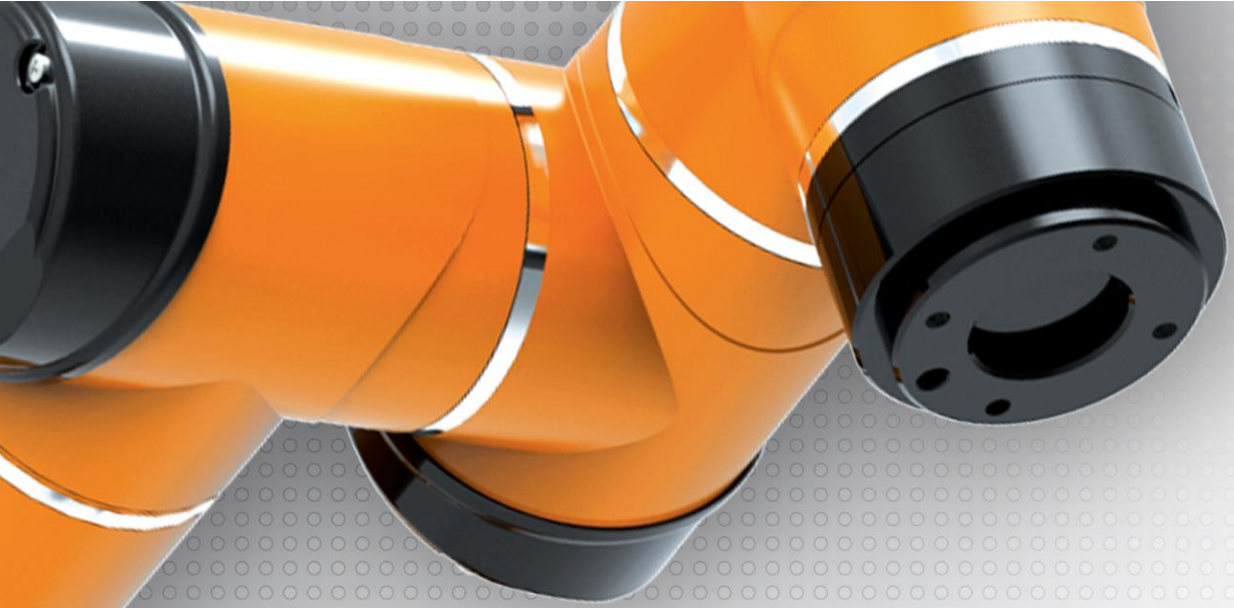


AUBORPE

Reference Guide

遨博智能
AUBO

AUBORPE

Reference Guide

Original Version 1.0.0

AUBO (Beijing) Robotics Technology Co., Ltd.

Content

Find the tool center point 5
Set the center of gravity and payload 7
New Project 8
Save Project 9
Load project 10
Run Project 11
Insert action and waypoint 12
Position and orientation parameter adjustment 14
Change the speed and acceleration of the movement 16
Select the type of move 17
Insert MoveJ 19
Insert MoveL 21
Insert Arc Move 23
Insert Cir Move 25
Insert Arc move with orientation 27
Insert cir move with orientation 30
Insert B-spline curve 32
Insert moveP 34
Arrive Ahead Configuration 35
Insert setting command 41
Keypad use 42
Insert wait command 44
Variable configuration 45
Insert a reminder 48
Insert the If...else command 49
Insert loop command 50
Insert switch/case command 51
Smooth transition between multiple waypoints 52
Insert multithreaded command 54
Insert GOTO command 55

Insert timing command 57
Insert Offline Record condition 60
Script file configuration 61
Insert line comment and block comment commands..... 63
Create subproject commands 64
Record Track..... 65
Insert track record command..... 66
Tool calibration..... 67
Set the plane coordinate system..... 69
Add relative offset..... 70
Tool IO 72
Automatically run the default project..... 74

Find the tool center point

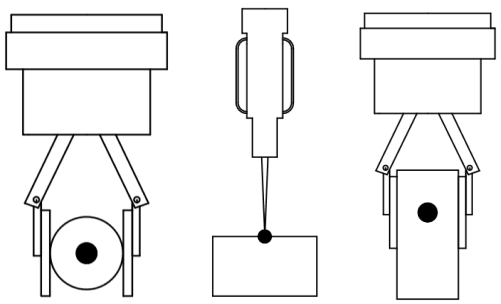
Purpose

The tool center point must be defined on the teach pendant to ensure optimal performance of the robot and prevent protective stops.

Definition

The tool center point is the part of the tool that is in contact with the workpiece. For example, the center of the fixture, the end of the soldering tool. You can also use the software to move the tool from 4 different angles to a fixed reference point to calculate TCP, creating 4 different locations with the same TCP (see Tool Calibration for specific operations).

Example: You can locate a tool center point (TCP) by finding the point of contact between the tool and the workpiece. TCP is at the center between the touch points.



Steps

Step	Operations
1	Click the “Settings” button In the teach pendant interface.
2	Click on the tool calibration under the robot.
3	Click kinematics cal the kinematic parameters below
4	Determine XYZ according to the flange_center coordinate system shown in the figure below 
5	Enter the kinematic name and tool XYZ value (position parameter, offset of XYZ joint), or RXRYRZ value (pose parameter, angle of rotation around the XYZ joint)

Set the center of gravity and payload

Purpose

The center of gravity of the attached tool must be defined in the teach pendant interface to calculate the applied force and avoid a protective stop. The robot needs to determine the payload in order to calculate how much force should be used and avoid a protective stop.

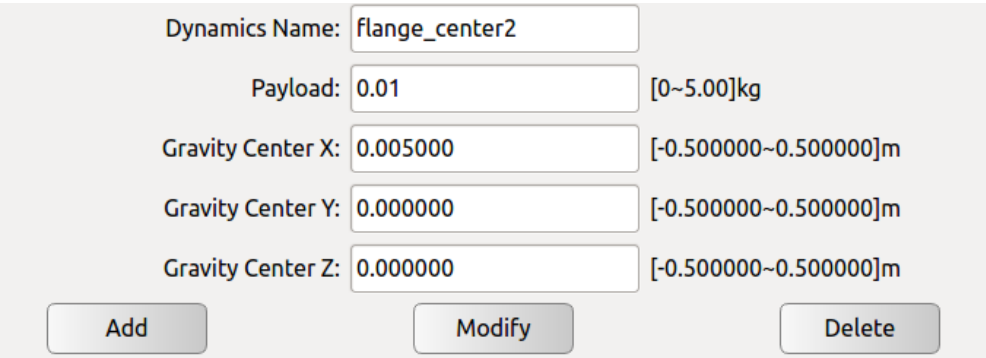
Definition

The center of gravity is the point on the tool, and the weight is evenly distributed on each side (point).

The payload is the weight of the additional tool (+ the weight of the workpiece).

Step

Please follow the steps below to set the center of gravity and payload.

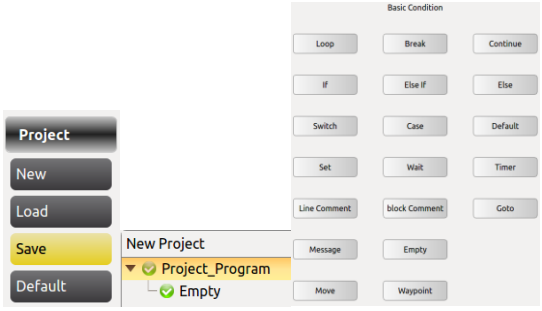
Step	Operations
1	Teach the device interface, respectively, click [Settings] - [Robot] - [Tool Calibration] to enter the tool calibration interface.
2	Click [Dynamic Calibration] to enter the dynamic calibration interface.
3	Enter the value of the dynamic name, payload, and center of gravity. 
4	Click the Add button.

New Project

Purpose

Before you edit the command, you need to create a new project.

Step

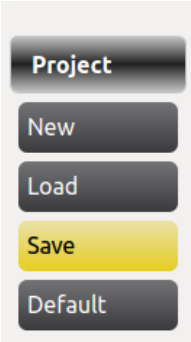
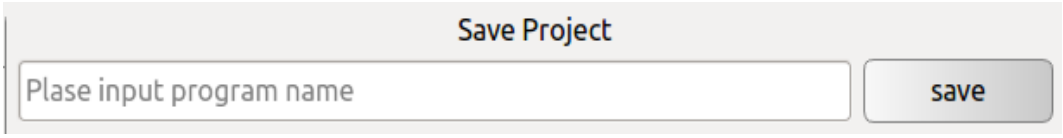
Step	Operations
1	Click the Project button in the Online Programming tab.
2	Click the New button under the Project button.  The new project file is displayed in the program logic and the base conditions are displayed on the right side of the interface.

Save Project

Purpose

The project needs to be saved after the project is written.

Step

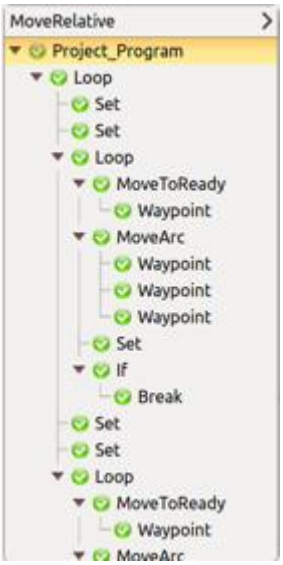
Step	Operations
1	Click the save button under the project button 
2	Enter the project name in the window that appears 
3	Click Save.

Load project

Purpose

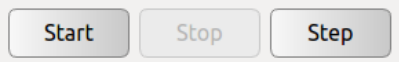
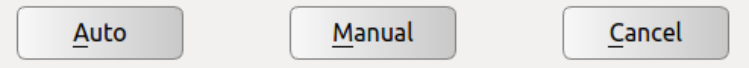
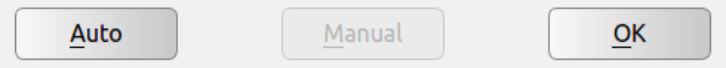
The saved project file will be displayed in the loading project interface, and the project file can be displayed in the project logic by loading.

Step

Step	Operations
1	Click the load button under the project button
2	Select the file name, click Load to load the previously saved project file, and the project content is displayed in the project logic. 

Run Project

Step

Step	Operations
1	Save project file or open project file
2	Click start at the bottom of the interface. 
3	Long press the auto button until the cancel button is displayed as ok  
4	Click the start button again to get the project running.

Note: For unconfirmed project files, you need to select the simulation mode in the working mode in the teach pendant interface to simulate the program file to prevent the expected movement track of the robot arm due to programming errors.

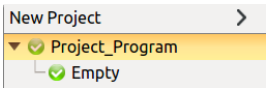
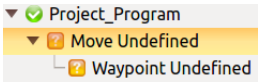
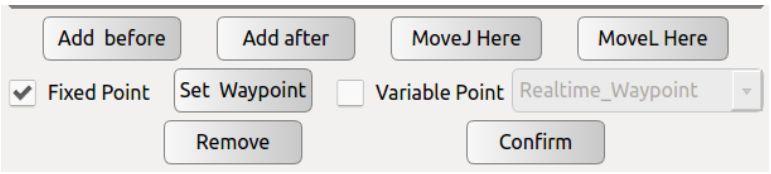
Insert action and waypoint

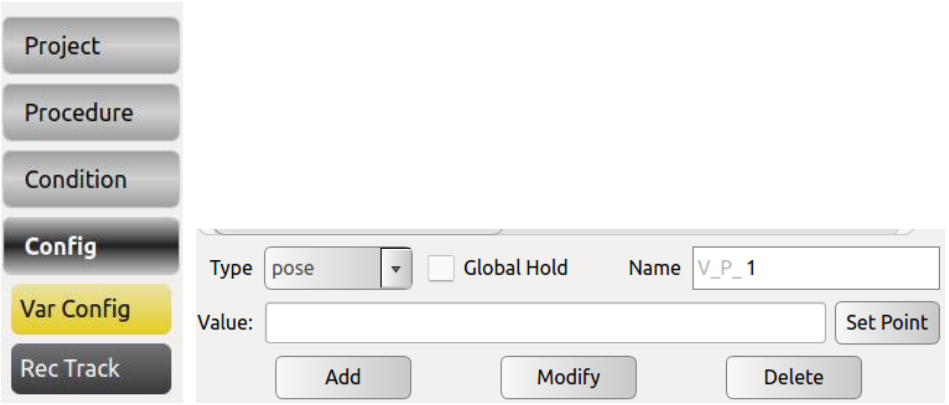
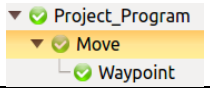
Purpose

In general, robot programs consist of waypoints and movements between waypoints.
Definition: The waypoint specifies the (next) position to which the robot moves, and the type of move determines the path or trajectory that the robot moves as it moves to the waypoint.

Step

Follow the steps below to insert actions and waypoints：

Step	Operations
1	In the teach pendant interface, click [Online Programming] - [Project] - [New] button to create a new project file (or load an existing project), and the corresponding program logic appears on the left side. 
2	Click the [Condition] - [Basic Conditions] button, click the move condition on the right, and the program logic inserts move and waypoint. 
3	Click on waypoint to set this waypoint information on the right side.  Click Add to before/after to add a waypoint before/after this waypoint Click MoveJ to get here, you can reach this waypoint with the fastest speed through each joint. Clicking MoveL here will move the end to a straight path and move to this waypoint. Select a fixed point and click to set the waypoint. You can set the position and posture information of this waypoint through the teaching interface. Select the variable point. After confirming, this waypoint is the waypoint set in the variable setting. When the waypoint in the variable is changed, all the waypoints in the project file will be changed. This function can change the parameters of the same waypoint in batches. , saving programming time. The variab

<table>
<tr><td></td><td> 
</td></tr>
<tr><td>4</td><td>Click Move Undefined, set the move condition, click the OK button, and click OK in the pop-up dialog box. Click Waypoint Undefined, click the Set Waypoint button, set the waypoint, click the confirm button, return to the waypoint condition interface, click the confirm button, and click OK in the pop-up dialog box.
 
