# Table of Contents

# Preface

**Purpose**

This Document describes the functions, , installation guide of DobotStudio2020, making it easy for users to fully understand and use it.

**Intended Audience**

This document is intended for：

- Customer
- Sales Engineer
- Installation and Commissioning Engineer
- Technical Support Engineer

**Change History**

| Date | Change Description |
| --- | --- |
| 2020/05/20 | The first release |

**Symbol Conventions**

The symbols that may be founded in this document are defined as follows:

| Symbol | Description |
| --- | --- |
|  DANGER | Indicates a hazard with a high level of risk which, if not avoided, could result in death or serious injury |
|  WARNING | Indicates a hazard with a medium level or low level of risk which, if not avoided, could result in minor or moderate injury, robotic arm damage |
|  NOTICE | Indicates a potentially hazardous situation which, if not avoided, can result in robotic arm damage, data loss, or unanticipated result |
|  NOTE | Provides additional information to emphasize or supplement important points in the main text |

# 1. Product Introduction

DobotStudio2020 is a multi-functional control software of robot arm independently developed by our company, which can control various types of robot arms of our company, such as Dobot M1, Dobot Magician, Dobot Magician Lite, Dobot CR5, MG400, etc.. The interface is simple and easy to understand which can help users quickly master the usage of various mechanical arms.

Currently, this version only supports Dobot M1 and MG400. Other types of robotic arms are under development and will be opened later.

# 2. DobotStudio2020 Installation

The supported OSs are as follows:

- Win7
- Win8
- Win10

The Download path of DobotStudio2020 is：https://cn.dobot.cc/downloadcenter.html

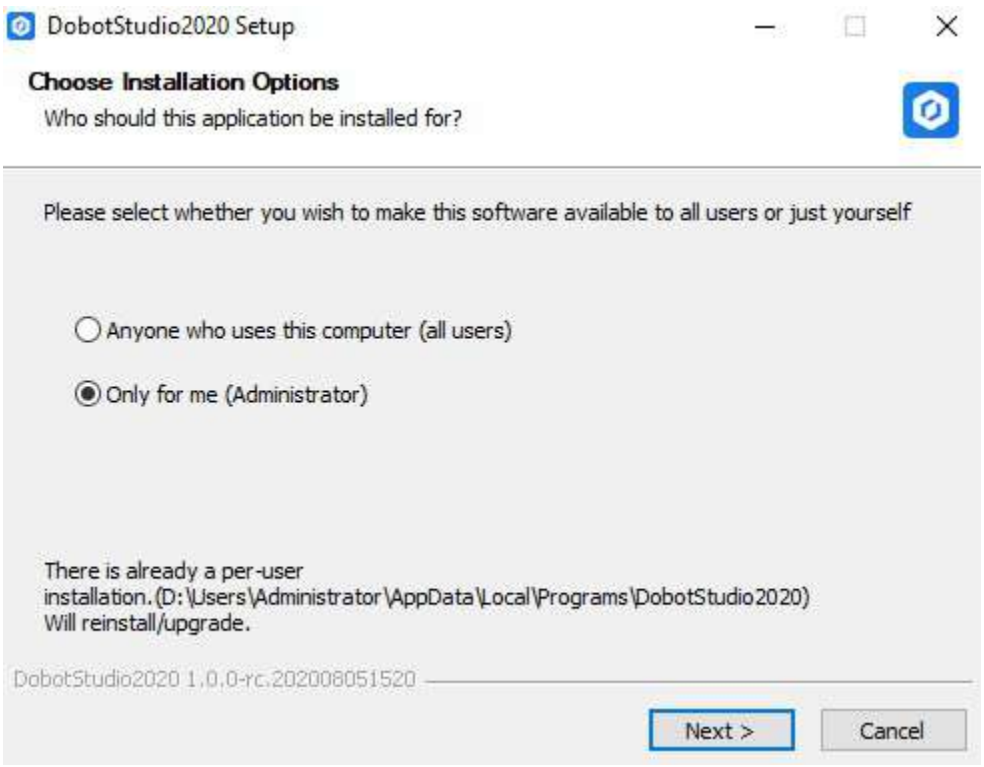
**Prerequisites**

You have obtained the DobotStudio2020 .

