INOVANCE

Quick Start Guide

Inovance Robot System

User Guide

A00
Data code 19010949

Contents

I. Basic Terminology 3
1 Coordinate System 4
1.1 Joint Coordinate System 4
1.2 Base Coordinate System 4
1.3 Tool Coordinate System 5
1.4 User Coordinate System 5
1.5 Other Coordinate Systems 6
2 Position Variable 6
* Position variable for special processes 8
3 Motion and Transition 8
II. Basic Operation 9
1 Software Usage Instructions 10
1.1 Basic Operating Process 10
1.2 System Startup 11
1.3 Functions of the Main Interface 13
1.4 Manipulation on Robot Movement 17
2 Programming and Running 21
2.1 Programming Panel 21
2.2 Running Panel 23
3 Settings 24
3.1 BasePos 25
3.2 Coordinate System Settings 26
3.3 Motion Settings 30
3.4 Peripheral Configurations 31
3.5 System Settings 32
4 Monitoring 41
4.1 Basic Operations on the Mon Page 41
4.2 Global Variable Monitoring 43
4.3 Local Variable Monitoring 44
4.4 I/O Monitoring 47
4.5 Communication State 50
4.6 Servo State 51
4.7 Log 51

4.8 Version Information....52
III. Programming 53
1. Robot Program Files 54
1.1 Position Variables in the Program....54
1.2 Commands in the Program....54
2 Variables 55
2.1 Global and Local Variables 55
2.2 Signal Variables 58
3 Motion Programming....59
3.1 Motion Commands....59
3.2 Form and Parameters of Motion Commands 60
3.3 V and Acc 61
3.4 Z 61
3.5 User....63
3.6 Tool....64
3.7 NWait 65
3.8 Until 65
3.9 I/O in the Motion Process....66
3.10 Jump and JumpL 66
4. I/O Programming 67
4.1 Preparations Before I/O Programming 67
4.2 I/O Read-Write 68
4.3 I/O Group Read-Write 69
4.4 Other I/O Commands 69
5 Logic Control Programming....70
5.1 Jump 70
5.2 Selection Type 70
5.3 Loop Type....72
5.4 Subprogram Call....74
6 TCP/IP Communication Programming....74
6.1 Communication Between an External Device and a Controller over TCP/IP 74
6.2 TCP/IP Communication Commands 75
Appendix: Robot Error and Solution List.... 77

I. Basic Terminology

1 Coordinate System 4

1.1 Joint Coordinate System 4

1.2 Base Coordinate System 4

1.3 Tool Coordinate System 5

1.4 User Coordinate System 5

1.5 Other Coordinate Systems 6

2 Position Variable 6

* Position variable for special processes 8

3 Motion and Transition 8

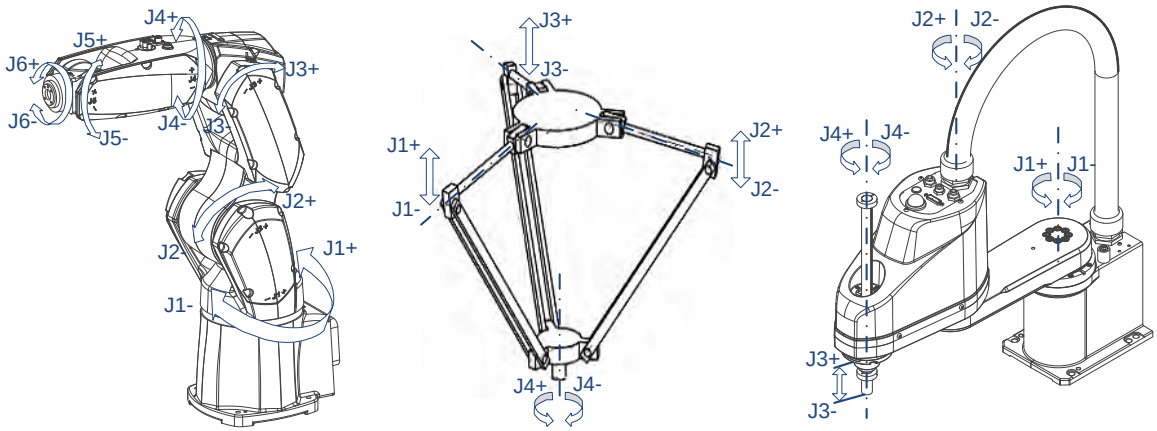
1 Coordinate System

Seven coordinate systems are available in Inovance robot system.

Coordinate system No.	Coordinate system name
1	Joint coordinate system
2	Base coordinate system
3	Tool coordinate system
4	User coordinate system
5	Fixed camera field of view coordinate system
6	Mobile camera field of view coordinate system
7	Object coordinate system on the conveyor belt

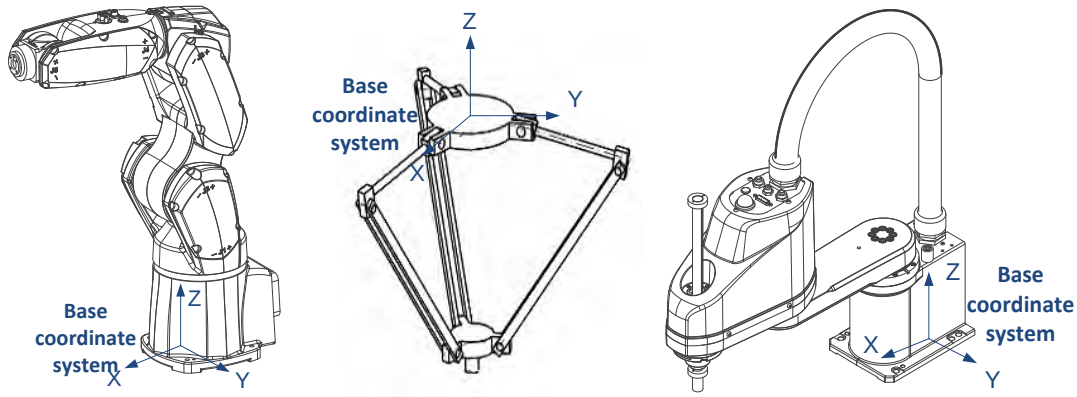
1.1 Joint Coordinate System

The joint coordinate system is located at the joints of the robot.



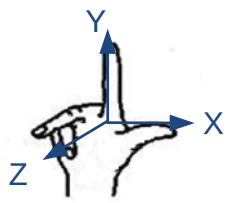
1.2 Base Coordinate System

The base coordinate system is also called the robot coordinate system and is generally located at the base of the robot.



The base coordinate system is a Cartesian coordinate system. (The tool and user coordinate systems are also Cartesian coordinate systems.)

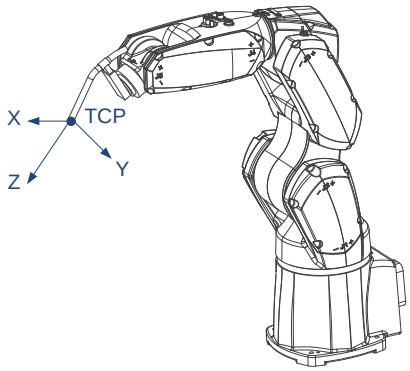
Method for determining the direction of a Cartesian coordinate system: Determine the X and Z directions and then the Y direction using the right-hand rule.



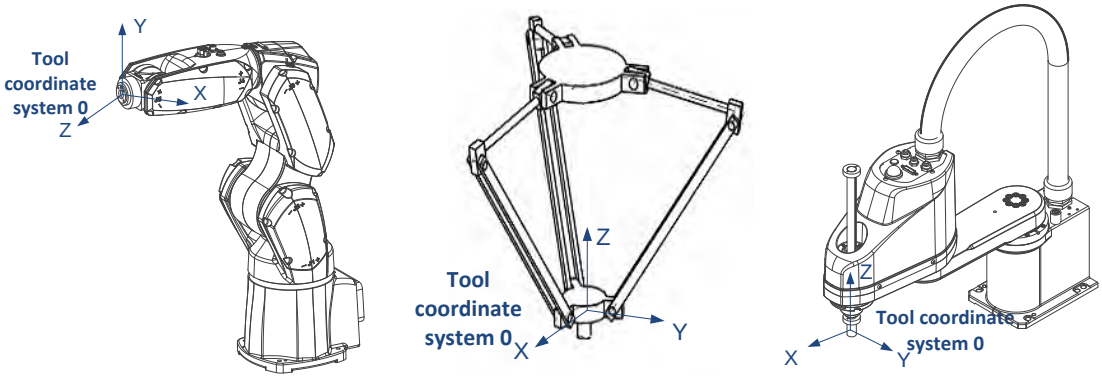
Right-hand rule

1.3 Tool Coordinate System

The tool coordinate system is attached to a tool. The tool center point (TCP) is a reference point for a position where the robot reaches. The tool endpoint is generally taken as the TCP and the direction can be freely defined.



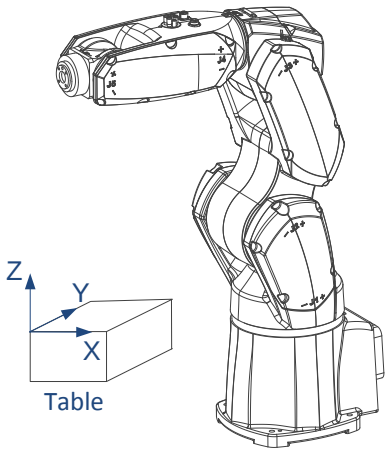
Up to 16 tool coordinate systems can be defined in Inovance robot control system. Tool 0 indicates that no tool is used, in which case, the tool coordinate system is located at the end of the robot body.



Tools 1-15 can be user defined.

1.4 User Coordinate System

The user coordinate system is a user-defined coordinate system whose origin parameter is described as coordinate values under the robot base coordinate system. The user coordinate system can be set discretionarily in essence, but in any specific application, one can set it to being on specific objects such as the table and the conveyor belt.



Up to 16 user coordinate systems can be defined in Inovance robot control system. User 0 indicates that the base coordinate system is also used as the user coordinate system. Users 1-15 can be user defined.

1.5 Other Coordinate Systems

Some specialized coordinate systems are used for some special processes.

Coordinate System No.	Coordinate System	Applicable Process	Meaning
5	Fixed camera field of view coordinate system	Specially used for the field of view function	Points (X, Y, θ) within fixed camera field of view are defined in this coordinate system.
6	Mobile camera field of view coordinate system	Specially used for the field of view function	Points (X, Y, θ) within mobile camera field of view are defined in this coordinate system.
7	Object coordinate system on the conveyor belt	Specially used for the tracking process	Points (X, Y, Z, A, B, C) relative to the object coordinate system on the conveyor belt are defined in this coordinate system.

2 Position Variable

In Inovance robot control system, spatial points are expressed using a "position variable" that stores information about coordinate values, arm parameters, coordinate system, tool No. and user No.

```
P[0] = 26.969294, -107.023346, 0.000000, 13.795120, 12.265454, 3.645454;-1, 0, 0, 0; 1, 0, 0;
```

① Coordinate values:

The meanings of coordinate values are indicated by a coordinate system.

When the coordinate system is 1, coordinate values are joint values of the robot (J1, J2, J3, J4, J5, J6). Joint values are rotation angles of an arm relative to the zero position.

When the coordinate system is 2, coordinate values are poses (X, Y, Z, A, B, C) of the robot flange center point* relative to its base coordinate system.

When the coordinate system is 3, coordinate values are poses (X, Y, Z, A, B, C) of the TCP relative to its base coordinate system.

When the coordinate system is 4, coordinate values are poses (X, Y, Z, A, B, C) of the TCP relative to the user coordinate system.

*Flange center point: Center point on the flange end face of the last axis of the robot that represents the end reference position of the robot body. The same as below.

② Arm parameters:

The robot can reach the same pose in many ways. Arm parameters are used to distinguish these ways.

③ Coordinate system No.:

Coordinate system used to get points. The meanings of coordinate values are indicated by the coordinate system.

④ Tool No.:

Current tool used.

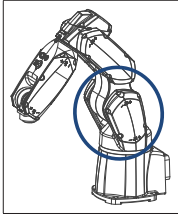
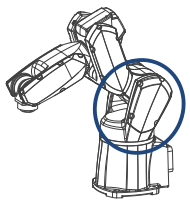
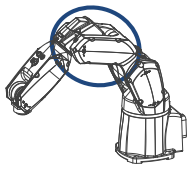
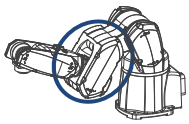
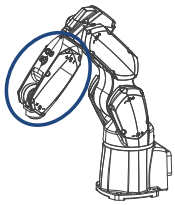
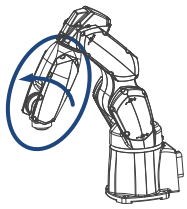
⑤ User No.:

Current user coordinate system used.

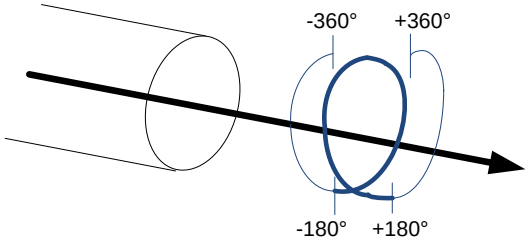
■ About arm parameters

When the robot control target point reaches the same pose, several different arm types of the robot may exist.

For the six-axis serial robot, there is an arm parameter at the waist, elbow and wrist joints and Axis 6.

Parameter 1		Parameter 2		Parameter 3	
-1	1	-1	1	-1	1
Waist forward	Waist backward	Elbow upward	Elbow downward	Wrist not flipped	Wrist flipped
					

Parameter 4		
-1	0	1
J6 (-360o to -180o)	J6 (-180o to 180o)	J6 (180o to 360o)



For the SCARA robot, arm parameters 1 and 4 are meaningful.

Parameter 1		Parameter 4 (Joint coordinate J4 value)		
-1	1	(-540 to -180)	(-180 to 180)	(180, 540)
Left arm orientation	Right arm orientation	-1	0	1
		When multi-turn is involved, the arm parameter increases by 1 per 360° increase and decreases by 1 per 360° decrease.		

For the Delta robot, arm parameter 4 is meaningful.

Delta arm parameter 4 (Joint coordinate J4 value)		
(-540 to -180)	(-180 to 180)	(180, 540)
-1	0	1
When multi-turn is involved, the arm parameter increases by 1 per 360° increase and decreases by 1 per 360° decrease.		

NOTE

◆ If arm parameters are modified after teaching, the robot will reach the same pose in an another arm type with a great change to motion status. Be careful during modification.

* Position variable for special processes

For some special processes, more position variable types are defined as follows:

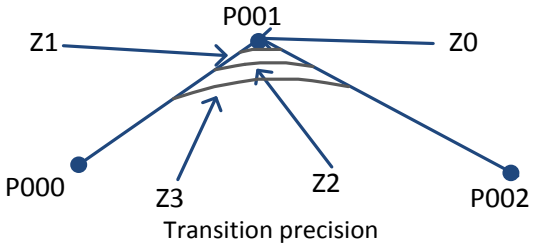
Position points within fixed camera field of view	The coordinate system No. is 5. Position points have coordinate values (X, Y, θ) within the plane of camera field of view. The corresponding storage form is (X, Y, 0, A, 0, 0). The tool No. is a used tool. The user No. is a used visual coordinate system No.	Specially used for the field of view function
Position points within mobile camera field of view	The coordinate system No. is 6. Position points have coordinate values (X, Y, θ) within the plane of camera field of view. The corresponding storage form is (X, Y, 0, A, 0, 0). The tool No. is a used tool. The user No. is a used visual coordinate system No.	Specially used for the field of view function
Used for the tracking process	The coordinate system No. is 7. Coordinate values (X, Y, Z, A, B, C) are used to define the synchronous motion of an object on the conveyor belt and represent the coordinate position of the object relative to the conveyor belt.	Specially used for the tracking process

3 Motion and Transition

Most of the time, in order to accelerate rhythm, accurate arrival is not required. At this moment, fly-by point is needed. This track approximation is expressed as "Z" in the industry.

Transition modes are as follows:

Type	Mode
Fine	It indicates accurate arrival. When the robot enters and stays within a set error range, it is deemed to be an accurate arrival and the subsequent motion is executed.
Z[0]	There is no transition, but the robot cannot be ensured to reach a target point accurately.
Z[1]	There is a transition at one transition unit length.
Z[2]	There is a transition at twice the transition unit length.
Z[3]	There is a transition at three times the transition unit length.
Z[4]	There is a transition at four times the transition unit length.
Z[5]	There is a transition at five times the transition unit length.
Z[CP]	There is a transition at the maximum transition length (Refer to the maximum transition length).



Interference area: Some areas where the end effector is prohibited from reaching are set to avoid collision.

II. Basic Operation

1 Software Usage Instructions 10

1.1 Basic Operating Process 10

1.2 System Startup 11

1.3 Functions of the Main Interface 14

1.4 Manipulation on Robot Movement 17

2 Programming and Running 21

2.1 Programming Panel 21

2.2 Running Panel 23

3 Settings 24

3.1 BasePos 25

3.2 Coordinate System Settings 26

3.3 Motion Settings 30

3.4 Peripheral Configurations 31

3.5 System Settings 32

4 Monitoring 41

4.1 Basic Operations on the Mon Page 41

4.2 Global Variable Monitoring 43

4.3 Local Variable Monitoring 44

4.4 I/O Monitoring 47

4.5 Communication State 50

4.6 Servo State 51

4.7 Log 51

4.8 Version Information 52

1 Software Usage Instructions

The teaching software is divided into three modules in terms of function:

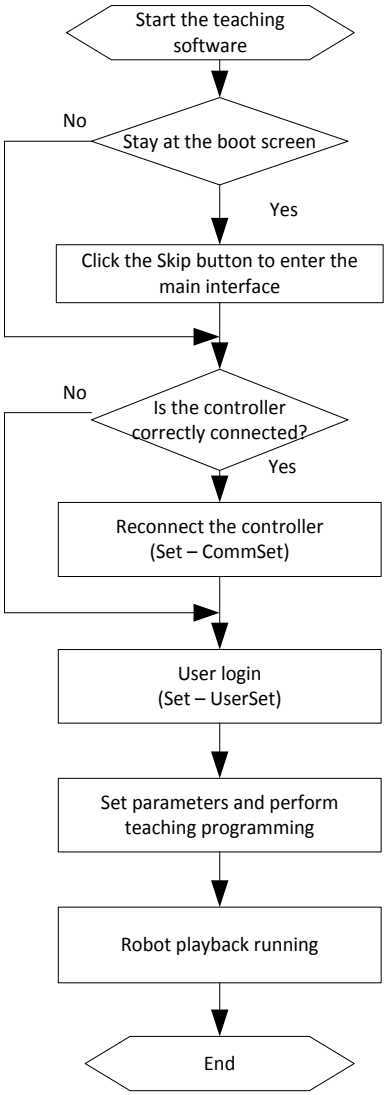
Programming module: Main operating module of teaching process that is mainly used for program editing.

Monitoring module: This module is used to observe data such as program parameters, interface status and logs. It also supports modification to variable parameters.

Setting module: This module is used to set robot-related parameters including system-related settings, robot structure/motion parameter settings and extended module settings.

1.1 Basic Operating Process

The following figure shows the basic operating process of teaching programming: After the teaching software is started, the system automatically performs connection according to the last connection address. If connection is successful, after checking that the address is correct, the system enters the **Set - UserSet** interface for login. After login, set related parameters (basically all other settings except system settings) and begin to teach. After teaching is complete, run the robot program.



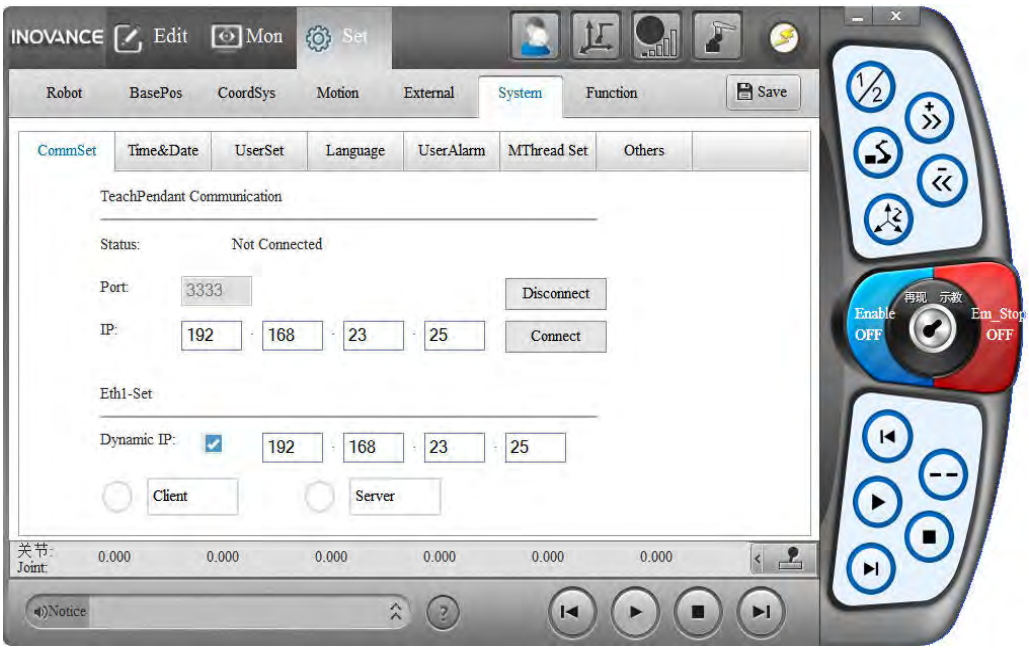
1.2 System Startup

a) Power-on and connection

Power on the system using the main switch of the electric control cabinet. After InoTeachPad is started, the system enters the connection interface:

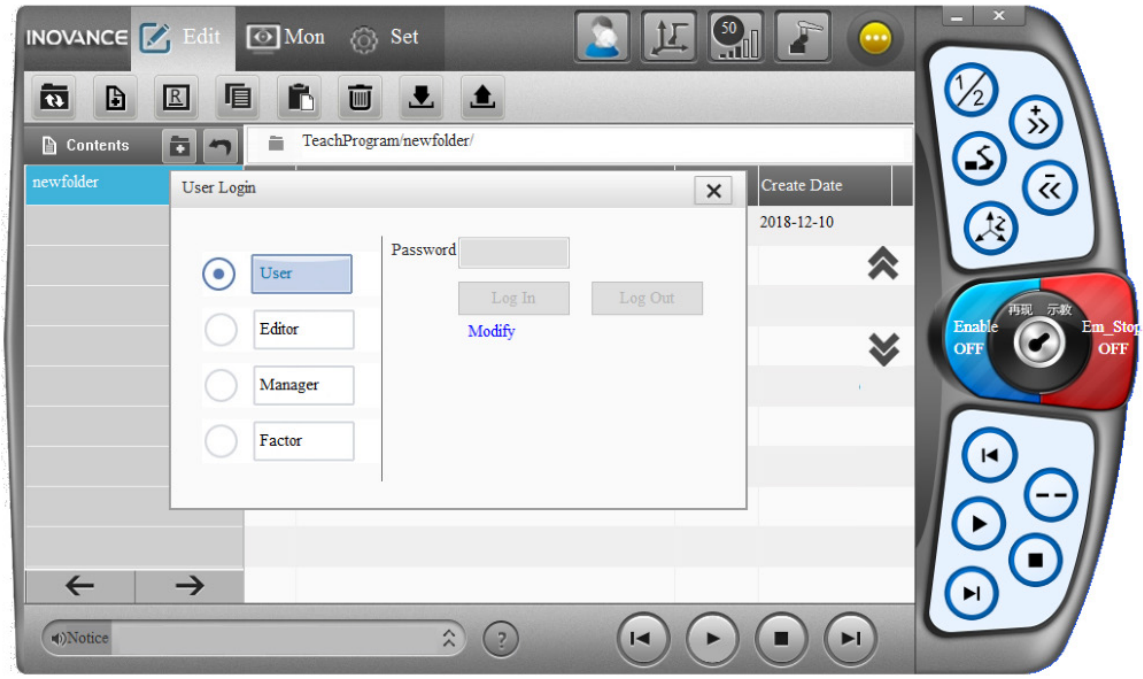
Generally, the system finally displays the message "Controller startup successful" and enters the main interface.

For hand-held InoTeachPad, it is directly connected to the controller by default and after power-on, is automatically connected. For non-direct connection, especially the PC version of InoTeachPad which is not connected to the controller, you need to click the Skip button, go to the Set - CommSet page, and enter an IP address for reconnection.



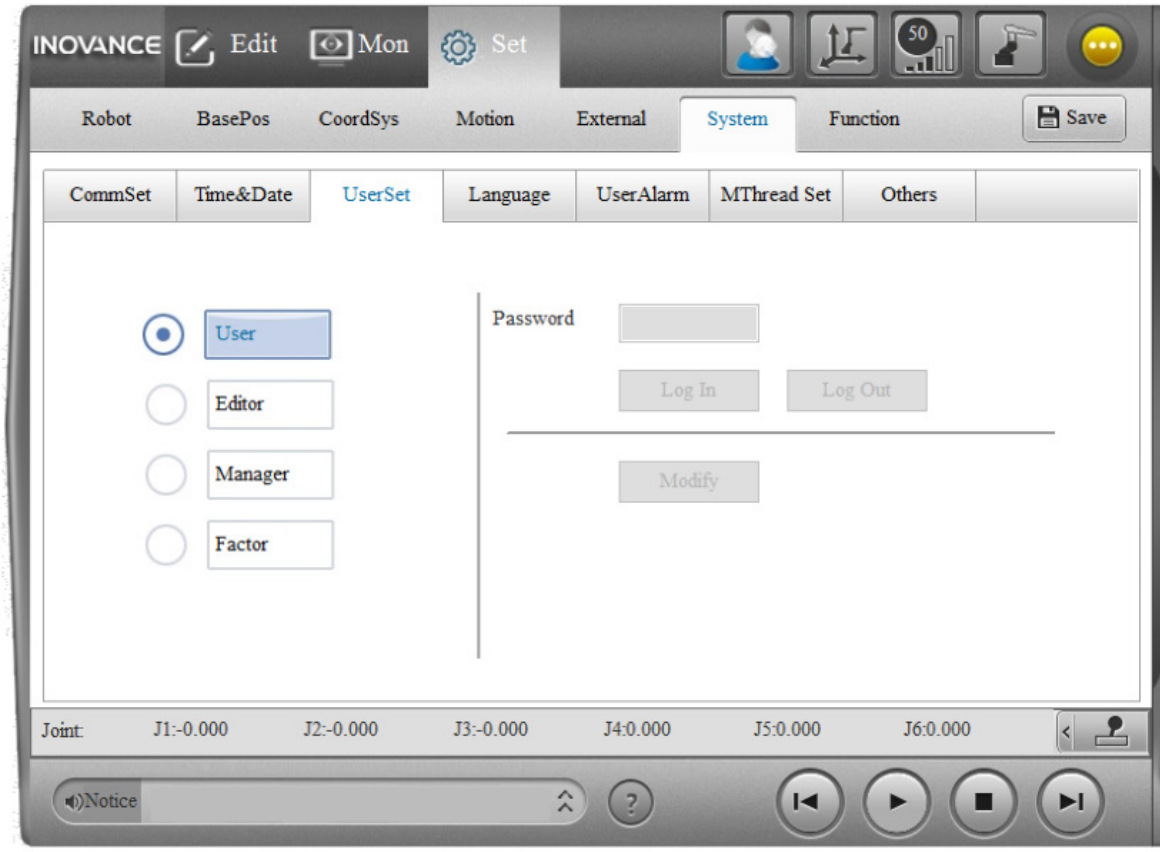
b) User login

User login is located in the Set - UserSet page. You can directly log in by clicking the button at the top right corner.



Select a mode according to the current user's role and log in.

Mode	Audience	Initial Default Password
User	Operators on the production line. Users can directly control robot motion and run a program.	No password is required.
Editor	Teaching programmer. Compared with the user mode, the program editing function is added so that users can perform teaching programming. Authorities such as equipment control and mechanical locking are also added.	The initial password is 000000.
Manager	Advanced users. Most system operations are included besides the preceding operation authorities.	The initial password is 000000.
Factory	The manufacturer's maintainers. Maintainers have the highest authority and can perform operations such as upgrading and joystick calibration.	Factory reserved



c) Inspection

After connection is successful, check the state indicator at the top right corner and the message bar at the bottom.

It is normal only when the state indicator is in standby or enabled state.

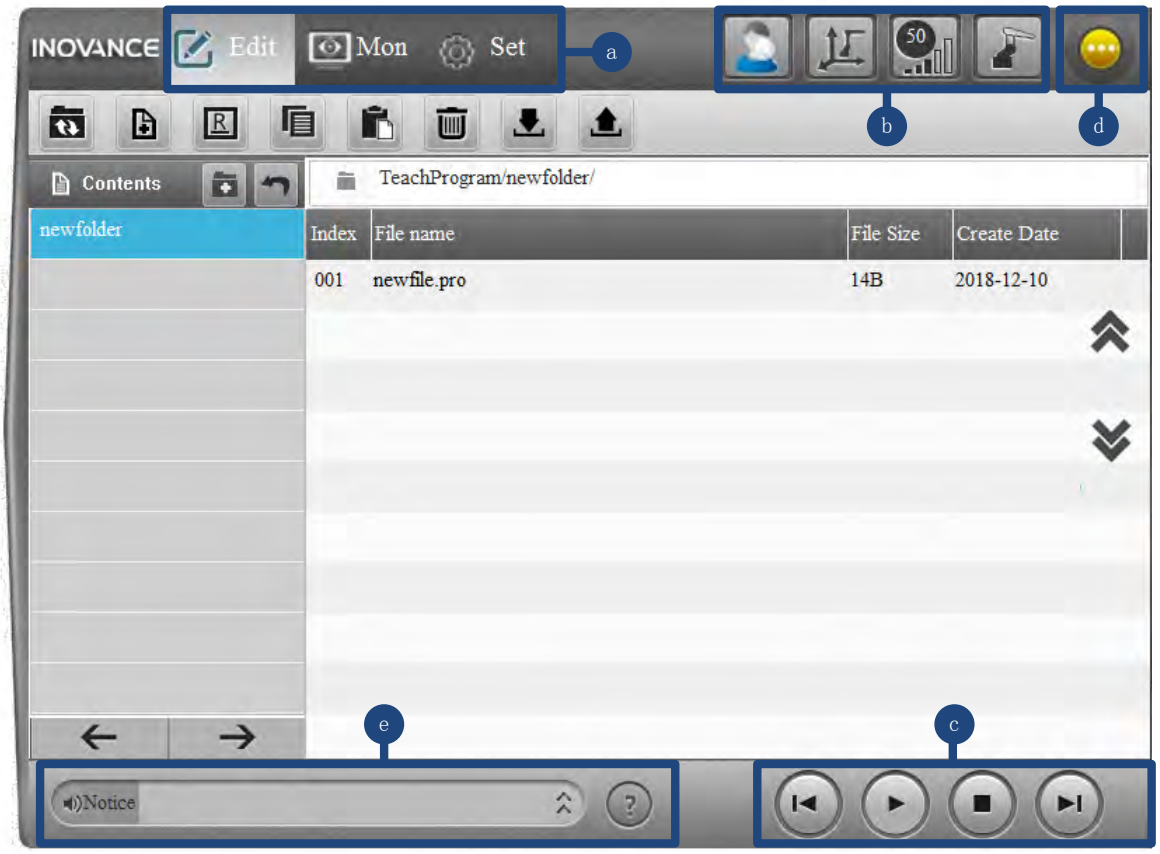
Other cases indicate abnormalities that shall be rectified according to the message bar at the bottom.

(a) Normal

(b) Abnormal

1.3 Functions of the Main Interface

The following main interface appears after the teaching software is started and connection is successful.



Panel switching bar

Different operating panels including the editing/running, monitoring and setting panels can be displayed using the panel switching bar.

a) Control toolbar

Four types of buttons including user mode buttons, coordinate system switching buttons, speed multiplying ratio/step selection buttons and axis group switching buttons exist on the control bar.

User mode buttons		User
		Editor
		Manager
		Factory
Coordinate system switching buttons		Joint coordinate system
		Base coordinate system
		Tool coordinate system The number at the top is a selected tool No.
		User coordinate system The number at the top is a selected user No.

Speed multiplying ratio/ step selection buttons		Running at 5% of a set speed
		Running at 25% of a set speed
		Running at 50% of a set speed
		Running at 100% of a set speed
		Return to speed adjustment multiplying ratio
		Switching to the step setting buttons
		Set a step value. Joint step: 0.2° Position step: 0.2 mm Rotation step: 0.2°
		Set a step value. Joint step: 1° Position step: 1 mm Rotation step: 1°
		Set a step value. Joint step: 5° Position step: 5 mm Rotation step: 5°
		User-defined step value (Take a value from Set - Motion - Precision - Step)
Axis group switching buttons		Axis group controlled by the joystick J1/J2/J3 (X/Y/Z)
		Axis group controlled by the joystick J4/J5/J6 (A/B/C)

b) Motion control bar

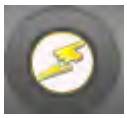
The motion control bar is used to control program execution and includes start/pause, stop, single-step forward and single-step backward buttons. It can be used in teaching programming mode and playback running mode.