</td></tr>
</table>

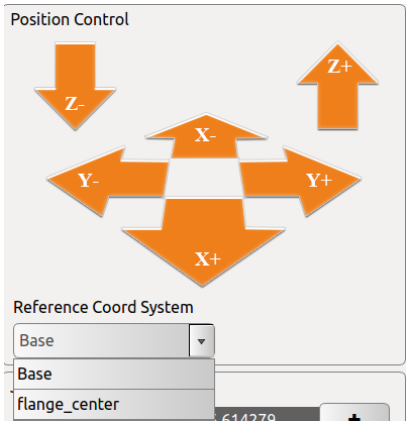
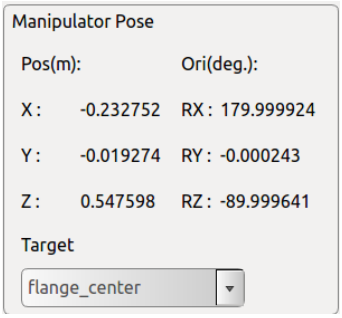
Position and orientation parameter adjustment

Purpose

Pose control is used to fine tune the position of the robot on the waypoint, for example to ensure that the fixture is picked up perpendicular to the workpiece.

Definition: When configuring a waypoint, you can use the pose control to configure the position and orientation of the robot arm very precisely.

Step

Step	Operations
1	In the project logic, click the waypoint to be edited, click the Set Waypoint button to enter the robotic arm teaching interface.
2	Select the corresponding reference coordinate system in position control and click the corresponding button to control the position of the arm. 
3	Select the corresponding target in the orientation control and click the corresponding button to control the orientation of the robot. 
4	In joint control, click on the corresponding button to control the joint of the robot.
5	After adjusting the posture of the robot, click the confirm button.

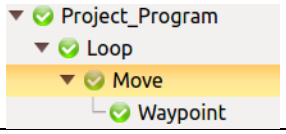
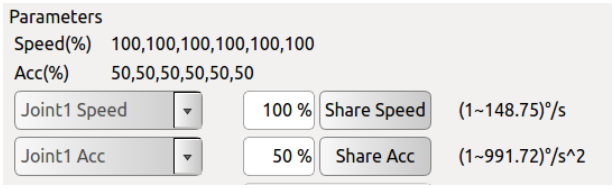
Change the speed and acceleration of the movement

Purpose

The speed and acceleration of the movement can be adjusted individually to move slower or faster in the relevant position in the robot program.

Step

Follow the steps below to change the speed and acceleration of the movement.

Step	Operations
1	In the project logic, click "move" to change the speed and acceleration. 
2	If the move type is joint, after inputting the speed and acceleration parameters, click the sharing speed and acc buttons. Click the OK button and click OK in the pop-up dialog box. 
3	If the move type is line, move track, enter the speed and acceleration parameters, click the confirmation button, and click OK in the pop-up dialog box. 

Select the type of move

Purpose

When the robot moves from one waypoint to another, the operator needs to select the type of move .

Definition

The move type determines the path or trajectory that the robot passes as it moves between waypoints. Different move types can be set for relative offset, speed and acceleration parameters, as well as advance in place. Please refer to the relevant chapter for specific use.

Type of move	Characteristics	Use
MoveJ	The fastest way to exercise Nonlinear move	The waypoints move quickly, regardless of the path of TCP movement between these waypoints. Move in the fastest way in a space-sufficient environment.
MoveL	The move mode belongs to the Cartesian space trajectory planning and needs to be solved by inverse kinematics. Therefore, there may be cases of no solution, multiple solutions, and approximation solutions; and due to the nonlinear relationship between joint space and Cartesian space, the axial move may exceed its maximum speed and acceleration limit.	Tool moves linearly between waypoints When the path to TCP is important In confined space
Move Track	The trajectory movement of multiple waypoints, the corresponding joint space or Cartesian space running speed and acceleration are continuous during the running, and the starting and ending waypoint speed is zero. When writing a trajectory move, at least three waypoints are required for each Move condition (there is no theoretical limit). When programming the trajectory and linear move of the robot arm, make sure that the adjacent	Arc move arc Circular move cir Circular move arcwithorirot with orientation(manipulator 6 joints need to support ±360° rotation) Circular move cirwithorirot with orientation(manipulator 6 joints need to support ±360° rotation) Circular arc smooth transition moveP B-spline curve B_spline

	waypoints of the two Move commands are continuous, that is, the last waypoint of the previous Move command is the same as the first waypoint of the next Move command. It is worth noting that when the robot arm makes a circular move, the last waypoint of the Move command is actually the first waypoint (the first and last waypoints coincide). When there is a loop loop command in the program logic list, the first waypoint of the first Move command should be kept the same as the last waypoint of the last Move command.	

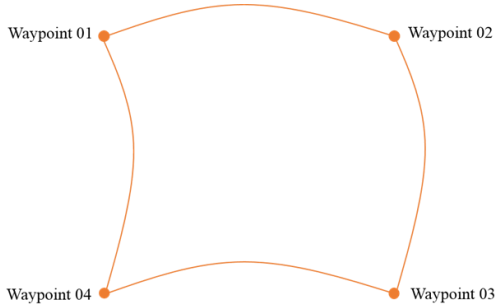
Insert MoveJ

Purpose

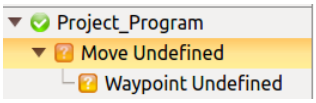
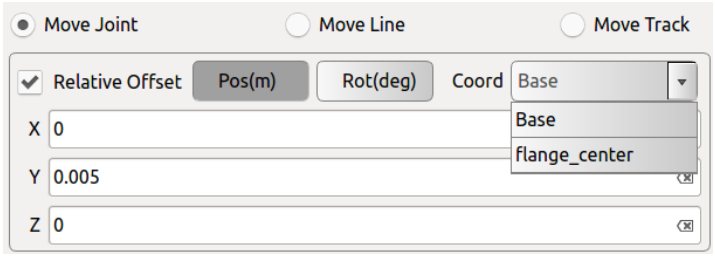
It is possible to move the robot arm quickly between waypoints without regard to the movement path of TCP between these waypoints.

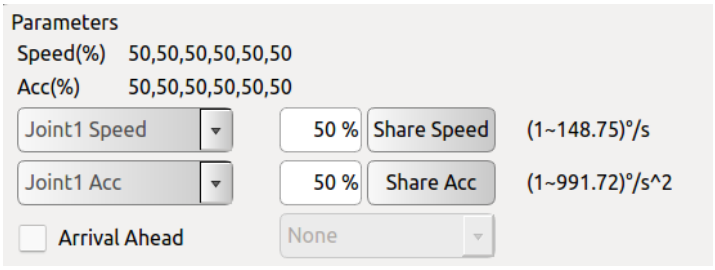
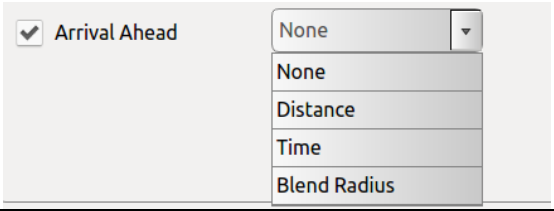
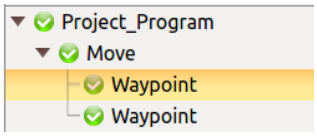
Example

Joint move is suitable for moving in the fastest way in a space-sufficient environment



Step

Step	Operations
1	In the teach pendant interface, click [Online Programming] - [Project] - [New], create a new project, click [Condition] - [Basic Conditions], in the basic condition interface, click the move button, add a move condition in the project tree. 
2	Click Move Undefined in the project tree to enter the move condition setting interface, select the move joint, select the relative displacement as needed, select the corresponding coordinate system, and enter the relative displacement value. 
3	In the parameter box, you can set the speed and acceleration of a joint. Click the sharing speed and acceleration button to make the joint speed and acceleration the same.

<table>
<tr><td></td><td> 
</td></tr>
<tr><td>4</td><td>Click Arrival Ahead, and the distance, time, and blending radius can be set in advance in the drop-down menu on the right. 
</td></tr>
<tr><td>5</td><td>Click Waypoint Undefined in the project tree to enter the waypoint setting interface, select the fixed point, click the set waypoint button, set the waypoint location, click the add to button, add the waypoint. 
</td></tr>
</table>

Insert MoveL

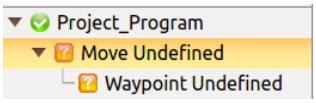
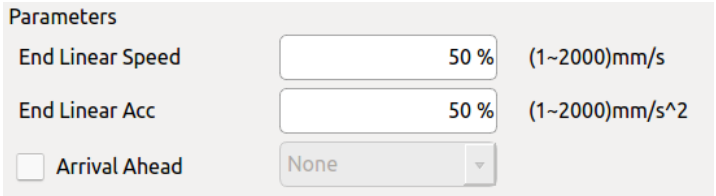
Purpose

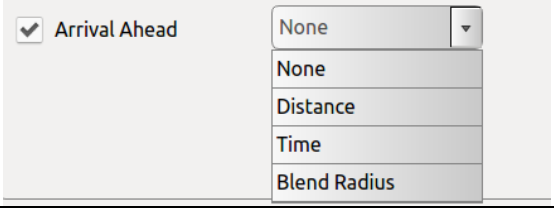
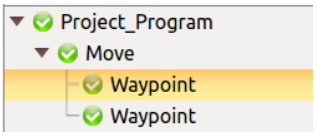
The tool can be moved linearly between waypoints.

Example

Each joint performs a more complex movement to keep the tool in a straight path.

Step

Step	Operations
1	In the teach pendant interface, click [Online Programming] - [Project] - [New], create a new project, click [Condition] - [Basic Conditions], in the basic condition interface, click the move button, add a move condition in the project tree. 
2	Click Move Undefined in the project tree to enter the move condition setting interface, select the joint move, select the relative displacement, select the corresponding coordinate system, and enter the relative displacement value. 
3	The end linear speed and acceleration can be set in the parameter box. 

4	Click Arrival Ahead, and the distance, time, and blending radius can be set in advance in the drop-down menu on the right. 
5	Click Waypoint Undefined in the project tree to enter the waypoint setting interface, select the fixed point, click the set waypoint button, set the waypoint location, click the add to button, add the waypoint. 

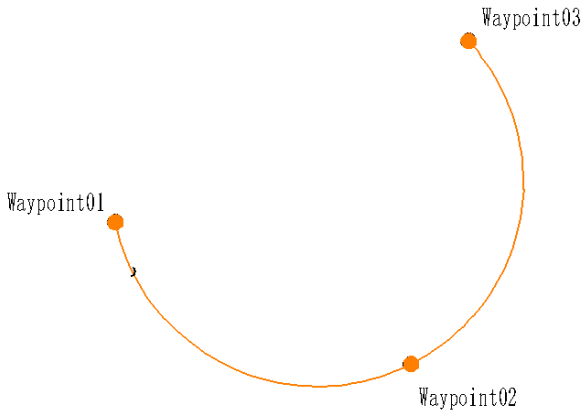
Insert Arc Move

Purpose

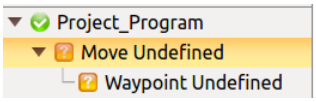
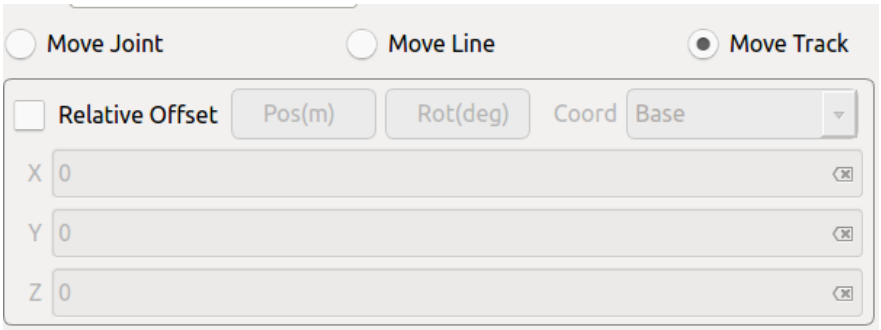
You can move the tool along the arc between the waypoints.

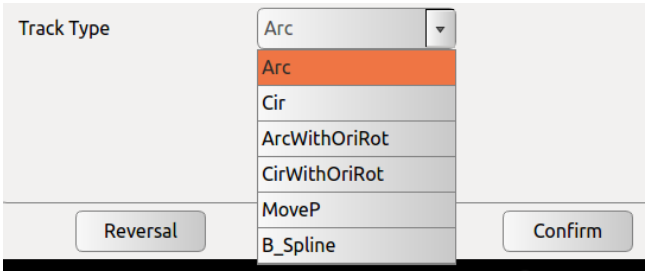
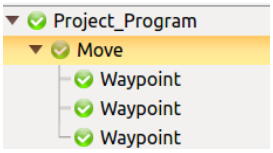
Example

The three-point method determines the arc and moves from the starting waypoint to the ending waypoint in order, which belongs to the Cartesian space trajectory planning.



Step

Step	Operations
1	In the teach pendant interface, click [Online Programming] - [Project] - [New], create a new project, click [Condition] - [Basic Conditions], in the basic condition interface, click the move button, add a move condition in the project tree. 
2	Click Move Undefined in the project tree to enter the move condition setting interface, select the joint move, select the relative displacement, select the corresponding coordinate system, and enter the relative displacement value. 
3	The end linear speed and acceleration can be set in the parameter box.

<table>
<tr><td></td><td> 
</td></tr>
<tr><td>4</td><td>In the track type drop-down menu, select Arc
 
</td></tr>
<tr><td>5</td><td>Click Waypoint Undefined in the project tree to enter the waypoint setting interface, select the fixed point, click the set waypoint button, set the waypoint location, click the add to button, add the waypoint.
 