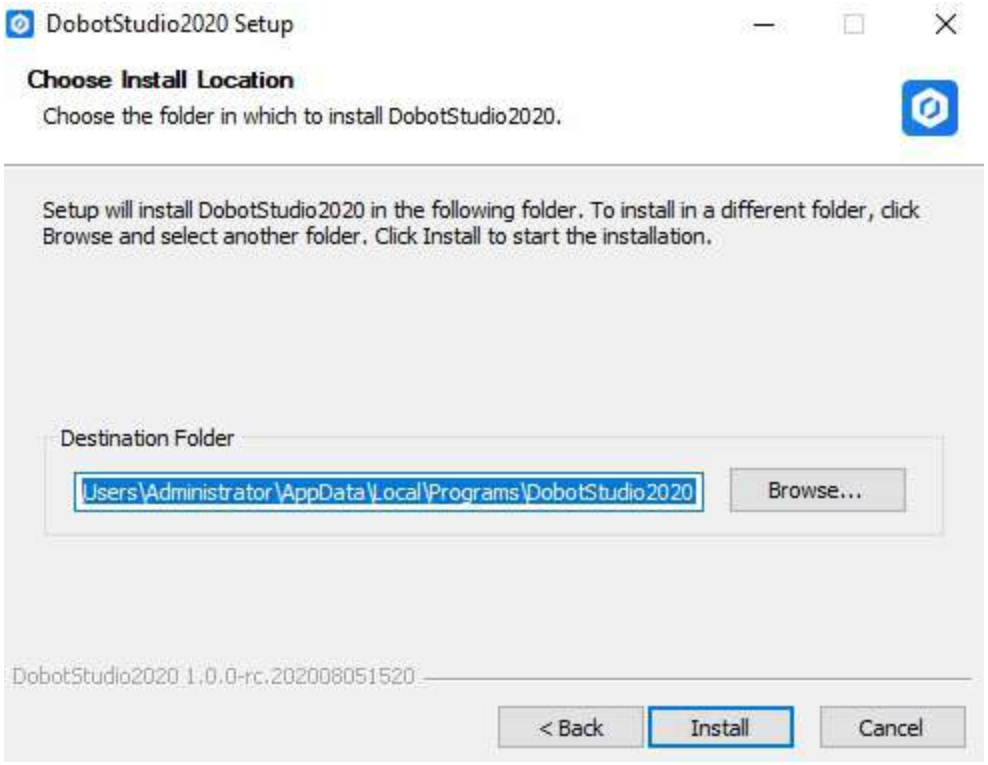
**Procedure**

**Step 1** Decompress the DobotStudio2020 software。If the directory containing decompressed DobotStudio2020 files is **E:\DobotStudio2020** . Please replace the directory based on site requirements.