Option Button	Option Function	Teaching programming mode	Playback running mode
	Single-step backward	Press this button to perform single-step backward. Release it to stop.	Disabled

Option Button	Option Function	Teaching programming mode	Playback running mode
	Start/Pause	Click this button to start, and a program continuously runs. Release it to pause.	Click this button to start. Re-click it to pause.
	Stop	Disabled (Once you release this button, a program stops running due to step mode under teaching.)	Once you press this button, a program stops running. The program is re-executed its top during the next startup.
	Single-step forward	Press this button to perform single-step forward. Release it to stop.	Disabled

State indicator

The state indicator is used to indicate the current state of the robot, including servo enabled, standby, emergency stop, error, warning and offline states. Note: The robot can move only in enabled state.

	Servo enabled: At this moment, the emergency stop button is released and servo is enabled. Motion can be performed only in enabled state.
	Emergency stop state: The emergency stop button is pressed and the robot cannot move.
	Standby state: The emergency stop button is released and servo is not enabled.
	Error state: An abnormality occurs and needs to be handled immediately.
	Warning state: An abnormality occurs and the system prompts users.
	Offline state: The network is disconnected and communication with the controller fails.

c) Message window

The message window displays prompt and error messages. For example, when you perform SD card formatting operation in user mode (SD card formatting requires management authority that is higher than user authority), a prompt message about authority appears in the message bars.

1.4 Manipulation on Robot Movement

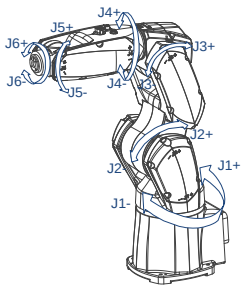
a) Procedure for manipulation on robot movement

Before the robot moves, select a coordinate system and speed (or step), hold down the Enabled switch (For the PC version of the teaching software, click the Enabled button to remain enabled), and click the

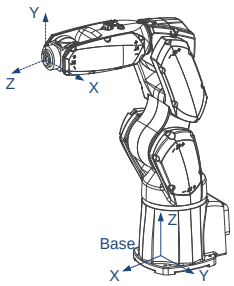
teaching panel to manipulate movement.

Step 1: Coordinate system, speed/step selection

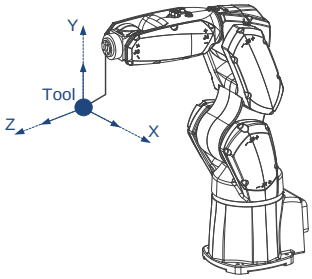
A coordinate system determines motion direction.



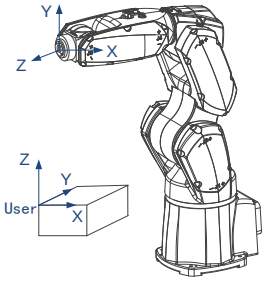
Move along the joint coordinate system



Move along the base coordinate system



Move along the tool coordinate system

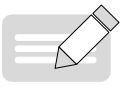


Move along the user coordinate system

Speed settings have four levels of 5%, 25%, 50% and 100% on the toolbar. You can also adjust it by the buttons and .

You can also select step. When you select step, the robot measures movement by step and moves one step at a time. Step has four levels of G1, G2, G3 and U, as shown below.

	Set a step value. Joint step: 0.2°, position step: 0.2 mm, rotation step: 0.2°
	Set a step value. Joint step: 1°, position step: 1 mm, rotation step: 1°
	Set a step value. Joint step: 5°, position step: 5 mm, rotation step: 5°
	User-defined step value (Take a value from Set - Motion - Precision - Step)



NOTE

◆ For movement in step mode, the movement speed is still related to the speed multiplying ratio!

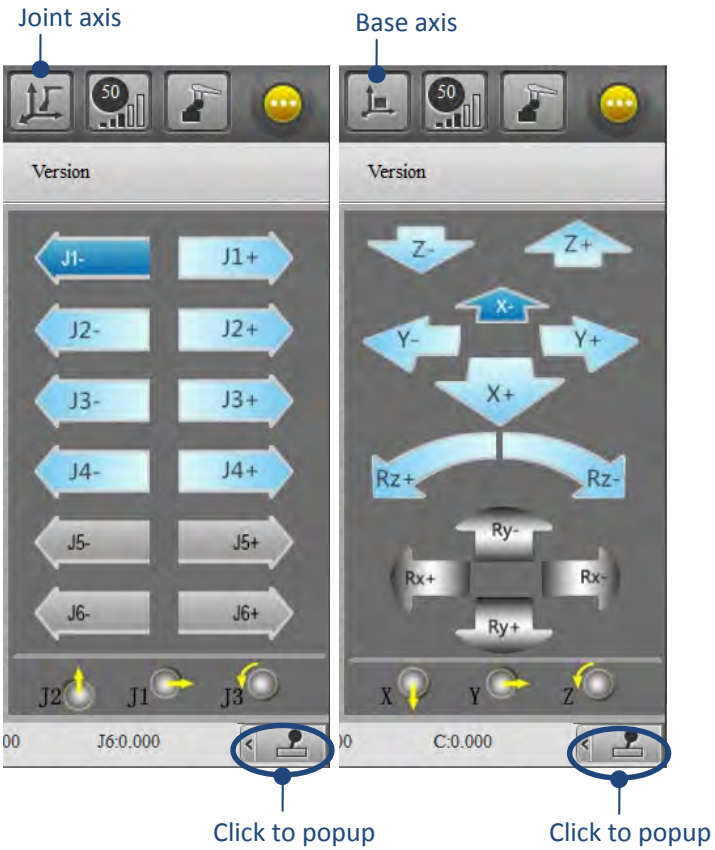
Step 2: If the robot moves along the tool or user coordinate system, pay attention to the current tool and user coordinate systems.

Step 3: Enable and manipulate movement

The robot can move only in enabled state.

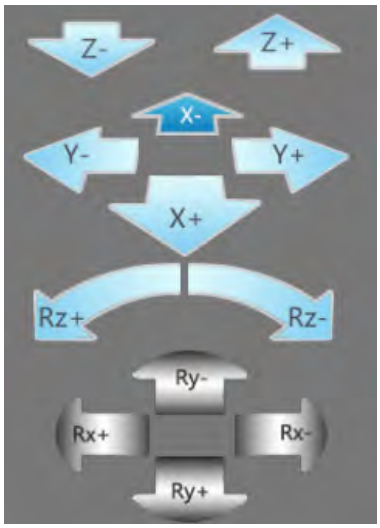
For hand-held InoTeachPad, always hold down the enabled switch. For the PC version of InoTeachPad, click the Enabled button to remain enabled!

A handle graphics button for manipulating robot movement is provided at the bottom right corner of most panels. Click this button to display the teaching panel that is used to control robot movement. The teaching panel has joint and non-joint types that are displayed according to the current coordinate system.



For IRTP80 InoTeachPad, you can also use its physical key.

For ITP100 InoTeachPad, you can use the joystick.



Step 1: Select a Speed button and hold down to enter step selection. Select G2 here.

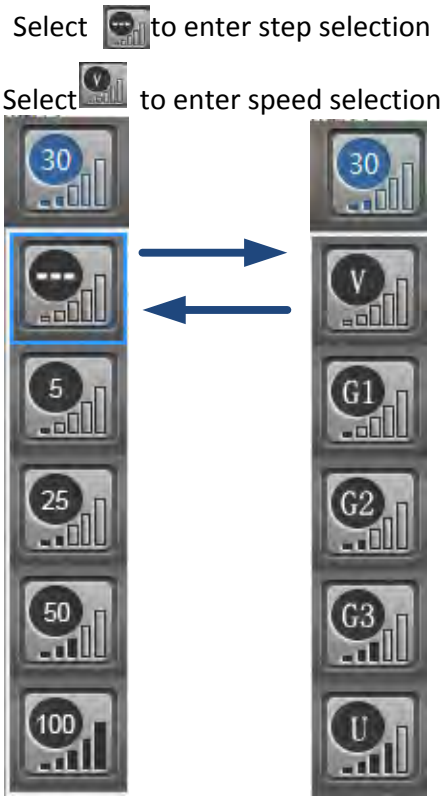
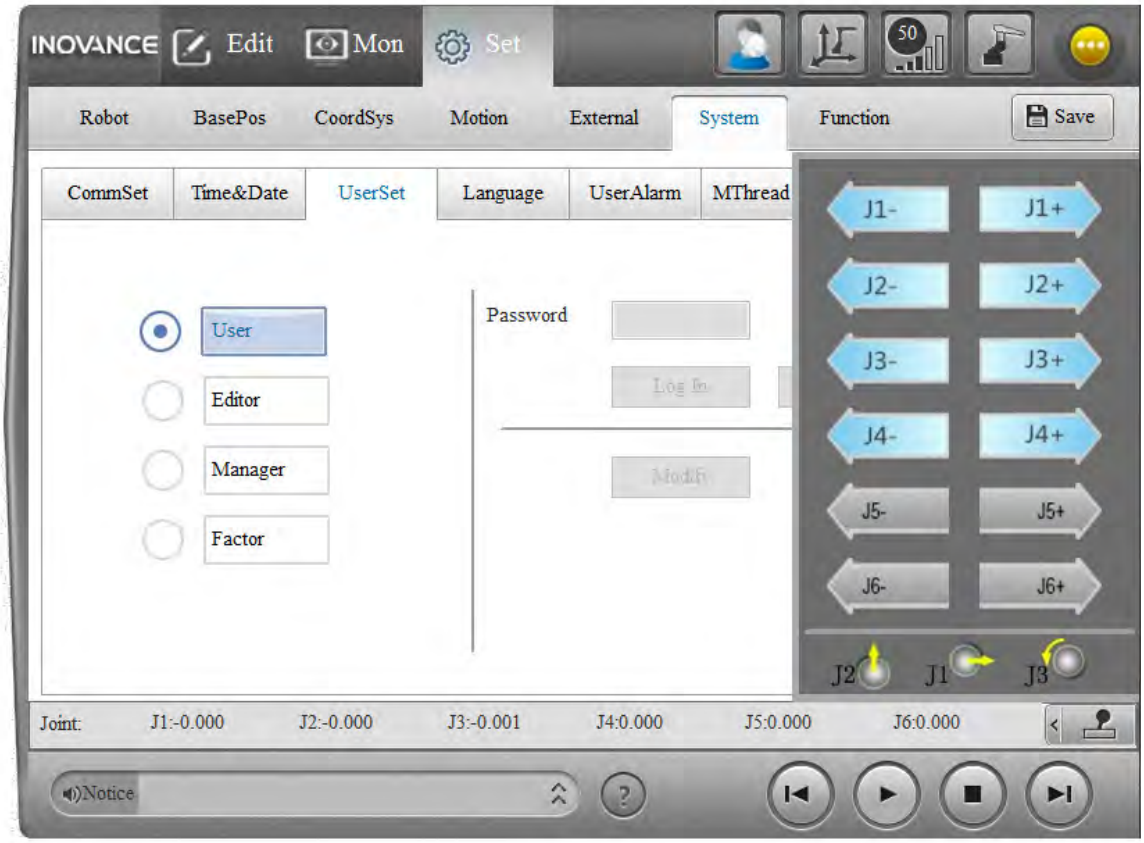


Figure – Switchover between speed/step selection

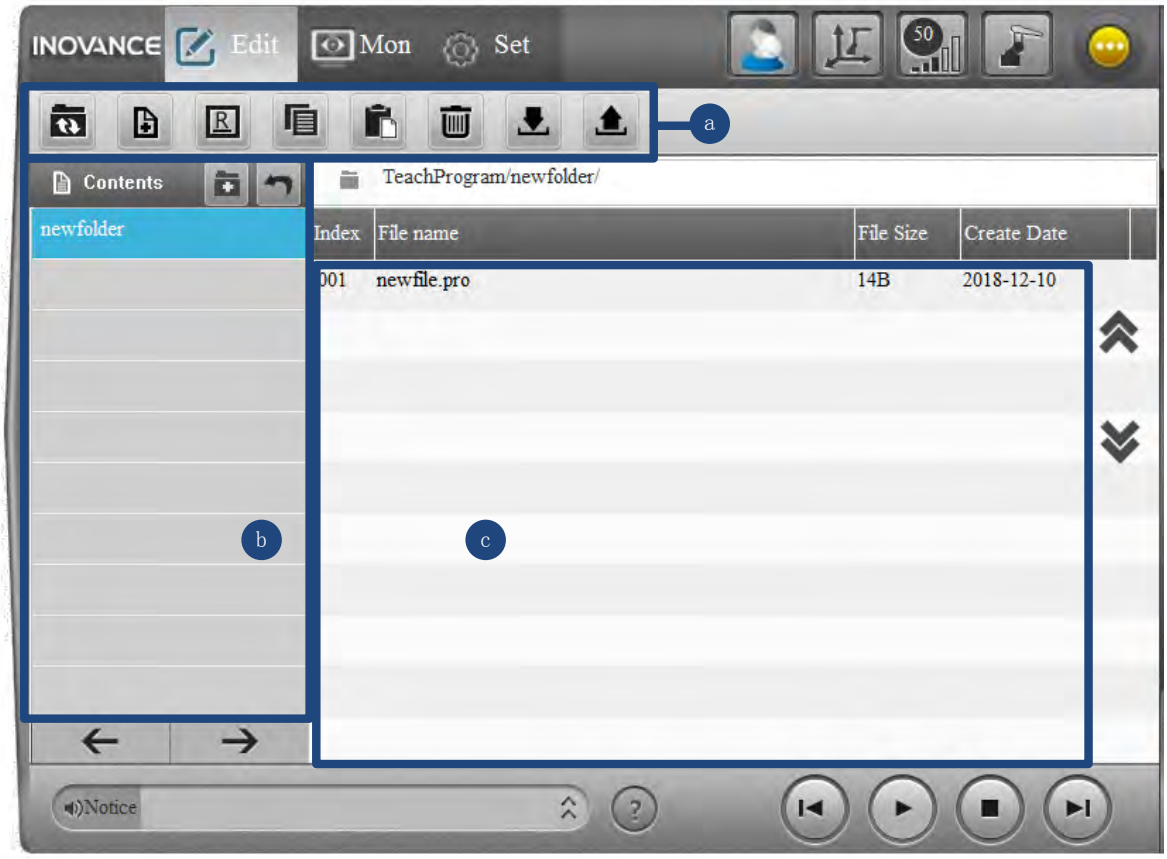
Step 2: Others are the same as speed movement. If you select the joint coordinate system, move after enablement.



2 Programming and Running

2.1 Programming Panel

The following figure shows the programming panel:



a) File editing toolbar: You can perform the following operations using the toolbar:

Refresh: Refresh the current directory.

New: Create a program file. (A folder is created using in the directory list.)

Rename: Rename a program file or folder.

Copy: Copy a program file or folder.

Paste: Paste a program file or folder.

Delete: Delete a program file or folder.

Import: Externally import a program file to the controller and in the offline case, to a local offline directory of InoTeachPad.

* The PC version of InoTeachPad software supports import by dragging.

Export: Export a program file from the controller. In the offline case, export a local offline directory of InoTeachPad.

b) Folder list

New: Create a folder.

Click: Show files in the current folder.

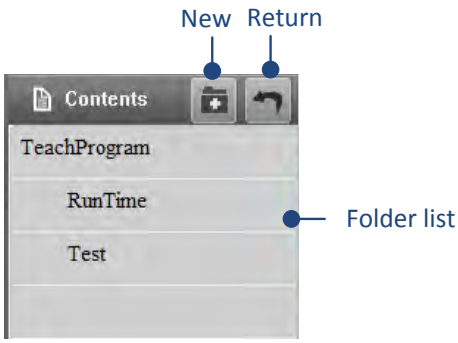
Double-click: Enter a folder.

Return: Return to the previous folder.

The rename, copy, paste and delete operations are performed using the buttons on the file editing toolbar.

■ **Folder specifications:**

A folder name consists of letters, numbers and underlines with a length of up to 16 characters, starting with a letter. Up to three layers of folder nesting are allowed. The number of files in every folder shall be limited to 512.



c) Program list and editing interface

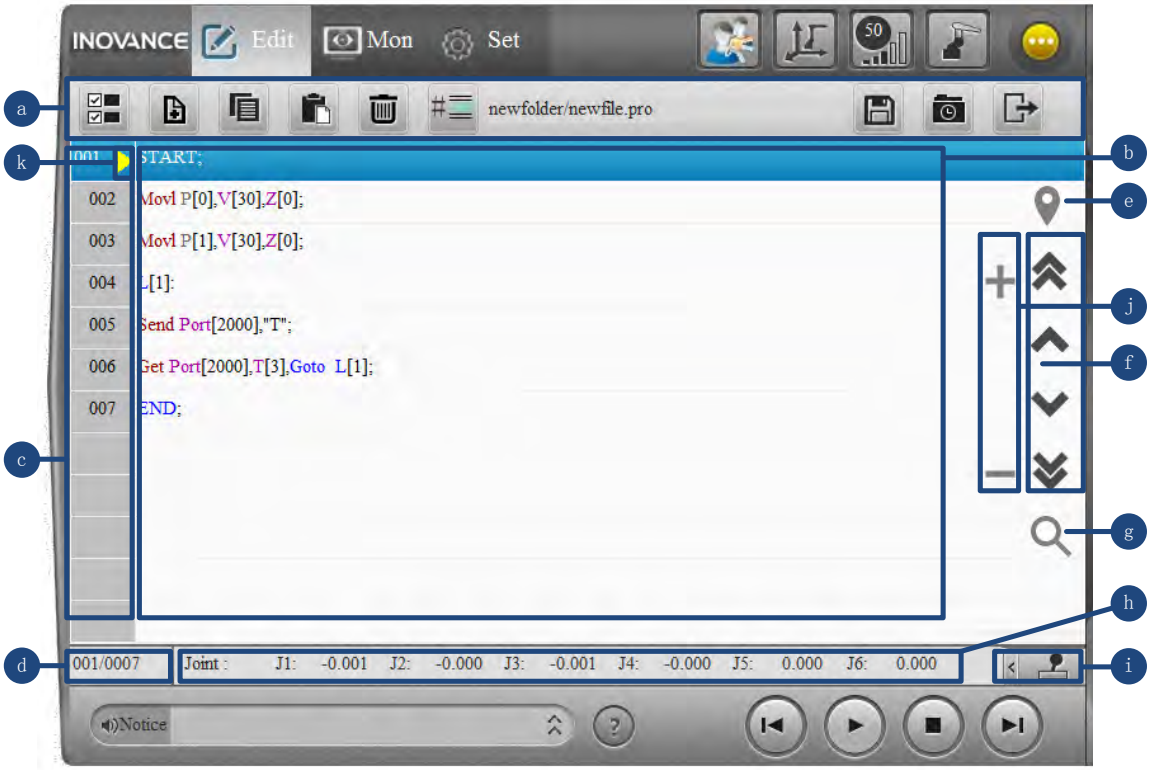
Program file naming specifications:

A filename consists of letters, numbers and underlines with a length of up to 32 characters, starting with a letter.

Letters are not case-sensitive.

The program list shows all programs in the current folder. Click the list header, and you can perform sorting by filename, size or date. and on the right are used to page up and down. Double-click a program to open it.

The interface is as follows:



No	Function	Description
a	Program editing toolbar	Perform editing operation on commands in the program file.
b	Program command editing area	Show specific contents of program commands. Click to select a line that then becomes blue. Double-click a line to modify commands in this line.
c	Program line No. area	Show the line No. of lines where every command is located.
d	Current line/Total number of lines	Show a selected command line No. and total number of lines.
e	Locate button	Enter a line No. to quickly jump.
f	Page turning button	Single-headed arrow: The cursor jumps to the previous or next line. Double-headed arrow: The program turns a page.
g	Search/Replace button	Find next: Find the next object. Replace: Replace the current object. Replace all: Replace all starting from the current line.
h	Coordinate display area	Show the current coordinate information of the robot (according to a coordinate system).
i	Teaching panel button	Show/Hide the teaching panel that is used to control robot motion.
j	Zoom button	Zoom the editing area (only applicable to ITP100 and the PC version of InoTeachPad).
k	Startup line identification	Identify a startup line with a yellow arrow that indicates the startup position of a program.

2.2 Running Panel

Switch to the running mode, and the system is automatically enabled. At this moment, you can run the current program.

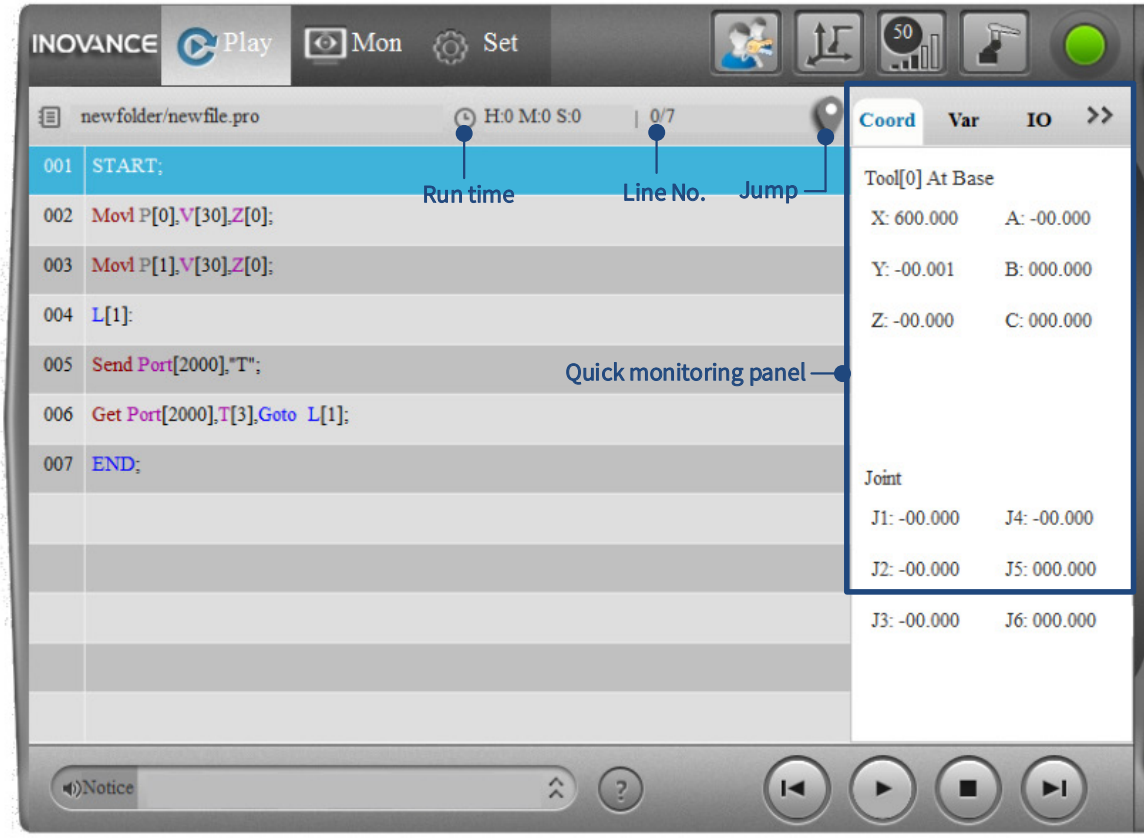
Note: For different InoTeachPad types, the switching button is located in different positions.

For ITP100 InoTeachPad, you can switch to the playback running mode using the mode switching knob on the electric cabinet.

For IRTP80 InoTeachPad, you can perform switchover using on InoTeachPad.

For the PC version of InoTeachPad, you can perform switchover using .

The following figure shows the Play interface.



Run time: Time from startup to stop. It is not the run time of only a single program. This is very helpful during continuous running of multiple programs (e.g. station appointment and primary subprogram call). It provides a total run time.

Jump: Used to set a cursor line. After a cursor line is set, a program starts to be executed from this line.

Line No.: Current line No./Total number of lines

Quick monitoring panel: Provide information about the current coordinate value and some collected data.

- Coord: Show values under the robot base coordinate system (Tool and User not included) and joint values.
- Var: Show the first ten items of collected BRD numerical variables.
- IO: Show the first ten items of collected IO numerical variables.

3 Settings

Initial settings have been completed in the factory. If you are a user, you generally need only to set the contents of [BasePos] and [CoordSys] items according to requirements before the product is put into operation. You can directly refer to the contents of the following two sections. If you are a manufacturer debugging person, you need to pay attention to more information.

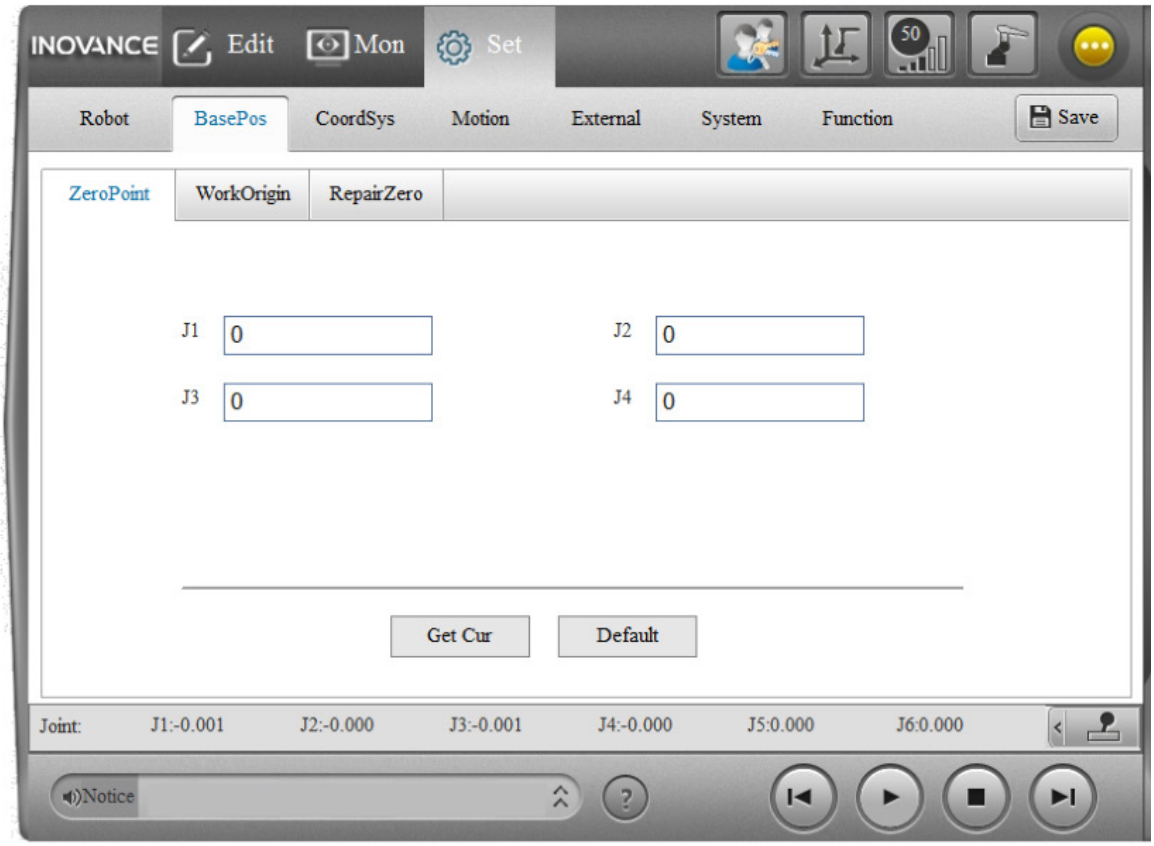
After modifying the current page parameters, remember to click **Save** to save the current page information!

3.1 BasePos

BasePos includes the ZeroPoint and WorkOrigin pages. For the SCARA robot, BasePos also includes the RepairZero page.

a) ZeroPoint

The following figure shows the ZeroPoint setting interface. Settings are finished in the following two steps:



Step 1: Adjust a robot joint to zero position using InoTeachPad. Click the **Get Cur** button, and the encoder values are displayed in the editing box of the panel.

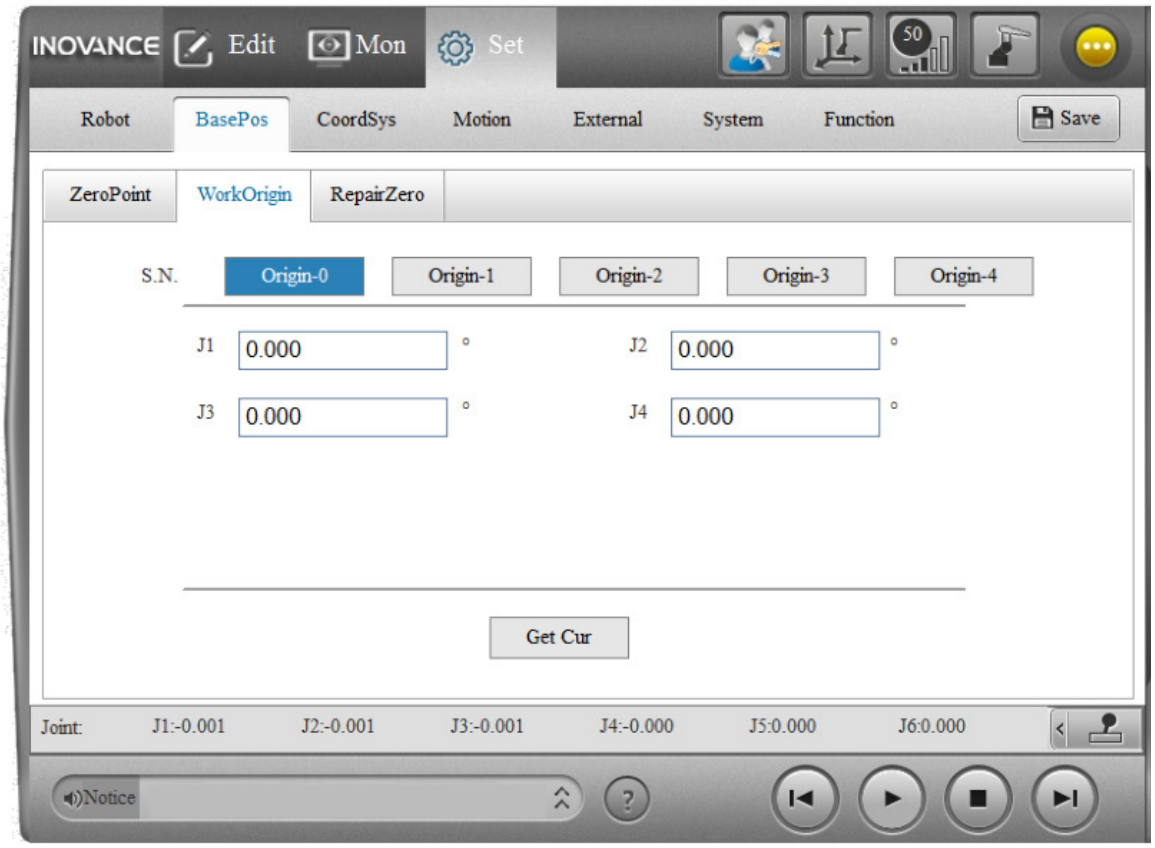
Step 2: Press the emergency stop button, click **Save**, and restart the system.

NOTE

◆ The J1-J6 encoder numeral range is -2^{21} to 2^{21}. If the encoder readings are too large at the zero point of the robot, it is recommended to clear the number of encoder turns and re-perform operations according to the preceding steps. For the operation of clearing the number of encoder turns, refer to related servo and encoder manuals.

b) Work Origin

A work origin is a user-defined position. You can return to a work origin in the program using the **Home** command.Five work origins can be set. A work origin can be manually entered or automatically obtained by clicking Get Cur after the robot is moved to an ideal position. Then click **Save**. The following figure shows the setting interface.



3.2 Coordinate System Settings

Tool and user coordinate systems are defined in CoordSys. Tool load, hand-held workpiece load and arm load must be configured for application that needs to give consideration to loads.

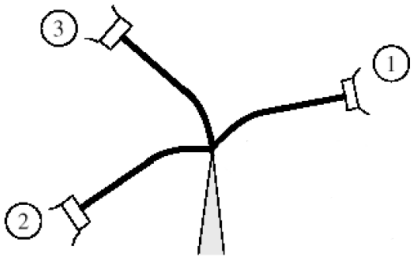
a) Tool coordinate system

The tool coordinate system is established on a tool. Its parameter is a value of the TCP relative to the end coordinate system of the robot body.

