROM.BIN), DSP program (MOTION.BIN), servo program (SERVO.BIN) and resource package (RC400.NCP) in ADT file of U disk;

3) The U disk is inserted into the USB port on the bottom of the teach pendant or the MEM port of RC400 controller;

4) Tick “All” (also check one of them), “Detail Info” and “Mandatory” in “Version Info” interface. It will take 3 minutes to complete upgrade after clicking “Upgrade”.



Version Info

☑ All

Project Name

System Versior

APP Version 1.04 ☑ Upgr

Resource ☑ Upgr

Servo Version 2178 ☑ Upgr

FPGA Version 0.271

DSP Version 1.0E ☑ Upgr

GPIO Version 1.02

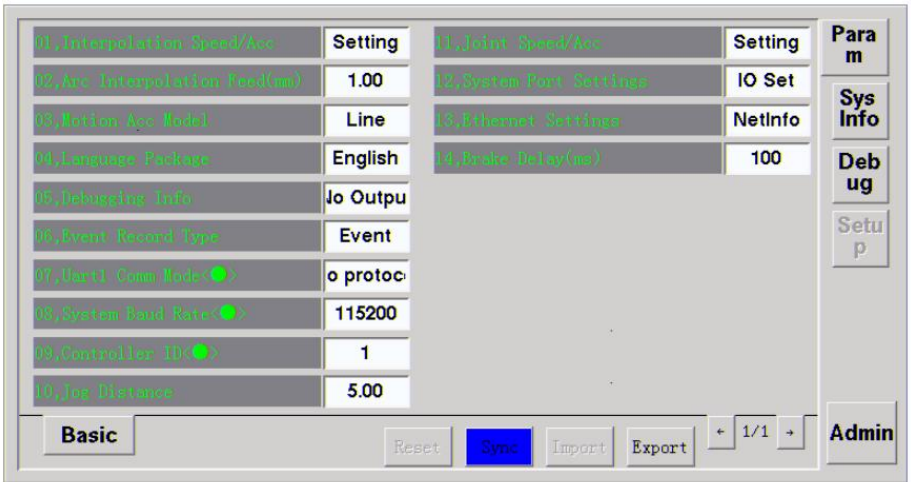
Flexpendant VeV603

```
Burn 2 area...
Burn 3 area...
Burn 4 area...
Complete Burning

Success to upgrade U/Motion.bin
运动库升级成功

Search resource file u:/ADT/RC400.N(
Success to upgrade U/RC400.NCP
Success to upgrade resource

Upgrade has finished,and succes:
```

☑ DetailInfo ☑ Mandatory

7. Parameter

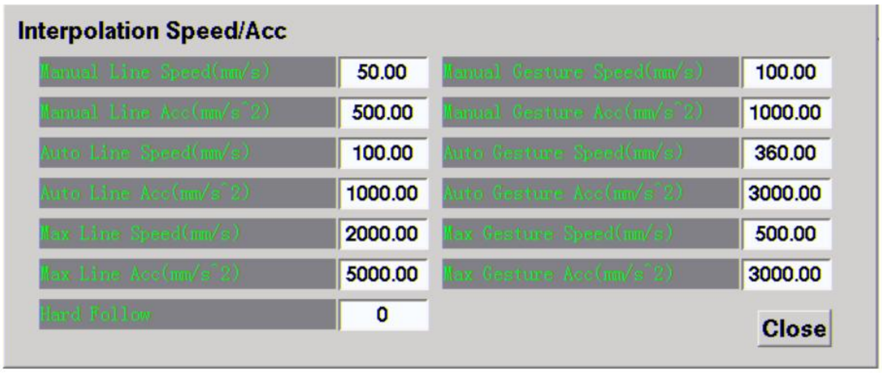
Many parameters of the RC400 controller are configured in this interface.



01,Interpolation Speed/Acc	Setting	11,Joint Speed/Acc	Setting
02,Arc Interpolation Feed(mm)	1.00	12,System Port Settings	IO Set
03,Motion Acc Model	Line	13,Ethernet Settings	NetInfo
04,Language Package	English	14,Brake Delay(ms)	100
05,Debugging Info	Jo Outpu		
06,Event Record Type	Event		
07,Uart1 Comm Mode<●>	o protoc		
08,System Baud Rate<●>	115200		
09,Controller ID<●>	1		
10,Jog Distance	5.00		

01, Interpolation Speed/Acceleration

Click "Setting" in the interpolation Speed/Acceleration column to enter the interface of interpolation speed/ Acc.