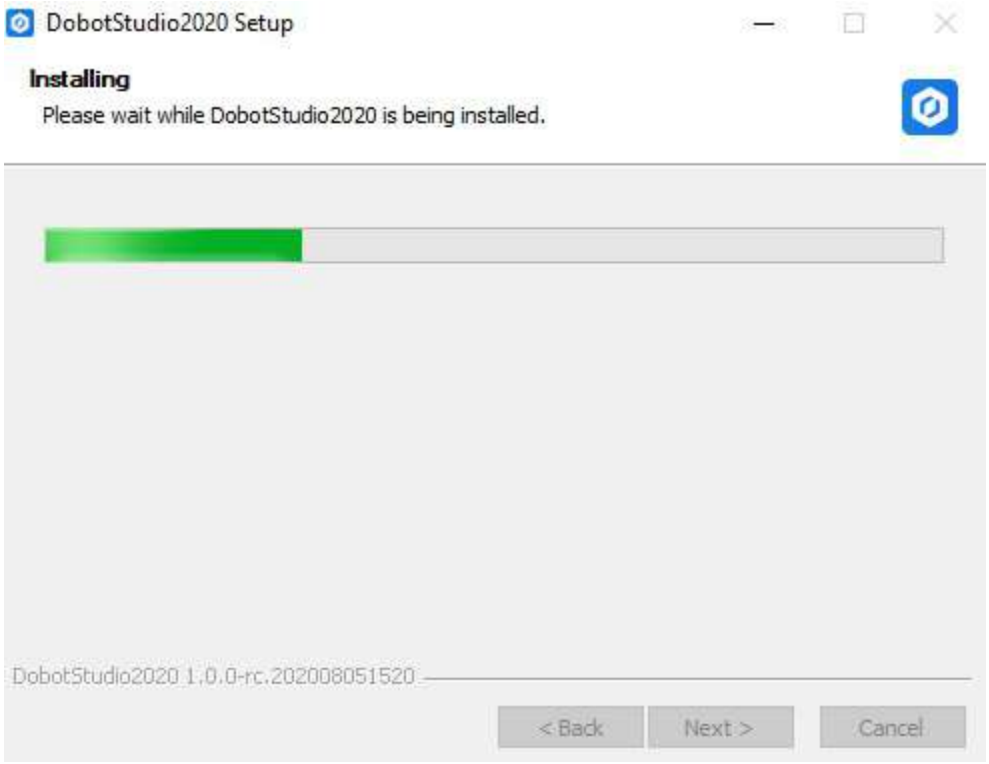
**Step 2** Double-click **DobotStudio2020.exe** in the **E:\DobotStudio2020directory** and click **Next**.



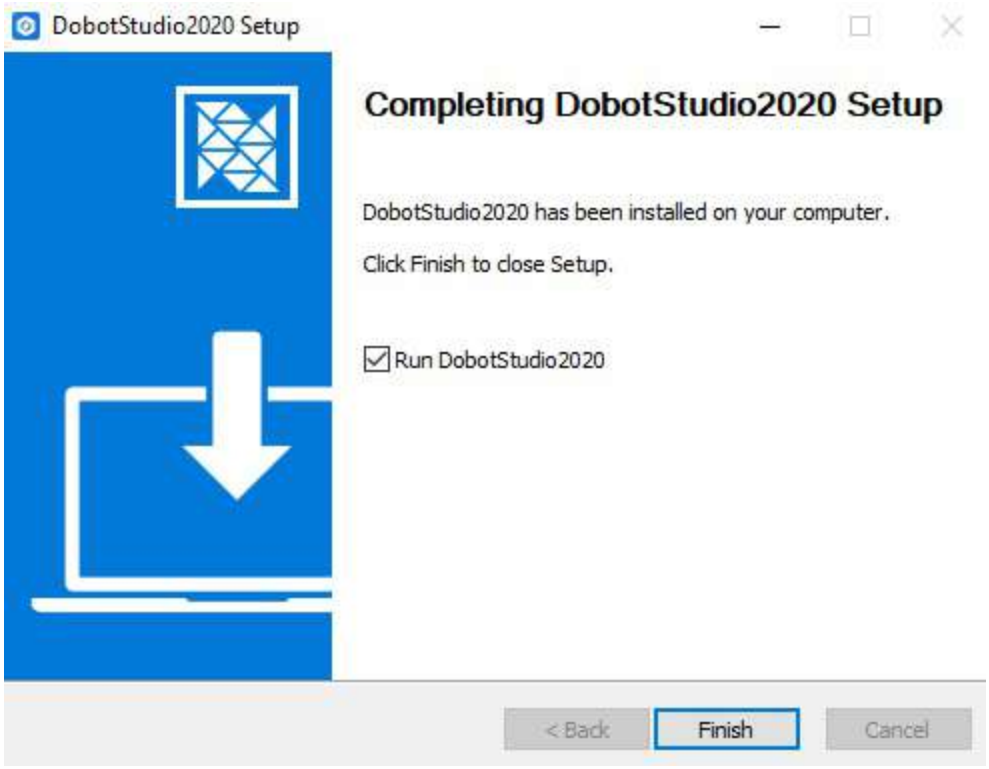

**Step 3** Select installation path and click **Install**.




**Step 4** click **Next** after finishing installing.




**Step 5** Click **Finish**.



# 3 Dobot M1

# 3.1 Overview

You can control Dobot M1 through the DobotStudio2020, and perform teaching and playback, Blockly, Script and other operations on the DobotStudio2020.