Method	Characteristics	Applicable Scenario
Direct	Directly enter coordinate system parameters.	Coordinate system parameters are known.
3P TCP	Obtain a tool position by 3P.	Position values of the tool coordinate system need to be manually calibrated.
5P TCP	Obtain a tool position by 5P.	Position values of the tool coordinate system need to be manually calibrated. It is more accurate than 3P TCP because more points are gotten.
3P TCP+ZX	Obtain a tool position by 3P and then a tool pose via three more points.	Position and pose values of the tool coordinate system need to be manually calibrated.
5P TCP+ZX	Obtain a tool position by 5P and then a tool pose via three more points.	Position and pose values of the tool coordinate system need to be manually calibrated. It is more accurate than 3P TCP+ZX because more points are gotten.
Lock screw 4P	By selecting four points in the horizontal plane, successively operate the robot so that the end of the electric screwdriver is aligned.	It applies to the lock screw robot.

Direct: Directly enter parameters.

3P TCP: After installing a tool at the end of the robot, adjust the tool pose so the TCP is aligned with one pose in three different directions. Record using the Get Pos buttons of the teaching software. Click **Generate** to obtain tool coordinate system parameters.



3P

The operations on the interface for 3P TCP are as follows:

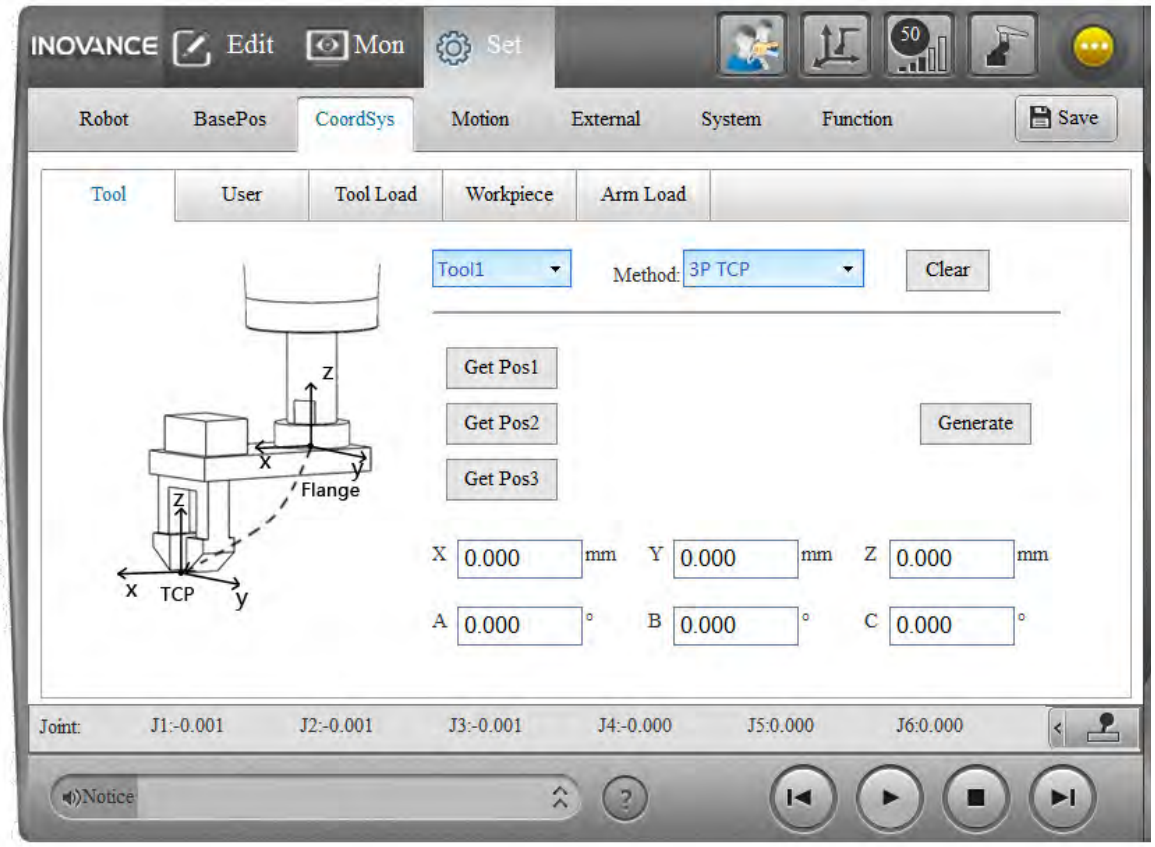
Step 1: Control robot motion so that the TCP is aligned with the reference point in Direction 1. Click Get Pos1.

Step 2: Control robot motion so that the TCP is aligned with the reference point in Direction 2. Click Get Pos2.

Step 3: Control robot motion so that the TCP is aligned with the reference point in Direction 3. Click Get Pos3.

Step 4: Click Generate, and tool coordinate system parameters are automatically generated.

Step 5: Click Save.



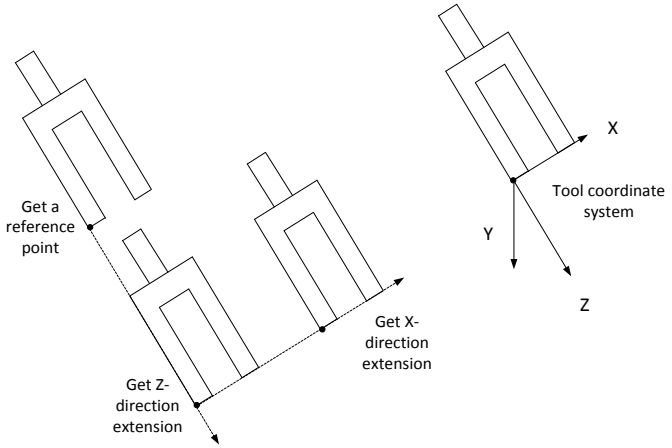
NOTE

◆ Three different poses selected by 3P shall have an as large gap as possible. A gap of over 20° is recommended. This holds true to 5P.

After click Generate, error parameters are displayed. It is generally deemed to be accurate that the maximum error is smaller than the absolute precision of the robot body+0.1mm. When the maximum error is too large, it is recommended to get points to generate tool coordinate system parameters again.

When the tool end of the SCARA and Delta robots is coaxial with the end axis (i.e. there is only a Z value), this Z position parameter cannot be obtained via the 3P TCP. This holds true to 5P.

3P TCP+ZX: Take three more points to calibrate poses of the tool coordinate system on the basis of 3P TCP. The Z extended point and the reference point compose the Z direction of the tool coordinate system. The X and Z extended point compose the X direction of the tool coordinate system. The Y direction is obtained from the Z and X directions by the right-hand rule. See the following figure.



The operations on the interface are as follows:

Step 1: Control robot motion so that the TCP is aligned with the reference point in Direction 1. Click Get Pos1.

Step 2: Control robot motion so that the TCP is aligned with the reference point in Direction 2. Click Get Pos2.

Step 3: Control robot motion so that the TCP is aligned with the reference point in Direction 3. Click Get Pos3.

Step 4: Control robot motion. Select the TCP at any moment as the reference point. Click Get RefPos.

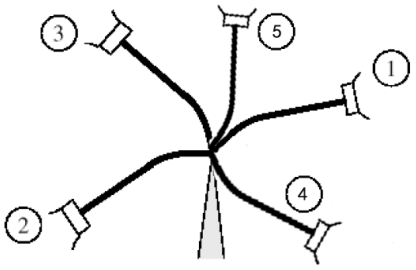
Step 5: Control robot motion so that the tool extends a distance in Z direction. Click GetZ-Extend.

Step 6: Control robot motion so that the tool extends a distance in X direction. Click GetX-Extend.

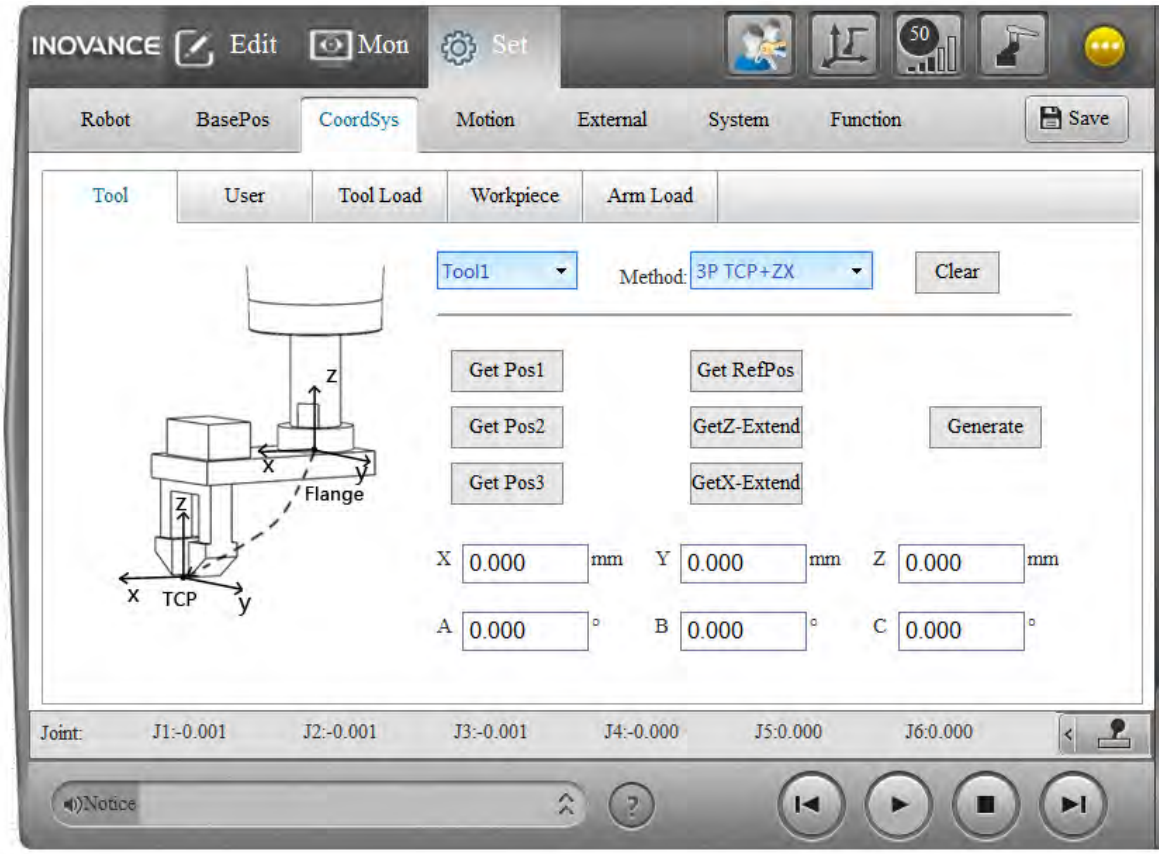
Step 7: Click Generate, and tool coordinate system parameters are automatically generated.

Step 8: Click Save.

5P TCP: It is similar to 3P except that the TCP is aligned with the reference point in five different directions.



Teaching tool coordinate system by 5P



5P TCP+ZX: Refer to 3P TCP+ZX. The difference is that taking the first three points is changed to being aligned with the reference point in five different directions and getting points.

b) User coordinate system

The setting parameters of the user coordinate system are coordinates of the user coordinate system in the base coordinate system.

Method	Characteristics	Applicable Scenario
Direct	Directly enter coordinate system parameters.	Coordinate system parameters are known.
3P	Calibrate the user coordinate system using three points.	Values of the user coordinate system need to be manually calibrated.
Rotate	Make marks on the turnplate. Rotate the turnplate and get points for teaching.	A turnplate is provided and the user coordinate system is located at the center of the turnplate.

Direct: Directly enter user coordinate system parameters.

3P:

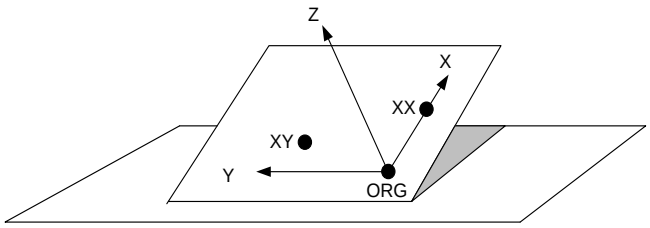
Step 1: Take the origin of the user coordinate system as Point 1. Click Get Pos1.

Step 2: Take a point in the positive direction of the X-axis of the user coordinate system as Point 2. Click Get Pos2.

Step 3: Take a point in the Y+ direction on the XY plane of the user coordinate system as Point 3. Click Get Pos3.

Step 4: Click Generate, and user coordinate system parameters are automatically generated.

Step 5: Click Save.



Points defined in the user coordinate system
ORG: Origin position
XX: Point on X-axis
XY: Point on the XY plane

Rotate:

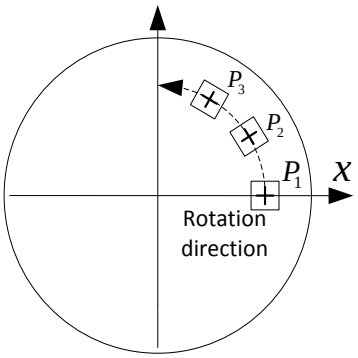
Step 1: Mark a fixed point on the turnplate. After making the TCP be aligned with this mark, click Get Pos1.

Step 2: Rotate by an angle. After making the TCP be aligned with this mark, click Get Pos2.

Step 3: Rotate by any angle again. After making the tool be aligned with this mark, click Get Pos3.

Step 4: Click Generate, and user coordinate system parameters are automatically generated.

Step 5: Click Save.



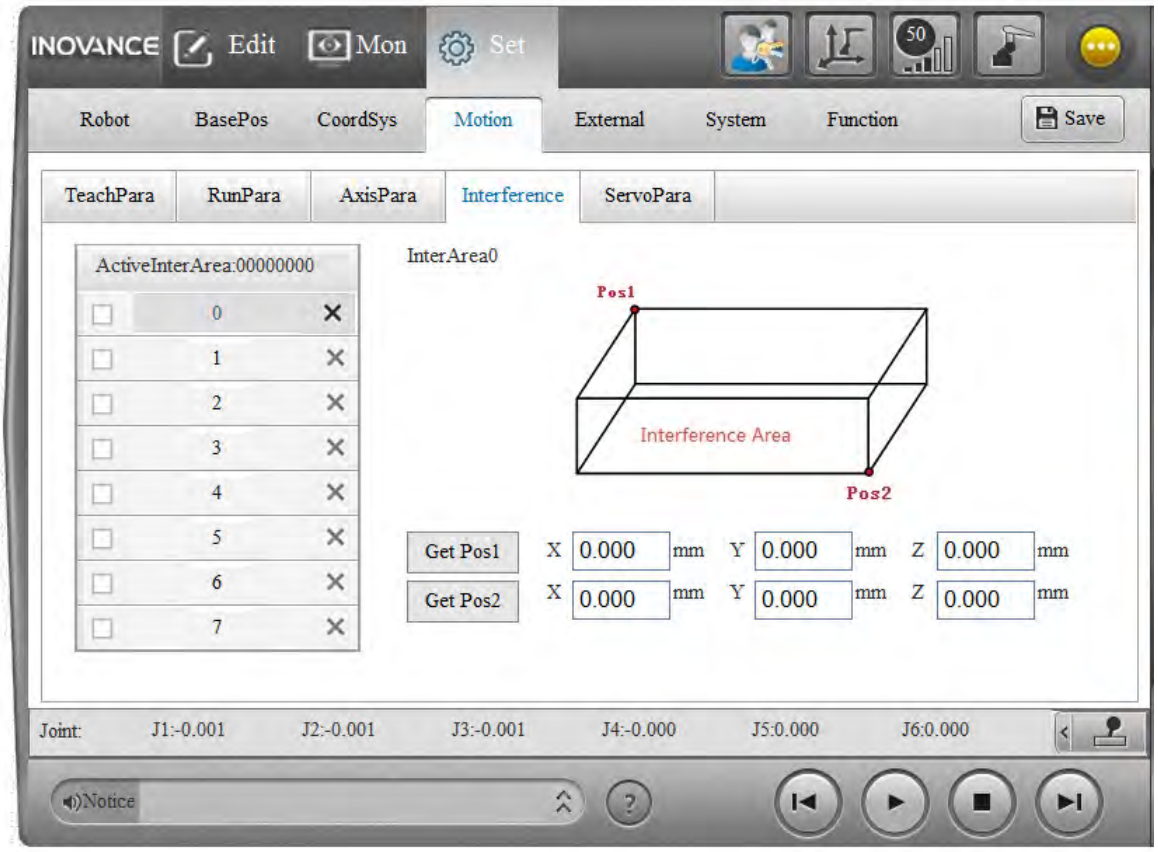
3.3 Motion Settings

a) Interference area settings

Interference areas are areas where the robot tool end is prohibited from reaching. Eight groups of interference areas can be set. Every group of interference areas is a cuboid area determined by XYZ of two points of a diagonal. Interference areas take only positions into account. When interference areas are activated, an error is generated during entry into interference areas. Multiple interference areas can be activated at the same time.

Editing interference areas: Select a number, and interference area values are displayed on the right. Edit interference areas.

Activating interference areas: Put a checkmark in a checkbox before interference areas to activate interference areas.



3.4 Peripheral Configurations

a) I/O configurations

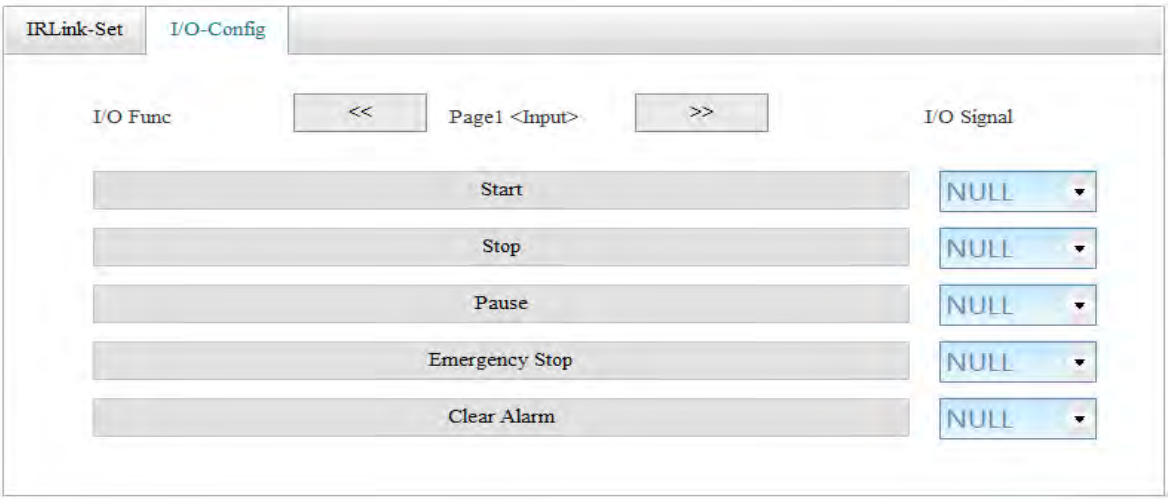
I/O configurations apply to remote I/O applications and are valid only when control equipment is selected as Remote IO.

I/O configurations are to link I/O signals to some robot functions to achieve the objective of controlling and monitoring the robot via I/O in the station appointment process.

Configurable I/O functions are as follows:

I/O Function		Description
Input (Used for external control)	Start	Start is controlled by external signals.
	Stop	Stop is controlled by external signals.
	Pause	Pause is controlled by external signals.
	Emergency Stop	Emergency stop is controlled by external signals.
	Clear Error	The sending of the **Clear Error** command is controlled by external signals.
	I/O Program	Five prioritized I/O programs can be set. The same program can be appointed only once. When the current program is running, re-appointment is invalid.
	Speed +	Increase one speed multiplying ratio level. Levels are 1%, 5%, 10%, 15%...100%, with an increment of 5%.
	Speed -	Decrease one speed multiplying ratio level. Levels are 1%, 5%, 10%, 15%...100%, with an increment of 5%.

Output (Used for display status)	Error Status	Output whether the program sends an error.
	Run Status	Output whether the program runs.
	Stop Status	Output whether the program is stopped.
	Start Finished	Output whether the controller startup is finished.
	Enable Status	Output whether the current item is enabled.



Precautions:

- I/O configurations require that Remote IO or Remote Modbus shall be selected as a control device. (You can set a control device to Remote IO or Remote Modbus under [Set] -[Others] - [Control Authority].
- Any output port linked to I/O functions is occupied by the system and cannot be controlled via a page or command.
- Output running/stop is program running/stop, not robot motion/stop. Once a program is started, it is always in running state unless a stop command is sent.

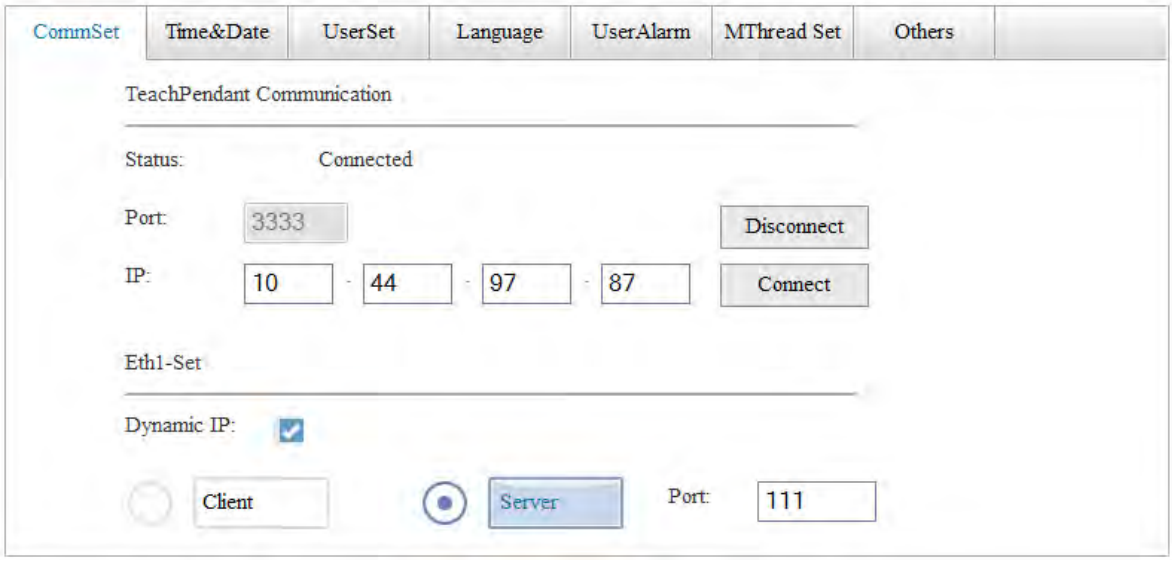
3.5 System Settings

a) Communication settings

InoTeachPad has two communication modes that correspond to the controller network ports EtherNet1 and EtherNet2. EtherNet1 is a dynamic IPnetwork port by default, that corresponds to the remote connection mode.. EtherNet2 is a static IP network port that corresponds to the direct connection mode.

Direct connection: InoTeachPad is directly connected to controller EtherNet2 using a network cable. The controller IP address is 192.168.23.25.

Remote connection: The controller and InoTeachPad are both in the local network, and they connect indirectly.



About EtherNet1 and EtherNet2:

For the IRCB300 electric cabinet, the LAN corresponds to EtherNet1 and the PC and InoTeachPad slot correspond to EtherNet2.

IP address setting of PC:

For the PC version of InoTeachPad, you must set an IP address on the PC so that it and the controller are located in the same subnet to connect to the controller. The following shows the modification procedure:

Step 1: Access Control Panel > Network and Internet > Network and Sharing Center, and click Local Area Connection.

Step 2: On the Local Area Connection Status page, click Properties.

Step 3: On the Local Area Connection Properties page, click Internet Protocol Version 4.

Step 4: Select "Obtain an IP address automatically". Use direct connction with IP address 192.168.23.xxx and subnet mask 255.255.255.0, and click OK to finish the setting.

Dynamic IP:

Dynamic IP of EtherNet1 is on by default, indicating that EtherNet1 is a dynamic port. However, you can set it to a static port as follows: Cancel the Dynamic IP option of EtherNet1 via EtherNet2 direct connection (e.g. via hand-held InoTeachPad). Define an IP address.

Application: When an external device needs to be used to connect a controller during the use of InoTeachPad, InoTeachPad is generally used to connect controller EtherNet2 and the external device to controller EtherNet1. At this moment, set the IP address to a static IP address by closing Dynamic IP. In this way, the external device can establish communication connection with the controller.

NOTE

- When a dynamic IP address is changed to a static IP address, the static IP address and the default static IP address shall not be located in the same subnet, i.e. the static IP address shall not be set to 192.168.23.xxx.
- After Dynamic IP is adjusted, the controller must be restarted before the setting takes effect..

Client and server:

The robot can be selected as a client or server during communication with an external device. This function is mostly used for vision and designates the role of the robot in the vision function as a client or server.

Server:

The robot controller as a server supports connection to up to five clients.

During the setting, you must designate a port No. that is in the range of 1024 to 9999, where 3333 cannot be used. After finishing the setting, restart the robot system, and the server is automatically turned on. At this moment, any external device as a client must be turned on before the robot program runs, otherwise an error is generated.

Client:

The robot controller is used as a client and an external device is used as a server. Note that in the vision programming application, you need to select a connected IP address using the Open Socket command in the program.

NOTE

◆ Support both Chinese and English. Language switching requires a restart!

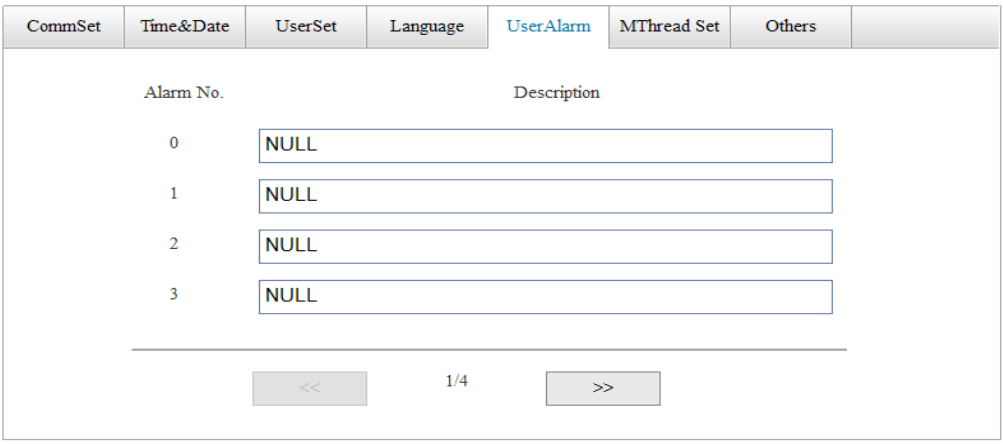
b) Language

To be valid, Chinese-English switching requires a restart!

c) User-defined error

16 user-defined errors can be set. Every error has up to 40 characters.

The Alarm command is used to call an error during programming. An error is also synchronously recorded in an error log.



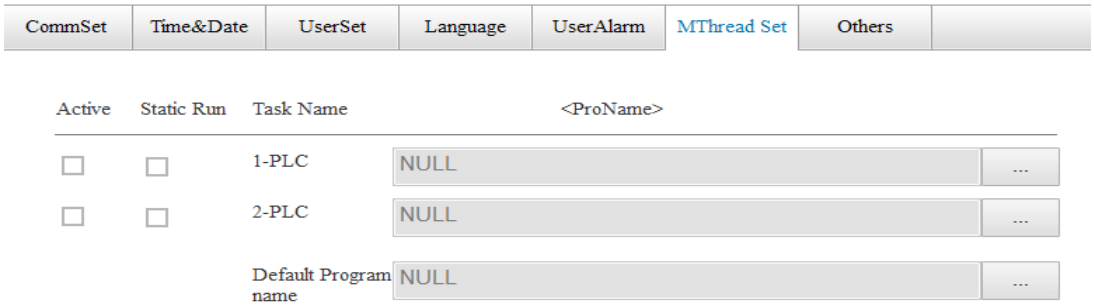
d) Multitask configurations

PLC[1] and PLC[2] are setting-type tasks.

About Active: Whether to activate this task.

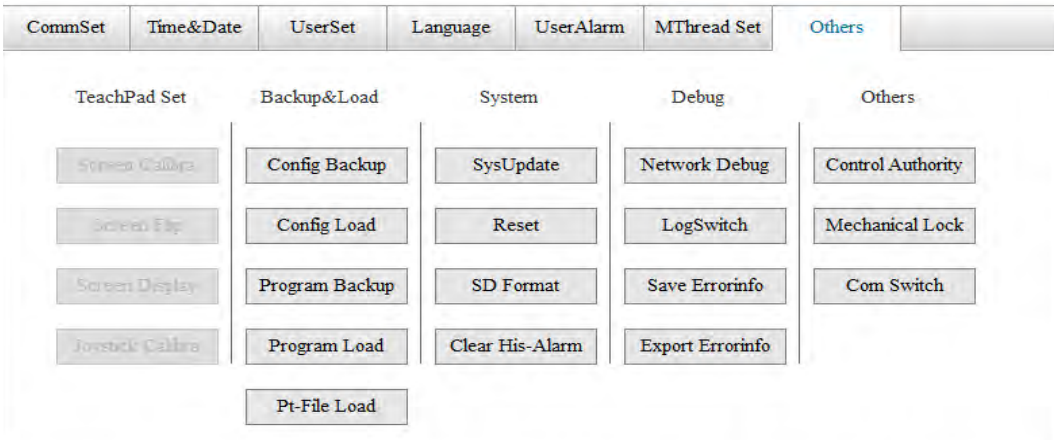
About Static Run: Static startup indicates that tasks are started immediately upon power-on of the robot and always run without being controlled by other commands. Non-static tasks are not started immediately upon power-on of the robot and are controlled by Start, Stop and Pause along with the main task (Task 0).

Default Program: Set a default program. When the robot controller is started, this program is started by default.

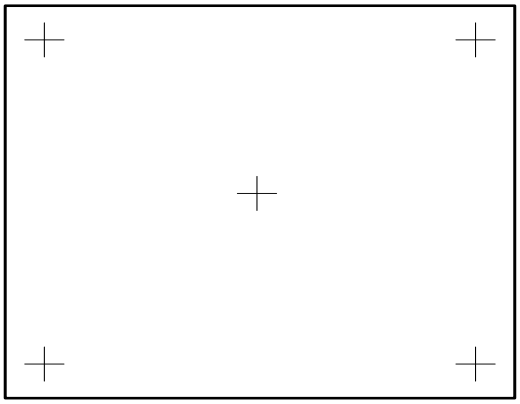


e) Other settings

Other settings include TeachPad Set, Back&Load, System, Debug and Others. TeachPad Set is valid for hand-held InoTeachPad only.



Screen Calibra: This function is used to calibrate the screen when screen touch is inaccurate. Use a nib to click "+" center points that appear successively (at the center and four corners of the screen). Then click any blank to finish settings.



Screen Flip: All default InoTeachPad displays apply to the left hand-held screen, but for ITP100 InoTeachPad, the display applies to the right hand-held screen using screen flip.

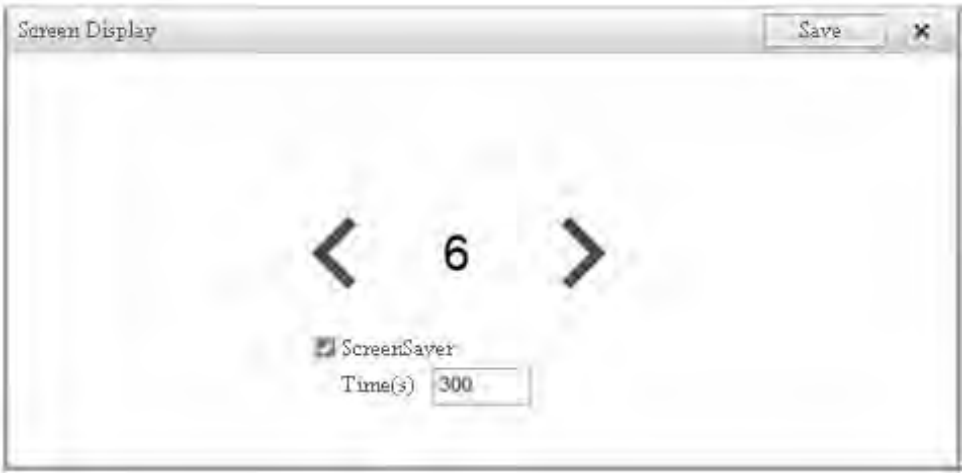
NOTE

- After confirmation, the system automatically enters the Screen Calibra interface and you must re-calibrate the screen

Brightness and screensaver:

Brightness: A total of six brightness levels 1-6 is available. You can freely select a brightness.