Interpolation Speed/Acc

Manual Line Speed(mm/s)	50.00	Manual Gesture Speed(mm/s)	100.00
Manual Line Acc(mm/s^2)	500.00	Manual Gesture Acc(mm/s^2)	1000.00
Auto Line Speed(mm/s)	100.00	Auto Gesture Speed(mm/s)	360.00
Auto Line Acc(mm/s^2)	1000.00	Auto Gesture Acc(mm/s^2)	3000.00
Max Line Speed(mm/s)	2000.00	Max Gesture Speed(mm/s)	500.00
Max Line Acc(mm/s^2)	5000.00	Max Gesture Acc(mm/s^2)	3000.00
Hard Follow	0		

- Manual Linear Speed: manual interpolation speed of X/Y/Z axis under the Descartes coordinate system;
- Manual Linear Acc: manual interpolation acceleration of X/Y/Z axis under the Descartes coordinate system;
- Manual Gesture Speed: manual interpolation speed of C axis under the Descartes Parameter coordinate system;
- Manual Gesture Acc: manual interpolation acceleration of C axis under the Descartes coordinate system;
- Auto Line Speed: automatic interpolation speed of X/Y/Z axis under the Descartes coordinate system;
- Auto Line Acc: automatic interpolation acceleration of X/Y/Z axis under theDescartes coordinate system;
- Auto Gesture Speed: automatic interpolation speed of C axis under the Descartes coordinate system;
- Auto Gesture Acc: automatic interpolation acceleration of C axis under the Descartes coordinate system;
- Max Line Speed: maximum speed of interpolation of line and arc for X/Y/Z axis;

- Max Line Acc: maximum acceleration of interpolation of line and arc for X/Y/Z axis;
- Max Gesture Speed: maximum speed of interpolation of line and arc for C axis;
- Max Gesture Acc: maximum acceleration of interpolation of line and arc for C axis;
- Hard follow: 0 is close; 1 is open;

02, Arc Interpolation Feed (mm): Arc resolution accuracy;

03,Motion Acc Model: LinearModel/CosineModel/ExponentModel;

04, Language Package: current language package used in system; controller is needed to be restarted after switching another language;

06, Event Record Type: Including servo shell, move shell, and operate shell;

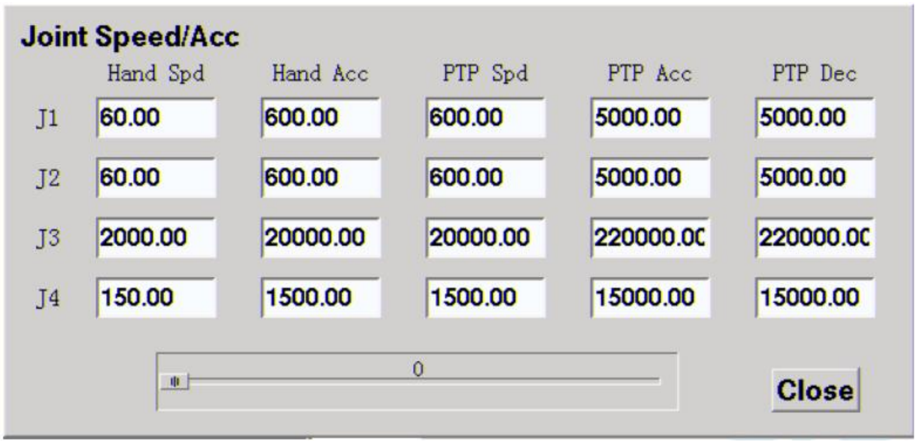
07, Uart1 Comm Mode: Asynchronous receiver transmitter, including Shell, Modbus, and no protocol;

08, System Baud Rate: COM2 baud rate which can also be modified in the program;

09, Controller ID: Configure the station number of controller for Modbus communication;

10, Jog Distance: Define the maximum value of a single step movement (default 5);