# 3.2 DobotStudio2020 Connection

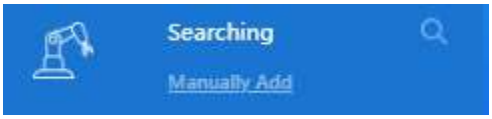
**Prerequisites**

- You have connected the Dobot M1 to the PC over a serial cable or a network cable.
- You have connected the Dobot M1 to the emergency stop switch.

**Procedure**

- If the Dobot M1 connects to PC over the serial cable, please select the right serial port and connect it on the DobotStudio2020.
- If the Dobot M1 connects to PC over the network cable (router or directly connection) and they are on the same network segment, please select the right IP address and connect it on the DobotStudio2020. If the device cannot be found, you need to add the device manually. Follow the steps as shown below.

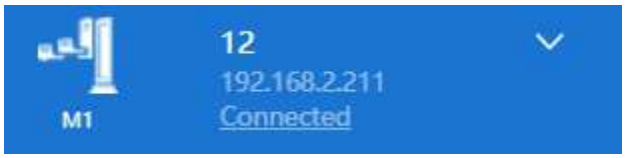
**Step 1** Click **Manually Add**.




**Step 2** Click **M1** to enter the Overwrite IP Configuration interface.


**Step 3** Click **Configure and connect**, The IP address of Dobot M1 will be forced set in the same network segment of the PC before connecting to the PC.