Screensaver: You can selectively activate the screensaver function and set screensaver time.



Config Backup

*About configuration file: All robot-related setting parameters are saved in the RobotInfo.cfg file.

Insert a USB drive into the controller/electric cabinet. Click this button to back up the robot configuration file to the root directory of the USB drive.

NOTE

- Before operation, insert a USB drive into the controller/electric cabinet and check connection status. If the monitored communication status in the teaching software displays "The USB controller has been inserted into the device and successfully mounted", it indicates that communication is normal. Otherwise, check the connection. Keep the USB drive in good communication during operation.

Config Load

Prepare the RobotInfo.cfg configuration file under the root directory of the USB drive in advance. Insert the USB drive into the controller/electric cabinet. Click this button to load it to the controller. Then power on the controller again to be valid.

NOTE

- Before operation, insert a USB drive into the controller/electric cabinet and check connection status. If the monitored communication status in the teaching software displays "The USB controller has been inserted into the device and successfully mounted", it indicates that communication is normal. Otherwise, check the connection. Keep the USB drive in good communication during operation.
- Loading configuration files of different models is not allowed.

Program Backup

*About the contents of program backup and loading:

	Before Version 15	Version 15 and later
Program backup folder	1) TeachProgram 2) PalletInfo	1) TeachProgram 2) PalletInfo 3) TecParameter
Program loading folder	1) TeachProgram 2) PalletInfo	1) TeachProgram 2) PalletInfo 3) TecParameter

TeachProgram: Program file folder containing all .pro program file.

PalletInfo: Pallet file folder containing information about the stacking and pallet. It needs to be used when a pallet is used.

TecParameter: Process folder containing information about the screw locking and gluing process.

Insert a USB drive into the controller/electric cabinet. Click this button to back up the program in the SD card to the root directory of the USB drive.

NOTE

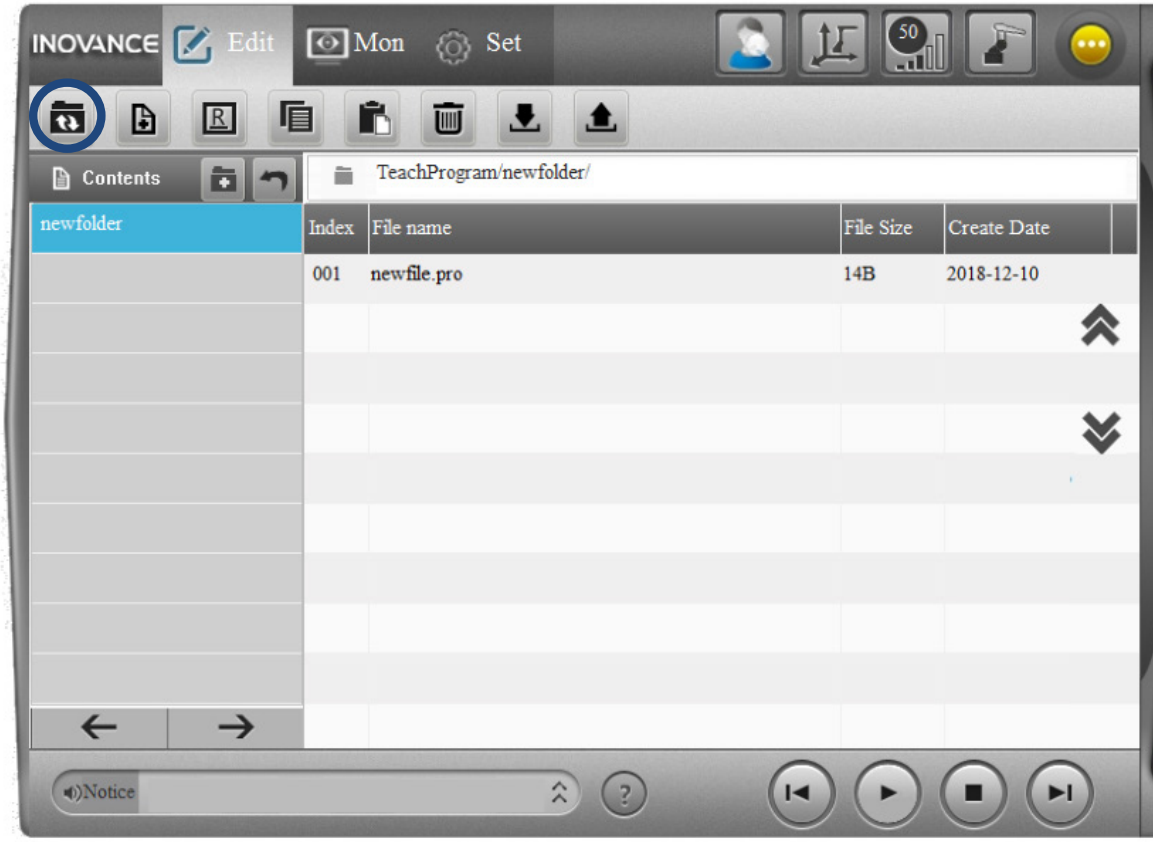
◆ Before operation, insert a USB drive into the controller/electric cabinet and check USB and SD card connection status. If the monitored communication status in the teaching software displays "The USB controller has been inserted into the device and successfully mounted" and "The SD card has been inserted and successfully mounted", it indicates that communication is normal. Otherwise, check the connection. Keep the SD card and USB drive in good communication during operation.

Program Load

Prepare a corresponding file under the root directory of the USB drive in advance. Insert the USB drive into the controller/electric cabinet. Click this button to load the program in the USB drive to the SD card. After loading is finished, enter the Edit interface for the program. Click the Refresh button at the top left corner to refresh the display.

NOTE

◆ Before operation, insert a USB drive into the controller/electric cabinet and check USB and SD card connection status. If the monitored communication status in the teaching software displays "The USB controller has been inserted into the device and successfully mounted" and "The SD card has been inserted and successfully mounted", it indicates that communication is normal. Otherwise, check the connection. Keep the SD card and USB drive in good communication during operation.



Pt-File Load

* About point file: A point file uses .pt as a name suffix. The file contents are data information of position variables. One line indicates one piece of position variable information. For the format of every line, refer to the definition of "position variable". Every line is divided into three subparagraphs. The first six parameters compose the first paragraph and are coordinate system values of the robot. The middle four parameters compose the second paragraph and are arm parameters. The last three parameters compose the third paragraph and are coordinate system No., tool No. and user No. Paragraphs are separated with ";". Numbers in a paragraph are separated with ",".

An example is shown below:

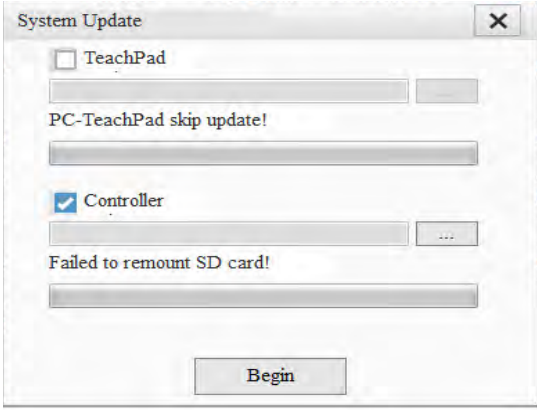
```
1.000000,  2.123355,  3.354556,  56.245514,  -7.985212,  6.354226;-1,0,0,0;1,0,0;
2.000000,  7.156114,  9.347547,  52.245514,  -17.9512,  6.354226;-1,0,0,0;1,0,0;
```

Prepare a point file in a USB drive in advance. Insert the USB drive into InoTeachPad. Browse a target point file and load it to the robot control system.

System Update:

Operation Method:

- Check an item(s) to perform upgrading. (You can check them at the same time. InoTeachPad and the controller can be upgraded simultaneously.)
- You can select a new version of the software package using the Browse button.
- After a selection is made, click Begin to perform upgrading.



NOTE

- The PC version of InoTeachPad cannot be upgraded. Only hand-held InoTeachPad can be upgraded.
- Do not cut off the power supply during controller upgrading, otherwise it may result in abnormality and the controller can be recovered only by refurbishing it.

SD Format:

Format the SD card on the controller into a program storage card suitable for the robot. This operation will clear the program in the SD card. Therefore, it is recommended to back up the program before operation.

NOTE

- Check the SD card communication status in advance. If the monitored communication status in the teaching software displays "The SD card has been inserted and successfully mounted", it indicates that communication is normal. Otherwise, check the connection.

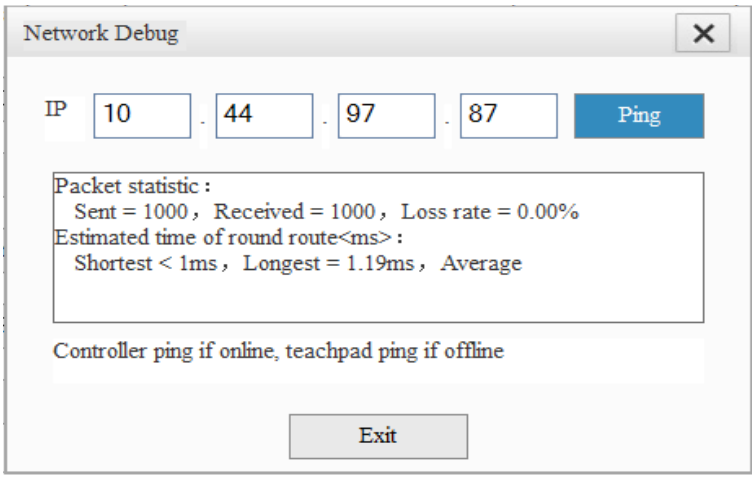
Clear His-Alarm:

Clear monitoring logs and error records.

Network Debug:

Debug communication of InoTeachPad and controller with other devices. It is equal to the Ping function.

Enter an IP address and click this button.

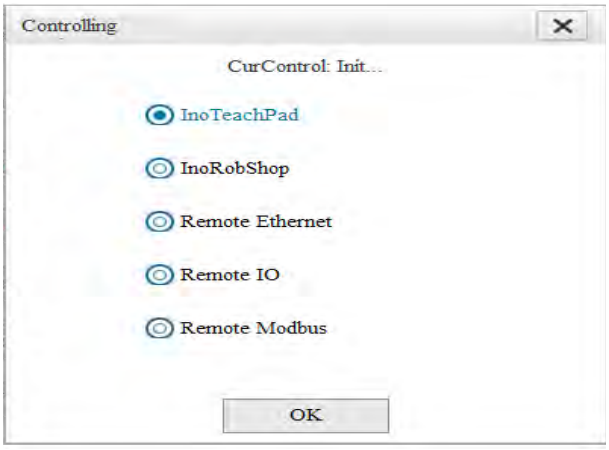


NOTE

- When InoTeachPad is connected to the controller, it is the controller Ping function that is used to detect network connection between the controller and other devices.
- When InoTeachPad is not connected to the controller, it is InoTeachPad Ping function that is used to detect network connection between InoTeachPad and other devices.

Control Authority:

Click the Control Authority button, and an interface appears. You can select which device has the control authority on the controller.



When InoTeachPad has no control authority, a "lock" identification appears on the teaching software toolbar. You cannot control the robot via InoTeachPad (including modifying parameters and running programs). In this case, InoTeachPad only plays a role of monitoring.

When Remote IO or Remote Modbus has the control authority, a startup speed percentage for the first running must be indicated in the startup speed bar. After the speed is modified via external I/O (speed +/- in remote I/O mode), this value becomes invalid and the robot runs at a set speed. This startup speed is generally smaller than 100, for example, in station appointment mode to ensure slow running for the first time.

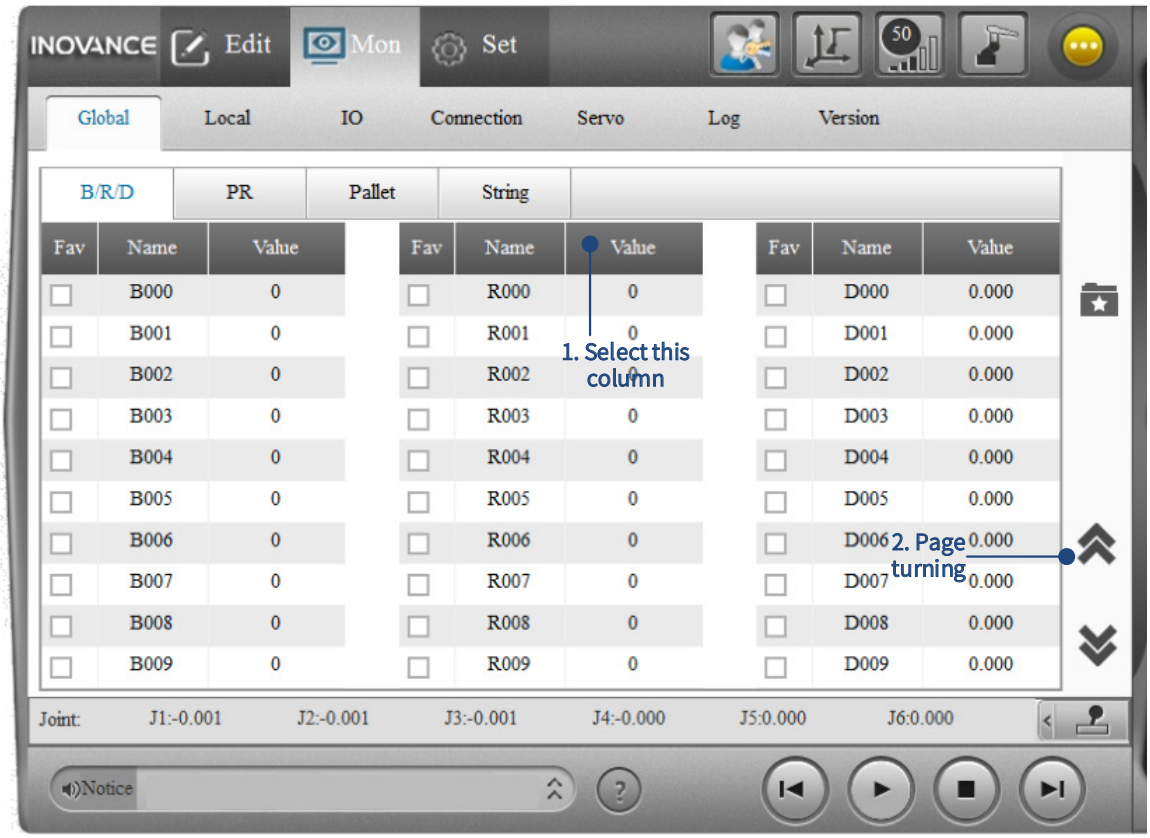
4 Monitoring

The Mon page is used to monitor various variables and various robot status properties.

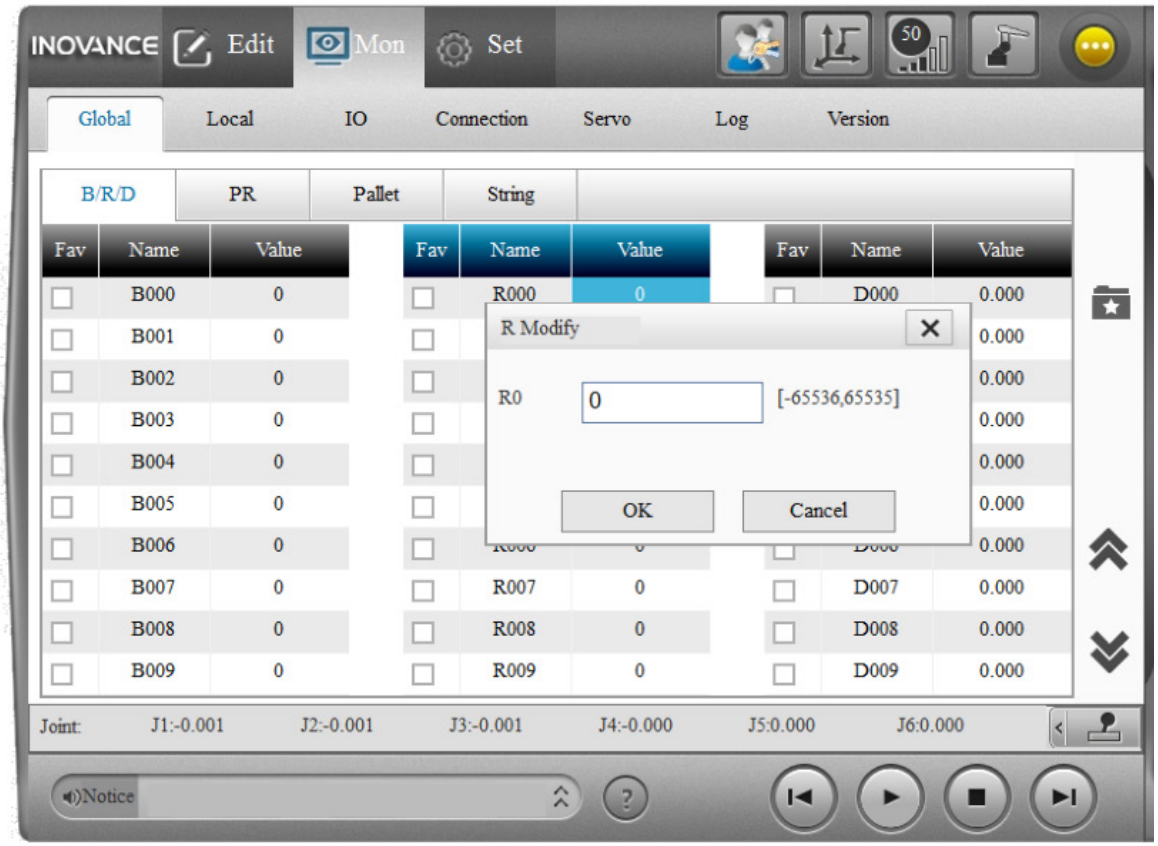
4.1 Basic Operations on the Mon Page

Several common basic operations on the Mon page are as follows:

Page turning: When a list header is blue, it indicates that this list is selected. You can turn a page of the current list using and on the right.

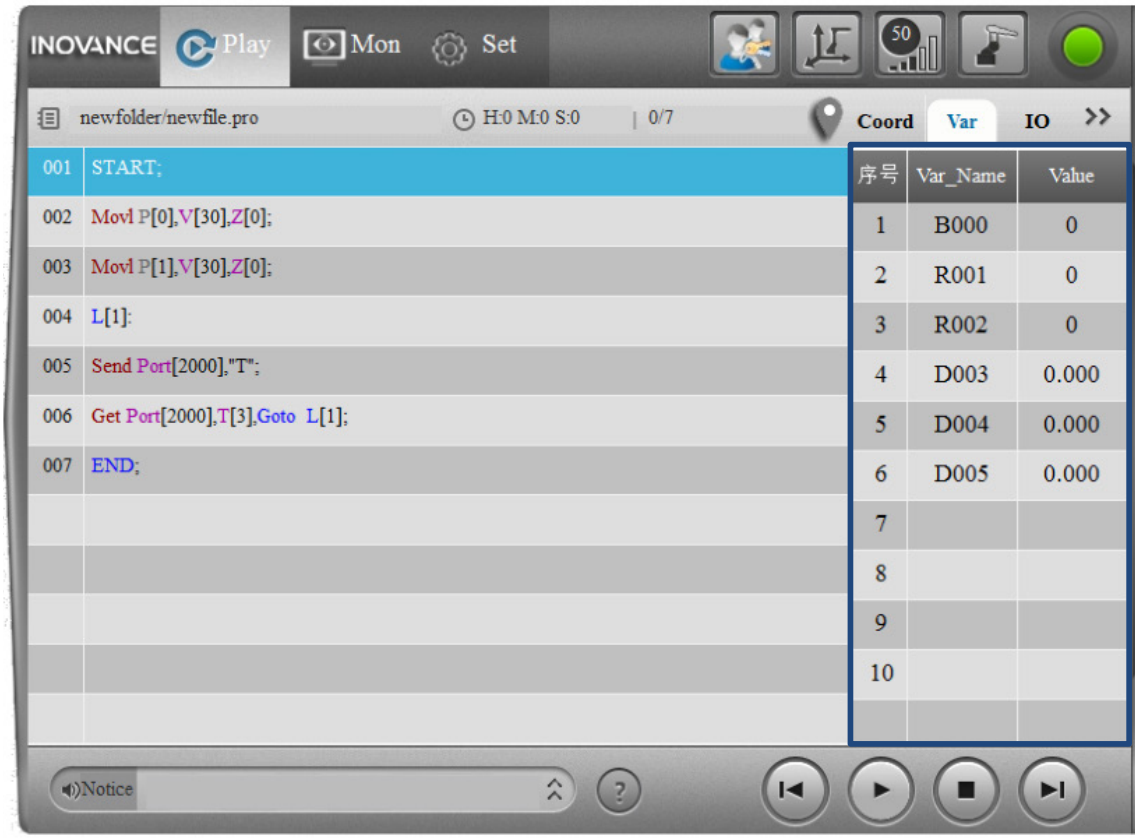


Modify: Double-click an object to directly modify it.



Fav: For global value variables and I/O, you can perform/cancel collection using a checkbox before a variable name. on the right is used to switch between all display and the display of only collected items.

In addition, the first ten collected items are displayed on the quick monitoring panel of the Play interface.



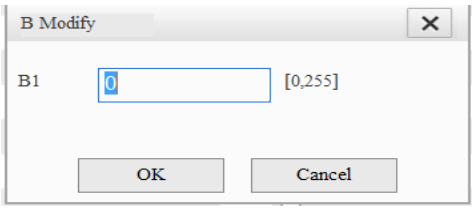
(The first ten collected variables and the first ten collected I/Os are displayed.)

4.2 Global Variable Monitoring

Global variables include numeral variables, translation variables, pallet variables and string variables.

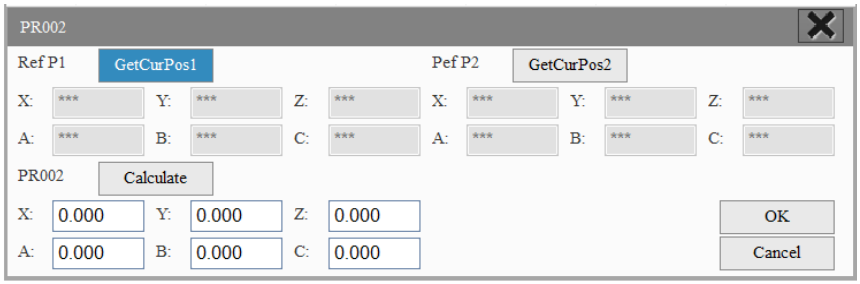
a) Global numeral variables

Double-click a variable, and the following window where you can modify this variable value appears.



b) Global translation variables

Double-click to modify a translation variable. You can directly enter a value or calculate it by taking two points.

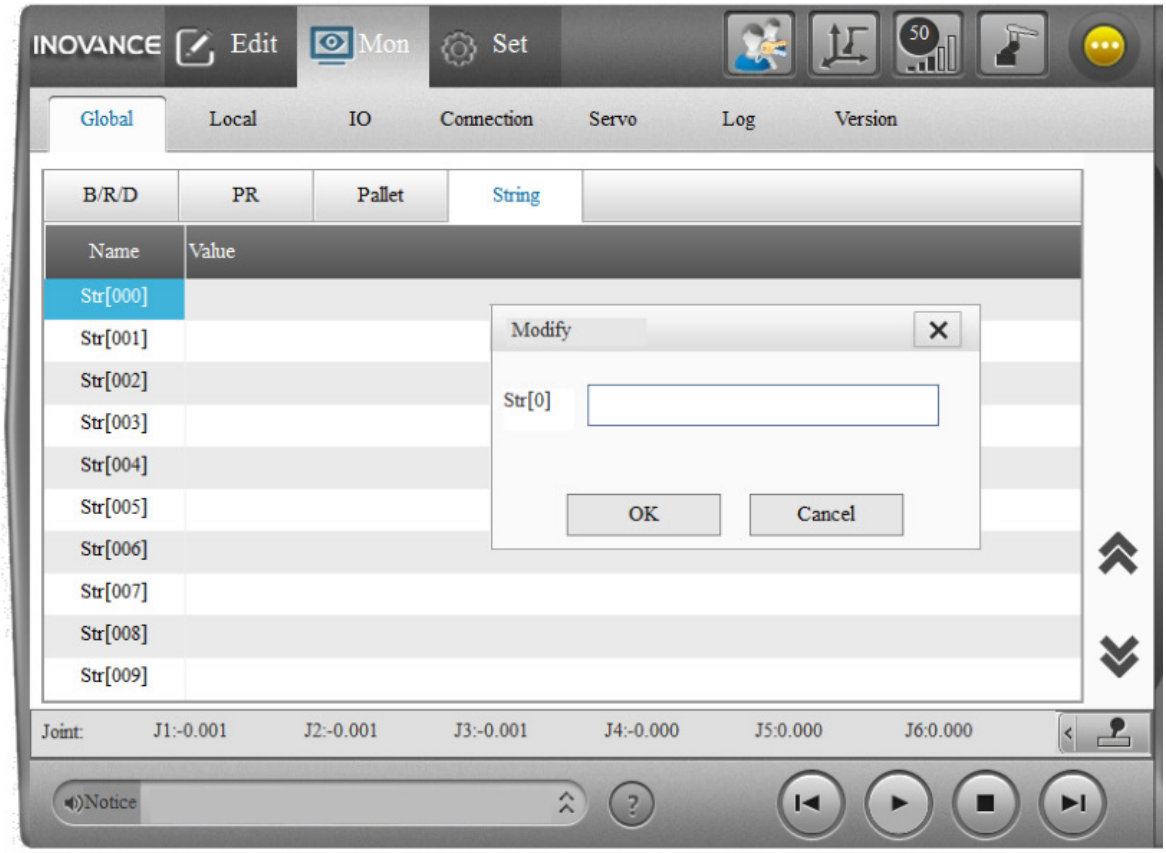


c) Global pallet variables

Global pallet variables are special pallet variables made using the stacking process.

d) Global string variables

Double-click to directly modify a variable.

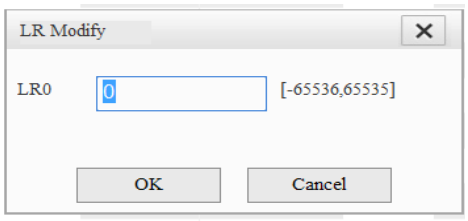


4.3 Local Variable Monitoring

Local variables include local numeral variables, position variables, local translation variables and local pallet variables.

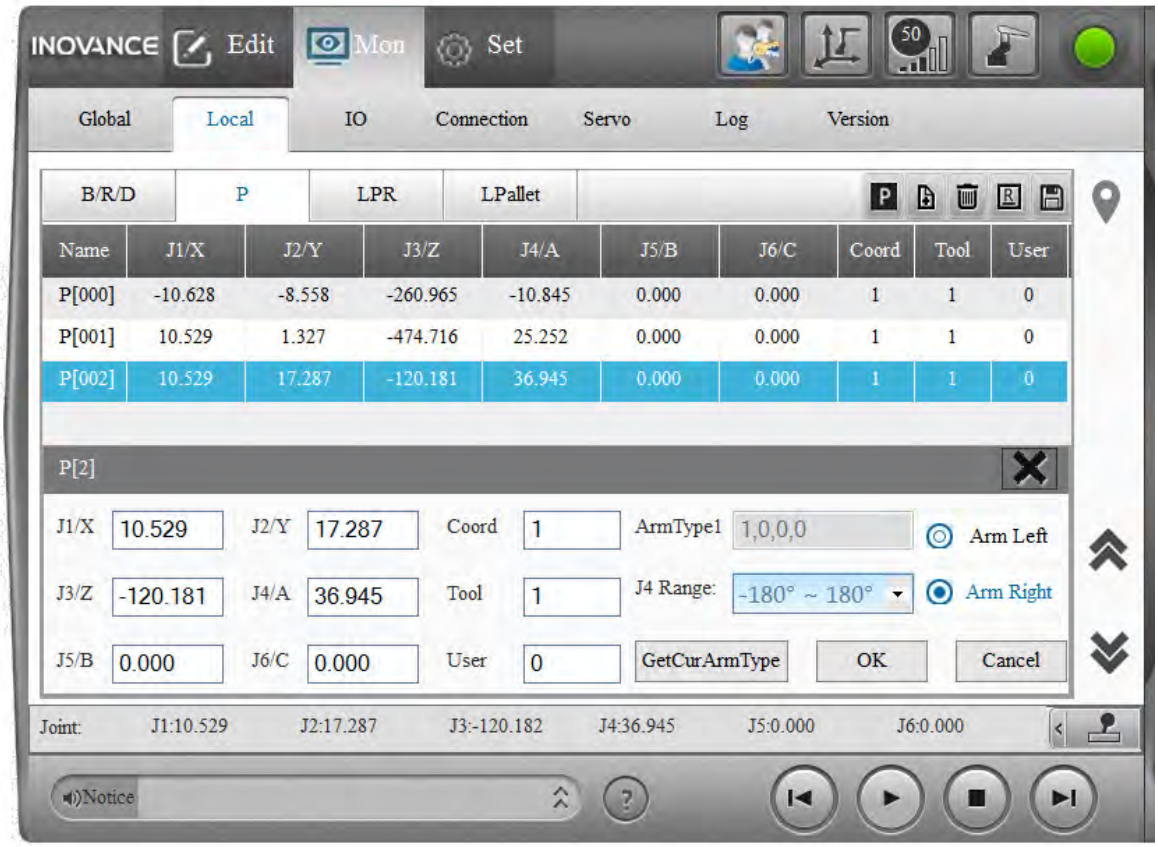
a) Local numeral variables

Double-click a variable, and the following window where you can modify this variable value appears.

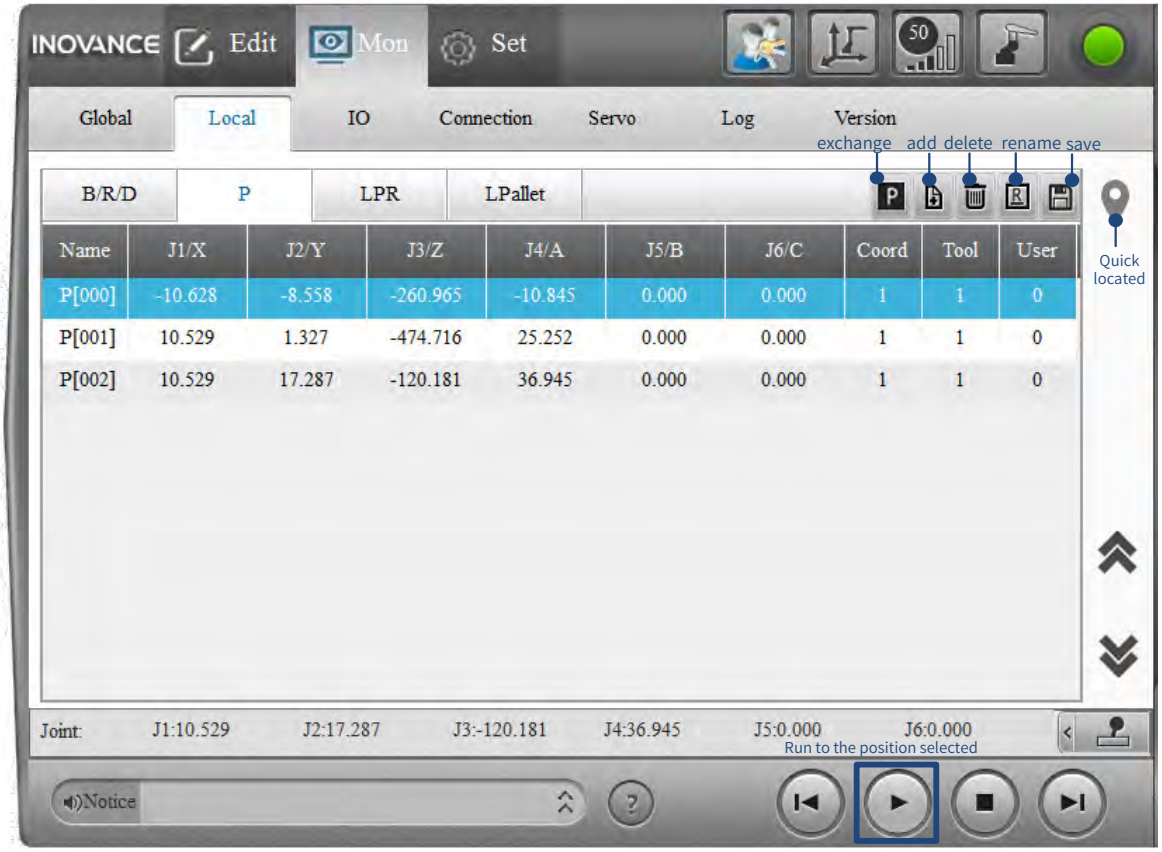


b) Position Variables

Position variable monitoring: Arm parameters are not displayed on the interface by default. If you want to see them, double-click a lien to display a detailed information interface.