</td></tr>
</table>

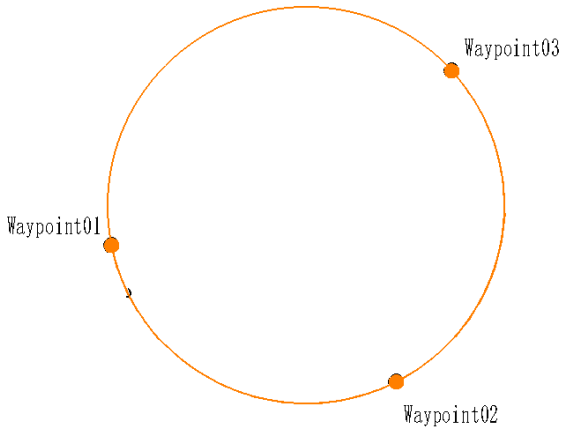
Insert Cir Move

Purpose

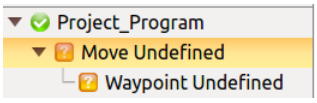
It allows the tool to make a circular move between waypoints.

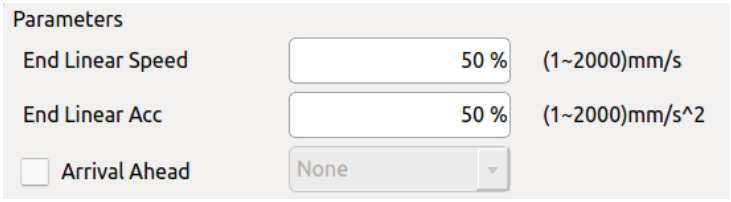
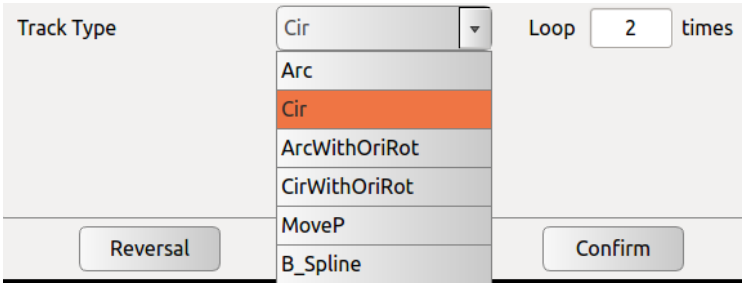
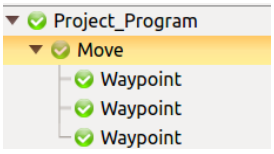
Example

The three-point method determines the full circle trajectory and the direction of move, and returns to the starting point after completing the entire circular move.



Step：

Step	Operations
1	In the teach pendant interface, click [Online Programming] - [Project] - [New], create a new project, click [Condition] - [Basic Conditions], in the basic condition interface, click the move button, add a move condition in the project tree. 
2	Click Move Undefined in the project tree to enter the move condition setting interface, select the joint move, select the relative displacement, select the corresponding coordinate system, and enter the relative displacement value. 

3	The end linear speed and acceleration can be set in the parameter box. 
4	In the Track Type drop-down menu, select Cir and enter the number of loops. 
5	Click Waypoint Undefined in the project tree to enter the waypoint setting interface, select the fixed point, click the set waypoint button, set the waypoint location, click the add to button, add the waypoint. 

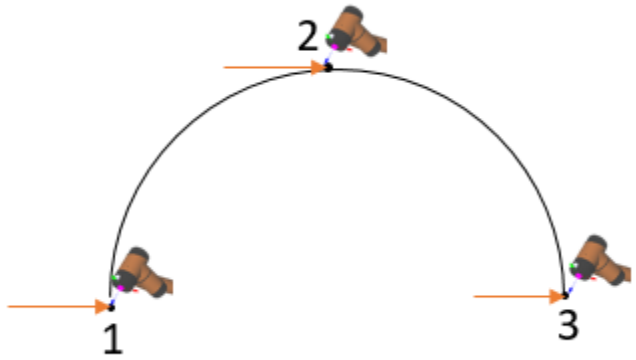
Insert Arc move with orientation

Purpose

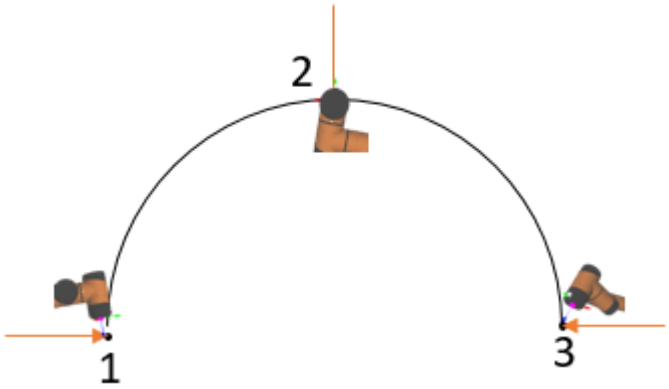
You can move the tool (different poses) along the arc between the waypoints.

This type of move is required when the tool end needs to maintain a fixed angle with the reference plane for circular move.

When there is no posture, the move track is as shown in the figure below, and the tool end orientation is based on the first waypoint set.

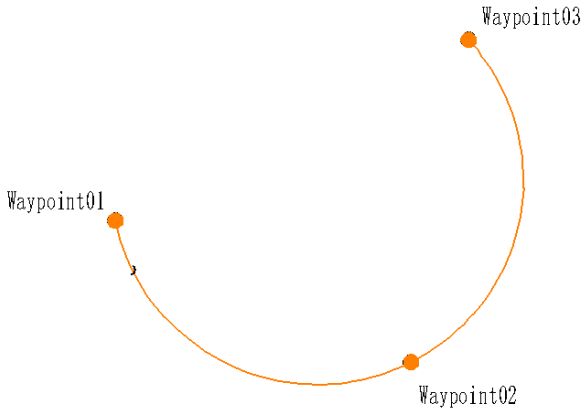


When the orientation is taken, the movement path of the tool end is as shown in the figure below, and the end of the tool moves with the set orientation of each waypoint.

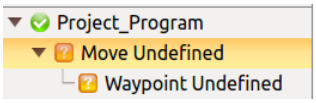


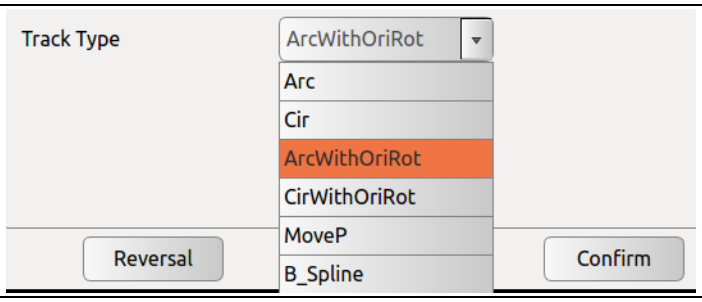
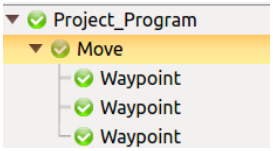
Example

The three-point method determines the arc and moves from the starting waypoint to the ending waypoint in order, which belongs to the Cartesian space trajectory planning.



Step

Step	Operations
1	In the teach pendant interface, click [Online Programming] - [Project] - [New], create a new project, click [Condition] - [Basic Conditions], in the basic condition interface, click the move button, add a move condition in the project tree. 
2	Click Move Undefined in the project tree to enter the move condition setting interface, select the joint move, select the relative displacement, select the corresponding coordinate system, and enter the relative displacement value. 
3	The end linear speed and acceleration can be set in the parameter box. 
4	In the Track Type drop-down menu, select ArcWithOriRot

	
5	Click Waypoint Undefined in the project tree to enter the waypoint setting interface, select the fixed point, click the set waypoint button, set the waypoint pose, click the add to button, add the waypoint. 

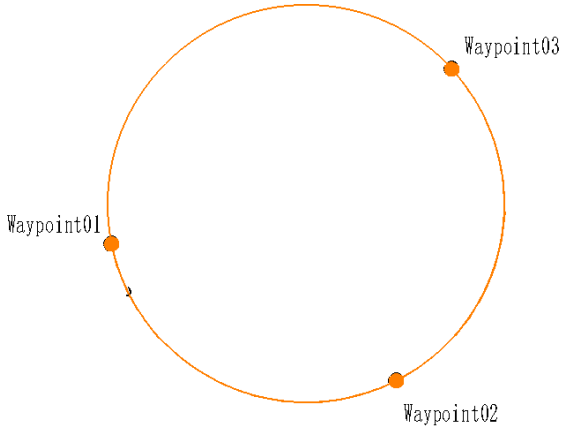
Insert cir move with orientation

Purpose

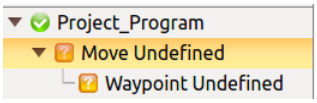
You can make the tool (different orientation) do circular move between waypoints.

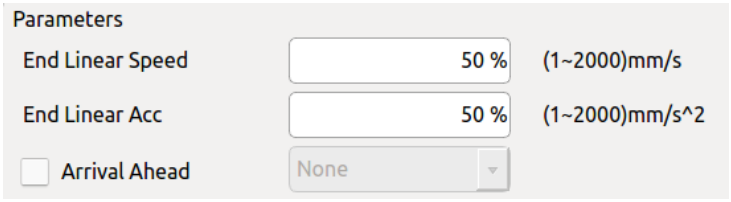
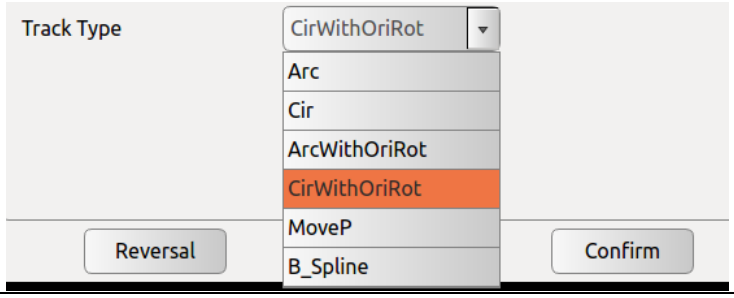
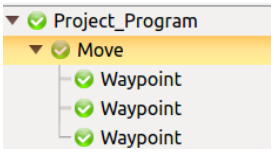
Example

The three-point method determines the full circle trajectory and the direction of move, and returns to the starting point after completing the entire circular move.



Step

Step	Operations
1	In the teach pendant interface, click [Online Programming] - [Project] - [New], create a new project, click [Condition] - [Basic Conditions], in the basic condition interface, click the move button, add a move condition in the project tree. 
2	Click Move Undefined in the project tree to enter the move condition setting interface, select the joint move, select the relative displacement as needed, select the corresponding coordinate system, and enter the relative displacement value. 

3	The end linear speed and acceleration can be set in the parameter box. 
4	In the Track Type drop-down menu, select CirWithOriRot. 
5	Click Waypoint Undefined in the project tree to enter the waypoint setting interface, select the fixed point, click the set waypoint button, set the waypoint location, click the add to button, add the waypoint. 

Insert B-spline curve

Definition

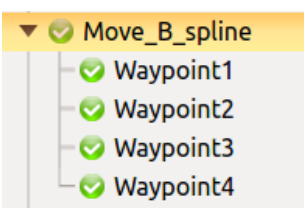
A path curve is fitted according to a given path point. The more waypoints are used to generate the fitted curve, the closer the fitted curve is to the expected。

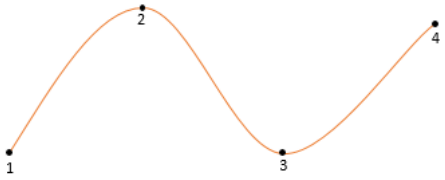
Description

The B-spline curve is a curve that smooth through all the given waypoints. Note that the beginning and end of the curve cannot be closed.

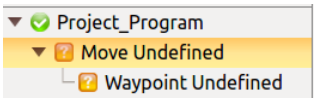
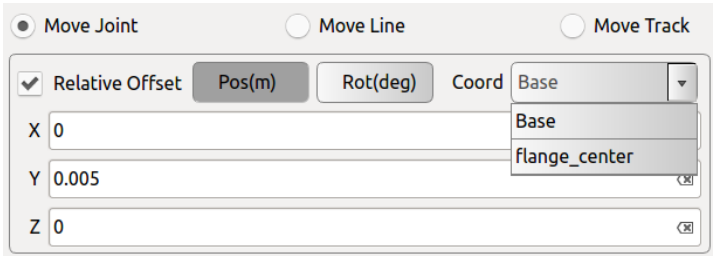
Example

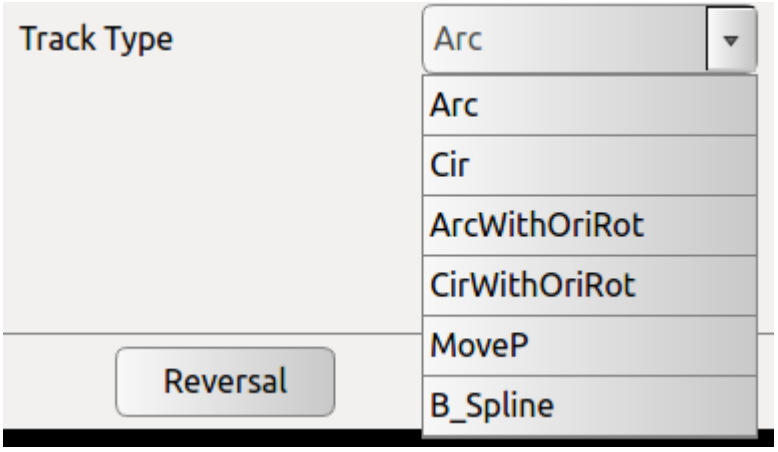
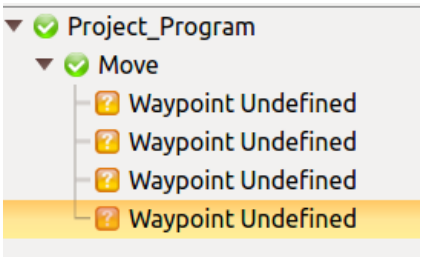
Insert a move and set the four waypoints as shown below (1, 2, 3, 4). After running the program, the B-spline curve runs as shown in the following figure.