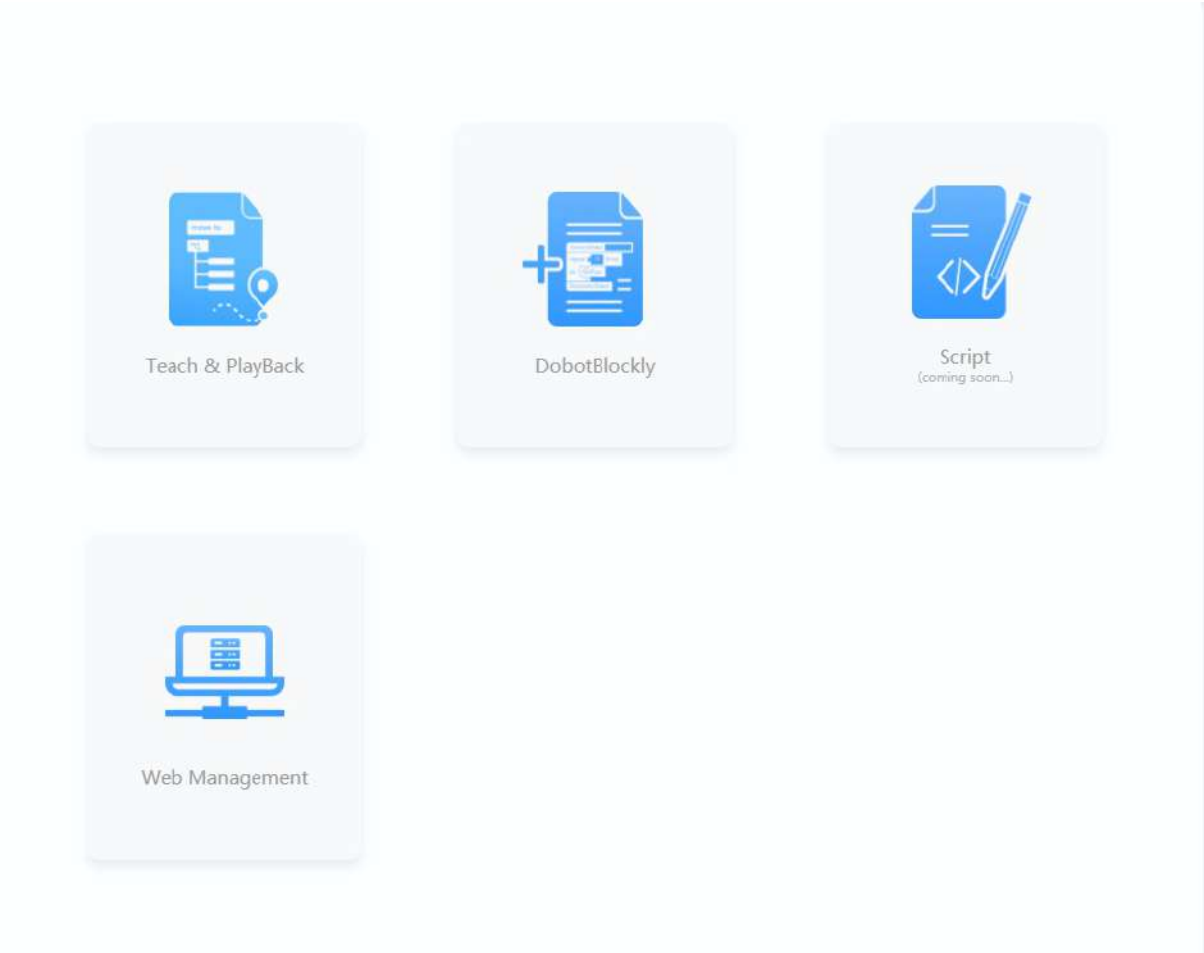

The status is **Connected** on the DobotStudio2020 after connecting successfully.




> [!NOTE] If multiple devices are configured at the same time, the system will randomly select a device to connect.

# 3.3 Function Description

Dobot M1 supports teaching, playback, script control, and Blockly graphics programming. You can use the DobotStudio2020 to control a Dobot M1. The corresponding applications on the M1Studio page is shown below.

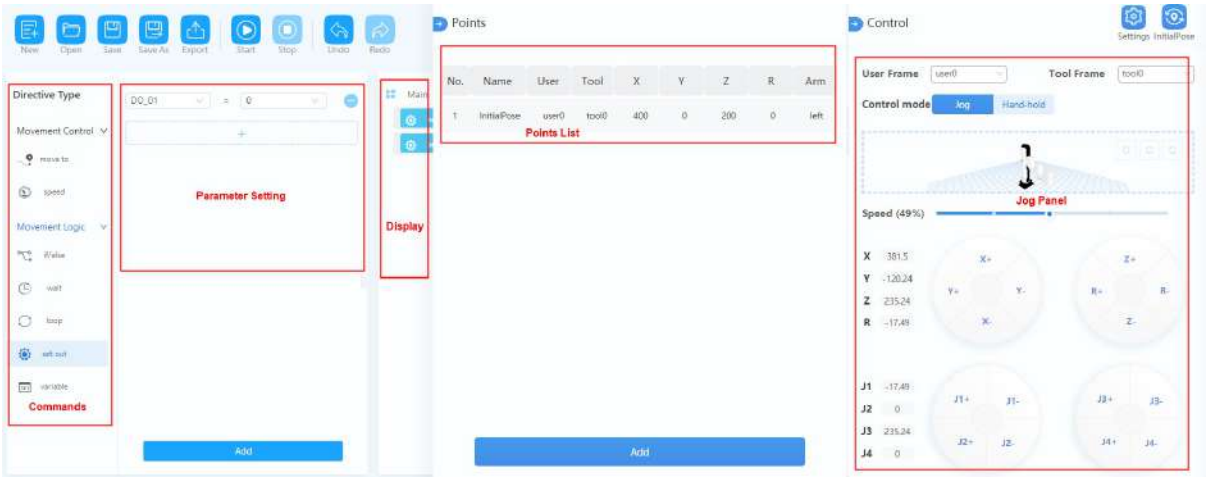



| Function | Description |
| --- | --- |
| Teach&PlayBack | Teach the Dobot M1 how to move and then record the movement to make the Dobot M1 accomplish the recorded movements. |
| DobotBlockly | Control the Dobot M1 by graphics programming. You can program through a puzzle interface which is intuitive and easy to understand. |
| Script | Control the Dobot M1 by the scripting language. |
| Web Management | Execute the saved points lists in the offline mode, and upgrade the firmware. |