Position variables have the following operation functions:



Replace: Select a position variable, click the Replace button, and the position variable value is replaced with the current position of the robot.

Add: Click the Add button to add a new position variable. The variable value is the current position of the robot.

Rename: Rename a position variable.

Delete: Delete a position variable.

Save: Different from other variables, after you finish editing a position variable, you must click Save to save it.

Quick locating: Enter a variable No., and you can quickly locate to a corresponding position.

Modify: Double-click to modify.

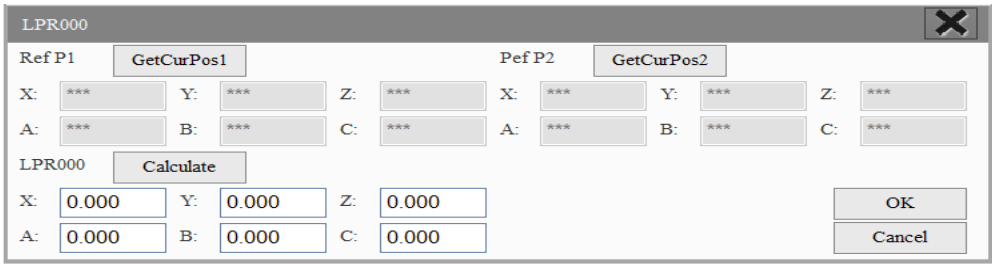
Run to a position variable: Select a variable in this line. Click the Run button at the bottom to run to the position variable.

NOTE

◆ Position variables displayed in the monitoring window can be refreshed only during pause or stop.

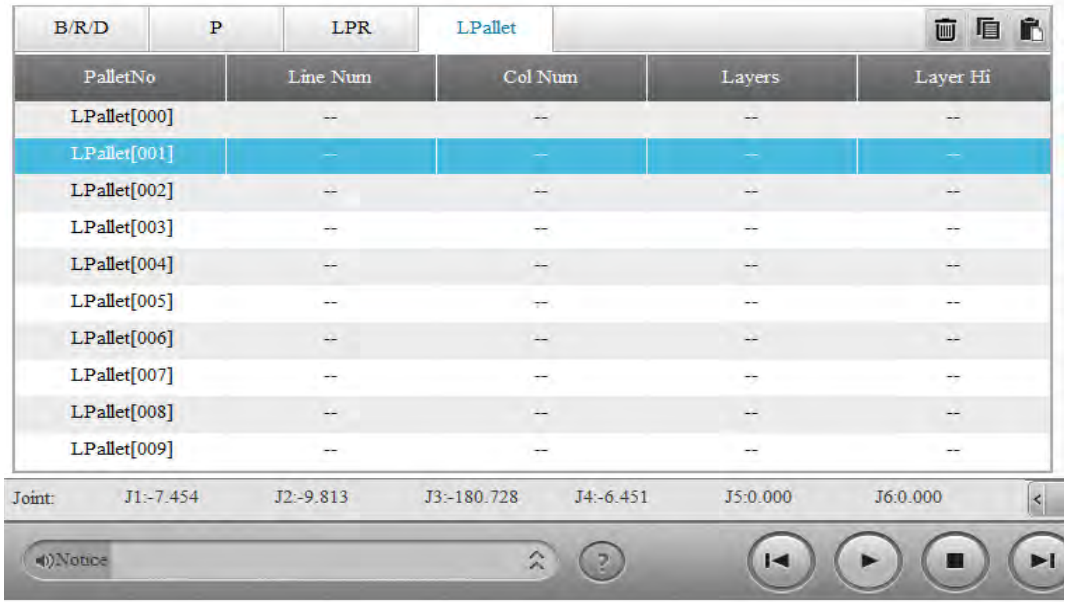
c) Local translation variables

Double-click to modify a translation variable. You can directly enter a value or calculate it by taking two points.



d) Local pallet variables

Local pallet variables can be edited by only the LPallet command. Here, the LPallet page plays a role of monitoring display. Local pallet variables are valid in only the current program. After entering a new program, local pallet variables in the current program are reset to empty.



4.4 I/O Monitoring

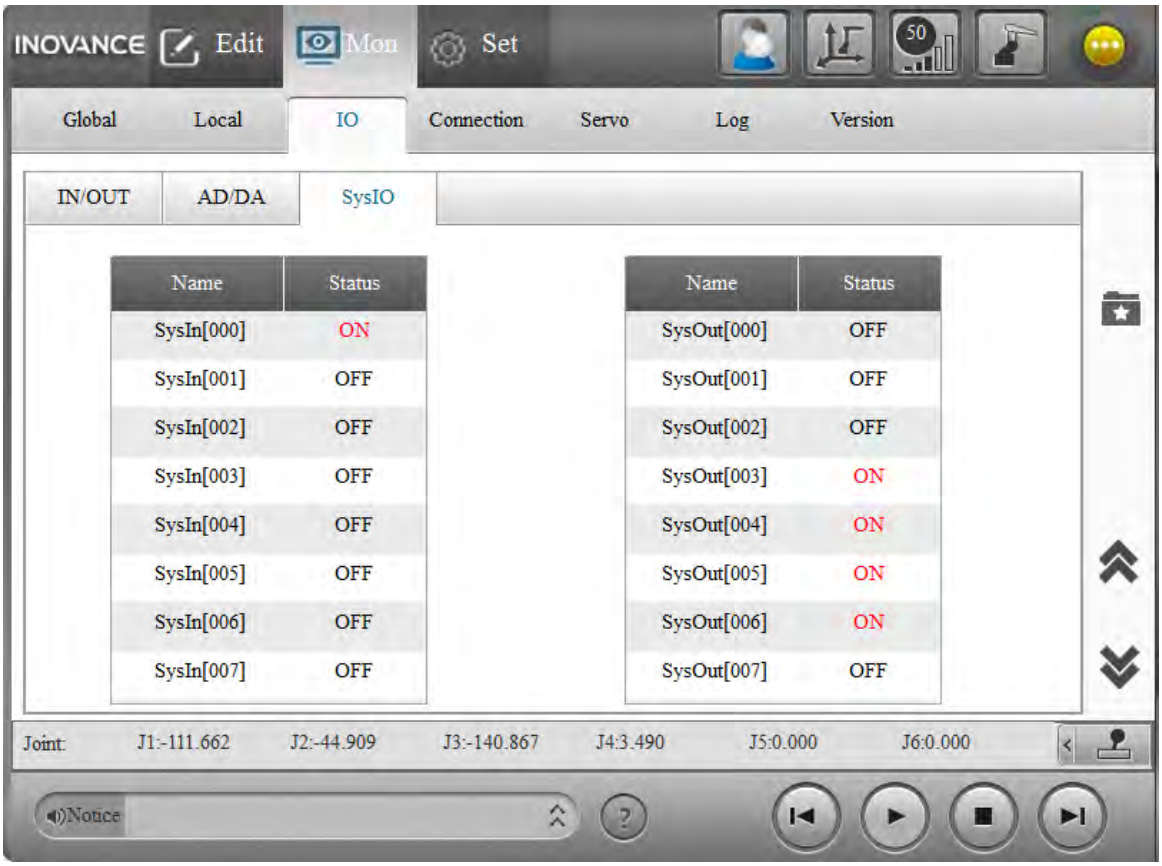
Precondition: Ensure that IRLink configurations are correct before using I/O monitoring!

For the IRCB10 electric cabinet, the IO interface contains IN/OUT and AD/DA parts. For the IRCB300 electric cabinet, the IO interface contains SysIO, IN/OUT and AD/DA parts.

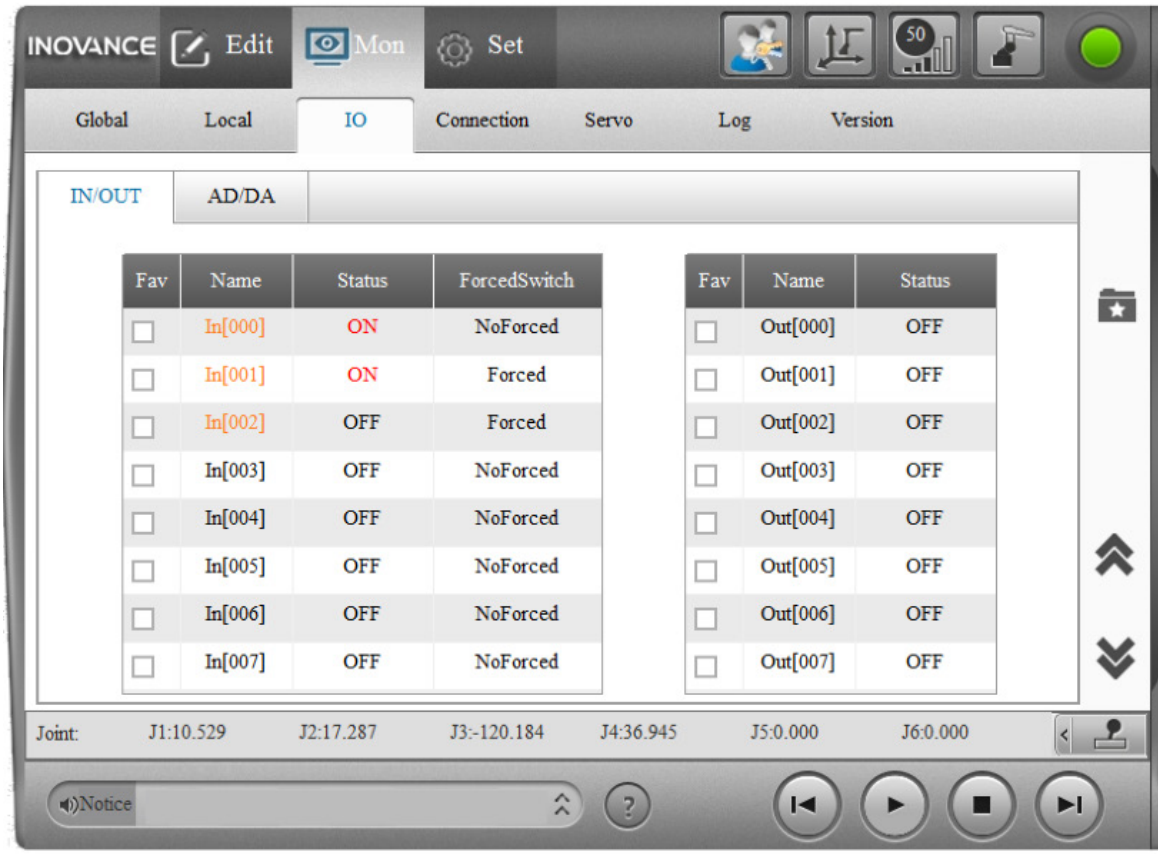
a) System I/Os

For the IRCB10 electric cabinet, System I/Os are In[0] to In[2] that are linked to emergency stop, enabled function and mode switching respectively and displayed on the IN/OUT interface. For the IRCB300 electric cabinet, System I/Os have a separate SysIO interface with 16 inputs and 16 outputs as follows:

System I/Os of the IRCB300 electric cabinet			
Input	Function	Output	Function
SysIn[0]	Emergency stop	SysOut[0]	System running indication
SysIn[1]	Enabled	SysOut [1]	System error indicator
SysIn[2]	Mode switching (Teaching/Playback)	SysOut [2]	System enabled indication
SysIn[3]	-	SysOut [3]	EtherNet1 connection indication On: Connected; Off: Disconnected
SysIn[4]	-	SysOut [4]	EtherNet1 frame transmission indication Flash: Data is transferred; Normally on: Connected, but no data is transferred
SysIn[5]	Security gate	SysOut [5]	EtherNet2 connection indication On: Connected; Off: Disconnected
SysIn[6]	ITP100 enabled	SysOut [6]	EtherNet2 frame transmission indication Flash: Data is transferred; Normally on: Connected, but no data is transferred
SysIn[7]	-	SysOut [7]	-
SysIn[8]	-	SysOut [8]	Robot body enabled indicator
SysIn[9]	-	SysOut [9]	-
SysIn[10]	-	SysOut [10]	-
SysIn[11]	-	SysOut [11]	-
SysIn[12]	-	SysOut [12]	-
SysIn[13]	I/O board fault indication	SysOut [13]	-
SysIn[14]	I/O board fault indication	SysOut [14]	-
SysIn[15]	I/O board fault indication	SysOut [15]	-



b) IN/OUT



IN forced switch: Input signal state is determined by an external source by default. However, IN can be changed to "Forced" state using the forced switch. In this case, signals can be manually forced to be ON or OFF.

State control: Set signal state ON or OFF to opposite state by directly clicking it.

Orange identification:

It indicates that this I/O is occupied by others and cannot be controlled or used.

IN orange - It appears for only the IRCB10 electric cabinet. IN[000]-IN[002] system I/Os are occupied.

OUT orange - Output ports are occupied and the control authority is not RC_ACTIVE.

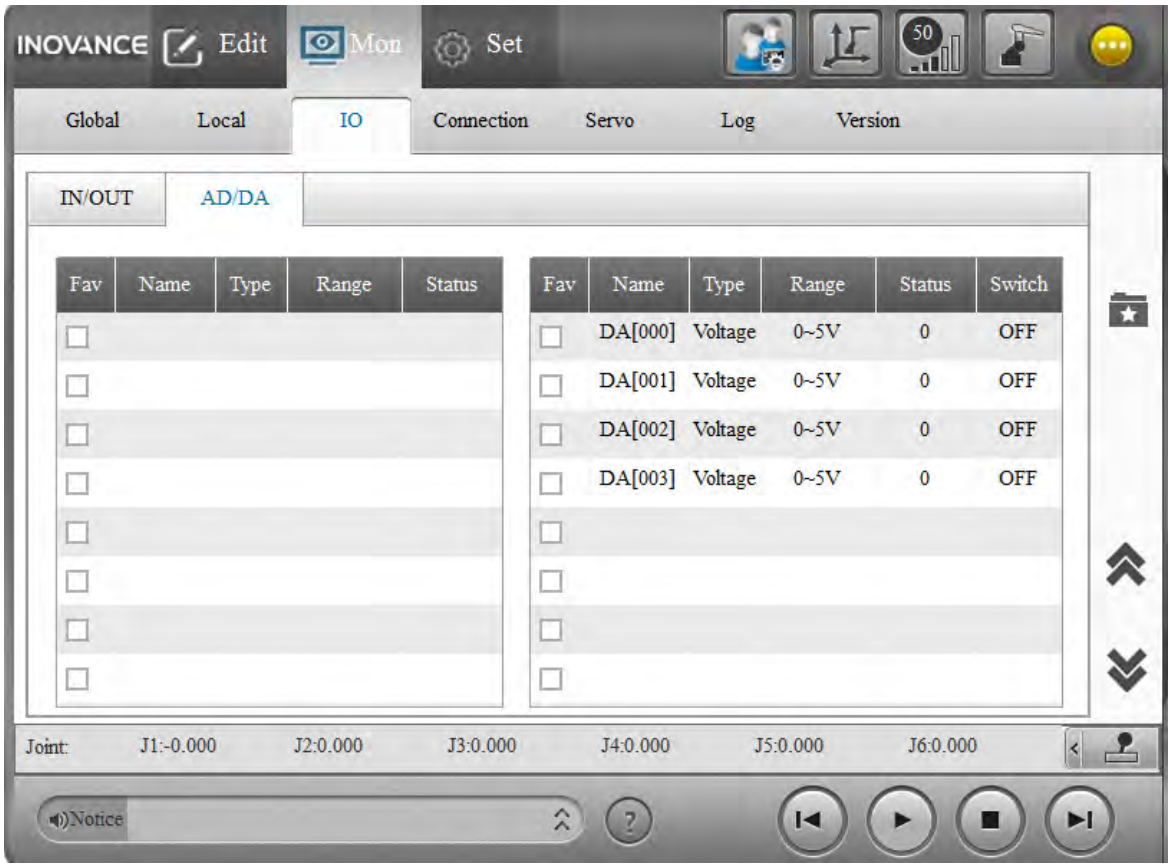
Control authority list:

Control Authority	
RC_STATIC	It indicates occupation by the RC system. That is, Out is bound to system functions under [**External**] - [**I/O-Config**]. In this case, output port signals are related to functions only and no signal state can be manually changed.
RC_ACTIVE	It indicates normal RC control state. In this case, signals are normally controlled. For example, signals are changed under [**IO**] or using the **Set** command.
PLC_ACTIVE	It indicates PLC control state. In this case, signals are controlled by PLC software like InoRobShop.

NOTE

◆ From the foregoing, when an I/O is orange, it cannot be controlled. If you hope that any controlled Out is in orange occupied state, check whether the control authority of this Out is configured as PLC under [External] - [I/O-Config] or changed to PLC by InoRobShop.

AD/DA



Output type: It indicates that an analog signal is current or voltage.

Range: Analog signal range. Every analog port has several selection ranges according to IRIInk product models.

Status: Analog parameter value in V for voltage signals and mA for current signals. Click a status item in the list to directly modify it.

DA switch: It determines whether a DA status value is valid. Manual switchover can be performed by clicking.

Orange identification:

DA orange - Output ports are occupied and the control authority is not RC_ACTIVE.

Control authority of output signals:

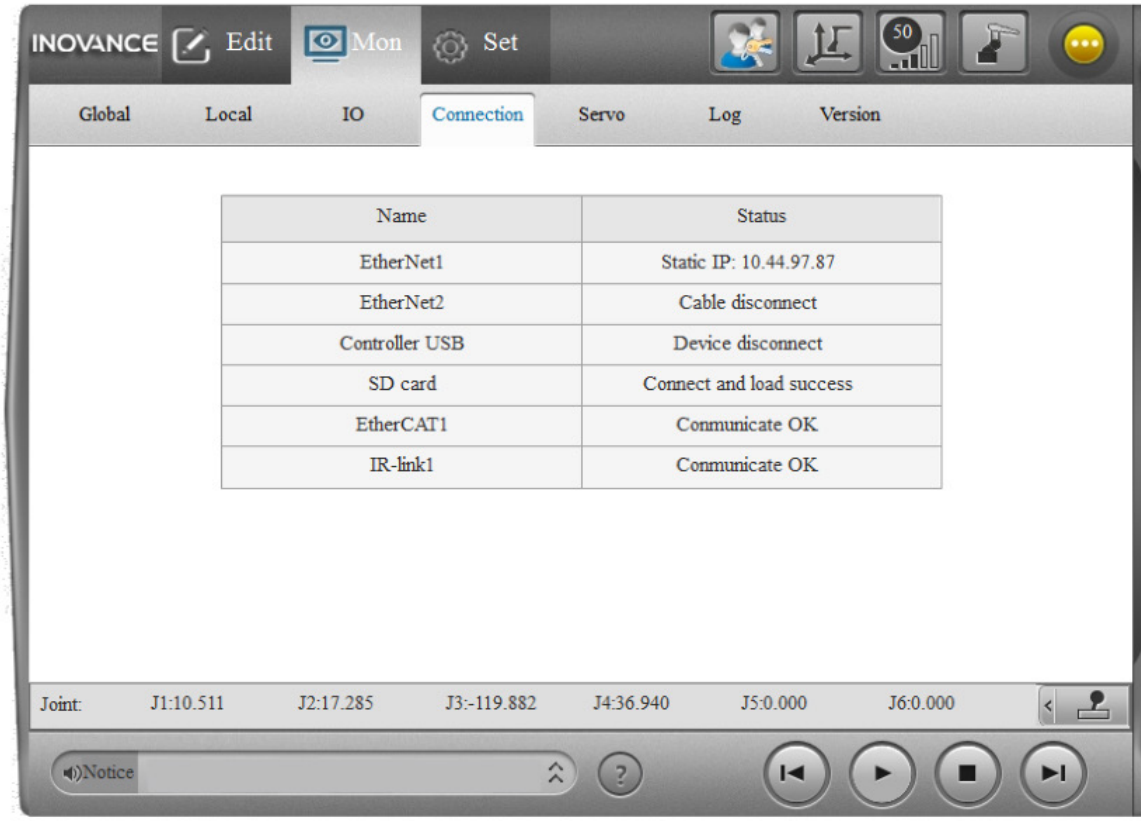
Control Authority	
RC_STATIC	It indicates occupation by the RC system. That is, Out is bound to system functions under [**External**] - [**I/O-Config**]. In this case, output port signals are related to functions only and no signal state can be manually changed.
RC_ACTIVE	It indicates normal RC control state. In this case, signals are normally controlled. For example, signals are changed under [**IO**] or using the **Set** command.
PLC_ACTIVE	It indicates PLC control state. In this case, signals are controlled by PLC software like InoRobShop.

NOTE

◆ From the foregoing, when an I/O is orange, it cannot be controlled. If you want to control the DA whose color is orange (means occupied), check whether InoRobShop occupys it.

4.5 Communication State

You can view port communication states on the controller on the interface as follows.



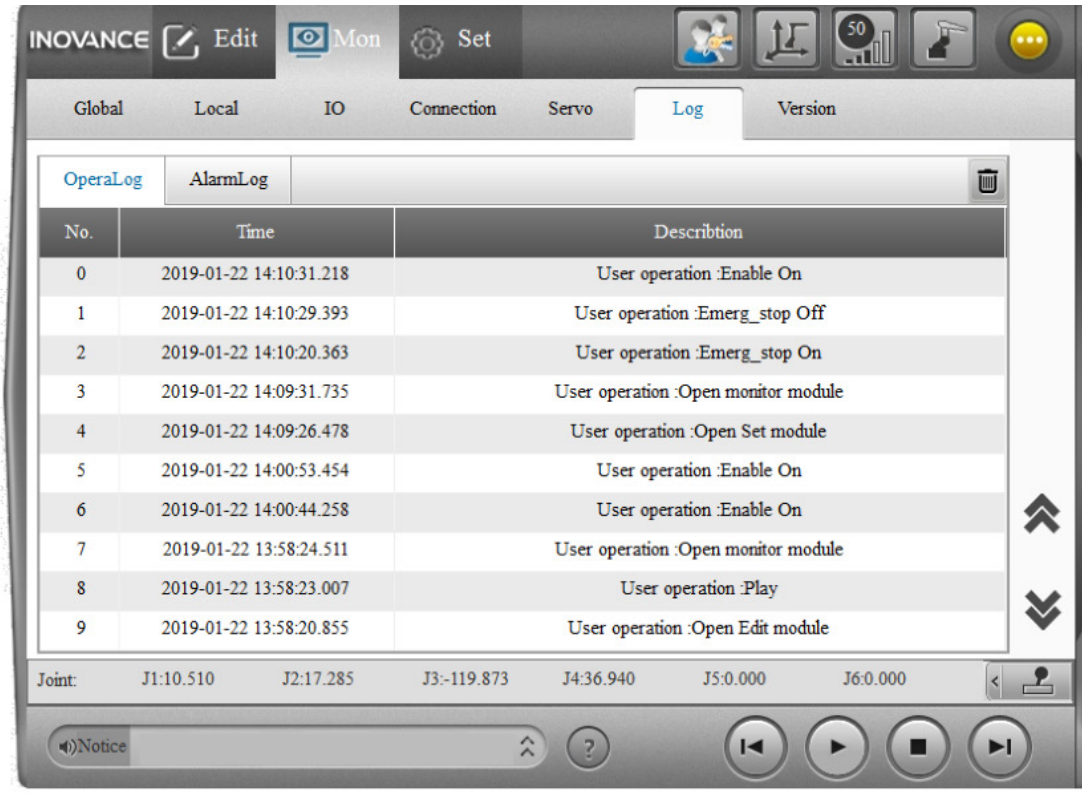
4.6 Servo State

You can monitor every servo parameter in real time on the servo state panel.

No.	CurValue	Name	Remark
00	1020	Servo software version	H0100
01	0	Servo Software unstandard-version	H0002
02	23110	Encoder Software Version	H0004
03	14101	Motor type	H0000
04	2	Servo type	H0102
05	0	Absolute pos mode	H0201
06	0	Actual motor speed	H0B00
07	0	Torque command	H0B01
08	0	Average load rate	H0B12
09	2449108	Absolute position feedback	H0B17

4.7 Log

Logs record the contents and operation time of every operation. Error records record abnormalities during operation. Logs and error records provide support information for fault diagnosis. For the robot error handling methods, refer to the *"Appendix: Robot Error and Solution List"*.



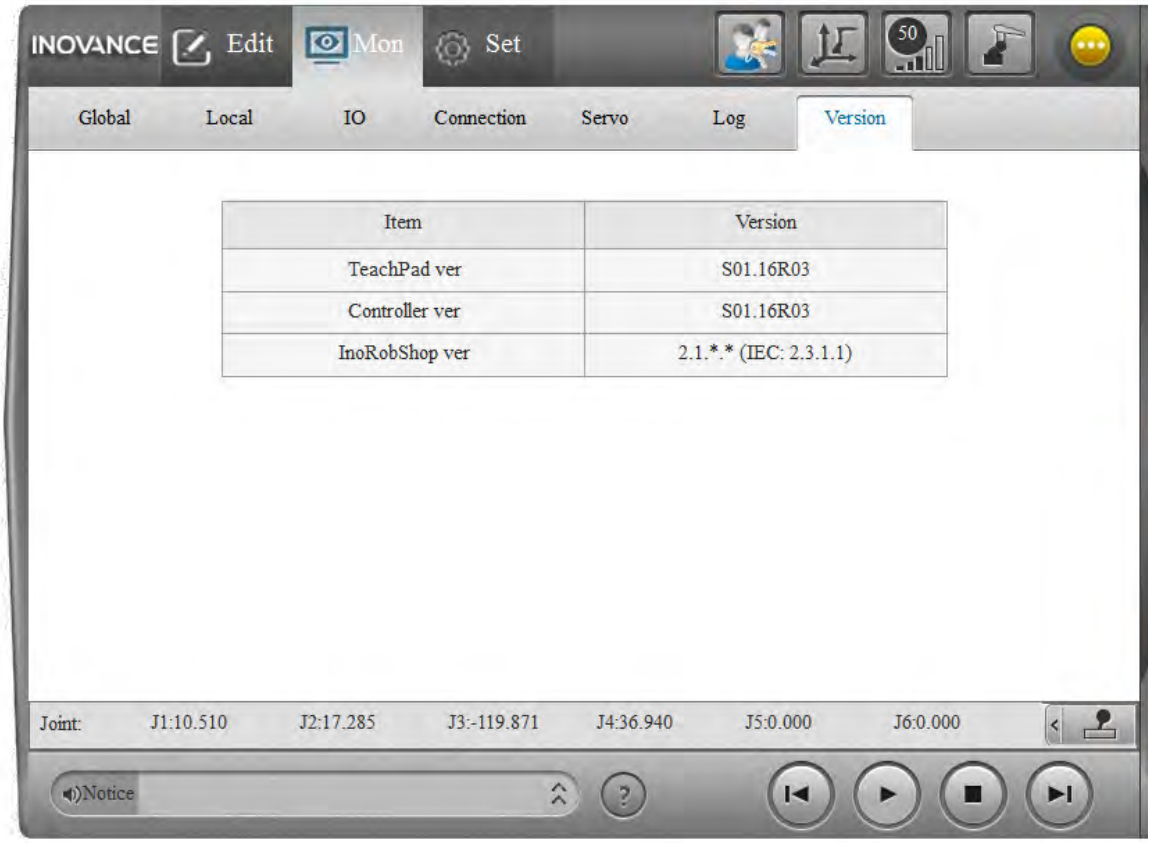
No.	Time	Describtion
0	2019-01-22 14:10:31.218	User operation :Enable On
1	2019-01-22 14:10:29.393	User operation :Emerg_stop Off
2	2019-01-22 14:10:20.363	User operation :Emerg_stop On
3	2019-01-22 14:09:31.735	User operation :Open monitor module
4	2019-01-22 14:09:26.478	User operation :Open Set module
5	2019-01-22 14:00:53.454	User operation :Enable On
6	2019-01-22 14:00:44.258	User operation :Enable On
7	2019-01-22 13:58:24.511	User operation :Open monitor module
8	2019-01-22 13:58:23.007	User operation :Play
9	2019-01-22 13:58:20.855	User operation :Open Edit module

NOTE

◆ Errors are cleared by clicking the error indicator at the top right corner. However, if problems are not solved, some errors still exist after being cleared. If a limit error appears, reverse motion is required to clear it.

4.8 Version Information

You can view version info as shown in the following figure.



Item	Version
TeachPad ver	S01.16R03
Controller ver	S01.16R03
InoRobShop ver	2.1.*.* (IEC: 2.3.1.1)

NOTE

◆ When the versions of InoTeachPad and controller are inconsistent, an error is generated.

◆ You can see more information in factory mode.

III. Programming

1. Robot Program Files 54

1.1 Position Variables in the Program 54

1.2 Commands in the Program 54

2 Variables 55

2.1 Global and Local Variables 55

2.2 Signal Variables 58

3 Motion Programming 59

3.1 Motion Commands 59

3.2 Form and Parameters of Motion Commands 60

3.3 V and Acc 61

3.4 Z 61

3.5 User..... 63

3.6 Tool 64

3.7 NWait 65

3.8 Until 65

3.9 I/O in the Motion Process 66

3.10 Jump and JumpL 66

4. I/O Programming 67

4.1 Preparations Before I/O Programming..... 67

4.2 I/O Read-Write 68

4.3 I/O Group Read-Write..... 69

4.4 Other I/O Commands..... 69

5 Logic Control Programming.....70

5.1 Jump 70

5.2 Selection Type 70

5.3 Loop Type 72

5.4 Subprogram Call 74

6 TCP/IP Communication Programming.....74

6.1 Communication Between an External Device and a Controller over TCP/IP 74

6.2 TCP/IP Communication Commands..... 75

In this part, you can know the programming concept of Inovance robots and learn the programming method of typical commands.

1. Robot Program Files

Robot program files are files with .pro name suffix that are stored in the TeachProgram directory of the controller SD. Position variables and commands are included in the files. Position variables are stored before commands.

The following figure shows a typical program file:

```
P[0] =     0.000000,     0.000000,     0.000000,     0.000000,     0.000000,     0.000000;-1, 0, 0, 0; 1, 0, 0;
P[1] =    10.000000,    20.000000,     0.000000,     0.000000,     0.000000,     0.000000; 1, 0, 0, 0; 1, 0, 0;
P[2] =    10.000000,    30.000000,     0.000000,     0.000000,     0.000000,     0.000000; 1, 0, 0, 0; 1, 0, 0;
START;
Movj P[0],V[30],Z[0];
Movj P[1],V[30],Z[0];
Movj P[2],V[30],Z[0];
END;
```

NOTE

◆ Only the command part of a program is displayed in InoTeachPad while position variables are displayed on the Mon page.

Program file naming convention: The name suffix is .pro. A filename consists of letters, numbers and underlines with a length of up to 32 characters, starting with a letter. Letters are not case-sensitive.

Program file size: The command part of a program can accommodate up to 2000 lines of commands.

Relevance: Several paths for obtaining program files.

Obtaining a single program file: **[Edit] - [Upload][Download] on the file editing toolbar**

Obtaining all program files: **[Set] - [System] - [Others] - [Program Backup][Program Load]**

The controller can be accessed using the Ftp file transfer function.

1.1 Position Variables in the Program

Position variables record information about the position and pose of a spatial point and the arrival of the robot at this point. A position variable is expressed as "(Coordinate values)+(Arm parameters)+(Coordinate system)+(Tool No.)+(User No.). For the meanings of the parameters, refer to *"2 Position Variable"*.

1.2 Commands in the Program

Commands in the program start with START and end with END. Commands are stored in the program file in the form of line. Every line ends with ";".

```
P[0] =      0.000000,     0.000000,     0.000000,     0.000000,     0.000000,     0.000000;-1, 0, 0, 0; 1, 0, 0;
P[1] =     10.000000,    20.000000,     0.000000,     0.000000,     0.000000,     0.000000; 1, 0, 0, 0; 1, 0, 0;
P[2] =     10.000000,    30.000000,     0.000000,     0.000000,     0.000000,     0.000000; 1, 0, 0, 0; 1, 0, 0;
START;
Movj P[0],V[30],Z[0];
Movj P[1],V[30],Z[0];
Movj P[2],V[30],Z[0];
END;
```

2 Variables

2.1 Global and Local Variables

Global variables are used in all programs while local variables are used in only the current program.