Step

Step	Operations
1	In the teach pendant interface, click [Online Programming] - [Project] - [New], create a new project, click [Condition] - [Basic Conditions], in the basic condition interface, click the move button, add a move condition in the project tree. 
2	Click Move Undefined in the project tree to enter the move condition setting interface, select the joint move, select the relative displacement, select the corresponding coordinate system, and enter the relative displacement value. 
3	The end linear speed and acceleration can be set in the parameter box.

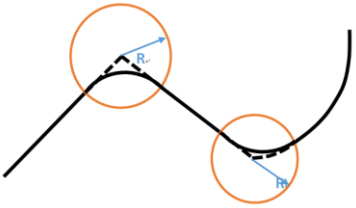
<table>
<tr><td></td><td> 
</td></tr>
<tr><td>4</td><td>In the Track Type drop-down menu, select B_spline.
 
</td></tr>
<tr><td>5</td><td>Click Waypoint Undefined in the project tree to enter the waypoint setting interface, select the fixed point, click the set waypoint button, set the waypoint location, click the add to button, add the waypoint.
 
</td></tr>
<tr><td>6</td><td>Click OK after setting</td></tr>
</table>

Insert moveP

Purpose

Smooth transition between two lines

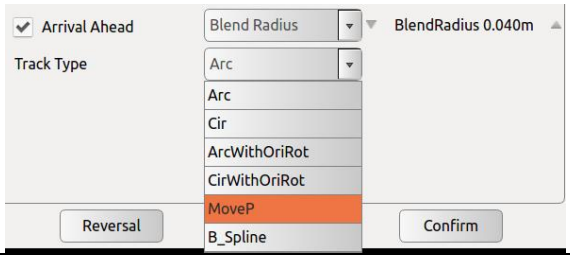
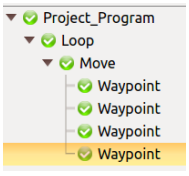
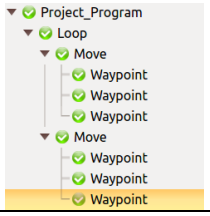
Description



The adjacent two straight lines are smoothly transitioned by the arc at the blending radius. The orientation change during the running is only affected by the beginning and ending points. The running characteristic of the blending radius is continuous move and will not stop at the waypoint. The operation mode is as shown.

Step

Please follow the steps below to set up movep

Step	Operations
1	Insert the move condition in the project logic, select the track for the move type, check the advance position, select the blend radius in the drop-down menu, select movep in the track type, and enter the blend radius value. 
2	Click the OK button and click OK in the pop-up dialog box.
3	Method 1: Implement with a move condition, add 4 waypoints in turn through the position control button, and make the first waypoint and the fourth waypoint the same waypoint. 
4	Method 2: Implemented with two move conditions, with the position control button, adding 3 waypoints for each move condition, and making the last waypoint of the first move the same as the first waypoint of the second move, and Make the last waypoint of the second move the same as the first waypoint of the first move. 

5	Click save on the left, enter the project name, and click save.

Arrive Ahead Configuration

Purpose

It is used to improve the working efficiency of the arm by selecting the distance from the target position, the distance from the target position, and the blending radius.

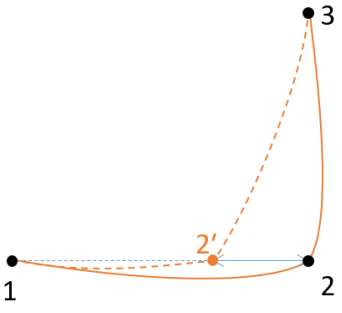
Definition

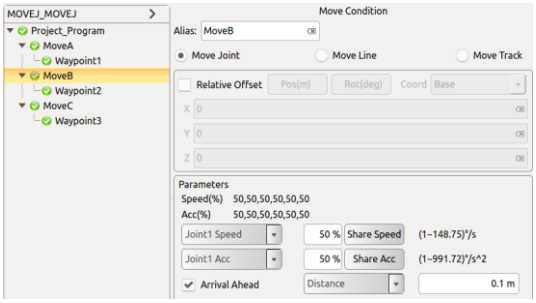
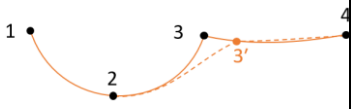
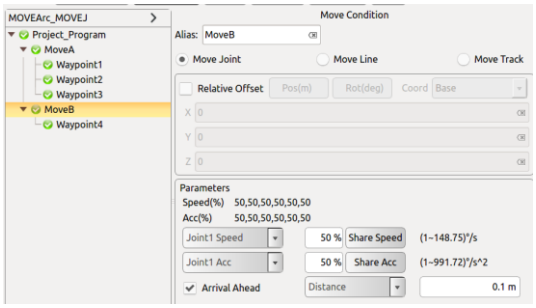
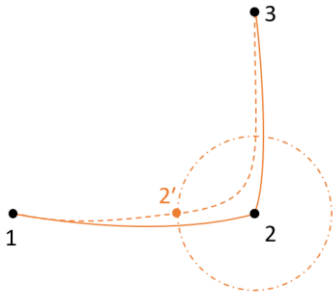
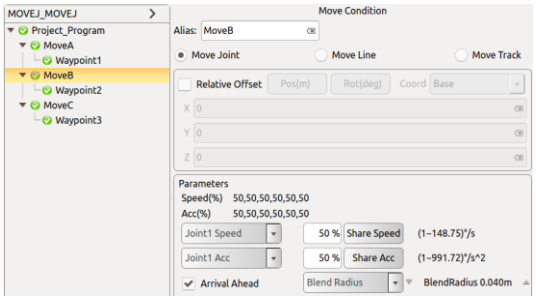
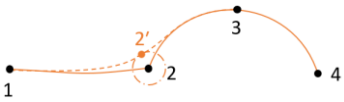
Arriving in advance will adjust the running track according to the distance or time set by the user and the blending radius, so as to improve the working efficiency of the arm, so there will be no one or more set waypoints.

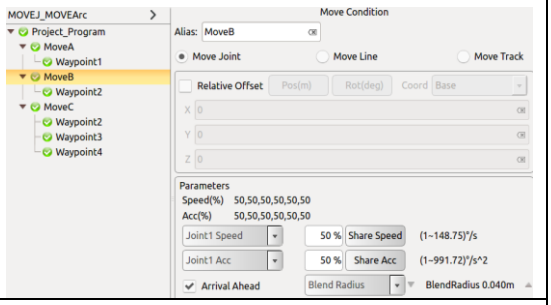
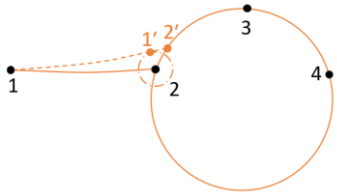
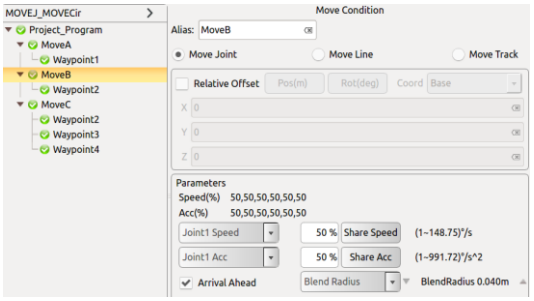
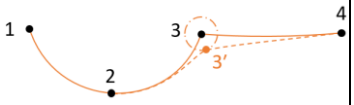
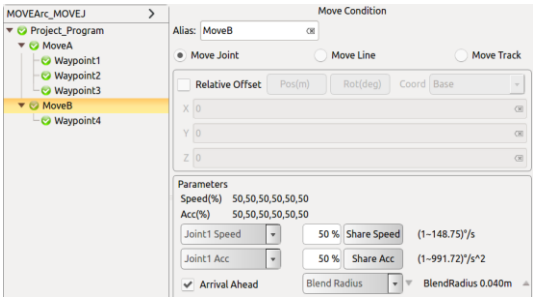
Type	Features	Scope of use
Distance	Check to get to this waypoint according to the set distance	Support Joint Move
Time	Check to get to this location early according to the set time	Support Joint Move
Blend Radius	Check to advance the position according to the set blend radius parameter	Supports joint move, linear move, arc move, circular move, arc move with orientation, circular move with orientation

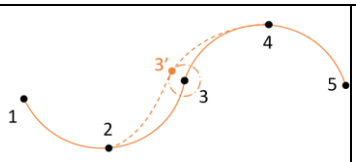
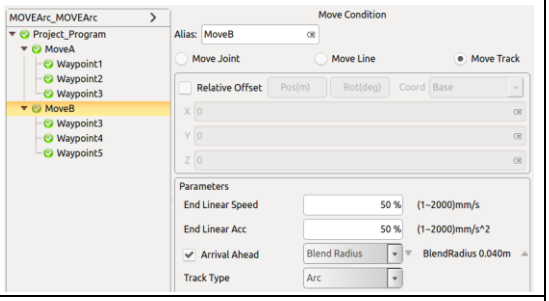
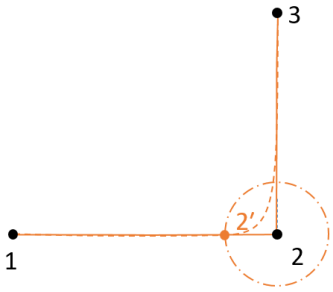
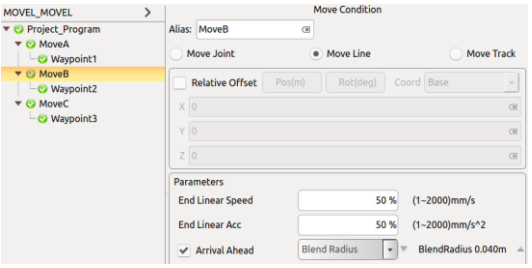
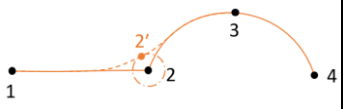
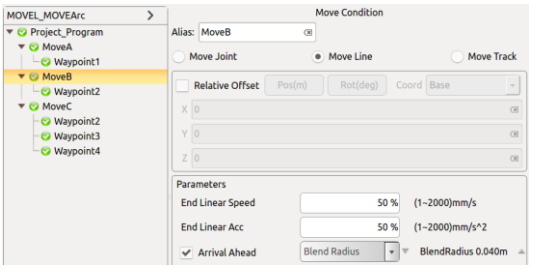
Note: The setting parameter is limited to the middle value of the two way points.
It is not recommended to add the first move in the first move in the project file.

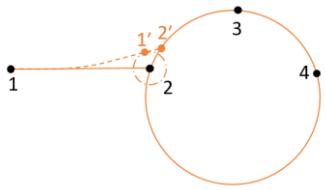
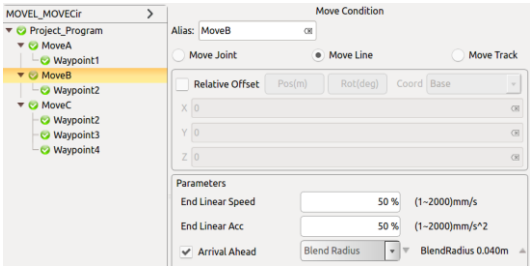
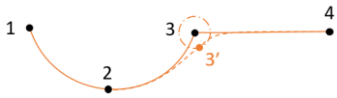
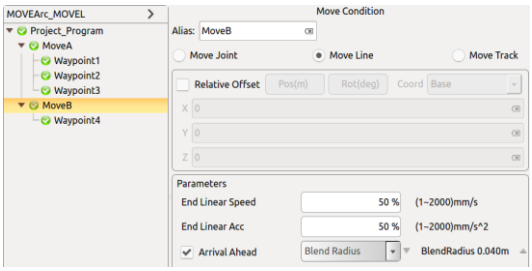
Example

Arrive Ahead	Type	Figure	Description
Distance/time	MoveJ-MoveJ		Insert three joint movements (moveA, moveB, moveC) and set three waypoints (1, 2, 3), Do not set in advance, the running track is 1-2-3 moveB check is placed in advance, after setting the distance or time, the running track is 1-2′-3

<table>
<tr><td></td><td></td><td></td><td> 
</td></tr>
<tr><td></td><td>Move arc-Move J</td><td> 
</td><td>Insert a circular move (moveA), set three waypoints (1, 2, 3), one joint move (moveB), set a waypoint (4)
Do not set in advance, the running track is 1-2-3-4
moveB check is placed in advance, after setting the distance or time, the running track is 1-2-3 ′ -4
 
</td></tr>
<tr><td>Blending radius</td><td>Move J-Move J</td><td> 
</td><td>Insert three joint movements (moveA, moveB, moveC) and set three waypoints (1, 2, 3),
Do not set in advance, the running track is 1-2-3
moveB check is placed in advance, after setting the distance or time, the running track is 1-2 ′ -3
 
</td></tr>
<tr><td></td><td>Move J-Move arc</td><td> 
</td><td>Insert two joint movements (moveA, moveB,) and set three waypoints (1, 2,),
Insert a circular move (moveC) and set three waypoints (2, 3, 4)
Do not set in advance, the running track is 1-2-</td></tr>
</table>

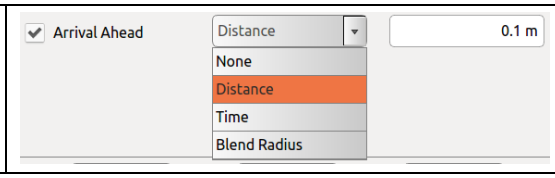
<table>
<tr><td></td><td></td><td></td><td>3-4
moveB check is placed in advance, after setting the distance or time, the running track is 1-2′-3-4
  </td></tr>
<tr><td></td><td>Move J-Move cir</td><td>  </td><td>Insert two joint movements (moveA, moveB,) and set three waypoints (1, 2,),
Insert a circular move (moveC) and set three waypoints (2, 3, 4)
Do not set in advance, the running track is 1-2-3-4
moveB check is placed in advance, after setting the distance or time, the running track is 1-1′-2′-3-4
  </td></tr>
<tr><td></td><td>Move arc-Move J</td><td>  </td><td>Insert a circular move (moveA), set three waypoints (1, 2, 3), one joint move (moveB), set a waypoint (4)
Do not set in advance, the running track is 1-2-3-4
moveB check is placed in advance, after setting the distance or time, the running track is 1-2-3′-4
  </td></tr>
</table>

	Move arc-Move arc		Insert two arc moves (moveA, moveB) and set three waypoints (1, 2, 3) (3, 4, 5) Do not set in advance, the running track is 1-2-3-4-5 moveB check is placed in advance, after setting the distance or time, the running track is 1-2-3 ′ -4-5 
	Move L-Move L		Insert three linear moves (moveA, moveB, moveC) and set three waypoints (1, 2, 3), Do not set in advance, the running track is 1-2-3 moveB check is placed in advance, after setting the distance or time, the running track is 1-2′-3 
	Move L-Move arc		Insert two linear moves (moveA, moveB,) and set three waypoints (1, 2,), Insert a circular move (moveC) and set three waypoints (2, 3, 4) Do not set in advance, the running track is 1-2-3-4 moveB check is placed in advance, after setting the distance or time, the running track is 1-2′-3-4 

<table>
<tr>
<td></td>
<td>Move L-Move cir</td>
<td> 
</td>
<td>Insert two linear moves (moveA, moveB,) and set three waypoints (1, 2,),
Insert a circular move (moveC) and set three waypoints (2, 3, 4)
Do not set in advance, the running track is 1-2-3-4
moveB check is placed in advance, after setting the distance or time, the running track is 1-1′-2 ′ -3-4
 
</td>
</tr>
<tr>
<td></td>
<td>Move arc-Move L</td>
<td> 
</td>
<td>Insert a circular move (moveA), set three waypoints (1, 2, 3), one linear move (moveB), set a waypoint (4)
Do not set in advance, the running track is 1-2-3-4
moveB check is placed in advance, after setting the distance or time, the running track is 1-2-3′ -4
 
</td>
</tr>
</table>

Steps

Step	Specific Operation
1	In the teach pendant interface, add a move condition to the program and enter the move condition setting interface. In the move condition, select the move type, check the advance arrival, select the distance or time or blending radius from the drop-down menu, enter the corresponding time value or distance value or blending radius value, click the confirmation button, and click OK in the pop-up dialog box.