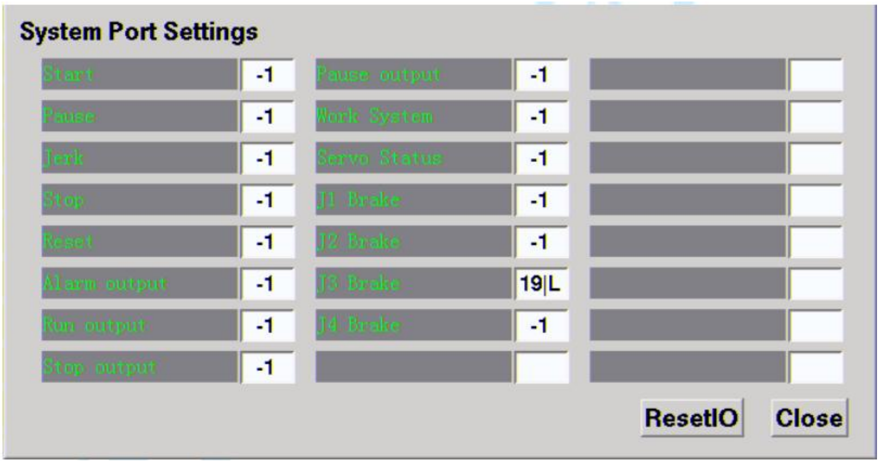
11, Joint Speed/Acc: configuration of manual joint speed, maximum speed of the point-to-point movement, and maximum acceleration;



Joint Speed/Acc

	Hand Spd	Hand Acc	PTP Spd	PTP Acc	PTP Dec
J1	60.00	600.00	600.00	5000.00	5000.00
J2	60.00	600.00	600.00	5000.00	5000.00
J3	2000.00	20000.00	20000.00	220000.0C	220000.0C
J4	150.00	1500.00	1500.00	15000.00	15000.00

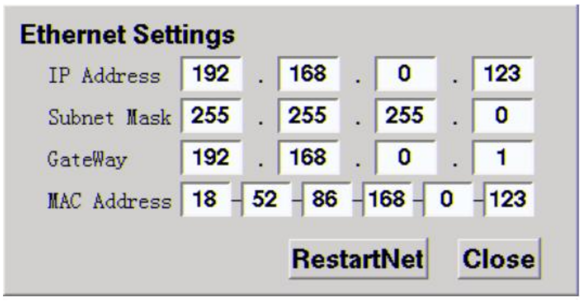
12, System Port Settings: for configuring the effective level of some inputs/outputs port, including the start, emergency stop and other integrated functions;



System Port Settings

Start	-1	Pause output	-1		
Pause	-1	Work System	-1		
Jerk	-1	Servo Status	-1		
Stop	-1	J1 Brake	-1		
Reset	-1	J2 Brake	-1		
Alarm output	-1	J3 Brake	19|L		
Run output	-1	J4 Brake	-1		
Stop output	-1				

13, Ethernet Settings: Configure system network IP Address, Subnet Mask address, Gateway, and MAC Address;



Ethernet Settings

IP Address 192 . 168 . 0 . 123

Subnet Mask 255 . 255 . 255 . 0

GateWay 192 . 168 . 0 . 1

MAC Address 18 - 52 - 86 - 168 - 0 - 123

14.After changing the value of the corresponding parameters, you need to click the “Sync” button; click on the parameter interface of the “Export” button to export parameters for backup.stop delaying time (ms) to prevent z axis falling down, the original setting is 100ms

Tips:

- ”11. joint speed/ accelerating speed “the PTP speed and PTP accelerating/ decreasing speed is limited by the setting of each axis maximum speed and accelerating speed in interface. 4.3
- After the parameter value is adjusted, click “synchronize”
- Click the “export” button to save a backup of parameter setting
- Lick the “import” button to import from the same robot parameter and use it. (administrator authorized needed)

8. Alarms Handling

There may come some alarming phenomenon when RC400 controller is in the use for some security protections; each alarm has a corresponding alarm code and related faulty content.

For actual use, to avoid unnecessary damage and safety problems, we should immediately stop running robot when an alarm appears. Refer to the error ID to eliminate errors one by one, then continue to restart robot.