Global Variable		
Name	Variable	Description
Global B variable	B[0] to B[255]	Byte variable in a value range of 0 to 255
Global R variable	R[0] to R[255]	Int variable in a value range of -65535 to 65535
Global D variable	D[0] to D[255]	Double variable in a value range of -9999999.999 to 9999999.999
Global translation variable	PR[0] to PR[255]	Six parameters describe spatial translation
Global pallet variable	Pallet[0] to Pallet[255]	Special variable for the stacking process
Global string variable	Str[0] to Str[255]	String type

Local Variable		
Name	Variable	Description
Position Variables	P[0] to P[9999]	For robot positions, refer to point position description
Local LB variable	LB[0] to LB[255]	Byte variable in a value range of 0 to 255
Local LR variable	LR[0] to LR[255]	Int variable in a value range of -65535 to 65535
Local LD variable	LD[0] to LD[255]	Double variable in a value range of -9999999.999 to 9999999.999
Local translation variables	LPR[0] to LPR[255]	Six parameters describe spatial translation
Local pallet variables	LPallet[0] to LPallet[255]	Pallet variable generated by the Pallet command
Local string variable	Name defined by commands	String type

a) Numeral variables

Global and local B\R\D variables. They are used as storage data that corresponds to Byte, int and double data respectively.

Programming example:

B1 = 42;

R1 = 1000;

D1 = 123456.789;

R2 = R1+2;

LD1=(Sin(30)+D1)/2.1;

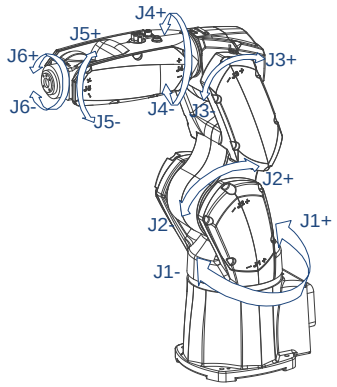
b) Translation variables

A translation variable is used to describe spatial motion. It can describe both J1-J6 joint motion under the joint coordinate system and any spatial change XYZABC of a pose relative to a coordinate system in 3D space. Please distinguish a translation variable according to the Offset command used with it:

1) OffsetJ

Function: Joint translation

Description: Six coordinate values of PR are joint rotation angles.



Programming examples:

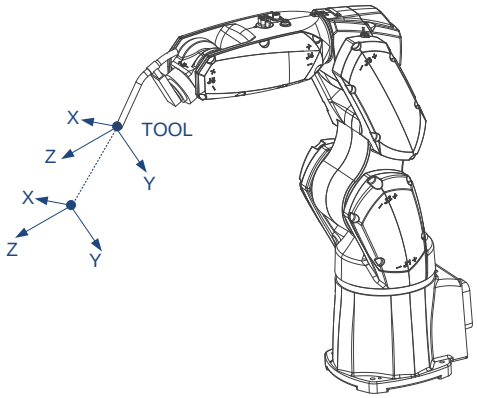
P[2]=OffsetJ(P[1],PR1);

MovjOffsetJ(P[1],PR1) ,V[30], Z[0];

2) OffsetT

Function: Translation along the current tool pose

Description: Translation is performed on the basis of the current tool coordinate system. PR is movement and rotation (X, Y, Z, A, B, C). Translation sequence: X > Y > Z > A > B > C



Programming example:

P[2]=OffsetT(P[1],PR1);

MovjOffsetT(P[1],PR1) ,V[30], Z[0],Tool[1];

NOTE

◆ If it is used with the motion command, e.g. MovOffsetT(P,PR)..., then there must be Tool, not User, in the motion command.

3) Offset

Function:

Offset indicates translation of a target point under a Cartesian reference system. A target point may be the TCP or flange center point. A reference system may be a user or base coordinate system.

Description:

Translation of a target point under a Cartesian reference system. PR is movement and rotation (X, Y, Z, A, B, C). Translation sequence: X > Y > Z > A > B > C

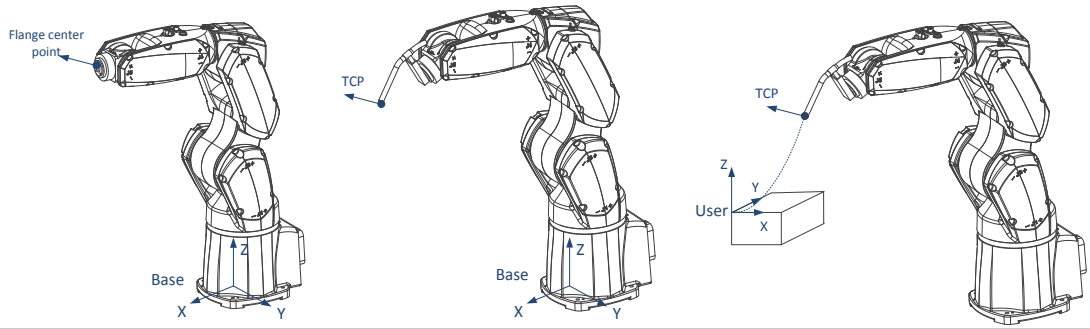
For P = Offset (P, PR), directly make a translation on coordinate values of position variables.

A translated target point and a reference coordinate system are designated by a coordinate system No. because the meanings of coordinate values are designated by a coordinate system.

For MovOffset(P,PR)…, a target point and a reference system are designated by Tool and User in a motion command respectively. In addition, a motion command has Tool or User different from P. Therefore, a tool or user coordinate system must be replaced before translation.

For operation details, refer to the Movj command and P = Offset.

Several offset translations



Programming example:

P[2]=Offset (P[1],PR1);

MovjOffset(P[1],PR1) ,V[30], Z[0],Tool[1],User[1];

c) Pallet variables

Global pallet variables are special variables for the stacking process. They are used only when the stacking process is involved! Global pallet variables contain constructed stacking information and a series of generated target points. For details, refer to the stacking process.

Local pallet variables can be generated using only the LPallet or LPallet4 command. They are valid in only the current program. After entering a new program, local pallet variables in the current program are set to empty. Local pallet variables contain constructed pallet information and a series of generated target points.

Programming example:

##Construct Pallet[1] in advance using the stacking process.

##Take the first point (with the index of Row 0, Column 0 and Layer 0) on the pallet. Assign it to P[0].

P[0]=Pallet(1,0,0,0);

##Construct a pallet variable LPallet[1] in Row 2, Column 3 and Layer 1 using P[1], P[2] and P[3].

LPallet 1,P[1],P[2],P[3],2,3,1,15;

##Take a point with the index of Row 1, Column 1 and Layer 0 on LPallet[1]. Assign it to P[10].

P[10]=LPallet(1,1,1,0);

d) String variables

The global string variable name is fixed to Str[***]. For local string variables, the name is defined in the program. To be used, it must be defined first.

Specifications for the string variable name:

The name consists of letters, numbers and underlines with a length of up to 10 characters.

- The name must start with a letter.
- The name shall not contain any Chinese and full-width characters.
- The name shall not be keywords or other existing variables such as "Movj", "V", "B1" and "P".

Specifications for string contents:

The contents consist of letters, numbers and underlines with a length of up to 100 characters.

- The contents must start with a letter.
- The contents shall not contain any Chinese and full-width characters.
- When you use InoTeachPad for programming, the string length is up to 50 characters due to interface limitation.

Programming example:

```
##Global string
Str[1] = "a1b2c3";
##Local string
String str1 = "abcd";        ##Make a definition first.
str1 = Left(str1,1);         ##Then use it.
```

2.2 Signal Variables

Signal variables are related to I/O and have four types: In, Out, AD and DA.

Global Variable		
Name	Variable	Description
Digital input signal	In[0] to In[63]	It has ON/OFF state.
Digital output signal	Out[0] to Out[63]	It has ON/OFF state.
Analog input signal	AD[0] to AD[15]	The value is a number and the range and unit are determined by IRLink module configurations.
Analog output signal	DA[0] to DA[15]	The value is a number and the range and unit are determined by IRLink module configurations.

NOTE

- Available variables are also restricted by any configured IRLink in actual use.
- In addition, every signal variable cannot be used in case of lack of authority.

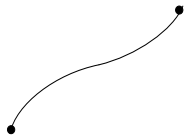
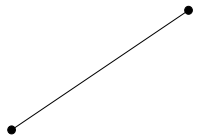
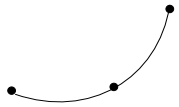
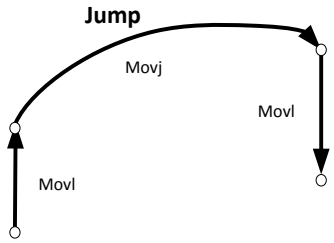
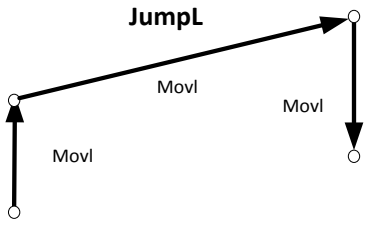
Programming example:

```
Set    Out[1],ON;
Set    DA[1],1.2;
Get    In[7],B1;
Get    AD[1],D1;
```

3 Motion Programming

3.1 Motion Commands

The following five robot motion-related commands are mainly available:

Motion Type	Track	Characteristics
Joint motion (Movj)		Point-to-point quick motion. All axes reach target positions at the same time. The track is not generally in a straight line. This motion is commonly used for applications such as spot welding and transportation.
Linear motion (Movl)		The TCP linearly moves to a given target position. The track is in a straight line. This motion is commonly used for applications such as track welding and surface mounting.
Arc motion (Movc)		The TCP moves in an arc to a given target position.
Jump motion (Jump)		The motion process is divided into three sections: Movl for the rising start section, Movj for the middle section and Movl for the end section. This motion is commonly used for picking up and placing objects by the SCARA robot.
Linear jump (JumpL)		The motion process is divided into three sections: Movl for the rising start section, Movl for the middle section and Movl for the descending end section. This motion is commonly used for picking up and placing objects by the SCARA robot.

Programming example:

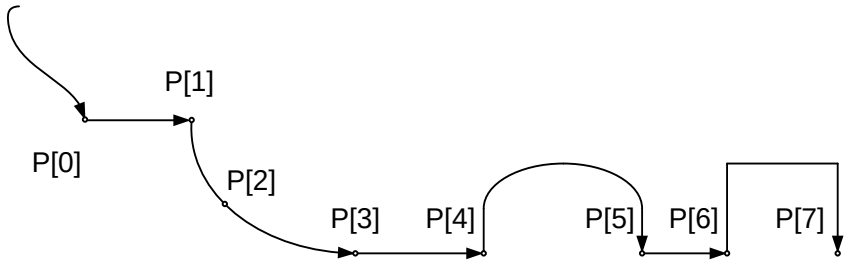
```
Movj P[0],V[30],Z[0];
Movl P[1],V[30],Z[0];
Movc P[2],V[30],Z[0];
Movc P[3],V[30],Z[0];
```

Movl P[4],V[30],Z[0];

Jump P[5],V[30],Z[0],LH[20],MH[-300],RH[20];

Movl P[6],V[30],Z[0];

Jump P[7],V[30],Z[0],LH[30],MH[-300],RH[30];



3.2 Form and Parameters of Motion Commands

Five Move series motions are mainly available in the following form:

Movj P, V, Z, [User], [Tool], [Acc], [NWait], [Until In == value], [Out(No,value,Type)…];

Movl P, V, Z, [User], [Tool], [Acc], [NWait], [Until In == value], [Out(No,value,Type)…];

Movc P, V, Z, [User], [Tool], [Acc], [NWait], [Out(No,value,Type)…];

Jump P, V, Z, [User], [Tool], [Acc], [NWait], [Out(No,value,Type)…], LH, MH, RH;

JumpL P, V, Z, [User], [Tool], [Acc], [NWait], [Out(No,value,Type)…], LH, MH, RH;

List of main parameters of motion commands:

List of main parameters of motion commands	
Parameter	Description
P	Target point to reach. It can also be in offset form and pallet point getting form.
V	Specifies a speed
Z	Arrival and transition parameter
User (Optional parameter)	Selected user coordinate system
Tool (Optional parameter)	Selected tool coordinate system
Acc (Optional parameter)	Specifies an acceleration rate
NWait (Optional parameter)	Non-waiting identification
Until In == value non-waiting	Detects signals in running and stops immediately after conditions are satisfied.
Out(No,value,Type)… (Optional parameter)	Outputs signals in parallel in running.

Among the preceding parameters, P, V and Z must be used and other parameters are optional and apply to some special cases.

Some key parameters are described below.

3.3 V and Acc

V specifies a speed and Acc specifies an acceleration rate. They affect motion in the form of percentage.

Running speed = Set maximum speed * Global speed percentage * Percentage set by a command

V in a motion command

Running acceleration rate = Set maximum acceleration rate * Percentage set by a command

Acc in a motion command

Programming example:

Movj P[0] ,V[70], Z[3]; ##70% of the speed, 100% of the acceleration rate by default

Movj P[1] ,V[30], Z[3],Acc[50]; ##30% of the speed, 50% of the acceleration rate

Movj P[2] ,V[30], Z[3],Acc[80]; ##30% of the speed, 80% of the acceleration rate

3.4 Z

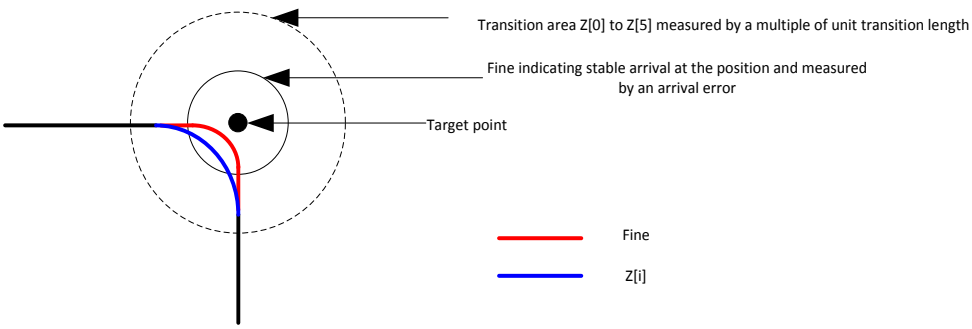
Z is an arrival and transition parameter. Seven options of Fine, Z[0], Z[1], Z[2], Z[3], Z[4] and Z[5] are provided.

Fine: It indicates accurate arrival. When the robot arrives at a stop point within arrival errors, it is deemed to be accurate and then the subsequent motion is executed.

Z[0] to Z[5]: They indicate that the robot does not stay at a target point, but smoothly pass a set transition area. Note that even Z[0] cannot ensure that the robot accurately reach a target point.

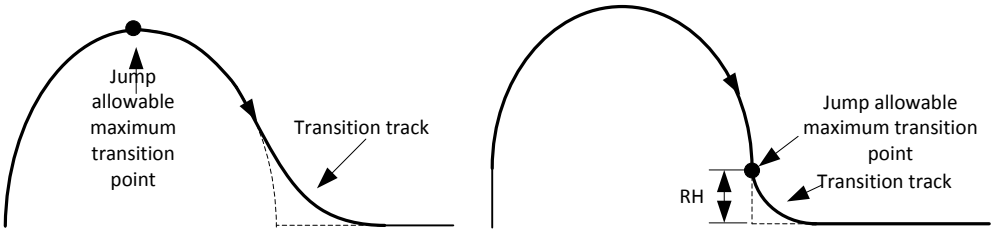
Transition is used to accelerate motion rhythm. The transition range is an area that uses a target point as a center of a circle and a transition radius as a radius. The transition length is equal to the product of the transition level and the transition unit length.

Type	Mode
Fine	It indicates accurate. When the robot enters and stays within a set error range, it is deemed to be accurate and the subsequent motion is executed.
Z[0]	There is no transition, but the robot cannot be ensured to reach a target point accurately.
Z[1]	There is a transition at one transition unit length.
Z[2]	There is a transition at twice the transition unit length.
Z[3]	There is a transition at three times the transition unit length.
Z[4]	There is a transition at four times the transition unit length.
Z[5]	There is a transition at five times the transition unit length.



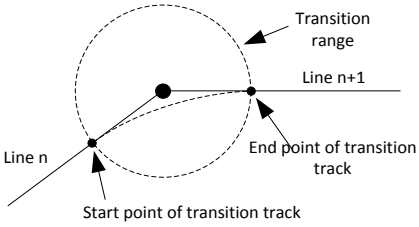
The unit transition length is divided into the joint transition unit and the linear transition unit. MoveJ and Jump commands use the joint unit (Three axes of the SCARA still use the linear unit) while other motion commands use the linear unit.

For specific motion, an allowable transition length is limited. When any set transition length exceeds an allowable maximum transition length, take the maximum transition length as an actual transition length. The maximum allowable transition length of MovL, MoveJ and MovC is a half of the total length. For Jump and JumpL commands, when RH (or LH) is 0, the allowable maximum transition length is a half of the total length, as shown in Figure 1. When RH (or LH) is not 0, the allowable maximum transition length is RH (or LH), as shown in Figure 2.



Jump transition at RH = 0 Jump transition at RH ≠ 0

Effect of transition on motion I/O: When the dotted transition area shown in the following figure exists, the set effective precision of motion I/O in the transition area cannot be guaranteed and the precision actually is effective in an uncertain position between the start point and end point of the transition track.



Programming example:

```
Movj  P[0] ,V[70], Fine;        ##Accurate arrival
Movj  P[1] ,V[30], Z[0];        ##No transition
Movj  P[2] ,V[30], Z[5];        ##Z[5] transition
```

3.5 User

Generally, Tool and User parameters need not be used because position variables have Tool and User. Of course, the effects of specifying the same Tool or User as P in motion commands are the same. For example,

P[1]=(100,0,0,0,0,0;1,0,0,0;1,2,3);

Movj P[1],V[30],Z[0]; or Movj P[1],V[30],Z[0],Tool[2],User[3];

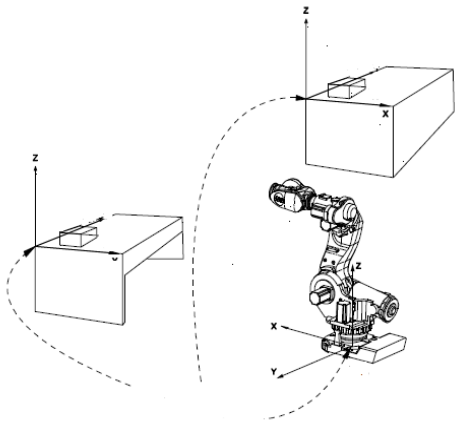
However, the tool changing function can be achieved using the Tool parameter. This is described below.

The switchover of the user coordinate system can be achieved using the User parameter in motion commands.

Application: A program has been taught and edited. After workpiece positions are changed, you need not perform teaching one by one again and can specify a new user coordinate system in motion commands.

NOTE

◆ The key precondition is that all points are gotten under the user coordinate system in the original program.



Case:

For an original batch of workpieces below the table, establish User1 on the table and use User1 as the current user coordinate system. Then perform teaching, point getting and programming under the user coordinate system.

```
Movj P[1],V[30],Z[0];          Movj P[1],V[30],Z[0],User[1];
Movl P[2],V[30],Z[3];    or    Movl P[2],V[30],Z[3],User[1];
Movl P[3],V[30],Z[3];          Movl P[3],V[30],Z[3],User[1];
……                             ……
```

When table placement positions are changed, you only need to establish User2 on a new table. You can directly change as follows:

MovjP[1],V[30],Z[0],User[2];

MovlP[2],V[30],Z[3],User[2];

MovlP[3],V[30],Z[3],User[2];

NOTE

◆ If you hope that the robot passes a singular position, you must use the Movj command. The coordinate system No. of P must be 1. Tool and User cannot be switched over.

3.6 Tool

Generally, Tool and User parameters need not be used because position variables have Tool and User. Of course, the effects of specifying the same Tool or User as P in motion commands are the same. For example,

P[1]=(100,0,0,0,0,0;1,0,0,0;1,2,3);

Movj P[1],V[30],Z[0]; or Movj P[1],V[30],Z[0],Tool[2],User[3];

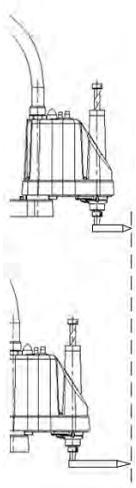
However, the tool changing function can be implemented using the Tool parameter. This is described below.

Tool switchover can be achieved using the Tool parameter in motion commands. When a new tool is used, the new TCP reaches the same pose.

Application: A processing program has been edited using an old tool. After the tool is changed, the same processing is performed using a new tool. You need not perform teaching one by one again and can specify a new tool in motion commands.

NOTE

◆ The tool changing function only requires that the tool No. of position variables is selected correctly. Position variables can be gotten under any coordinate system, even joint and base coordinate systems.



Case:

Tool1 has been used for teaching and point getting. Note that the tool No. of the position variable is selected as 1. Programming is as follows:

```
Movj P[1],V[30],Z[0];          Movj P[1],V[30],Z[0],Tool[1];
Movl P[2],V[30],Z[3];    or    Movl P[2],V[30],Z[3],Tool[1];
Movl P[3],V[30],Z[3];          Movl P[3],V[30],Z[3],Tool[1];
……                             ……
```

Now the tool is replaced and Tool2 is used for processing. You need not perform teaching again and can directly change Tool in the commands.

MovjP[1],V[30],Z[0],Tool2;

MovlP[2],V[30],Z[3],Tool2;

MovlP[3],V[30],Z[3],Tool2;

NOTE

◆ If you hope that the robot passes a singular position, you must use the Movj command. The coordinate system No. of P must be 1. Tool and User cannot be switched over. No translation shall be contained.

3.7 NWait

NWait is a non-waiting identification. This item indicates that subsequent non-motion commands are immediately executed without running to a target point. This causes subsequent non-motion commands such as operation, signal and logic judgment to be executed in advance until the next motion command is encountered.

The difference is as follows:

##Set a signal in advance without waiting until motion arrival. Movj Movj P[1],V[30],Z[0],NWait; Set Out[3],ON; ……	##Set a signal after motion is finished. Movj Movj P[1],V[30],Z[0]; Set Out[3],ON; ……

Programming example:

##A signal is set immediately at the beginning of motion without waiting until motion is complete.

```
Movj P[1],V[30],Z[0];
Movl P[2],V[30],Z[3],NWait;
Movl P[3],V[30],Z[3],NWait;
Set Out[3],ON;                    ## Out[3] is set in advance at the beginning of Movl P[2].
MovjMovl P[3],V[30],Z[0];
Set Out[4],ON;                    ##Wait until Movj P[3] ends before Out[4] is set.
```

3.8 Until

Input signals are detected in the motion process. If conditions are satisfied, motion in the current line is terminated immediately and the program in subsequent lines continues to be executed. If conditions are not satisfied, motion continues until the end point.

Until In == value:

In: Input signal

Value: Input signal value, ON or OFF

Programming example:

##In[5] is detected in the motion process. When it is ON, motion is terminated immediately, otherwise motion continues until the end.

```
Movj P[1],V[30],Z[0],Until In[5] == ON;
```

3.9 I/O in the Motion Process

The Out(No,value,Type)... parameter is used to output signals in parallel in the running process.

Up to two signals are output in parallel in a motion command.

Out(No,value,Type)...

No: Output signal No.

Value: Output signal value.

Type: Type of output signals in the motion process. Different motions can use different Types.

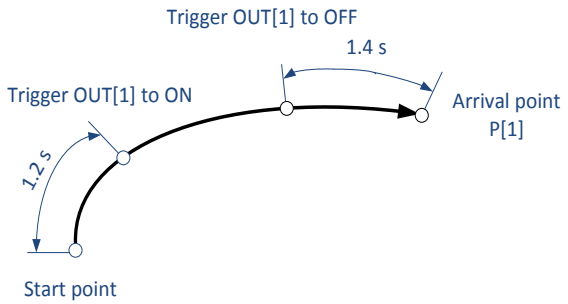
Type	Meaning	Applicable motion type
T[n]	Signals are output by time in the motion process. T[n] is time in seconds. The range of -65535.000 to 65535.000. n >= 0 indicates that signals are output n seconds after motion starts. n < 0 indicates that signals are output n seconds before arrival at a target point. When time is set beyond the motion range, motion is triggered at the start or end points of this travel section at most, so the set time is invalid.	Movj/ Movl/ Movc/ Jump/JumpL
D[n]	Signals are output by a path percentage in the motion process. D[n] indicates a path percentage. It indicates that signals are output from the begging to end of motion until n% of the path.	Movl/ Movc/Jump/ JumpL
S[n]	Signals are output by distance in the motion process. S[n] indicates distance in mm. The range is -65535 to 65535. n >= 0 indicates signals are output after the robot moves to n mm from the start point. n < 0 indicates signals are output before the robot moves to n mm from the end point.	Movl/ Movc/Jump/ JumpL

NOTE

- When motion I/O in the transition area is set, the precision cannot be guaranteed.
- When T[n]/D[n]/S[n] is set beyond the motion range, motion is triggered at the start and end points of this travel section at most.

Programming example:

Movj P[1] ,V[30], Z[3],OUT(1,ON,T[1.2]), OUT(1,OFF,T[-1.4]);



3.10 Jump and JumpL

The Jump command applies to only the SCARA and Delta robots. Both Jump and JumpL are jump commands in the rise-move-descend motion mode. They consist of Movj and Movl in essence. Their difference is that the middle section of Jump is Movj while the middle section of JumpL is Movl.

Jump format: Jump P, V, Z, [User], [Tool], [Acc], [NWait], [Out(No,value,Type)...], LH, MH, RH;

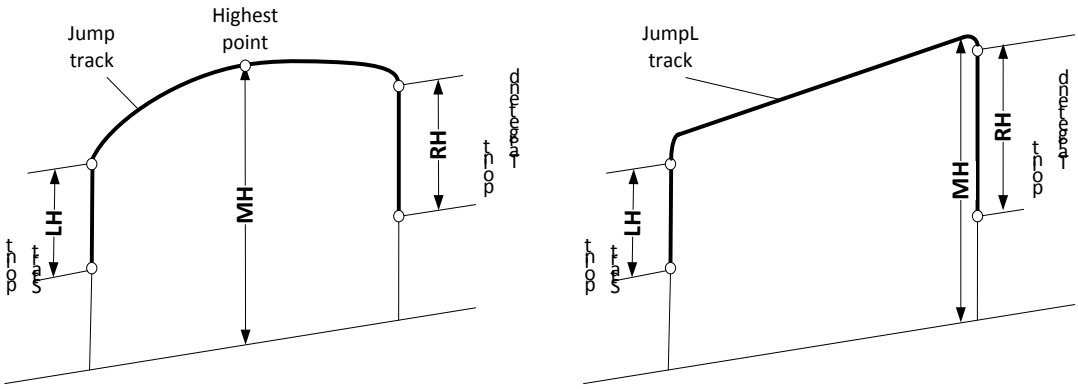
JumpL format: JumpL P, V, Z, [User], [Tool], [Acc], [NWait], [Out(No,value,Type)...], LH, MH, RH;

It is observed that most parameters are the same as Move series motions and only LH, MH and RH are unique:

LH: Lifting height at the start point in a value range of 0 to 2000.

MH: It is used to limit a height. It is the height of the highest point relative to the zero point of the base coordinate system in the running process. The value range is -2000 to 2000.

RH: Descending height to the end point in a value range of 0 to 2000.



NOTE

- The zero point of the base coordinate system of the SCARA robot is located on the upward side. The measurement of MH is an absolute value. Therefore, MH may be a negative value in most cases.
- Generally, the height parameter must satisfy MH > start point height + LH and MH > end point height + RH. For any abnormal setting that does not satisfy this condition, non-gate shaped motion may appear and can be used for special application.

Programming example:

Jump P[1] ,V[30], Z[3],User[1],Tool[1],LH[60],MH[-130],RH[60];

JumpL P[1] ,V[30], Z[3],User[1],Tool[1],LH[60],MH[-130],RH[62];

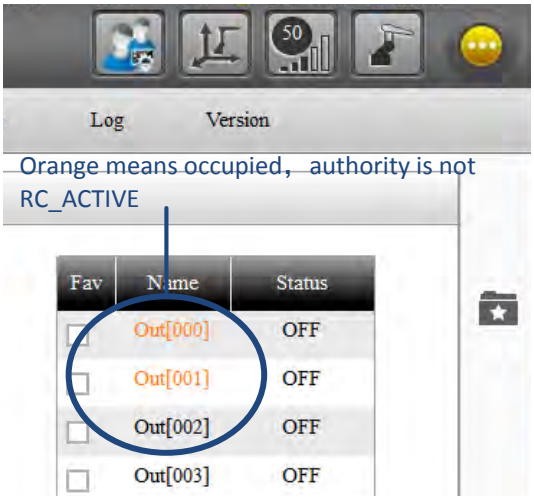
4. I/O Programming

4.1 Preparations Before I/O Programming

Ensure that IRLink configurations are correct before I/O programming. For output signal control, I/O control authority is required. We can easily confirm these preconditions on InoTeachPad. On the [Mon] - [IO] interface of InoTeachPad, observe that this I/O is available and not orange, which indicates the preconditions are satisfied.

For example,

when the following case appears, it indicates Out[2] programming cannot be used.



4.2 I/O Read-Write

The following commands are used to finish individual read-write of I/Os.

Command	Description	Remarks
Set Out,value;	Set an Out signal variable	
Set DA,Value;	Set a DA signal variable	
Get In,B;	Get an In signal variable	Signal values are put in the B variable. The value 0 is OFF. The value 1 is ON.
Get Out,B;	Get an Out signal variable	Signal values are put in the B variable. The value 0 is OFF. The value 1 is ON.
Get AD,D;	Obtain an AD signal variable	Signal values are put in the D variable.

NOTE

◆ For analog I/O, the value unit is determined according to the configurations of the IRLink module. If this signal is configured as current and voltage, the unit is mA and V respectively.

Programming example:

```
Set    Out[1],ON;          ##Set Out[1] output to ON.
Set    DA[1],1.2;          ##If DA[1] is configured as current type, this command is to output a 1.2 mA current.
Get In[7],B1;
Switch B1
        Case 0:
                Print "In[7] is OFF";
                Break;
        Case 1:
                Print "In[7] is ON";
                Break;
        Default:
                Print "Error";
EndSwitch;
Get Out[0],B2;          ##Obtain an Out[0] value.
Get    AD[1],D1;        ##Obtain an AD[1] value.
```

4.3 I/O Group Read-Write

OG and IG are used to represent a group of Out and In signal variables respectively. A group of signals contain eight signals by default. For example, OG[0] is Out[0] to Out[7], OG[1] is Out[8] to Out[15], and so on.

Values of a group of signals are stored in the B variable. The B variable value is Byte type and corresponds to an 8-bit binary number. Every bit from right to left represents a signal state from low bit to high bit.

Command	Description	Remarks
Set OG,B;	Set a group of output signals	The B variable contains values of a group of signals.
Get IG,B;	Obtain a group of input signals	The B variable contains values of a group of signals.
Get OG,B;	Obtain a group of input signals	The B variable contains values of a group of signals.

Programming example:

B1=7;

Set OG[0],B1; ##Set OG[0] output to 0000 0111, i.e. Out[0] to Out[7] are 1110 0000 respectively.

Get IG[1],B1; ## Read IG[1] signals.

##If signals In[15] to In[8] are OFF, OFF, OFF, OFF, OFF, ON, ON, ON successively, then B1 = 7.

Get OG [1],B2; ## Read OG [1] signals.

##If signals Out [15] to Out [8] are all OFF, then B2 = 0.

Signals in a group can be redefined using the Group command. The number of signals in a group can be smaller than 8. Note that this is valid in only the current program.

Format:

Group OG, n0, n1, n2..., n7;

Group IG, n0, n1, n2..., n7;

Parameters:

OG: Output group to be redefined, OG[0] to OG[8].

IG: Input group to be redefined, IG[0] to IG[8].

n0, n1, n2..., n7: Signal No. Up to 8, but not smaller than 2 signals can be selected.

Example:

Group OG[0] 1,2,7; ##Redefine OG[0] as Out[1], Out[2] and Out[7].

4.4 Other I/O Commands

Invert: Invert a signal

Wait: Wait for a signal

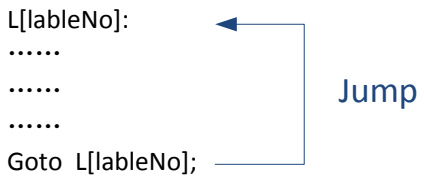
Plus: Pulse output

I/O in the motion process

5 Logic Control Programming

5.1 Jump

The L-Goto logic structure achieves the jump function. L is used to set a program tag and is used with the Goto command to finish the jump action.