Insert setting command

Purpose

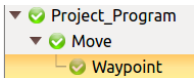
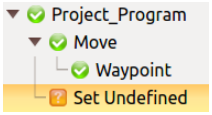
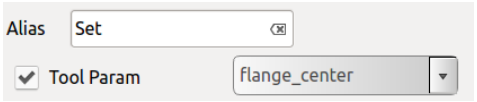
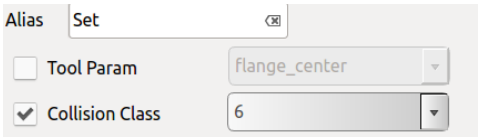
The set command can be used for tool parameters, collision levels, IO, and variable settings.

Definition

The set command sets a number, variable, or analog output to a specific value, sets a collision level, and tool parameters.

Step

Please follow the steps below to insert the set command

Step	Specific operation
1	In the project logic, click on the Waypoint to which you want to insert the set command (or at other conditions to insert the set condition). 
2	Click [Condition] - [Basic Conditions], select set click in the basic conditions, and insert the set condition in the project logic. 
3	Click set in the project logic to enter the set condition setting interface and enter the name. If you select a tool parameter, select the appropriate tool parameter. 
4	If you choose a collision level, select the appropriate collision level. 
5	If IO is selected, you can assign values to users, tools, PLCs, Modbus, and other digital and analog IOs. 
6	If the variable is checked, the variable can be assigned a value. 

7	Click the OK button and click OK in the pop-up dialog box.

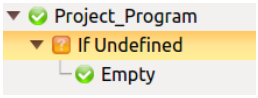
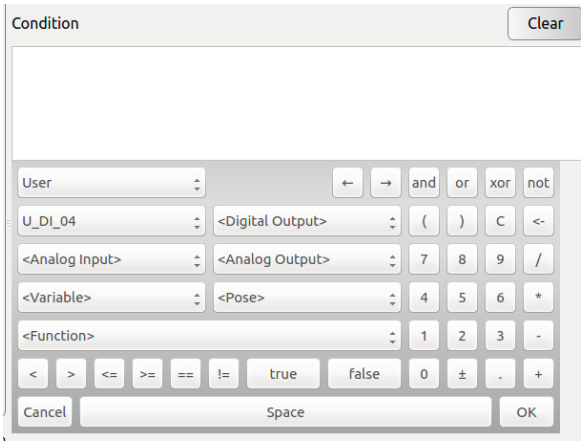
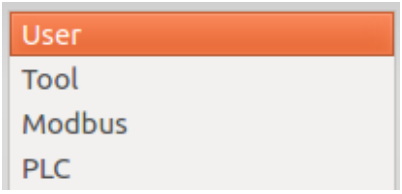
Keypad use

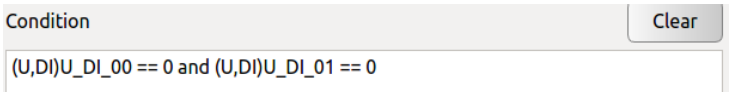
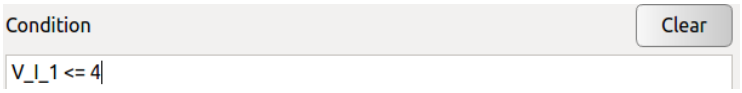
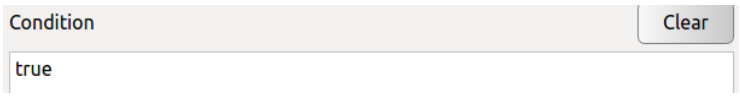
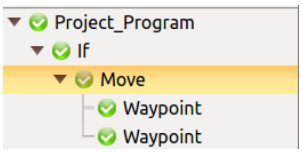
Purpose

The keypad is used for conditional input of certain conditions in the project tree, which is simple and convenient. Inputs of logical conditions such as if, loop, etc.

Steps

Follow the steps below to set the conditions.

Step	Specific operation
1	In the teach pendant interface, click [Online Programming] - [Project] - [New], click on Empty in the project tree, click [Condition] - [Basic Conditions], and click the if button in the basic condition interface. Click the If condition in the project tree to enter the condition setting interface. Click the blank box below the condition to display the keypad.  
2	Click on User on the keyboard to display Tool, Modbus, Plc, etc.  Now take the Tool as an example, click on the number on the keyboard, analog input and output, the corresponding number/analog will be displayed in the blank box, and the numeric key, logical operator and delete key in the keyboard will be assigned to the value.

<table>
<tr><td></td><td> 
</td></tr>
<tr><td>3</td><td>Click on the variable on the keyboard, the corresponding variable will be displayed in the blank box, and the numeric key in the keyboard, as well as the relational operator and the delete key, will be assigned to it.
 
</td></tr>
<tr><td>4</td><td>Click the logical value ture, false on the keyboard, the ture, false will be displayed in the blank box, click the confirm button.
 
</td></tr>
<tr><td>5</td><td>When the variable is used, the Arabic numeral indicates the value. In the signal input and output, 0 means invalid and 1 means valid.</td></tr>
<tr><td>6</td><td>Add a move condition to the project tree, save the project, and run the project.
 
</td></tr>
</table>

Insert wait command

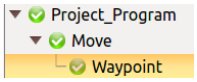
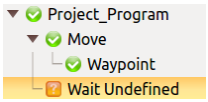
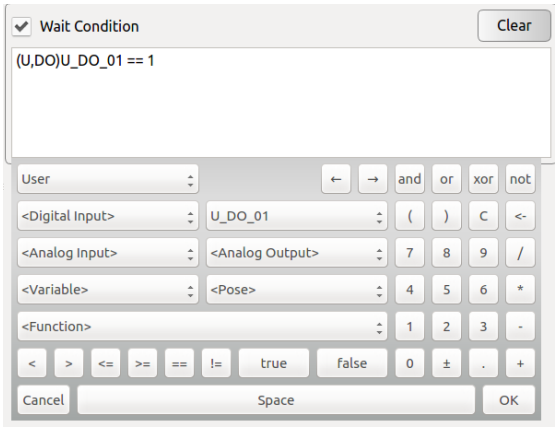
Purpose

A wait command can be inserted in the project logic to cause the robot to wait and respond to specific events.

Definition: The wait command causes the robot to wait for a specific time, to change a numeric or analog input to a specific value, or to change a variable to a specific value.

Step

Please follow the steps below to insert the wait command.

Step	Specific Operation
1	In the project logic, click on the Waypoint to which you want to insert the wait command (or at other conditions to insert the wait condition). 
2	Click [Condition] - [Basic Conditions], select wait in the basic conditions, and insert the wait condition in the project logic. 
3	Click wait in the project logic, enter the wait condition setting interface, and enter the nickname. If you select time, enter the time value. 
4	If you select a condition, enter numbers, simulations, and variable values. 
5	Click the OK button and click OK in the pop-up dialog box.

Variable configuration

Purpose

Variables are used by robotic applications to track related program information. E.g. Calculate the number of workpieces processed, or track the input of an external device.

Definition

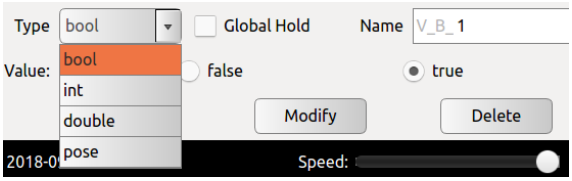
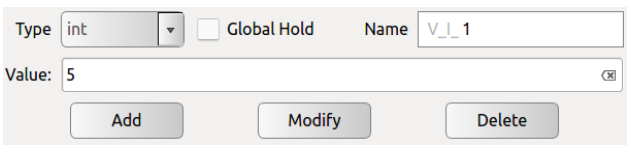
A variable is a "container" used to store information that can be referenced and modified in a robot program.

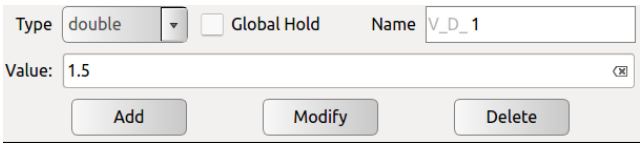
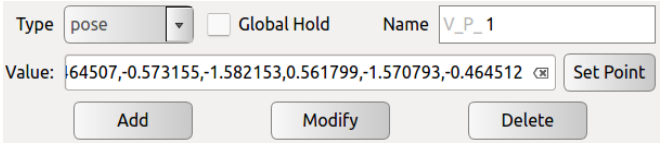
Description

Variable settings currently only support variables of type bool, int, double, and pose.
Bool: Define a bool type variable whose value is true/false. Click the variable value to assign the value.
Int: Defines an integer variable whose value is an integer and enters the value in the cell after the value of the variable.
Double: Defines a double variable whose value is a double-precision floating-point number. Enter the value in the cell after the variable value.
Pose: Define a position type variable whose value is the robot waypoint information. Click the [Set Waypoint] button behind the variable value cell to jump to the robot teaching interface. After completing the waypoint setting, click [OK] to complete the variable. Assignment.

Steps

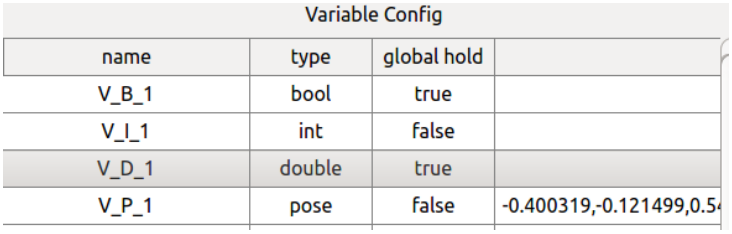
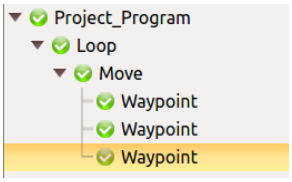
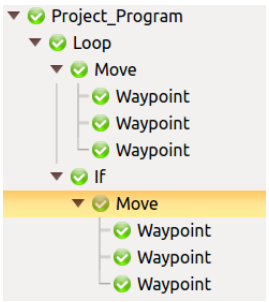
Follow the steps below to insert or initialize a variable.

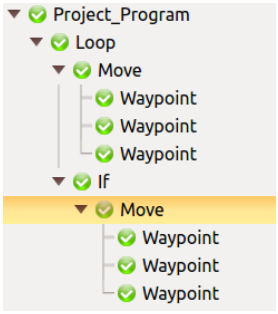
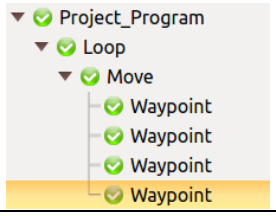
Step	Specific Operation
1	Click [Online Programming] - [Configuration] - [Variable Configuration] in the Teach Pendant interface to enter the variable configuration interface.
2	Bool variable configuration, select bool in the type drop-down menu, select global hold as needed, enter the variable name, check ture/false, click the add button. 
3	Int variable configuration, select int in the type drop-down menu, select global hold as needed, enter variable name, enter integer, click the add button. 
4	Double variable configuration, select double in the type drop-down menu, select global hold as needed, enter variable name, enter double precision floating point number, click

	the add button. 
5	Pose variable configuration, select pose in the type drop-down menu, select global hold as needed, enter the variable name, set the corresponding waypoint, click the add button.  Variable waypoints can be given by set variable waypoints

Examples

Follow the steps below to use the variables.

Step	Specific Operation
1	In the teach pendant interface, click [Online Programming] - [Configuration] - [Variable Configuration] to add 4 variables. Click [Status] - [Variable Status] to view the status of the variable. 
2	Click [Project] - [New] to create a new project file. 
3	For logical conditions: Click on the move in the program logic, click [Condition] - [Basic Conditions], select Add to the back, click the if button, click the if in the program logic, write V_D_1 <= 4 in the if condition, click Confirm button. In the program logic, under the if command, add a move instruction, save the project, and run the project. 

4	For constants: Click on the move in the program logic, click [Condition] - [Basic Conditions], select Add to the end, click the switch button, click the switch in the program logic, write V_I_1 in the switch condition, and click the OK button. In the program logic, under the default command, add a move command, save the project, and run the project. 
5	For waypoints: Click on the waypoint in the program logic, click [Condition] - [Basic Conditions], select Add to the back, click the Add after button, check the variable point, select V_P_1 in the drop-down tab, click the confirm button, save the project , running the project. 