Error ID		
11003	Encoder is not connected	
	Analysis	Selected encoder type in servo software is wrong or motor encoder wiring of motor is connected wrong.
	Handle	●Check if the encoder type selection is correct in the servo software. ●Refer to the encoder wiring in the electrical manual to check whether the encoder wiring is correct.
11007	Motor stuck	
	Analysis	The possible reasons for this alarm are: protection conditions of stuck in servo software are set too strict; motor with brake, but brake is not open; the selected motor is in low power for heavy load; Mechanical clamping.
	Handle	Firstly, amplify the stuck protection conditions in servo software; If the alarm still exists, then check whether the mechanical structure is stuck; If there is normal, maybe motor power does not match.
11008	Bus voltage is too high	
	Analysis	Bus voltage is instability
	Handle	Power loads in day and night of a factory are different; Generally, bus voltage will rise in the evening, so it is better to check whether the bus voltage set in servo protection parameters is correct.
11009	Bus voltage is too low	
	Analysis	Bus voltage is instability
	Handle	Bus voltage will decline when robot run with load or high speed. In this case, it maybe causes an alarm for bus voltage being too low; You can change the “Minimal allowed bus voltage” to 180V. if the alarm still exists, please contact manufacture.
11013	A phase current is too high	
11014	B phase current is too high	
11015	C phase current is too high	
	Analysis	Motor’s phase current exceeds the protection range.
	Handle	● Check whether motor’s power wiring is correct. ● If power wiring is right, you can decrease speed or reduce the load to see whether the alarm is relieved; If the alarm is cleared which means that the load is too heavy or the running speed is over the maximal allowed velocity.

11016	Motor current is too high	
	Analysis	The actual current of the motor exceeds the protection range
	Handle	Check whether power cable wiring is correct
11020	Position deviation is over limit	
	Analysis	The position deviation exceeds maximal allowed position error.
	Handle	● Check whether the maximal allowed position error set in the protection parameter is too small (5~10 times of the actual position deviation should be set). ● Position loop gain is set improperly: you can increase position loop gain appropriately under the condition of ensuring no mechanical jitter. ● Acceleration is set unreasonable: you can reduce acceleration or deceleration.
11021	Velocity deviation	
	Analysis	The velocity deviation exceeds the limit
	Handle	●Check whether the power cable wiring is correct. ●Check whether the maximal allowed velocity error set is too small (should be set as 5~10 times of the actual speed deviation). ●Velocity loop parameter setting is not appropriate: in the case, increase the gain of velocity loop appropriately to ensure that running robot has no noise. ●Check whether the shielding line of power cable is connected reliably and check whether the motor with brake is connected with a brake plate filter.
11027	IPM module error	
	Analysis	Module abnormal
	Handle	Please contact the manufacturer
1028	Selected encoder type is not supported	
	Analysis	Encoder type is not correct
	Handle	Check whether the encoder type selected in the servo software is correct, and please contact manufacture to ensure whether RC400 controller support this type controller.
11035	Drive power supply module is disconnected	
	Analysis	Abnormal power supply
	Handle	Check whether the 220V power supply has fluctuations or abnormal
19999	Encoder communication error	
	Analysis	Encoder is abnormal
	Handle	Check whether the encoder wiring is correct, the encoder shield wire connection is reliable.
19998	BISS protocol encoder communication error	
	Analysis	Encoder exception
	Handle	●Check whether the encoder wiring is correct, the encoder shield wire connection is reliable. ●For magnetic encoder, if the battery has low voltage under the condition of alarms not being cleared, it will also cause this alarm; in this case, you

		need to change a new battery, and then clear the alarm by command form. Notice that the robot is required to calibrate the origin again.
11090	Warning of low Battery voltage	
	Analysis	Battery of the encoder is abnormal
	Handle	Check whether the battery is in low voltage. If the battery is in
11036	overload	
	Analysis	The actual current of the motor exceeds the overload protection range
	Handle	● If the speed is reduced, the motor run normally, which indicates that the load is too heavy or type of motor isn’t fit. In this case, you need to robot’s running speed or change another motor.
11037	Motor overload of 1.2 times	
11038	Motor overload of 1.5 times	
11039	Motor overload of 2 times	
11040	Motor overload of 2.5 times	
11041	Motor overload of 3 times	
	Analysis	Motor current exceed the current limit and continue for a period
	Handle	If reduce speed, motor running normally. It shows that the load is too heavy or selected motor does not be matched or deceleration set is inappropriate. In this case, it is better to reduce speed or change another motor with higher power.
11042	Battery error	
	Analysis	Error alarm about multi loop information of motor
	Handle	●After installation of absolute encoder with the battery, you need to clear the battery error alarm (FlexPandent with an interface to clear this alarm) when the robot starts for the first time. ●If this error comes up when not first time use after installation, it shows that the origin has been.
11043	Error alarm of CRC checksum	
	Analysis	Encoder data is abnormal
	Handle	Check whether the encoder line is connected wrong or check whether shield line of encoder is not connected.
11057	Input speed of pulse is too large	
	Analysis	Speed of sending pulse for controller is over the protection range
	Handle	●Check whether the maximum allowable speed of servo protection parameter is reasonable; ●Check whether the pulse sent by controller is normal.
11058	Bus between FPGA and DSP is abnormal	
	Analysis	Data bus or address bus between FPGA and DSP is abnormal
	Handle	Please contact factory
20005	No axis existing	
20004	Axis used conflict	
	Analysis	Error in axes’ parameters
	Handle	Check whether parameters in AR program are set correctly.