Programming example:

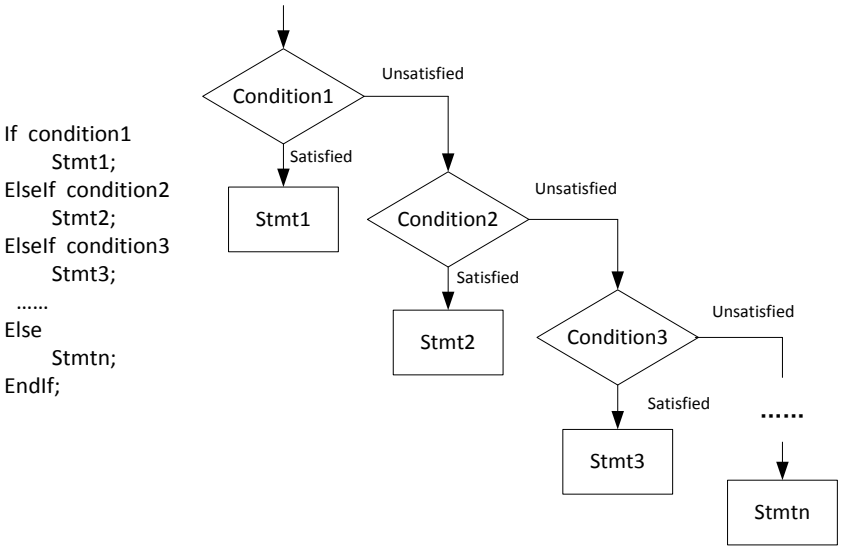
```
START;
Movj P[0],V[30],Z[3];
L[1]:                          #Set Label 1.
Movl P[1],V[30],Z[3];
Movl P[0],V[30],Z[3];
Goto L[1];                     #Go to Label 1
END;
```

##Execution effects of this paragraph: Run to P[0] and perform reciprocating motion between P[1] and P[0].

5.2 Selection Type

a) If -EndIf

Make a conditional judgment one by one. If a condition is satisfied, execute the corresponding statement and end. If a condition is not satisfied, continue to make a judgment. When all conditions are not satisfied, execute the statement after Else.



In this structure, If -EndIf is essential and ElseIf and Else can be omitted. Thus, only one judgment is executed. If a condition is satisfied, corresponding statements are executed.

Judgment conditions can be the combination of multiple conditions.

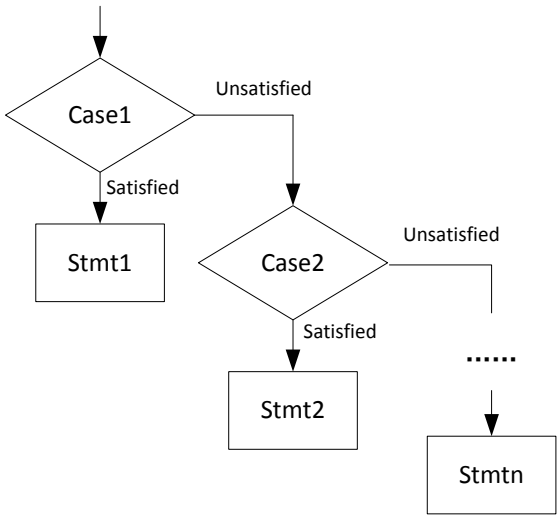
Programming example:

```
START;
If IN[1]==ON And IN[2]==ON ;
        Movj P[1],V[50],Z[3];
Else
        Movj P[2],V[50],Z[3];
Movj P[3],V[50],Z[3];
EndIf;
End;
```

b) Switch-Case-Default-EndSwitch

Match Switch objects using Case values one by one. If a condition is satisfied, execute corresponding statements and end. If a condition is not satisfied, continue to perform matching. If all Cases do not satisfy conditions, execute the statement after Default.

```
Switch Var
  Case  value1:
      Stmt1;
      Break;
  Case  value2:
      Stmt2;
      Break;
  ......
  Default:
      Stmtn;
      Break;
EndSwitch;
```



Extension:

Case A To B: Case A To B can be used to replace Case Value. Case A To B represents a selection within the range of Integer A to Integer B (A and B included). When a variable value is within this range, it indicates that this Case is matched.

NOTE

- As a general rule, every Case (Default included) paragraph had better end with Break, otherwise the program does not break and continues to execute the next Case paragraph until Break.
- The Default statement can be omitted. If an entire conditional selection paragraph ends with Default, then Break must follow after the end of the Default contents. If an entire paragraph does not use Default, i.e. only Case is used, then Break must follow after the end of the contents of the last Case.

Programming example:

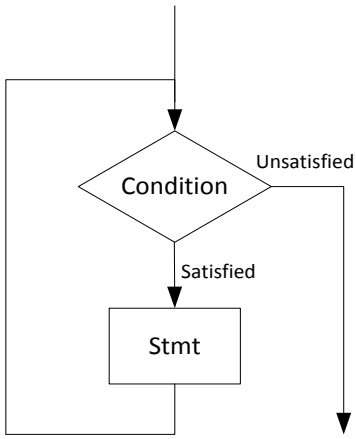
```
START;
Switch B0
    Case 1:
        Movj P[1],V[50],Z[3];
        Break;
    Case 2 To 4:
        Movj P[1],V[50],Z[3];
        Movj P[2],V[50],Z[3];
        Break;
    Case 5:
        Movj P[3],V[50],Z[3];
        Break;
    Default:
        Movj P[4],V[50],Z[3];
        Break;
EndSwitch;
End;
```

5.3 Loop Type

a) While-EndWhile

Judge a condition. If the condition is satisfied, execute statement in the loop body and then make a judgment again, and so on. Once a condition is not satisfied, the program jumps out of the loop.

```
While Condition
    Stmt;
EndWhile;
```



Description:

Judgment conditions can be the combination of multiple conditions.

The Stmt statements can be several lines of commands.

Programming example:

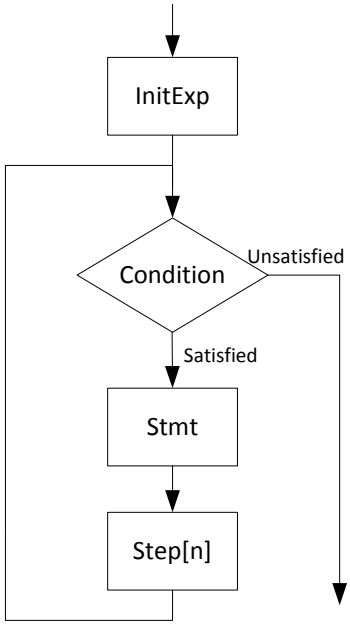
```
START;
Movj P[1],V[50],Z[3];
While LB[0]<3
        Movj P[2],V[50],Z[3];
Movj P[1],V[50],Z[3];
Incr LB[0];
EndWhile;
End;
```

Description: The preceding running commands have 3 cycles. The entire paragraph of the program executes to-and-fro movement between P[1] and P[2] for three times.

b) For-EndFor

Loop statement with number of executions. Execute an initialization assignment statement and judge a condition. If the condition is satisfied, execute the contents of the loop body and execute a step so that the variable in the initialization assignment statement achieves auto increment. Judge a condition again. If the condition is satisfied, continue the preceding steps until the program jumps out of the loop when a condition is not satisfied.

```
For  InitExp, Condition, Step[n]
       Stmt;
EndFor;
```



Programming example:

```
START;
For B0=0,B0<5,Step[2]
Movj P[2],V[50],Z[3];
Movj P[3],V[50],Z[3];
EndFor;
End;
```

Description: Two Movj commands are under loop execution for three times.

5.4 Subprogram Call

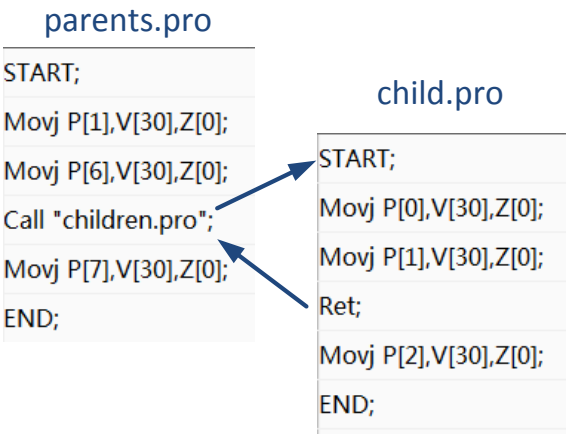
a) Call and Ret

Another subprogram can be called in a program. Thus, the system enters the subprogram. If the Ret command is encountered in the subprogram, the system returns to the main program to continue execution. If no Ret command is encountered, the system does not return to the main program after running to the end of the subprogram.

Up to three layers of nesting call of a subprogram are supported.

Programming example:

##The subprogram child.pro is called in the main program parents.pro.



This subprogram child.pro runs only the first two Mov command and the system returns to the main program. The entire motion process is as follows:

Main program P[1] → **Main program P[6]** → **Subprogram P[0]** → **Subprogram P[1]** → **Main program P[7]**

6 TCP/IP Communication Programming

6.1 Communication Between an External Device and a Controller over TCP/IP

The controller and peripherals can communicate with each other over TCP/IP.

Operations:

1) A local controller is used as a server and an external device is used as a client:

You need to set EtherNet1 of the controller as a server and specify a port No.

It is started during the system startup by default. An external device as a client can communicate with the controller. You only need to write commands for data receiving, sending and processing.

2) A local controller is used as a client and an external device is used as a server:

You need to set EtherNet1 of the controller as a client.

You must use the Open Socket command to open the client before writing commands for data receiving, sending and processing. Finally, you need to close the client using the Close Socket command.

6.2 TCP/IP Communication Commands

Typical commands in the communication process are as follows:

a) Open Socket

Format 1: Open Socket(IP, SvrPort, ClientPort)

Function: (Only used for the case that a robot controller is used as a client and an external Ethernet device is used as a server)

Open communication to connect the server. Enter a server IP address and port No. to specify a connection object. Enter a client port No. to specify a communication client port.

Tips: Why must a client port No. be specified in a command?

To support opening multiple client ports in a program to connect different servers.

Format 2: Open Socket(IP, SvrPort, ClientPort,B)

Function: Compared with the previous format, a B parameter is added, indicating whether a client port is opened successfully. During use, loop statements are generally used to perform loop execution, judge the B parameter and jump out of the loop after successful opening.

b) Close Socket

Format: Close Socket,ClientPort

Function: Close a client port No. It is only used for the case that a robot controller is used as a client and an external Ethernet device is used as a server)

c) Send Port

Format: Send Port[Port],String

Function: Send a string to a remote network port. This port is a client port No.

In particular, when a local controller is used as a server and an external device is used as the only client whose port No. is unknown, you can use a port No. 4444.

d) Get Port

Format: Get Port[Port],T[Time],Goto L[Index]

Function: Receive data from a remote port within a certain period of time and put data in a buffer. When no data is received within a specified time, jump to a specified tag. This port is a client port No.

In particular, when a local controller is used as a server and an external device is used as the only client whose port No. is unknown, you can use a port No. 4444.

e) GetPortbuf

Format:StrVar = GetPortbuf(StartBit , Num, Port)

Function: Read a specified number of characters from a specified position of a receiving buffer and store them to a specified string variable.

f) Others

Data is received in the form of string. String processing related commands like the StrGetData string segmentation command are usually required. For more information, refer to the list of commands.

g) Communication programming example

The robot is used as a client and the local port No. is set to 1026 in the command. The external vision is used as a server and the server port No. is set to 1025 under [CommSet].

```
START;
PR0=(0,0,10,0,0,0);
TxBuf = "TA";
RxBuf = " ";
L[2]:
Movj P[0],V[30],Z[0];
L[0];
Open Socket("10.44.53.13",1025,1026,B0);
If B0 == 0
   Goto  L[0];
EndIf;
Send Port[1026],TxBuf;
L[1]:
Get Port[1026], T[10], Goto L[1];
RxBuf = GetPorbuf(0,100,1026);
B1 = StrGetData(RxBuf,"#",P10);
Cnvrt(P[10],P[20],World]);
Movl Offset(P[20],PR0),V[30],Z[0];
Set Out[1],ON,T[0];
Delay T[1];
Jump P[1],V[100],Z[0],LH[10],MH[-750],RH[10];
Set Out[1],OFF,T[0];
Delay T[1];
Goto L[2];
Close Socket,1026;
END;
```

(Open)

↓

Send a request

↓

Receive data

↓

Process received data and control the motion

↓

(Close)

The robot is used as a server and the server port No. is set to 1025 under [CommSet].
The external vision is used as a client and the port No. is set to 1026.

```
START;
TxBuf = "TA";
RxBuf = " ";
L[2]:
Movj P[0],V[30],Z[0];
Send Port[1026],TxBuf;
L[1]:
Get Port[1026], T[10], Goto L[1];
RxBuf = GetPorbuf(0,100,1026);
B1 = StrGetData(RxBuf,"#",P10);
Cnvrt(P[10],P[20],World]);
Movl Offset(P[20],PR0),V[30],Z[0];
Set Out[1],ON,T[0];
Delay T[1];
Jump P[1],V[100],Z[0],LH[10],MH[-750],RH[10];
Set Out[1],OFF,T[0];
Delay T[1];
Goto L[2];
END;
```

Appendix: Robot Error and Solution List

Description: System faults are mainly divided into three levels:
0: Error with enablement lost
1: Error with enablement not lost
2: Warning with enablement not lost

Fault Code	Level	Description	Possible Causes	Solution
0x0001	0	Initialization fails.	1) Creating or opening the ParaFile.PF file fails; 2) Creating or opening the ComErrorFile.PF file fails; 3) Creating or opening the ServoWarnFile.PF file fails.	Check system hardware; power off and restart the system.
0x0002	0	InoTeachPad communication module scheduling fails.	The teaching box communication thread is not started normally or the hardware is damaged.	Power off and restart the system or replace the hardware.
0x0003	0	Vision communication module scheduling fails.	The vision communication thread is not started normally or the hardware is damaged.	Power off and restart the system or replace the hardware.
0x0004	0	Internal communication module scheduling fails.	The DSP communication thread is not started normally or the hardware is damaged.	Power off and restart the system or replace the hardware.
0x0005	0	Playback and teaching function module scheduling fails.	The ARM scheduling thread is not started normally or the hardware is damaged.	Power off and restart the system or replace the hardware.
0x0006	0	Data interpolation module scheduling fails.	The interpolation test thread is not started normally or the internal test function is not open.	Check the software version.
0x0007	0	EtherCAT communication opening fails.	1) Configuration file error occurs; 2) The EtherCat slave is not consistent with system configurations.	1) Restore factory settings and power on the system; 2) Check slave configurations.
0x0008	0	Opening the parameter configuration file fails.	Parameter configuration file development fails or the file is damaged.	Restore factory settings and power on the system.
0x0009	0	Decoding error	Program syntax error	Check programming specifications.
0x000A	0	Decoding row No. error	The row No. sending command of the teaching box exceeds the range.	Check whether the teaching program is incorrect.
0x000B	0	I/O waiting timeout	I/O waiting time exceeds any set time.	1) Check I/O ports; 2) Reset waiting time.
0x000C	0	Command reading error	1) The teaching file is damaged; 2) The preparation of the teaching file does not conform to specifications.	1) Re-teach the file. 2) Check teaching programming specifications.
0x000D	0	The subprogram does not allow any nesting call.	The nesting call of the subprogram exists.	Change the teaching program.
0x000E	0	Motion command decoding error	Motion command decoding error	Check the teaching program.
0x000F	0	The initialization file cannot be found.	The initialization file is damaged or lost.	1) Restore factory defaults and power on the system again; 2) Replace the hardware.

Fault Code	Level	Description	Possible Causes	Solution
0x0010	0	Playback data calculation error	1) Teaching point getting error occurs; 2) A teaching point is within a singular range.	Reselect a teaching point.
0x0011	0	Creating an axis interpolation thread fails.	1) Creating a test interpolation thread fails. 2) The internal test function is not open.	1) Replace the hardware; 2) Replace the software version.
0x0012	0	The jump command fails.	Point data calculation error occurs in the jump command.	Reselect a teaching point.
0x0013	0	IRLink initialization fails.	1) Configuration error of the number of IRLink slaves; 2) Configuration error of IRLink slave sequence.	Reconfirm IRLink configurations.
0x0014	2	Teaching program saving fails.	The SD card is loose or cannot be identified.	Check the SD card.
0x0015	0	Communication error between the system and the motion module	DSP software running error	Power on the controller again.
0x0016	0	Internal enablement error	DSP software running error	Power on the controller again.
0x0017	0	Program running error in the system motion module	DSP software running error	Power on the controller again.
0x0018	0	Homing fails.	The relative encoder homing fails.	Perform homing again.
0x0019	0	Enablement loss	The enablement of the running state is lost.	Check whether the system is in enabled state.
0x001A	1	Data recovery fails.	Ferroelectric data recovery fails.	Check whether any ferroelectric exists in the system.
0x0020	2	It is too fast from stop to startup.	It is too fast from stop to startup.	Switch stop to startup at a slow speed.
0x0021	0	Input parameter error	Decoding input parameter error	Check command parameters.
0x0022	0	No command line can be found.	The input No. exceeds the range.	Select a command line that is running.
0x0023	0	No point data can be found.	No point is defined.	Check whether a point is defined.
0x0024	0	Data inverse solution calculation error	This point is a singular point of the robot.	Modify the coordinate of this point.
0x0025	0	Parameter error of the point data coordinate system	The coordinate system values exceed the range.	Get a point again.
0x0026	0	A sudden change in the arm parameter of a straight line or arc command is not allowed.	A sudden change in the arm parameter of the MOVL and MOVC commands is not allowed.	Get a point again or add a joint transition point.
0x0027	0	The V parameter exceeds the range.	The V parameter exceeds the range (1-100).	Modify the V parameter.
0x0028	0	The Z parameter exceeds the range.	The Z parameter exceeds the range (0-5).	Modify the Z parameter.
0x0029	0	The Tool parameter exceeds the range.	The Tool parameter exceeds the range (0-15).	Modify the Tool parameter
0x002A	0	The User parameter exceeds the range.	The User parameter exceeds the range (0-15).	Modify the User parameter.
0x002B	0	The Acc parameter exceeds the range.	The Acc parameter exceeds the range (1-100).	Modify the Acc parameter.

Fault Code	Level	Description	Possible Causes	Solution
0x002C	0	The Until parameter exceeds the range.	The I/O No. exceeds the range (0-255).	Modify the Until In parameter.
0x002D	0	Pallet parameter error	PNo,i,j and k of Pallet (PNo,i,j,k) are greater than or equal to 0.	Modify the Pallet parameter.
0x002E	0	No pallet is defined.	No pallet No. is defined.	Use a pallet after it is defined.
0x002F	0	Repeat parameter error	The Repeat parameter exceeds the range.	Modify the Repeat parameter.
0x0030	0	Decoding error in running	Command parse error in running	Check any commands that are running. Modify according to a prompt.
0x0031	0	Data inverse solution operation error during step of the base coordinate system	A step target position under the base coordinate system exceeds the running space or is located in a singular area.	1) Check the size of the step; 2) Check the step direction.
0x0032	0	Data inverse solution operation error during step of the tool coordinate system	A step target position under the tool coordinate system exceeds the running space or is located in a singular area.	1) Check the size of the step; 2) Check the step direction.
0x0033	0	Data inverse solution operation error during step of the user coordinate system	A step target position under the user coordinate system exceeds the running space or is located in a singular area.	1) Check the size of the step; 2) Check the step direction.
0x004D	2	Ethernet communication error	Ethernet communication data is not complete.	Check the communication line environment and retransfer data.
0x004E	0	The network is interfered with.	The Ethernet is connected to multiple terminals.	Check whether multiple terminals are connected to the same controller.
0x004F	0	Vision command error	Vision command opening error	Check the vision command.
0x0050	1	The SD card is removed.	The SD card is removed after connection.	Check the connection of the SD card.
0x0051	0	EtherCAT is disconnected.	EtherCAT is disconnected after connection.	Check EtherCAT communication.
0x0052	0	IRLink is disconnected.	IRLink is disconnected after connection.	Check IRLink communication.
0x0053	0	The interval between startup and pause is too short.	The interval between startup and pause is short.	Restart operation.
0x0054	0	Obtaining vision characteristic values fails.	Obtaining vision characteristic values during command execution fails.	Obtain vision characteristic values again after checking vision processing.
0x0055	0	Decoding is not finished.	Command character or syntax error may exist in the program.	Check the command syntax of the edited program.
0x0056	0	Robot type error	The robot has no related type firmware.	Restart the system or check the DSP firmware.
0x0057	0	System firmware initialization error	FPGA firmware error or startup abnormality	Restart the system or check the FPGA firmware.
0x0058	0	System firmware initialization error	FPGA firmware error or startup abnormality	Restart the system or check the DSP firmware.
0x0059	2	Control device switchover is not allowed in running state.	A control device is switched over during running.	A control device can be switched over only after stop.
0x005A	0	I/O parameter setting error	I/O related parameter setting error	Check I/O parameter settings.

Fault Code	Level	Description	Possible Causes	Solution
0x005B	0	IP address error	IP address obtaining or setting error	Check network line connection.
0x005C	2	InoTeachPad has no IRLink configuration right.	The secondary development platform has been configured with IRLink.	Use the current configuration or cancel the secondary development configuration.
0x005D	0	Shared memory mapping error	Shared memory mapping error of RC and PLC	Contact the agent or Inovance.
0x005E	0	(User-defined error 1)	The system triggers a user-defined error.	Check the user error.
0x005F	0	(User-defined error 2)	The same as above.	The same as above.
0x0060	0	(User-defined error 3)	The same as above.	The same as above.
0x0061	0	(User-defined error 4)	The same as above.	The same as above.
0x0062	0	(User-defined error 5)	The same as above.	The same as above.
0x0063	0	(User-defined error 6)	The same as above.	The same as above.
0x0064	0	(User-defined error 7)	The same as above.	The same as above.
0x0065	0	(User-defined error 8)	The same as above.	The same as above.
0x0066	0	(User-defined error 9)	The same as above.	The same as above.
0x0067	0	(User-defined error 10)	The same as above.	The same as above.
0x0068	0	(User-defined error 11)	The same as above.	The same as above.
0x0069	0	(User-defined error 12)	The same as above.	The same as above.
0x006A	0	(User-defined error 13)	The same as above.	The same as above.
0x006B	0	(User-defined error 14)	The same as above.	The same as above.
0x006C	0	(User-defined error 15)	The same as above.	The same as above.
0x006D	0	(User-defined error 16)	The same as above.	The same as above.
0x006E	0	Error in interference area 1	The robot is located in the interference area.	Check robot positions and interference area setting values.
0x006F	0	Error in interference area 2	The same as above.	The same as above.
0x0070	0	Error in interference area 3	The same as above.	The same as above.
0x0071	0	Error in interference area 4	The same as above.	The same as above.
0x0072	0	Error in interference area 5	The same as above.	The same as above.
0x0073	0	Error in interference area 6	The same as above.	The same as above.
0x0074	0	Error in interference area 7	The same as above.	The same as above.
0x0075	0	Error in interference area 8	The same as above.	The same as above.
0x0076	0	Error in interference area 9	The same as above.	The same as above.
0x0077	0	Error in interference area 10	The same as above.	The same as above.
0x0078	0	Error in interference area 11	The same as above.	The same as above.
0x0079	0	Error in interference area 12	The same as above.	The same as above.
0x007A	0	Error in interference area 13	The same as above.	The same as above.
0x007B	0	Error in interference area 14	The same as above.	The same as above.
0x007C	0	Error in interference area 15	The same as above.	The same as above.

Fault Code	Level	Description	Possible Causes	Solution
0x007D	0	Error in interference area 16	The same as above.	The same as above.
0x007F	2	Data flow mode is not closed.	The system is in data flow mode.	Release data flow mode.
0x0080	0	The emergency stop button is pressed in running mode.	The emergency stop button is pressed.	Release the emergency stop button and clear an error.
0x0081	0	No fallback data is available.	The execution of fallback data has been finished.	Terminate the back operation.
0x0082	0	Port No. closing error	The port No. is not within a set range or is not opened at all.	Check the port No.
0x0083	0	TCP port overflow	Peripheral TCP application connections are excessive.	Close useless TCP connections.
0x0084	0	API processing error	The API channel is occupied by other applications for a long time or the previous API application processing is blocked due to fault.	Close or reduce the previous API application process.
0x0085	0	The arc track is uncontrolled.	The arc start point is uncertain.	Add other motion commands before and after the arc command.
0x0089	2	IP address conflict	IP address settings conflict.	Reset the IP address.
0x008A	0	The SD card does not identify a correct file system.	The file system on the SD card is incorrect.	Replace or reformat the SD card.
0x008B	0	Reading servo parameters fails.	The controller fails to read servo parameters.	Optimize the connection environment between the servo and the controller.
0x008C	0	A screw locking process group is not within a range.	Only 16 process groups are provided and the setting is not within the range.	Reset the process group to within the range of 0 to 15.
0x008D	0	I/O configurations are illegal.	Configured I/O is controlled by a PLC or does not exist.	Reset I/O.
0x008E	2	I/O setting operations are illegal.	The set I/O lacks the I/O control authority.	Check the I/O control authority.
0x008F	2	I/O does not exist	Used I/O No.exceeds the total number of configured I/Os.	Check whether I/O exists: Check real object connection. Check IRlink configurations. Check whether I/O used by the program is correct.
0x0090	0	Screw locking startup fails	Sending screw locking startup parameters to the server fails.	1) Check the electric screwdriver servo firmware matches with the controller. 2) Perform clearing and restart.
0x0091	0	The electric screwdriver fails to stop.	Sending screw locking stop parameters to the server fails.	1) Check the electric screwdriver servo firmware matches with the controller. 2) Perform clearing and restart.

Fault Code	Level	Description	Possible Causes	Solution
0x0092	0	Screw locking status detection fails.	Reading screw locking status from the server fails.	1) Check the electric screwdriver servo firmware matches with the controller. 2) Perform clearing and restart.
0x0093	0	Reading screw locking setting parameters fails.	Reading screw locking setting parameters from the server fails.	1) Check the electric screwdriver servo firmware matches with the controller. 2) Perform clearing and restart.
0x0094	0	Writing screw locking setting parameters fails.	Writing screw locking setting parameters to the server fails.	1) Check the electric screwdriver servo firmware matches with the controller. 2) Perform clearing and restart.
0x0095	0	Screw locking data display fails.	Obtaining screw locking display data from the server fails.	1) Check the electric screwdriver servo firmware matches with the controller. 2) Perform clearing and restart.
0x0096	0	Screw locking count clearing fails.	Writing a screw locking counter clearing flag to the server fails.	1) Check the electric screwdriver servo firmware matches with the controller. 2) Perform clearing and restart.
0x0097	0	Server error,failed to get thread	Server error,failed to get thread	Restart the robot.
0x0098	0	Unscrewing startup fails	Sending screw removal startup parameters to the server fails	1) Check the electric screwdriver servo firmware matches with the controller. 2) Perform clearing and restart.
0x0099	0	Writing unscrewing setting parameters fails.	Writing screw removal setting parameters to the server fails.	Check the electric screwdriver servo firmware matches with the controller.
0x009A	0	Reading unscrewing setting parameters fails.	Obtaining screw removal setting data from the server fails.	Check the electric screwdriver servo firmware matches with the controller.
0x009B	0	Unscrewing fallback point setting ultralimit	Fallback point setting ultralimit	Set the unscrewing fallback point to within a limit.
0x00A0	0	A read-write TCP port is not opened.	The read-write TCP port does not correspond to the opened TCP port.	Check the command and match the ports.
0x00A1	0	Dynamic vision is not closed.	Ordinary vision is used without closing dynamic vision.	Close dynamic vision.
0x00A2	0	Conveyor belt or camera pixel error	Conveyor belt or camera data transfer type error	Reset the Conveyor belt No. or camera data transfer type.

Fault Code	Level	Description	Possible Causes	Solution
0x00A3	0	Dynamic vision coordinate transformation error	Error occurs when the pixel coordinate is transformed into the robot coordinate.	Check whether the calibration results of a corresponding vision No. are correct.
0x00A4	0	No fallback data is available.	No fallback data is available.	Clearing any error.
0x00A5	0	There is no Movj or Movl before arc motion.	The command before arc motion is not Movj or Movl. (This appears in only the single-step teaching.)	Ensure that the command before the arc command is not Movj or Movl.
0x00A6	0	A specified file cannot be found.	The file does not exist or the file path is incorrect.	Check whether the file exists or the file path is correct.
0x00A7	2	State saving error	State saving error of the screw locking machine	Re-save the state or check the parameter range.
0x00A8	2	State export error	State export error of the screw locking machine	Re-export the state.
0x00A9	2	File quantity ultralimit	The file (folder included) quantity in the current folder is excessive.	Delete excessive files. Ensure the number of files+folders is not greater than 100.
0x00AA	0	Buffer data sending fails during pause.	Buffer data sending fails from pause to startup.	Restart from stop.
0x00AB	0	Multitask No. does not exist.	Multitask No. is written incorrectly or the range of application is incorrect.	Check task No.
0x00AC	0	Multitask No. is incorrect or occupied.	Task No. is incorrect or occupied.	Check task No.
0x00AD	0	Multitask creation error	System creation fails due to busy system or other unknown reasons.	Recreate multitasks.
0x00AE	0	Opening or closing the position latching function fails.	Opening or closing the position latching function fails.	Check hardware connection or restart the system.
0x00AF	2	Controller debug function invocation fails.	The controller debug function module fails.	Check whether the configuration of the controller debug switch is correct.
0x00B0	2	Load parameter setting fails.	The load parameter fails or exceeds the range.	Reset or check the parameter.
0x00B1		The read-write serial port is not opened.	The read-write serial port does not correspond to the opened serial port.	Check the command and match the serial ports.
0x1001	2	A directory is recreated.	The directory to be created has existed in this path.	Replace the directory name to be created.
0x1002	2	Memory operation error occurs. No previous directory exists.	The current created directory has no parent directory.	Replace a path and recreate a directory.
0x1003	2	The directory to be renamed does not exist.	Your are trying to rename a directory that does not exist.	Refresh the directory and check whether the directory exists.
0x1004	2	A directory to be deleted does not exist.	The directory to be deleted does not exist.	Refresh the directory and check whether the directory exists.
0x1005	2	Directory sending error (It is not a directory or does not exist)	The directory to be sent to the upper computer is illegal.	Refresh the directory and check whether the directory exists.
0x1006	2	Memory and path error during file creation	Creation path error	Refresh the file and check the creation path.

Fault Code	Level	Description	Possible Causes	Solution
0x1007	2	The directory to be renamed does not exist.	Your are trying to rename a directory that does not exist.	Refresh the file and check whether the file exists.
0x1008	2	A directory to be deleted or a path does not exist.	The directory to be deleted does not exist.	Refresh the file and check whether the file exists.
0x1009	2	A given file path does not exist.	The given created file path is illegal.	Check the creation path for legality.
0x100A	2	A non-file is sent.	A non-file is sent.	The hand-held box inspection requires that a file should be sent.
0x100B	2	A non-directory is sent.	A non-directory is sent.	The hand-held box inspection requires that a directory should be sent.
0x100C	2	Frame sequence error	Frame sequence error in the process of sending a large file	Resend the file.
0x100D	0	Abnormality causes network cable disconnection.	1) The hand-held box is not closed according to normal operations; 2) The network is disconnected abnormally.	1) For any error due to abnormal operations, please actively disconnect the network cable; 2) For abnormality, check the error causes in conjunction with the existing error codes.
0x100E	0	Time format setting error	Time format setting error	A correct time format shall be given according to the user manual.
0x100F	0	System timing execution error	System time calculation circuit error	Check the current network connection environment.
0x1010	0	RTC timing execution error	RTC external batteries do not exist or are low.	Replace batteries or check the current hardware.
0x1011	2	Copying execution error	Misoperation during file copying	Perform copy operations again by referring to the user manual.
0x1012	2	Cutting execution error	Misoperation during file cutting	Perform cut operations again by referring to the user manual.
0x1013	0	Vision device communication is abnormal.	1) The vision device cannot be connected; 2) The communication network is disconnected abnormally.	1) Check whether the vision device is normal; 2) Check the current network connection environment.
0x1101	0	Internally exchanged data length is invalid.	When the DSP is requested to provide data, the DSP does not return valid data as required.	Check the DSP software version or replace the hardware.
0x1102	0	Internally exchanged data check error	When the DSP is requested to provide data, the DSP does not return valid data as required or returned data has error codes.	Check the DSP software version or replace the hardware.
0x1103	0	Error when internally writing fast data to the system	The GPMC channel is abnormal.	Check the DSP software version or replace the hardware.
0x1104	0	Error when internally reading fast data from the system	The GPMC channel is abnormal.	Check the DSP software version or replace the hardware.