Insert a reminder

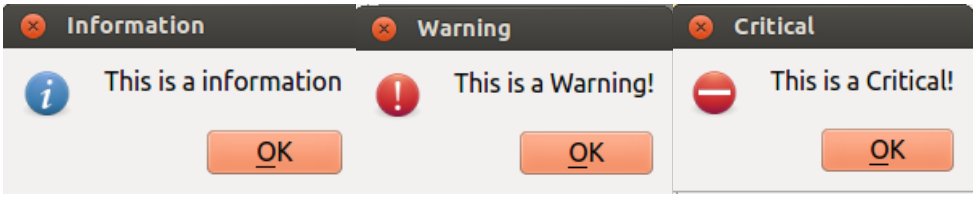
Purpose

A bullet box is a message that is displayed on the teach pendant when the program moves to a pop-up command in the program logic.

Principle

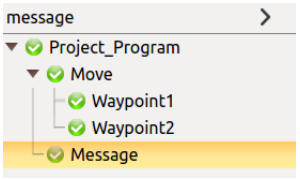
When a pop-up window is displayed, the robot waits for the operator to press the "OK" button under the pop-up window before continuing the program.

There are three types of pop-up windows, which can be selected according to different needs.

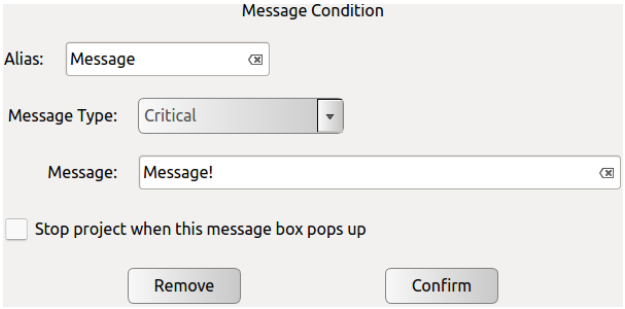


Example

Pop-up window after Waypoint2



Steps

Step	Specific Operation
1	Click on the message under the basic conditions
2	Click on the message logic at the program logic.
3	Enter the message condition interface 

Insert the If...else command

Purpose

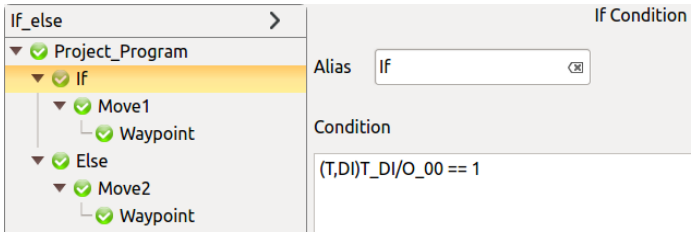
If else is used to determine whether the given condition is satisfied, and one of the two operations given is determined according to the result of the decision (true or false). For example, the value of the variable, the signal of the end tool.

Principle

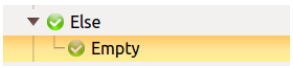
Under the If command, when the given condition is true, the subcommand under the If condition is executed. When the given condition is false, the subcommand under the else condition is executed.

Example

If (T, DI) T_DI/O_00 is true, the arm performs the action of Move1, otherwise, the action of Move2 is executed.



Steps

Step	Specific Operation
1	Click the condition button in the online programming tab
2	Click the If...else button under basic conditions Result: If command is displayed at the program logic
3	Click If at the program logic, and write down the condition that the subcommand to be executed must be true in the blank space under the condition.
4	Click the subroutine of < Empty > under If condition and enter the command to execute
5	Optional: To insert an else statement, click the If statement in the project logic. Click the If condition under Program Logic and click the Add Else button. Result: Else condition is inserted 
6	Click the <empty> subcommand under the else statement and enter the command to execute

Insert loop command

Purpose

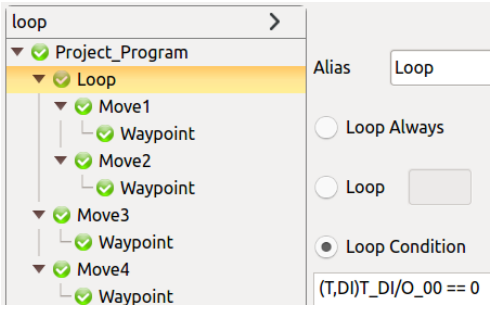
Loops can be used for program code that is repeated multiple times in succession.

Principle

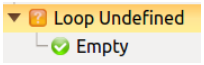
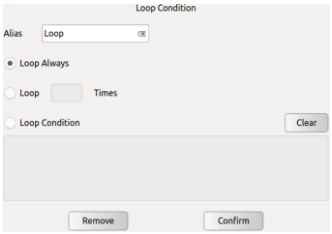
The program code that needs to be repeated is placed in the loop command. The loop command can be configured to repeat indefinitely, a specific number of times, or the expression is true (such as a variable or input signal)

Example

(T,DI)T_DI/O_00 == 0, run move1 and move2, in other cases, run move3 and move4



Steps

Step	Specific Operation
1	Click the condition button in the online programming tab
2	Click the loop button under basic conditions Result: If command is displayed at the program logic 
3	Click loop undefined
4	Configuring the loop condition 

Insert switch/case command

Purpose

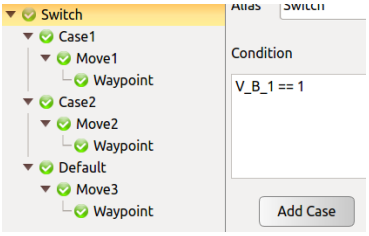
The switch / case command can be used to control the flow of a program based on the value of the variable.

Principle

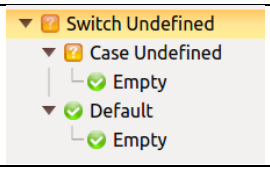
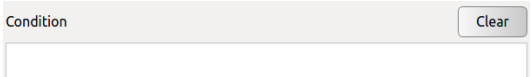
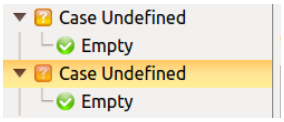
The Switch command is related to variables. You can use the Case command to define multiple possible actions based on the value of a variable. You can also insert the default. If the variable is not equal to any value specified in the Case command, the default will be executed.

Example

The Switch command checks the value of the variable V_B_1. If the value is 1, execute the command move1 under case1. If the value is 2, execute the command move2 under case2. If the value is neither 1 or 2, the command move3 under Default will be executed.



Steps

Step	Specific Operation
1	Click the condition button in the online programming tab
2	Click the switch button under basic conditions Result: the switch command is displayed at the program logic 
3	Click on the switch at the program logic
4	Variable assignment in the input box under conditions 
5	Click on the case of the program logic
6	Type the value of the variable for which you want to assign an action in the input box under Conditions, and then click add case.
7	You can insert as many Case commands as you need. 
8	Click the <empty> subcommand under the case undefined statement and enter the command to execute

Smooth transition between multiple waypoints

Purpose

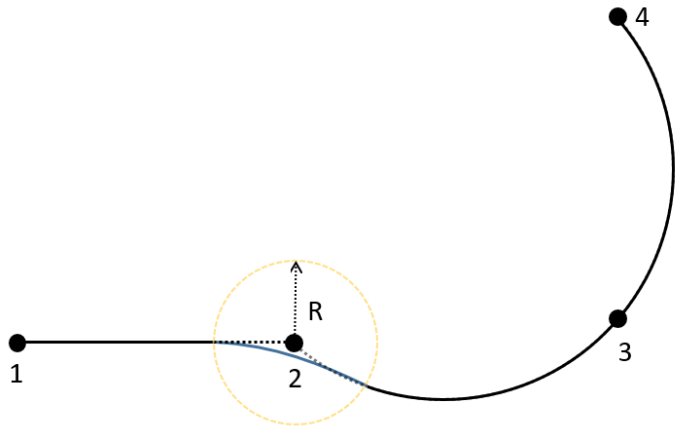
It is used to make the robot TCP perform the smooth transition of the waypoint, so that the robot TCP can quickly pass the waypoint, and the blending radius can be added to the waypoint to reduce the cycle time.

Principle

The blending radius moves the robot arm trajectory around the waypoint rather than passing through the waypoint, allowing the robotic arm to stop at that point. Thereby reducing cycle time

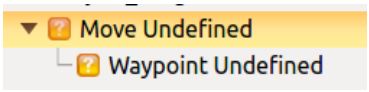
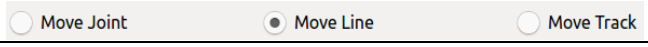
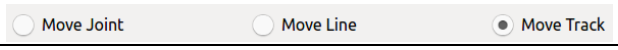
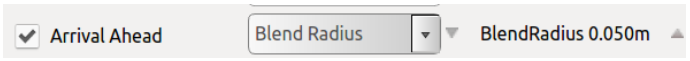
Example

MoveL is a straight line, MoveArc is a circular arc, the following example is a smooth transition of the robot TCP from the straight line to the arc without going through the waypoint 2.



Waypoints 1-4 correspond to waypoint1-4 in the project file

Steps

Step	Specific Operation
1	Click the condition button in the online programming tab
2	Click the move button under basic conditions
3	Result: the move command is displayed at the program logic 
4	Click move undefined
5	Select the line and click to confirm 
6	Set waypoint1 and waypoint2
7	Continue adding move
8	Click move undefined to select movetrack 
9	Select arc at track type 
10	Set the blend radius value, check the arrival ahead to select the blend radius, and adjust the blend radius value. 
11	Click to confirm.

Insert multithreaded command

Purpose

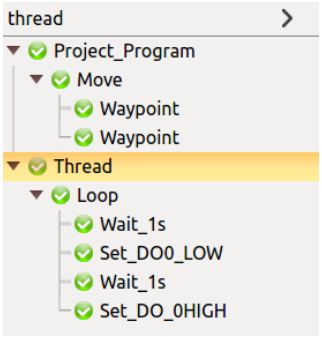
Thread is a process that is parallel to the robot program

Principle

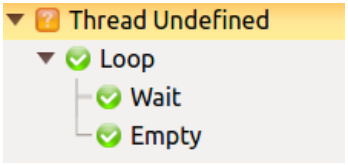
In the Thread block, there must be a Loop loop command in which parallel control with the main program can be implemented.

Example

During the move process, the DO_0 signal is set every 1s.



Steps

Step	Specific Operation
1	Click the condition button in the online programming tab
2	Click the thread button under advanced conditions
3	Result: the thread command is displayed at the program logic 
4	Click thread undefined to confirm the thread name
5	Determine the loop condition
6	Determine the wait condition
7	Create the required command under empty

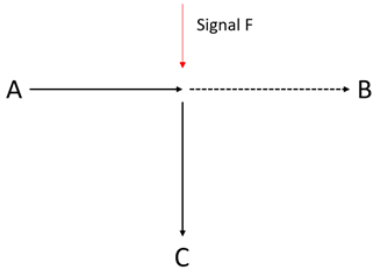
Insert GOTO command

Purpose

In some industrial scenarios, the robot needs to interrupt whatever it is doing and turn to other tasks.

Principle

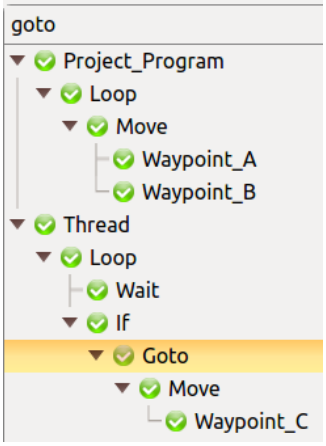
The robot actions are as follows:



The robot is programmed to move from A to B, but it receives a signal F on the way to B, it stops moving in direction B and immediately goes to C.

Example

The GOTO command is designed to interrupt the current move. It must be used in a threaded program, as an example, if the if condition is true, then the waypoint A-B move is jumped and run to waypoint C.



The "If" command is set to "DI == 0" or any other input signal. Please note that in order to ensure that GOTO works properly, it requires a "wait" command of at least 0.01s. The lack of it can lead to unpredictable problems and stop the robot.

Steps

Step	Specific Operation
1	Click the condition button in the online programming tab
2	Click the goto button under basic conditions

3	Result: the goto command is displayed at the program logic 
4	Click the empty add command

Insert timing command

Purpose

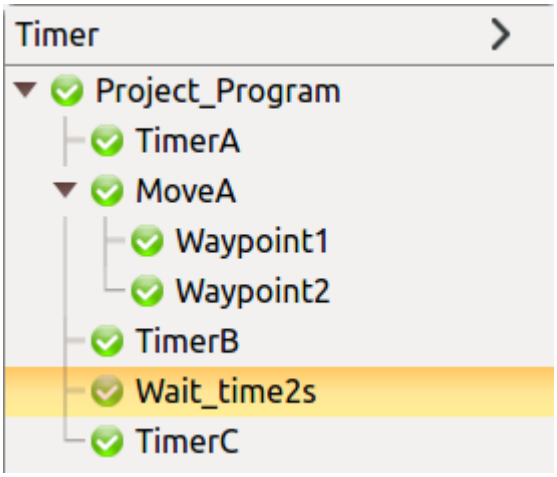
The timer records the time and number of times the project file starts to run to the node. It measures the time required for the arm to move.

Definition

Two nodes of the project file are inserted into the timer to calculate the running time between the two nodes.

Example

Create a new project file as shown below, moveA is joint move, wait command condition is set to 2s, note that the timer nickname cannot be repeated



Run the project file, in the programming-state-timer, you can display timerA, timeB: the time after running waypoint2, and the time after timerC runs wait-time2s.

Timer Condition State

Alias	Time	Times
TimerA	0.006s	1
TimerB	1.455s	1
TimerC	3.455s	1

Times indicates the number of times this Timer command was executed in the program file.

The interval between nodes can be calculated below the interface.