# 3.3.1 Teaching and Playback

# 3.3.1.1 Overview

The teach and Playback function supports tree programming teaching, and users can perform teach and Playback through tree programming. According to different program instructions, the interface displays different parameter settings. The instruction description is shown in the table below.




| Instruction | Description | Setting |
| --- | --- | --- |
| move to | Motion instructions. Move to a certain point or follow a certain trajectory | Choose different motion mode: movj, movl, jump, arc, circle |
| speed | Speed instruction. you can set speed and acceleration of the robot arm | Set the robot arm speed acceleration ratio |
| if/else | Logical instruction. Set Judgment conditions to trigger robot movement | Logical processing based on I/O or variable setting |
| wait | Waiting instructions. The time can be set to make the robot arm wait | Set the wait time |
| loop | Loop instruction | Set the number of loop |
| set out | I/O instruction. You can set the state of I/O | Set the state of the I/O |
| variable | variable. You can create and set variable values | Create a new variable, and assign an initial value to the variable according to the variable type |

# 3.3.1.2 Alarms Description

If teaching or saving point is incorrect, for example, the Dobot M1 moves to where a point is at a limited position or a singular position, the Dobot M1 will generate an alarm. For details, please see the table as shown below. When an alarm is generated, the red LED indicator on the base will be on.

| **Alarm Condition** | **Clear Method** |
|---|---|
| **Jogging** | |
| The Joint axis is limited | Jog the limited Joint towards the opposite direction, and the alarm will be automatically cleared |
| The Cartesian axis is limited | Jog the Joints towards the opposite direction, and the alarm will be automatically cleared |
| The point is at the singular position when clicking the Cartesian coordinate buttons | Jog joint2, and the alarm will be automatically cleared |
| **Playback** | |
| The starting point or the end point is a singular point in the MOVL mode | Clear the alarm manually and modify the point |
| A point in the trajectory is a singular point in the MOVL mode | Clear the alarm manually and modify the point |
| Modify the arm orientation of the saved point in the MOVL mode | Clear the alarm manually and modify the arm orientation |
| The middle point or the end point is a singular point in ARC mode | Clear the alarm manually and modify the point |
| A point in the trajectory is a singular point in the ARC mode | Clear the alarm manually and modify the point |
| Any two of the three points of the arc coincide in the ARC mode | Clear the alarm manually and modify the point |
| The three points of the arc are in a line in the ARC mode | Clear the alarm manually and modify the point |
| The trajectory is out of range of the workspace in all modes | Clear the alarm manually and modify the point |
| The joint is limited in all modes | Clear the alarm manually and modify the point |

> [!NOTE]
>
> - Singular point: If the directions of the joint1 and joint2 are collinear, the resultant velocity of joint1 and joint2 is not in any direction, but in the direction of joint1 (joint2). Namely, the degrees of freedom of Dobot M1 are degraded. The singular point is at the position where joint2 is located at ±10°. In JUMP and MOVJ mode, the movement of Dobot M1 is joint movement, Dobot M1 will not generate an alarm about singular point.
> - Generally, if you save a point where an alarm is generated when implementing jogging, the saved point is unavailable. You need to jog Dobot M1 towards the opposite direction under the Joint coordinate system to clear the alarm, and then save the point. However, if an alarm about singular point is generated when implementing jogging, the saved point is available in JUMP and MOVJ mode.
> - In the MOVJ or JUMP mode, if the two points are the same, only different in arm orientations, J1 or J4

> may be limited when moving the Dobot M1, resulting in an alarm generated. You need to modify and resave these points and then clear the alarm manually.