Fault Code	Level	Description	Possible Causes	Solution
0x1105	0	The internal block data buffer of the system is full.	The FPGA buffer is full and cannot accept new data.	Retry to write data after delay for a period of time.
0x1106	0	Internal channel opening error: abnormal or occupied	The GPMC channel is abnormal or has been occupied.	Restart
0x1107	0	Internal channel opening error: abnormal or closed	The GPMC channel is abnormal or has been closed.	Restart
0x1108	0	The system is internally busy.	The DSP does not respond to the ARM command.	Check the DSP status for whether to be paused or terminated.
0x1109	0	Error when the system tries to internally obtain channel resources	The CPMC channel is frequently occupied so that currently the system cannot apply for use of CPMC resources.	Retry to use them after delay for a period of time.
0x110A	0	Timeout when the system internally waits for a response from the motion module	The response time to ARM commands from the DSP exceeds the set maximum waiting time.	Check the DSP status for whether to be paused or terminated.
0x110B	0	The system motion module cannot execute commands effectively.	The execution of ARM commands by the DSP fails.	Check the DSP software version or replace the hardware.
0x110C	0	Parameters set for the motion module are illegal.	Parameters set by the ARM for the DSP are incorrect, e.g. Exceeding the parameter range.	Check function call interface parameters to ensure that they are correct.
0x110D	0	Commands set for the motion module are illegal.	Commands requested by the ARM to the DSP are invalid.	Check the name to ensure that there is processing on this command in the DSP.
0x110E	0	The number of axes configured by the system is not consistent with that scanned online.	The model is not matched or servo is disconnected.	Check the model and servo connection to ensure that the current robot is consistent with the configured robot.
0x110F	0	Axis data set by the system is not consistent with that re-obtained from the motion module.	Data check error	Check the software version or replace the hardware.
0x1110	0	I/O data set by the system is not consistent with that re-obtained from the motion module.	Data check error	Check the software version or replace the hardware.
0x1111	0	Error when an EtherCAT command controls servo to enter the homing mode.	Servo cannot enter the homing mode.	Check the software version or replace the hardware.
0x1112	0	Error when an EtherCAT command controls servo to exit the homing mode.	Servo cannot exit the homing mode.	Check the software version or replace the hardware.
0x1113	0	Error when an EtherCAT command sets servo homing parameters	The set homing parameters are not accepted by servo.	Check the software version or replace the hardware.
0x1114	0	System setting parameter check error	Data check error	Check the software version or replace the hardware.
0x1115	0	Data channel opening error	The GPMC channel No. is incorrect.	The default channel is 0. Ensure that the channel No. is correct.

Fault Code	Level	Description	Possible Causes	Solution
0x1116	0	Data channel mapping error	The GPMC data channel cannot map data to the memory.	Check the software version or replace the hardware.
0x1117	0	Data channel mapping error	The GPMC channel is abnormal.	Check the software version or replace the hardware.
0x1118	0	Data channel mapping error	The GPMC channel is abnormal.	Check the software version or replace the hardware.
0x1119	0	Device error	The GPMC channel is abnormal.	Check the software version or replace the hardware.
0x111A	0	Data channel opening error	The GPMC channel is abnormal.	Check the software version or replace the hardware.
0x111B	0	Data communication inconsistency	The GPMC channel is abnormal.	Check the software version or replace the hardware.
0x111C	0	Data communication inconsistency	The GPMC channel is abnormal.	Check the software version or replace the hardware.
0x111D	0	System application memory error	The system cannot assign a required memory.	Check whether the system memory is close to a consumption limit or whether the memory to apply for is too large.
0x111E	0	Error of I/O data offset information configured on the IR-LINK bus	Data check error	Check the software version or replace the hardware.
0x111F	0	Error of AD data offset information configured on the IR-LINK bus	Data check error	Check the software version or replace the hardware.
0x1120	0	Error of DA data offset information configured on the IR-LINK bus	Data check error	Check the software version or replace the hardware.
0x1121	0	Error of Encoder data offset information configured on the IR-LINK bus	Data check error	Check the software version or replace the hardware.
0x1122	0	Error of AD parameters (ranges) configured on the IR-LINK bus	Data check error	Check the software version or replace the hardware.
0x1123	0	Error of DA parameters (ranges) configured on the IR-LINK bus	Data check error	Check the software version or replace the hardware.
0x1124	0	Error of module numerical codes set on the IR-LINK bus	Data check error	Check the software version or replace the hardware.
0x1125	0	Error when synchronizing the controller planned position and the encoder feedback position	The current axis does not exist or the GPMC channel or DSP firmware is abnormal.	Check the software version or replace the hardware.
0x2001	0	Paragraph data coincides.	The target positions of the previous input and this input are the same.	Re-teach points.

Fault Code	Level	Description	Possible Causes	Solution
0x2002	0	Arc input parameter calculation error	Arc interpolation information cannot be calculated because: 1) At least two points are too close; 2) Three points are approximately in the same line; 3) Pose transformation is too large; 4) Error exists in the transition with other tracks. For example,transition is performed near a singular point.	Re-teach other points to calculate the arc.
0x2003	0	Line input parameter calculation error	Line interpolation information cannot be calculated because: 1) Pose transformation is too large; 2) Error exists in the transition with other tracks. For example,transition is performed near a singular point.	Re-teach other points to calculate the line.
0x2004	0	Inverse solution operation error	The robot is located in a singular area or beyond an accessible range.	Turn off servo and switch to the joint mode. After clearing the error, turn on servo and move the robot to a proper position.
0x2005	0	Singular position error error	The robot moves to a singular area.	Switch to the joint mode and move the robot out of the singular point.
0x2006	0	Servo disconnection occurs in the playback motion.	1) Power failure of a drive occurs; 2) A drive is wired incorrectly; 3) A drive fails.	Check whether the drive is abnormal.
0x2007	0	Reserved	The pose change is too large.	Re-teach other points to calculate the planning of this section.
0x2008	0	The I/O index access range is exceeded.	The I/O access physical terminal does not exist.	Check whether a corresponding I/O physical module is available.
0x2009	0	Jump parameter setting error	MH is beyond the limit range of Axis 3.	Modify a corresponding parameter and limit height or reselect a start or end position.
0x200A	0	Arm type parameter error	Three arm type parameters before the end point are not consistent with those of the start point in line or arc motion.	Change the latter motion command to MovJ or reselect points to ensure consistent arm type.
0x200B	0	The set motion characteristic parameter is unreasonable.	The motion parameter input range is unreasonable.	Remodify the motion parameter.
0x200C	0	DA operation error	The channel is configured as current output, but the voltage command is used. Or the channel is configured as voltage output, but the current command is used.	Operate the DA port using a command consistent with the configuration.

Fault Code	Level	Description	Possible Causes	Solution
0x200D	0	The servo enabled command is sent, but actual feedback indicates that servo is not enabled.	1) The main servo power supply is not switched on; 2) The joint may be in deceleration process; 3) The joint is in motion state and does not arrive at the position.	1) Check whether the strong current button of the control cabinet is be pressed; 2) The interval between servo stop and startup or power-on; 3) Check whether the joint arrives at the position. Amplify the arrival error or adjust the servo parameter.
0x200E	0	Joint motion input parameter error	1) Beyond the space range for the Delta model; 2) MoveJ and MOVL or MOVC performs transition near a singular point.	1) Check whether the point position is beyond the space range for the Delta model; 2) Check whether MoveJ and MOVL or MOVC performs transition near a singular point.
0x200F	0	The robot does not perform the homing operation.	The robot does not perform the homing operation when an incremental encoder is used.	For a relative encoder, perform the homing operation first.
0x2010	0	The robot oversteps the boundary in the radius direction.	The X and Y combined radius at the end of the robot is greater than the set radius.	Under the rectangular coordinate system, step so that the robot end moves in the direction of reduction of the X and Y combined radius.
0x2011	0	The robot oversteps the boundary in the positive direction of Z.	The Z at the end of the robot is greater than the setpoint.	Under the rectangular coordinate system, step so that the robot end moves in the negative direction of Z.
0x2012	0	The robot oversteps the boundary in the negative direction of Z.	The Z at the end of the robot is smaller than the setpoint.	Under the rectangular coordinate system, step so that the robot end moves in the positive direction of Z.
0x2013	0	The robot oversteps the boundary.	The teaching point oversteps the boundary in online running.	Change the teaching point so that it is within the work space of the robot.
0x2016	0	The included angle between Axes 2 and 3 of the stacking robot is too small.	The included angle between Axes 2 and 3 of the stacking robot is too small.	Rotate Axis 3 in positive direction or Axis 2 in negative direction in teaching mode.
0x2017	0	The included angle between Axes 2 and 3 of the stacking robot is too greater.	The included angle between Axes 2 and 3 of the stacking robot is too greater.	Rotate Axis 3 in negative direction or Axis 2 in positive direction in teaching mode.

Fault Code	Level	Description	Possible Causes	Solution
0x2018	0	The robot speed is abnormal.	The robot joint speed exceeds twice the allowable maximum speed.	1) Ensure that the settings of allowable maximum speed and acceleration of joints on the setting interface are reasonable; 2) Ensure that the robot is not near a singular position; 3) Reduce the line motion speed.
0x2019	0	Motion parameter error	The motion planning parameters are abnormal.	Check whether the settings of motion parameters are reasonable.
0x201A	0	The robot position or pose speed exceeds a setpoint.	The robot end motion exceeds the set position or pose speed.	1) If the MoveJ command is used, set the MoveJ speed coefficient to a smaller value; 2) Check whether the set pose speed is consistent with the J4 joint speed.
0x201D	0	Arm type change error	The arm type change command (specially used for SCARA) is not supported near a singular position.	Replace the point with a point far from the singular area and then call the arm type change command.
0x201E	0	The robot acceleration is abnormal.	The robot joint speed exceeds 50 times the allowable maximum acceleration.	1) Ensure that the settings of allowable maximum speed and acceleration of joints on the setting page are reasonable; 2) Ensure that the robot is not near a singular position; 3) Reduce the speed and acceleration of line motion.
0x201F	0	Speed settings are abnormal.	Too small set speed of joints or too large set speed of spatial tracks causes too many constraint points.	1) Increase the joint speed or reduce the set speed of spatial tracks. 2) Ensure that the robot does not perform spatial track motion near a singular position.
0x2021	0	The robot exceeds the upper boundary of the operating range.	The robot exceeds the set operating range in the tracking process.	Adjust the upper boundary according to the description of conveyor belt parameter settings.
0x2022	0	The robot exceeds the lower boundary of the operating range.	The robot exceeds the set operating range in the tracking process.	Adjust the lower operating boundary and pick up the upper boundary according to the description of conveyor belt parameter settings; increase the robot speed; reduce the conveyor belt speed.

Fault Code	Level	Description	Possible Causes	Solution
0x2023	0	The conveyor belt speed is too large.	The conveyor belt speed exceeds a reasonable range.	The conveyor belt speed exceeds the maximum speed limit (1 m/s for the linear conveyor belt and 180o/s for the conveyor turntable).
0x2024	0	The conveyor belt speed fluctuates excessively.	The conveyor belt speed fluctuates.	Check whether the conveyor belt motor speed fluctuates excessively or whether the conveyor belt is abnormal.
0x2025	0	Vision data waiting timeout	If no returned data is received for a long period of time after vision triggering signals are sent, then the vision processing period is greater than the shooting time interval.	Check whether the vision processing period is greater than the shooting triggering period.
0x2026	0	Robot coordinate type error	A static coordinate is used in the tracking command or a dynamic coordinate is used in ordinary motion.	Check whether the command point position type is 7 (dynamic object coordinate system) in the tracking process or whether a Type 7 point position is used during the non-tracking period.
0x2027	0	Dynamic point coordinate error	The given dynamic target position is incorrect, singular or out of bounds.	Check whether the coordinate given by vision is within a reasonable range.
0x2028	0	Syntax error in the conveyor belt tracking command	RefConvyor or RefBase is continuously used.	Check whether RefConvyor and RefBase are jointly used.
0x2029	0	Establishing a workpiece grabing coordinate system fails.	GetCnvObject or RefConvyor is executed.	Call the GetCnvObject command and then track.
0x202A	0	Conveyor belt vision port error	Multiple vision conveyor belts are used.	Check whether more than two vision inputs are used at the same time.
0x202B	0	Single-step teaching is not allowed.	The conveyor belt tracking-related commands do not allow single-step teaching.	Single-step operation on commands between RefConvyor and RefBase is not allowed.
0x202C	0	A disabled conveyor belt is used in the command.	A disabled conveyor belt is used in the command.	Check whether the conveyor belt No. used in the program is disabled.
0x202D	0	PTP motion is not allowed in the tracking process.	Joint motions such as MovJ and JumP are used in the tracking process.	Check whether the MovJ or jump command is used between RefConvyor and RefBase. (Note: The jump command can be used in Delta.)
0x202E	0	Conveyor belt speed direction error	The conveyor belt speed is detected to be a negative value.	1) Check whether the encoder direction parameter on the conveyor belt interface is correct. 2) Check whether the conveyor belt shows the evidence of slipping.

Fault Code	Level	Description	Possible Causes	Solution
0x202F	0	The right-angle teaching start position is located in the singular position where inverse solution cannot be executed.	The right-angle teaching start position is located in the singular position where inverse solution cannot be executed.	Remove the singular position for joints.
0x2030	0	The latching counters of the axes are not consistent.	When the robot position is latched, the latching counters of the axes are not consistent.	Check whether the latching signals of the axes are active.
0x2031	0	The latching buffer is full.	When the robot position is latched, the latching buffer is full.	There are too many latches that are not used.
0x2041	0	Reserved (Defined after 15D)	The planned track has a limit.	Check the planned track of a corresponding section. Check whether a limit is triggered at an intermediate process point.
0x2042	0	There is a singular position in the planned track.	There is a singular position in the planned track.	Check the planned track of a corresponding section. Check whether a singular position is triggered at an intermediate process point.
0x0243	0	A position where inverse solution cannot be executed exists in the planned track.	Inverse solution cannot be executed at an intermediate point of the planned track.	Modify point getting in the error section.
0x2044	0	The position goes beyond the lower operating boundary during the dynamic tracking preprocessing.	The target point is detected to go beyond the lower operating boundary during the dynamic tracking preprocessing.	1) Check whether the received vision data point is within a reasonable range; 2) Check whether the dynamic coordinate of the given motion command goes beyond the lower operating boundary.
0x2045	0	A limit is triggered at Joint 1 in the planned track.	A limit is triggered at Joint 1 in the track.	Check the planned track of a corresponding section. Check whether a limit is triggered at an intermediate process point.
0x2046	0	A limit is triggered at Joint 2 in the planned track.	A limit is triggered at Joint 2 in the track.	Check the planned track of a corresponding section. Check whether a limit is triggered at an intermediate process point.
0x2047	0	A limit is triggered at Joint 3 in the planned track.	A limit is triggered at Joint 3 in the track.	Check the planned track of a corresponding section. Check whether a limit is triggered at an intermediate process point.
0x2048	0	A limit is triggered at Joint 4 in the planned track.	A limit is triggered at Joint 4 in the track.	Check the planned track of a corresponding section. Check whether a limit is triggered at an intermediate process point.

Fault Code	Level	Description	Possible Causes	Solution
0x2049	0	A limit is triggered at Joint 5 in the planned track.	A limit is triggered at Joint 5 in the track.	Check the planned track of a corresponding section. Check whether a limit is triggered at an intermediate process point.
0x204A	0	A limit is triggered at Joint 6 in the planned track.	A limit is triggered at Joint 6 in the track.	Check the planned track of a corresponding section. Check whether a limit is triggered at an intermediate process point.
0x2071	0	A limit is triggered at Joint 1 in the planned track during teaching.	A limit is triggered at Joint 1 in the track.	Check the planned track of a corresponding section. Check whether a limit is triggered at an intermediate process point.
0x2072	0	A limit is triggered at Joint 2 in the planned track during teaching.	A limit is triggered at Joint 2 in the track.	Check the planned track of a corresponding section. Check whether a limit is triggered at an intermediate process point.
0x2073	0	A limit is triggered at Joint 3 in the planned track during teaching.	A limit is triggered at Joint 3 in the track.	Check the planned track of a corresponding section. Check whether a limit is triggered at an intermediate process point.
0x2074	0	A limit is triggered at Joint 4 in the planned track during teaching.	A limit is triggered at Joint 4 in the track.	Check the planned track of a corresponding section. Check whether a limit is triggered at an intermediate process point.
0x2075	0	A limit is triggered at Joint 5 in the planned track during teaching.	A limit is triggered at Joint 5 in the track.	Check the planned track of a corresponding section. Check whether a limit is triggered at an intermediate process point.
0x2076	0	A limit is triggered at Joint 6 in the planned track during teaching.	A limit is triggered at Joint 6 in the track.	Check the planned track of a corresponding section. Check whether a limit is triggered at an intermediate process point.
0x2077	0	A position where inverse solution cannot be executed exists in the planned track in teaching state.	Inverse solution cannot be executed at an intermediate point of the planned track.	Modify point getting in the error section.
0x2078	0	Line input parameter calculation error in teaching state	Line interpolation information cannot be calculated.	Re-teach other points to calculate the line.
0x2079	0	The arm type is inconsistent with the current point during preprocessing in teaching state and space interpolation cannot be performed.	ARM_TYPE is found to be inconsistent with the current point during preprocessing and space interpolation cannot be performed.	Modify point getting in the error section.
0x207A	0	Pose motion is found to be too large in teaching state.	The pose motion angle is found to be greater than 180oin the planned track.	Reduce the pose change range of a single motion.

Fault Code	Level	Description	Possible Causes	Solution
0x207B	0	There is a singular position in the planned track in teaching state.	There is a singular position in the planned track.	Check the planned track of a corresponding section. Check whether a singular position is triggered at an intermediate process point.
0x2101	0	Axis 1 positive limit error	The joint limit position is reached.	The robot moves in the opposite direction. In case of non-joint mode, switch to the joint mode first.
0x2102	0	Axis 1 negative limit error	The joint limit position is reached.	The robot moves in the opposite direction. In case of non-joint mode, switch to the joint mode first.
0x2103	0	Axis 1 drive error	The drive sends an error.	For servo error codes, refer to Mon->Servo->J1-No.12 and 13. Perform troubleshooting by referring to the servo manual according to fault codes.
0x2104	0	Axis 1 planning overflow error	The planned value exceeds the maximum calculation range (-1073741823 to 1073741824).	Check whether the absolute origin position is selected to be near the counting limit edge. If the zero point is near the boundary, clear the driver turns.
0x2105	0	Error due to excessive tracking error of Axis 1	The difference between the planned and actual positions is too large.	Adjust the servo parameter and reduce the corresponding hysteresis.
0x2106	0	Axis 1 overspeed error	The running speed is greater than the set maximum speed.	Reduce the maximum speed in Cartesian space.
0x2107	0	Strong current of Axis 1 is not switched on.	Strong current is not switched on.	Check the drive circuit for whether strong current is not switched on.
0x2111	0	Axis 2 positive limit error	The same as Axis 1	The same as Axis 1
0x2112	0	Axis 2 negative limit error		
0x2113	0	Axis 2 drive error		
0x2114	0	Axis 2 planning overflow error		
0x2115	0	Error due to excessive tracking error of Axis 2		
0x2116	0	Axis 2 overspeed error		
0x2117	0	Strong current of Axis 2 is not switched on.		
0x2201	0	Axis 3 positive limit error	The same as Axis 1	The same as Axis 1
0x2202	0	Axis 3 negative limit error		
0x2203	0	Axis 3 drive error		
0x2204	0	Axis 3 planning overflow error		
0x2205	0	Error due to excessive tracking error of Axis 3		
0x2206	0	Axis 3 overspeed error		
0x2207	0	Strong current of Axis 3 is not switched on.		

Fault Code	Level	Description	Possible Causes	Solution
0x2211	0	Axis 4 positive limit error	The same as Axis 1	The same as Axis 1
0x2212	0	Axis 4 negative limit error		
0x2213	0	Axis 4 drive error		
0x2214	0	Axis 4 planning overflow error		
0x2215	0	Error due to excessive tracking error of Axis 4		
0x2216	0	Axis 4 overspeed error		
0x2217	0	Strong current of Axis 4 is not switched on.		
0x2301	0	Axis 5 positive limit error	The same as Axis 1	The same as Axis 1
0x2302	0	Axis 5 negative limit error		
0x2303	0	Axis 5 drive error		
0x2304	0	Axis 5 planning overflow error		
0x2305	0	Error due to excessive tracking error of Axis 5		
0x2306	0	Axis 5 overspeed error		
0x2307	0	Strong current of Axis 5 is not switched on.		
0x2311	0	Axis 6 positive limit error	The same as Axis 1	The same as Axis 1
0x2312	0	Axis 6 negative limit error		
0x2313	0	Axis 6 drive error		
0x2314	0	Axis 6 planning overflow error		
0x2315	0	Error due to excessive tracking error of Axis 6		
0x2316	0	Axis 6 overspeed error		
0x2317	0	Strong current of Axis 6 is not switched on.		

Fault Code	Level	Description	Possible Causes	Solution
0x8001	0	Error due to no network equipment	ECAT communication initialization error	Check whether the FPGA and DSP firmware is loaded successfully.
0x8002	0	Error due to no master		
0x8003	0	Error due to invalid domain		
0x8004	0	Error because this slave is unavailable		
0x8005	0	Error due to invalid process data		
0x8006	0	Error due to invalid service data		
0x8007	0	Error due to invalid entry object		
0x8008	0	Domain memory address assignment error		
0x8009	0	Error due to master activation failure		
0x800A	0	Service data public error		
0x800B	0	Registration cycle callback error		
0x800C	0	Process communication configuration error		
0x800D	0	Module initialization error		
0x800E	0	Parse configuration error		
0x800F	0	Configuration channel parameter error		
0x8010	0	Domain registration error		
0x8011	0	Timer creation error		
0x8012	0	Timer startup error		

Fault Code	Level	Description	Possible Causes	Solution
0x8013	0	ECAT communication cycle configuration error	ECAT slave configuration error	Check ECAT configuration information. Ensure that the system is restarted after configurations are correctly saved.
0x8014	0	ECAT version selection configuration error		
0x8015	0	ECAT servo slave quantity configuration error		
0x8016	0	ECAT I/O slave quantity configuration error		
0x8017	0	ECAT I/O module quantity configuration error		
0x8018	0	ECAT I/O type configuration error		
0x8019	0	Configuration error of nonsupport for ECAT I/O		
0x801A	0	Configuration error of ECAT memory application		
0x801B	0	Configuration error of ECAT error shared memory		
0x801C	0	Configuration error of ECAT servo operation mode		
0x801D	0	Configuration error of ECAT register		
0x801E	0	Configuration error of mismatching between the ECAT I/O quantity and the online I/O quantity		
0x801F	0	Configuration error of mismatching between the ECAT servo quantity and the online servo quantity		
0x8020	0	Configuration error of nonsupport for ECAT servo supplier codes		

Fault Code	Level	Description	Possible Causes	Solution
0x8028	0	Buffer write error	ECAT communication running error	Check ECAT network connection. Check whether the ECAT slave is powered off.
0x8029	0	Start command write error		
0x802A	0	State register read error		
0x802B	0	Data link state read error		
0x802C	0	Service data channel read error		
0x802D	0	Service data length read error		
0x802E	0	Service data length error		
0x802F	0	Service data receiving error		
0x8030	0	Error due to busy service data channel		
0x8031	0	Service data packet error		
0x8032	0	Process data communication read error		
0x8033	0	Process data length read error		
0x8034	0	Process data length error		
0x8035	0	Process data receiving error		
0x8036	0	Network device opening error		
0x8037	0	Bottom communication error		
0x8038	0	Bottom communication error		
0x8039	0	Bottom communication error		
0x803A	0	Sent time stamp read error		
0x803B	0	Received time stamp read error		
0x803C	0	Process data surplus read error		
0x803D	0	Application time stamp read error		
0x803E	0	Field bus LED opening error		
0x803F	0	Field bus LED IOCTRL error		
0x8040	0	Bottom hardware invalid		
0x8041	0	Bottom hardware invalid		
0x8042	0	ECAT slave disconnection error		Ensure that the ECAT slave network cable is correctly connected; power off and restart the system.
0x8N42	0	ECAT slave N disconnection error		
0x8043	0	Non-ECAT slave device connection error		Insert the ECAT network cable into the ECAT network port.
0x8044	0	ECAT network port 1 disconnection error		
0x8045	0	ECAT network port 2 disconnection error		

Fault Code	Level	Description	Possible Causes	Solution
0x8046	0	ECAT startup time setting error	ECAT periodic communication startup error	Check whether slave status is abnormal.
0x805C	0	Toggle bit unchanged error	ECAT slave service data communication error	Check a request for service data like operation data format of servo function codes.
0x805D	0	SDO protocol timeout error		
0x805E	0	Error due to invalid or unknown commands		
0x805F	0	Error due to inaccessible objects		
0x8060	0	Error when retrying to read one write-only object		
0x8061	0	Error when retrying to write one read-only object		
0x8062	0	Error that an object does not exist in the object directory		
0x8063	0	Error that an object cannot be mapped as process data		
0x8064	0	Error that the length of mapped objects exceeds that of process data		
0x8065	0	Basic parameter incompatibility error		
0x8066	0	Internal incompatibility error in the device		
0x8067	0	Error that the hardware causes access failure		
0x8068	0	Service parameter length mismatching error		
0x8069	0	Error that the service parameter is too long		
0x806A	0	Error that the service parameter is too short		
0x806B	0	Error that the sub-index does not exist		
0x806C	0	Error that the parameter value is beyond the range		
0x806D	0	Error that the written parameter value is too large		
0x806E	0	Error that the written parameter value is too small		
0x806F	0	Error that the maximum value is smaller than the minimum value		
0x8070	0	Error that data cannot be transferred or stored		
0x8071	0	Error that data cannot be stored due to local control		
0x8072	0	Error that data cannot be stored due to device status		
0x8073	0	Object directory error or object directory unavailable		

Fault Code	Level	Description	Possible Causes	Solution
0x807A	0	IR-LINK communication period configuration error	IR-LINK slave configuration error	Check IR-LINK configuration information. Ensure that the system is restarted after configurations are correctly saved.
0x807B	0	IR-LINK version selection configuration error		
0x807C	0	IR-LINK servo slave quantity configuration error		
0x807D	0	IR-LINK slave quantity configuration error		
0x807E	0	IR-LINK module quantity configuration error		
0x807F	0	IR-LINK type configuration error		
0x8080	0	Configuration error of nonsupport for IR-LINK		
0x8081	0	Configuration error of IR-LINK memory application		
0x8082	0	Configuration error of IR-LINK error shared memory		
0x8083	0	Configuration error of IR-LINK servo operation mode		
0x8084	0	Configuration error of IR-LINK register		
0x8085	0	Configuration error of mismatching between the IR-LINK I/O quantity and the online I/O quantity		
0x8086	0	Configuration error of mismatching between the IR-LINK slave quantity and the online slave quantity		
0x8087	0	Configuration error of nonsupport for IR-LINK servo supplier codes		

Fault Code	Level	Description	Possible Causes	Solution
0x8090	0	Buffer write error	IR-LINK running error	Check IR-LINK network connection; Check whether the IR-LINK slave is powered off.
0x8091	0	Service data channel read error		
0x8092	0	Service data length read error		
0x8093	0	Service data length error		
0x8094	0	Service data receiving error		
0x8095	0	Error due to busy service data channel		
0x8096	0	Service data packet error		
0x8097	0	Process data communication read error		
0x8098	0	Process data length read error		
0x8099	0	Process data length error		
0x809A	0	Process data receiving error		
0x809B	0	Network device opening error		
0x809C	0	GPMC IR-LINK read error		
0x809D	0	GPMC IR-LINK write error		
0x809E	0	IR-LINK slave disconnection error		Ensure that the IR-LINK slave network cable is correctly connected; power off and restart the system.
0x8N9E	0	IR-LINK slave N disconnection error		
0x809F	0	Non-IR-LINK slave device connection error		
0x80A0	0	IR-LINK network port 0 disconnection error	IR-LINK network port 0 disconnection	Insert the IR-LINK network cable into the IR-LINK network port.
0x80A1	0	IR-LINK network port 1 disconnection error		
0x80A2	0	IR-LINK startup time setting error	IR-LINK periodic signal startup error	Check whether slave status is abnormal.
0xE001	0		Normal running warning	No processing is required.
0xE002	0	Firmware loading fails.	Abnormality occurs when the FPGA_FW.bin firmware is read from the first partition of the SD card because this file may be unavailable or damaged.	Check whether the FPGA_FW.bin firmware exists in the SD card.
0xE003	0	Firmware loading fails.	Abnormality occurs when the FPGA_FW.bin firmware is read from the SPI flash robotfw partition because this file may be unavailable or damaged.	Check whether the FPGA_FW.bin firmware exists in SPI flash robotfw.
0xE004	0	Firmware resetting fails.	After the ARM sends a reset signal to the FPGA chip, no response is received.	Check the FPGA and ARM communication pin configurations.

Fault Code	Level	Description	Possible Causes	Solution
0xE005	0	Firmware loading fails.	Abnormality occurs when the ARM sends data to the FPGA chip.	Check the FPGA and ARM communication pin configurations.
0xE006	0	Firmware loading fails.	After the ARM finishes sending data to the FPGA, no normal running report is received from the FPGA.	Check whether the FPGA chip or firmware is correct.
0xE007	0	Control channel 1 runs normally.	Normal running warning	No processing is required.
0xE008	0	Memory assignment failure in control channel 1	System software error	Check the software version and power off and restart the system.
0xE009	0	Firmware loading fails.	Firmware in control channel 1 cannot be found in the SD card.	Check whether firmware in control channel 1 is stored in the SD card.
0xE00A	0	Firmware loading fails.	Firmware in control channel 1 cannot be found in the SPI flash robotfw.	Check whether firmware in control channel 1 exists in the SPI flash robotfw.
0xE00B	0	The firmware length in control channel 1 exceeds the range.	Currently, firmware is assumed to have a maximum length of 1 Mbyte. This firmware has exceeded this length.	Check whether this firmware has exceeded this length.
0xE00C	0	Firmware loading fails.	The FPGA may operate abnormally.	Check whether the FPGA running indicator blinks normally.
0xE00D	0	Control channel 1 resetting failure	In control channel 1, it may fails to start firmware program from the SPI.	Check the hardware startup mode in control channel 1.
0xE00E	0	SPI communication failure	SPI communication failure between the ARM and control channel 1	Check the SPI hardware lines on the ARM side and on the control channel 1 side.
0xE00F	0	SPI communication failure	SPI communication failure between the ARM and control channel 1	Check the SPI hardware lines on the ARM side and on the control channel 1 side.
0xE010	0	SPI speed change failure in control channel 1	SPI communication failure between the ARM and control channel 1	Check whether control channel 1 and the ARM chip are abnormal.
0xE011	0	Data loading timeout in control channel 1	Configurations error occurs or the firmware is damaged during firmware production.	Check whether the produced firmware is correct or complete.
0xE012	0	Control channel 1 running response is abnormal.	Too long initialization time of control channel 1 causes abnormal response.	The firmware must be updated.
0xE013	0	Control channel 0 runs normally.	Normal running warning	No processing is required.
0xE014	0	Memory assignment failure in control channel 0	System software error	Check the software version and power off and restart the system.
0xE015	0	Firmware loading fails.	Firmware in control channel 0 cannot be found in the SD card.	Check whether firmware in control channel 0 is stored in the SD card.
0xE016	0	Firmware loading fails.	Firmware in control channel 0 cannot be found in the SPI flash.	Check whether firmware in control channel 0 exists in the SPI flash.
0xE017	0	The firmware length in control channel 0 exceeds the range.	Currently, firmware is assumed to have a maximum length of 1 Mbyte. This firmware has exceeded this length.	Check whether this firmware has exceeded this length.

Fault Code	Level	Description	Possible Causes	Solution
0xE018	0	Firmware loading fails.	The FPGA may operate abnormally.	Check whether the FPGA running indicator blinks normally.
0xE019	0	Control channel 0 resetting failure	In control channel 0, it may not be configured as starting a firmware program from the SPI.	Check the hardware startup mode in control channel 0.
0xE01A	0	SPI communication failure	SPI communication failure between the ARM and control channel 0	Check the SPI hardware lines on the ARM side and on the control channel 0 side.
0xE01B	0	SPI communication failure	SPI communication failure between the ARM and control channel 0	Check the SPI hardware lines on the ARM side and on the control channel 0 side.
0xE01C	0	SPI speed change failure in control channel 0	SPI communication failure between the ARM and control channel 0	Check whether control channel 0 and the ARM chip are abnormal.
0xE01D	0	Data loading timeout in control channel 0	Configurations error occurs or the firmware is damaged during firmware production.	Check whether the produced firmware is correct or complete.
0xE01E	0	Control channel 0 running response is abnormal.	Too long initialization time of control channel 0 causes abnormal response.	The firmware must be updated.

Shenzhen Inovance Technology Co., Ltd.
Add.: Building E, Hongwei Industry Park, Liuxian Road, Baocheng No. 70 Zone, Bao'an District, Shenzhen
Tel: +86-755-2979 9595
Fax: +86-755-2961 9897
Service Hotline: 400-777-1260
http: //www.inovance.com

Suzhou Inovance Technology Co., Ltd.
Add.: No. 16 Youxiang Road, Yuexi Town, Wuzhong District, Suzhou 215104, P.R. China
Tel: +86-512-6637 6666
Fax: +86-512-6285 6720
Service Hotline: 400-777-1260
http: //www.inovance.com

Copyright © Shenzhen Inovance Technology Co., Ltd.