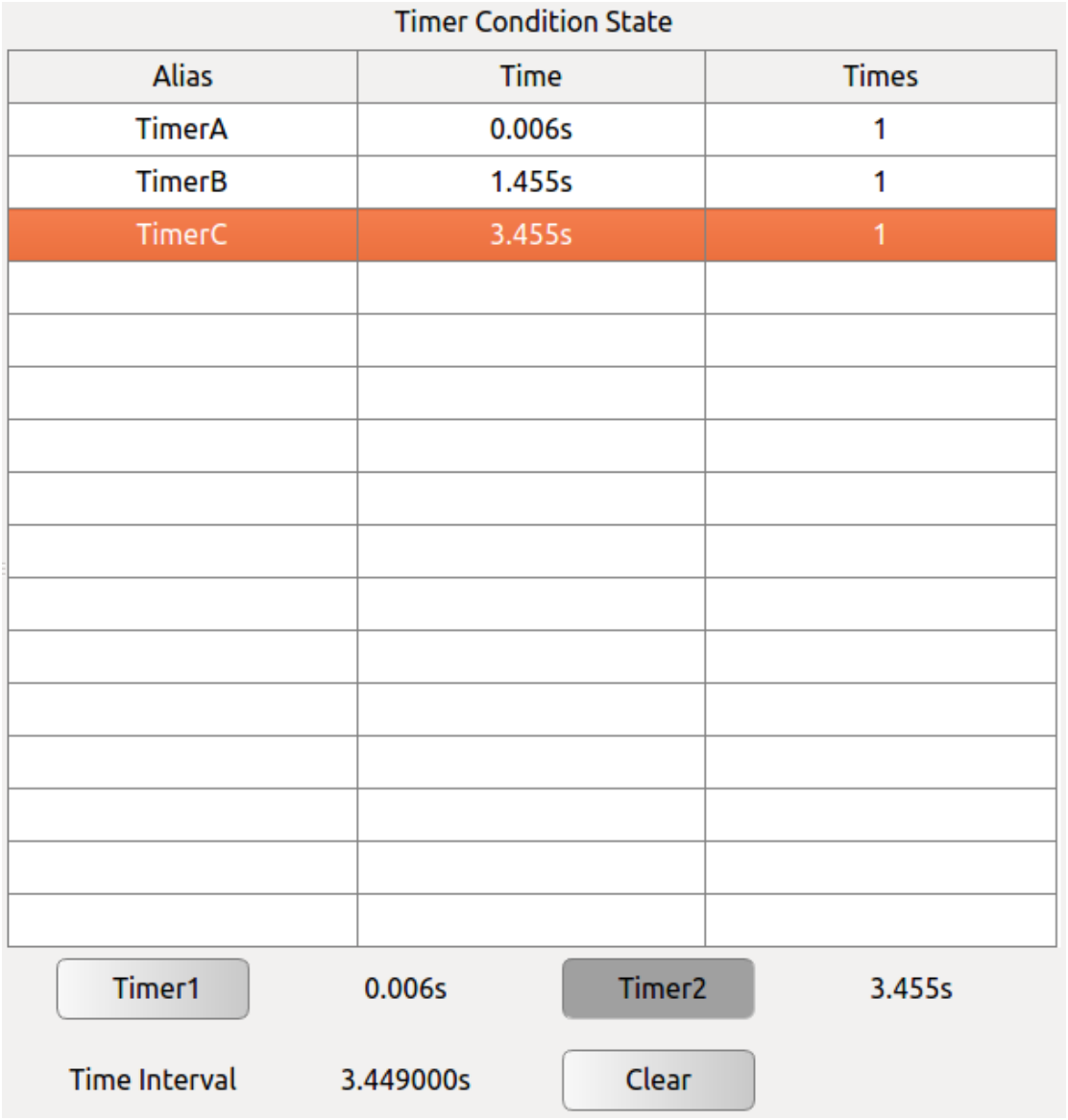
Timer1 and Timer2 are the corresponding time displayed when a Timer command is selected. Click the timer name in the list, the lower timer button will be darker, and the time corresponding to

the timer name in the list will be displayed on the right side. The time interval between time1 and timer2 will be calculated automatically at the time interval.

For example, if you first click timeA in the list, the corresponding time on the right side of timerA is displayed on the right side of the gray timer button.

Then click timeC in the list, then the corresponding time 3.455s on the right side of timerC is displayed on the right side of the gray timer button.

The Timer interval shows the time interval between timer1 and timer2, that is, the time between the first move of waypoint1 and the first time of wait_times2s.



Note that the display of timer1 and timer2 has a relationship with the order of the condition names in the click list, which is based on the gray color of the button, and has no relationship with the display order of the time name.

Steps

Please follow the steps below to set the timer

Step	Specific Operation
1	In the teach pendant interface, click [Online Programming] - [Project] - [New] to create

	a new project file.
2	Click the condition button, add a conditional command, and write a project file.
3	Run the project file and click [Status] - [Timer] to view the project running time and number of times.

Insert Offline Record condition

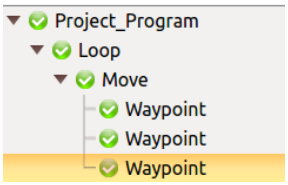
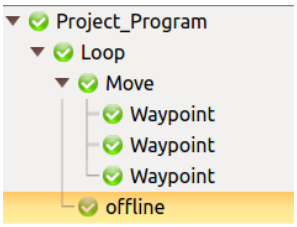
Purpose

The Offline Record command embeds the trace files generated by the offline programming software into online programming.

Definition: Import the trace file generated by the offline programming software into the offline track directory, which can be embedded into the online programming through the Offline Record condition.

Steps

Please follow the steps below to insert the Offline Record condition.

Step	Specific Operation
1	Import the trace file with the suffix ending in .offt in the Home\AuboRobotWorkSpace\teachpendant\share\teachpendant\offlinetrack directory.
2	In the teach pendant interface, click [Online Programming] - [Project] - [New] to create a new project file. 
3	Click [Condition] - [Advanced Condition], select Add, click move in the program logic, click the Offline Record button, click Offline Record in the program logic to enter the condition setting interface, select the corresponding offline file, click OK, pop up Click OK in the dialog box to save the project and run the project. 

Note:

The imported track file format must contain six joint angles per line in radians.

Script file configuration

Purpose

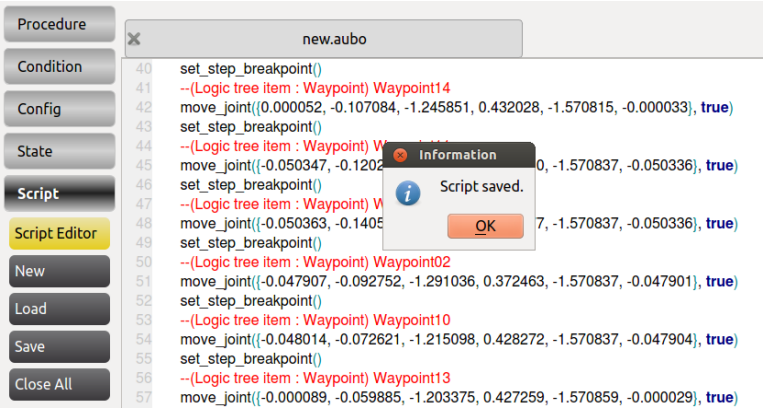
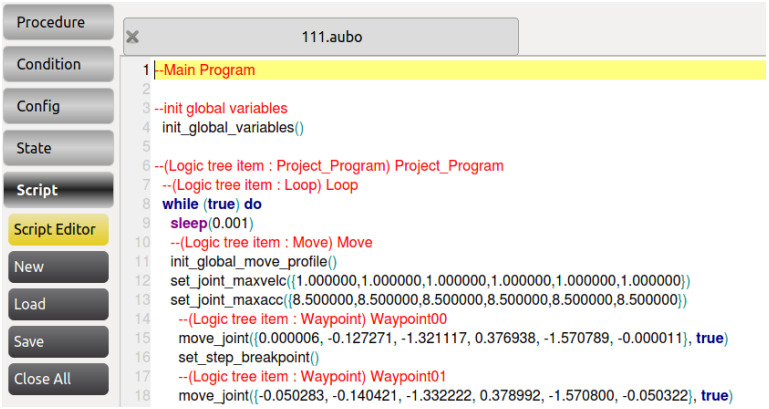
The script file is used in robot online programming to make the user's instructions more convenient and accurate.

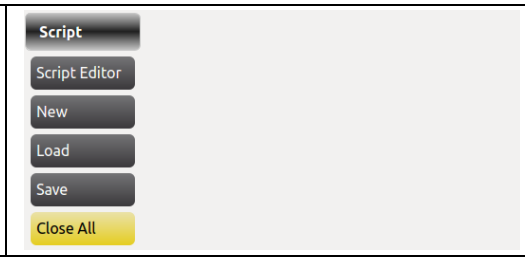
Definition

Script instructions conform to LUA syntax and can control arm movements.

Step

Please follow the steps below to configure the script file.

Step	Specific operation
1	In the teach pendant interface, click **[Online Programming]** - **[Script]**
2	Click the **New** button, enter the code command that conforms to the LUA syntax, click the **Save** button, enter the script name, click the **Save** button, and click **OK** in the pop-up dialog box. The script is created and saved successfully. 
3	Before loading the file, you need to click the **Update** button at the bottom of the interface, then select the corresponding script, click the **load** button, and the script loads successfully. Click the **Close All** button and the script file closes. The directory corresponding to the script file is: Home\AuboRobotWorkSpace\teachpendant\share\teachpendant\script. 



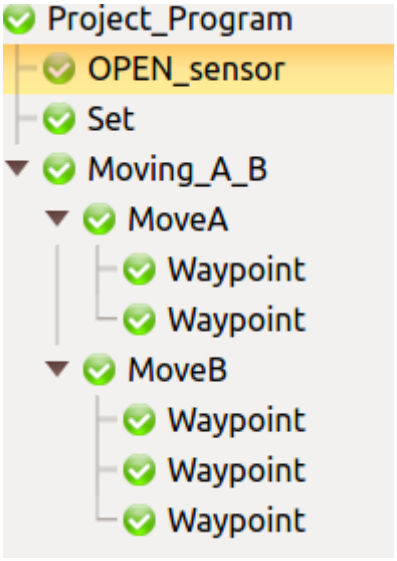
Insert line comment and block comment commands

Purpose

Line comments and block comment commands can make it easier to navigate and edit programs by annotating command usage or by creating folders for a few lines of commands.

Example

Open sensor and run action



Step

Step	Specific operation
1	Click the **condition** button in the online programming tab
2	Click the **line comment** or **block comment** button under **basic** conditions.
3	Click the line comment or block comment at the program logic and fill in the name and comment information on the right side.
4	**Empty** under **block comment** can enter the command to be executed

Create subproject commands

Purpose

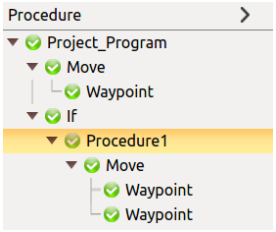
Subprojects can be used in many program files, can be used for stand-alone files in a task, or can be called multiple times in other program files

Principle

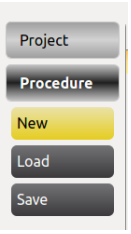
The program data contained in the subproject is only used as data when the subproject in the main program is activated. Subprojects can be called from one or more locations in the main program based on certain conditions, such as the value of a variable or the input signal of an external device.

Example

When the **if** condition is true, the subproject **procedure** will be called.



Step

Step	Specific operation
1	Create a subproject
2	Click the **process** button in the online programming tab
3	Click the **new** button in the process 
4	Save the subproject after saving the command.
5	Call subproject: Click the **condition** button in the online programming tab Click the procedure button under advanced conditions Result: The procedure command is displayed at the program logic Click the procedure undefined in the logic, select the subproject to be called in the procedure file that appears on the right, and the project content under this subproject will be displayed to the program logic.

Record Track

Purpose

Recording track enables the recording of the track of the robot during a period of time and is applied to the definition in the online programming environment.
Through the force control switch, enter the manual adjustment mode, adjust and record the movement of the robot, and apply it to the online programming environment.

Step

Follow the steps below to record the track.

Step	Specific operation
1	In the teach pendant interface, click [Online Programming] - [Config] - [Record Track] to enter the track record interface.
2	Press and hold the force control switch on the teach pendant to enter the force control mode. Click the Start button and drag the robot. When the track is complete, click the Finish button. Enter the track name and click the Save button to create a new track.
3	Select the track icon, click the load button, long press the Auto button, move the arm to the initial position of the track record, click the run button to play the track. During the track playback, you can click to move here or stop button to operate the track playback.
4	Manually drag the slider and progress bar to a certain position, click the cut head button to cut off the track before the position. Similarly, you can cut off the track after a certain position.
5	Adjust the interval time to adjust the playback speed of the track. For example, if the interval time is set to 50ms, the track will be played at twice the speed. If it is set to 200ms, it will be slowed down by 0.5 times.

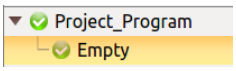
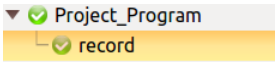
Insert track record command

Purpose

Recording track enables the recording of the track of the robot during a period of time and is applied to the definition in the online programming environment.
Through the force control switch, enter the manual adjustment mode, adjust and record the movement of the robot, and apply it to the online programming environment.

Step

Follow the steps below to insert the Record Track command.

Step	Specific operation
1	In the teach pendant interface, click [Online Programming] - [Project] - [New] to create a new empty project. 
2	Click on empty in the program logic, click [Condition] - [Advanced], click the Record Track button in the Advanced Condition interface, click Track_Record in the program logic, select the corresponding track in the Record Track condition, click the Confirm button, and pop up Click OK in the dialog box. 
3	Save the project and run the project.

Tool calibration

Purpose

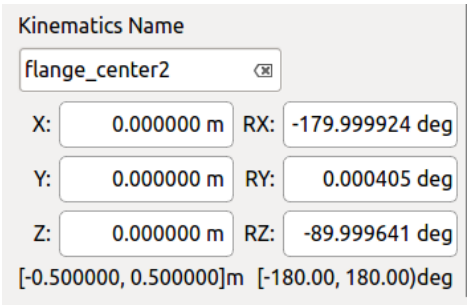
Tool center points must be defined in the teach pendant user interface to ensure optimal performance of the robot and prevent protective stops.