20006	Unable to reach for invalid area	
	Analysis	The target position is not in the reasonable working range of the robot. It is also possible that some positions are belonging to singular points, if so, it also report this alarm when move robot with straight line.
	Handle	●Determine whether the target position is outside the range of robot’s motion, or in the singular point position. ●If the position data is imported from somewhere, it is required to check whether the arm lengths are the same.
20009	Unable to track motion for singular region	
	Analysis	Singular point refers to interference region of the robot body. This warning is generally generated when check whether the current point is reasonable before moving to it. Usually due to move robot with a line movement, and the current point stay in the zero point or interference.
	Handle	Change the line command to point-point or arch movement. The current point out the critical point (manual arm all in a line, on behalf of the critical point), and then execute the motion instructions
20010	Unable to track motion for different hand	
	Analysis	The target position and the current position are not in the same hand while performing the line motion.
	Handle	To modify the current hand or target’s hand to ensure that they are in the same hand.
20013	Interpolation queue is full, please wait	
20014	Event queue is full, please wait	
20016	Undefined Order	
	Analysis	Operating environment is abnormal
	Handle	Please contact the manufacturer
20018	External IO Trigger Alarm	
	Analysis	The system is equipped with external emergency stop, and the emergency stop signal is detected to be effective, which leads to the protection of the alarm.
	Handle	Check whether external emergency stop is effective. Check whether the emergency stop port and the effective level is set reasonable
20019	Motion Stop	
	Analysis	Self locking protection alarm, which needs to be cooperated with other alarm information at the same time to analyze it.
	Handle	● According to other alarm information to determine the reason.
20020	Security detection of ARM system is abnormal, and DSP is self protection	
	Analysis	For abnormal ARM running, so DSP watchdog creates an alarm to enter self protection
	Handle	Please contact the manufacturer
20021	J1 Soft Limit	
20022	J2 Soft Limit	
20023	J3 Soft Limit	

20024	J4 Soft Limit	
	Analysis	Operating position is out of range
	Handle	Check whether the set range is reasonable
20025	Motor enable error	
	Analysis	Disable error during movement
	Handle	Cannot be carried out enable or disable operation; check whether the operation is in compliance with the specification.
20026	External encoder communication error	
20027	External encoder battery low voltage warning	
20028	External encoder battery warning	
	Analysis	External encoder (M5 or M6) is abnormal
	Handle	If it is 20026-error, check whether the connection is reliable and shielding is good. If the battery is low voltage, you need to change the battery. Please pay attention, the controller must be on power when change a new battery in avoid of losing the origin of the robot.
41001	Task Timeout	
	Analysis	Operating environment is abnormal
	Handle	Please contact the manufacturer
41002	HMI Connection Failed	
	Analysis	HMI communication cannot work properly, may be due to the version of the reasons or bad connection.
	Handle	Check the MCU version number and the welding line of the connecting head of teach pendent.
41003	HMI Detect Emergency Stop	
	Analysis	HMI emergency stop detection is effective.
	Handle	Observe whether it is really effective; if so, clockwise rotate the emergency stop switch to pop up and then press “Reset” button to relieve this alarm.
41004	Detect the external scram	
	Analysis	System is configured with an external emergency stop IO, and this IO is effective.
	Handle	● If there is an external emergency stop, you need to relieve the external emergency stop signal, and then press “reset” within the alarm interface. ● If the parameter is wrong, you can modify port number and active level of the IO.
41006	Abnormal DSP running time	
41006	Failed to create task	
	Analysis	Running environment is abnormal
	Handle	Please contact the manufacturer