Definition

The tool center point is the part of the tool that is in contact with the workpiece. For example, the center of the fixture, the end of the welding tool.

Step

Step	Specific operation
1	In the teach pendant interface, click [Settings] - [Robot] - [Tool Calibration], enter the tool calibration interface, click [Kinematics Calibration] - [Kinematic Calibration], enter the kinematics calibration interface.
2	Perform position calibration: Select the calibration point type as the position calibration point, click Add, perform move control on the teaching interface or hand guiding to determine the first position point, and the same method to determine the remaining 3 or more position calibration points. Note that each position point must keep the tool end position unchanged. Pos Calibration Point Index / Point Type Pos1 / Pos Calibration Pos2 / Pos Calibration Pos3 / Pos Calibration Pos4 / Pos Calibration
3	To perform the orientationcalibration: take the xOxy with reference point as an example. First, select the orientationcalibration method xOxy through the pull-down menu at the orientationcalibration method; then select the arbitrary calibration point as the reference point through the reference point drop-down menu below the orientationcalibration point; Type Select the orientationcalibration point, click Add, perform move control or drag teach on the teaching interface to determine the first pose point; use the same method to calibrate the second pose point. Note that each pose point must maintain the end of the tool posture. Ori Calibration Point Reference Point: Pos1 Index / Point Type Ori1 / Ori Calibration Ori2 / Ori Calibration
4	Check the calibration mode option. At this time, the calibration button is enabled, click the calibration, switch to the kinematics parameter interface, enter a tool

	kinematics name, click the add button, and add the tool kinematics calibration. This interface also supports manual input tool kinematics parameters. After entering the parameters manually, click Add to save the parameters. 

Set the plane coordinate system

Purpose

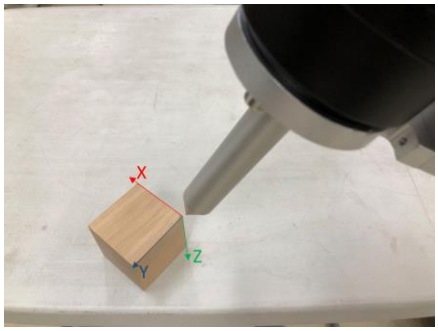
The plane feature is typically used when there is a need for a frame with high precision, e.g. when working with a vision system or doing movements relative to a table.

Principle

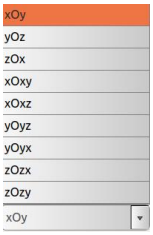
A plane feature is a coordinate system defined by three location points that indicate the Z, X and Y axes. The robot can perform movements relative to the plane feature/coordinate system, and the feature/coordinate system can be moved to another location

Example

A plane feature aligned with the surface of a workpiece:



Step

Step	Specific operation
1	Click on the Robot button in the Settings tab.
2	Click the coordinate calibration button under the robot
3	Select the desired coordinate type in the coordinate calibration method 
4	Enter the name of the coordinate system
5	Select the tool name to use
6	Select the Calibration Mode mode, select Point1, click Set Point, enter the teaching interface, and calibrate the coordinate system origin. Use the same method to calibrate Point2 and Point3. Point point meaning can be viewed at the interface coordinate type details.

Add relative offset

Purpose

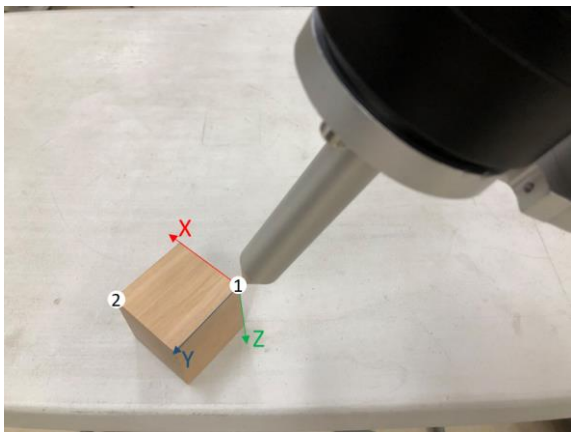
Moves can be inserted relative to a feature, which means that waypoints inside the move will automatically be relocated if the feature is moved/adjusted

Principle

When inserting a move, a feature can be associated with the move. For waypoints inside this move, the position and orientation of the tool is relative to the feature. If the feature is moved, the waypoints will be moved accordingly, still keeping the same position and orientation relative to the position of the feature

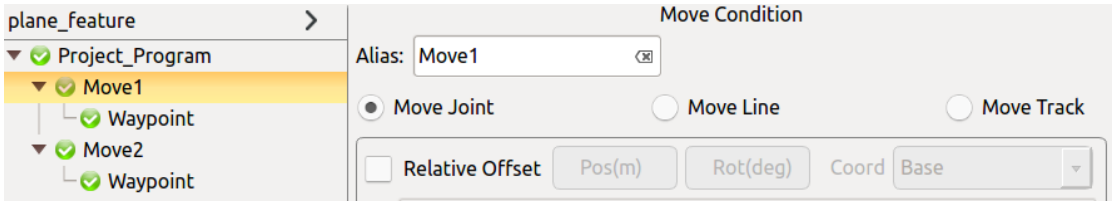
Example

The waypoints are inserted relative to feature Plane_1. When the feature and workpiece are moved left, the waypoints are automatically repositioned relative to the feature
As shown in the figure below, the end tool moves from waypoint 1 to waypoint 2, which can be run by selecting different coordinate systems by changing the waypoint.

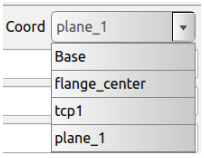


step

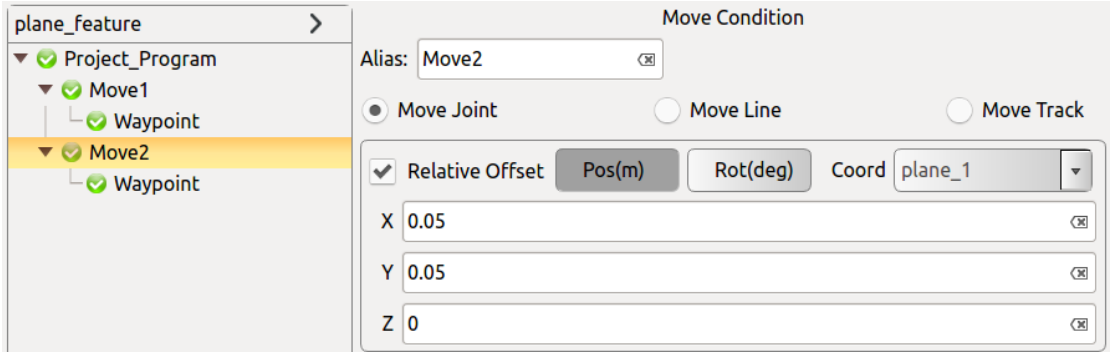
Set the coordinate system plane1 as shown above
Insert the move command as shown below, and waypoint is the location of waypoint 1.
Move1 is set to MoveJ



Move2 selects the MoveJ，
Check the relative offset and select plane1 in the position coordinate system.



Text box X is set to 0.05, Y is set to 0.05, and Z is set to 0.

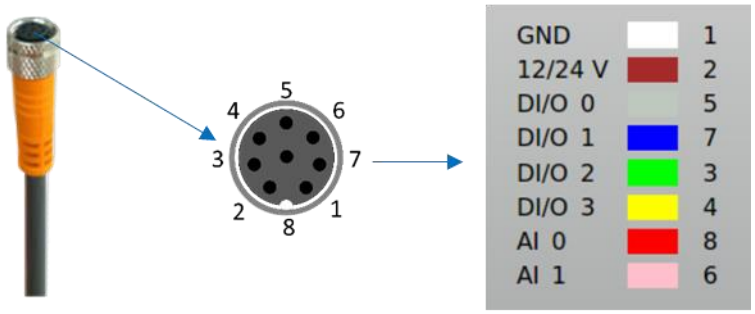


After running the project, the waypoint point parameter is unchanged, and the end tool will set the offset of the parameter according to the selected coordinate system. That is, it is operated by waypoint 1 to waypoint 2.

Tool IO

Purpose

The tool has an 8-pin small connector at the end that provides power and control signals for specific tools (clamps, etc.) used at the end of the robot with an electrical error of ±10%. The line sequence is as shown.



The cable uses the Lumberg RKMV 8-354 industrial cable, and the eight internal lines of different colors represent different functions, as shown in the table below.

Color	Signal	Pin
white	GND	1
brown	12/24V	2
gray	DI/O 0	5
blue	DI/O 1	7
green	DI/O 2	3
yellow	DI/O 3	4
red	AI 0	8
pink	AI 1	6

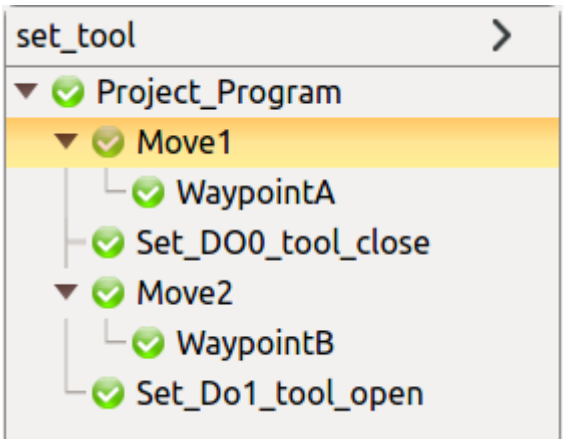
The electrical parameters of the analog terminal are：

Parameter item	Min.	Max.
Voltage input analog AI 0	0V	+10 V
Voltage input analog AI 1	0V	+10 V

Example:

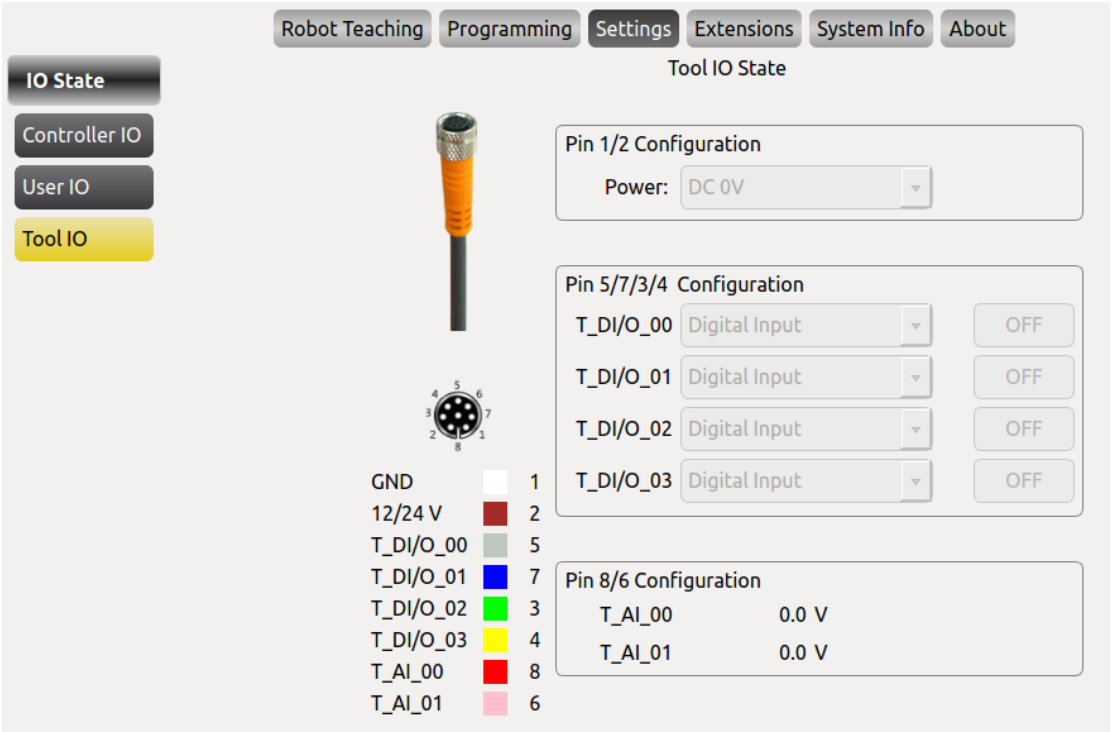
The end tool end signal can be set by the set command setting.

The end of the robotic tool is connected to the holder, which receives a signal to open or close the jaws, the gray line is connected to the closing signal of the holder, and the blue line is connected to the opening jaw of the holder. Write the project shown below:



The Set_DO0_tool_close condition is set to check IO, select tool, set DO_0 to low (this operation defaults the jaw closing signal is active low, if the jaw is active high, set DO_0 to high)

Set_tool_DO1_close condition is set to check IO, select tool, set DO_1 to low (this operation defaults the jaw open signal is active low, if the clamp open signal is active high, set DO_1 to high)



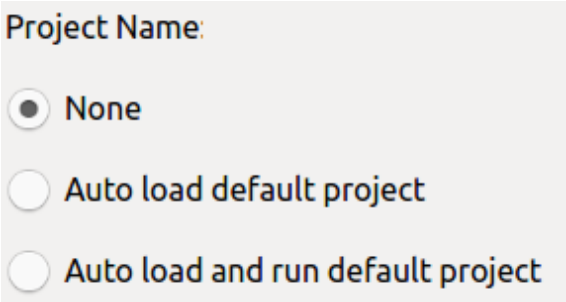
Run this project file, that is, after moving to waypointA, the jaws are closed, and after running to waypointB, the jaws open.

Automatically run the default project

Purpose

The project file can be automatically run after the robot arm is powered on. You can choose to load the project automatically (you need to start it manually), or run the project directly after booting.

Example



Step

- Click the Project button in the Online Programming tab
- Click on the default project under the Project button
- Select the project file that needs to be started by default
- Choose the default mode of operation
- Click to confirm