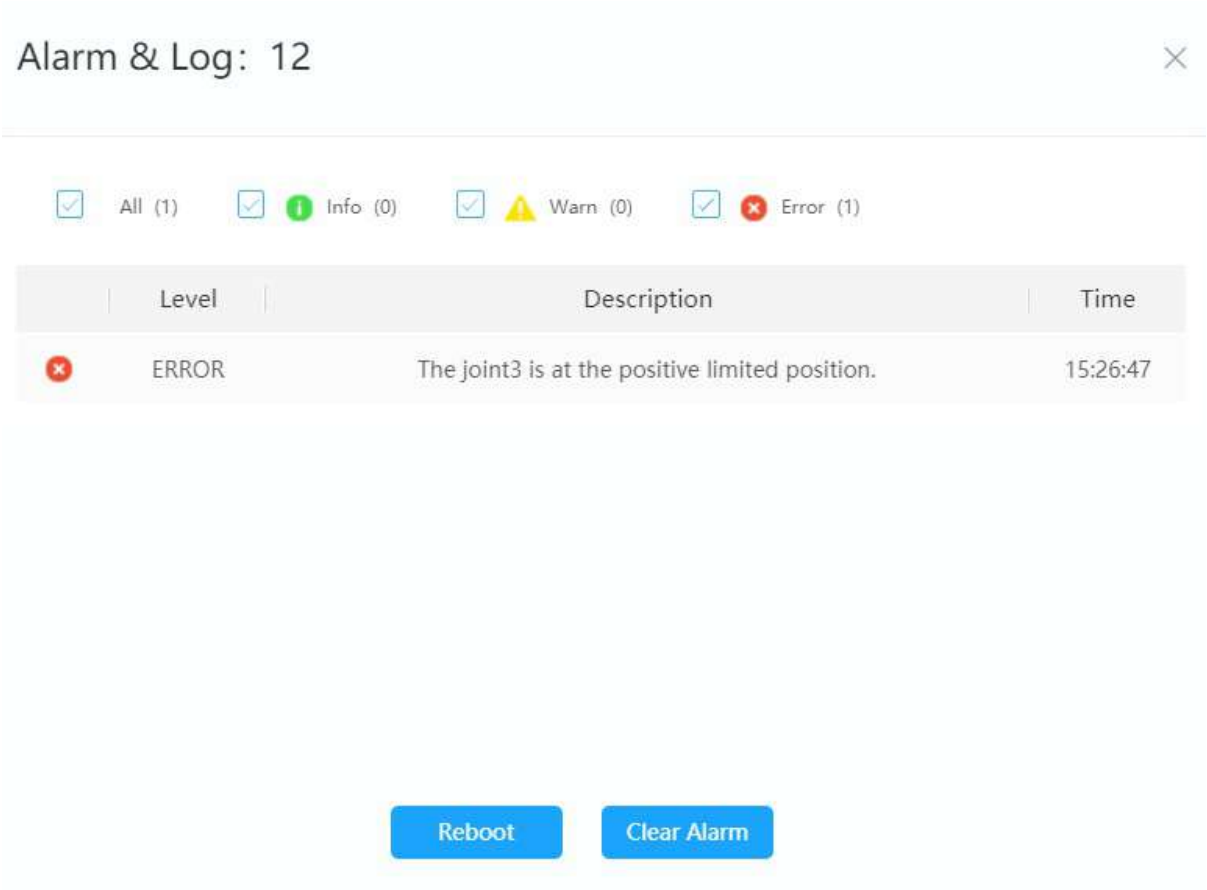
The method on how to clear the alarm is shown as follows.

**Prerequisites**

- The Dobot M1 has been powered on and connected to the DobotStudio2020 successfully.
- The Dobot M1 has been connected to the emergency stop switch.

**Procedure**

**Step 1** Click the alarm tip on the DobotStudio2020 page, as shown below. The **Alarm Log** is displayed.




**Step 2** Click the right alarm, the details on alarm will be displayed.

**Step 3** Please clear the alarm with the solution.

If there are no alarm tips on the DobotStudio2020 page, the alarm has been cleared.

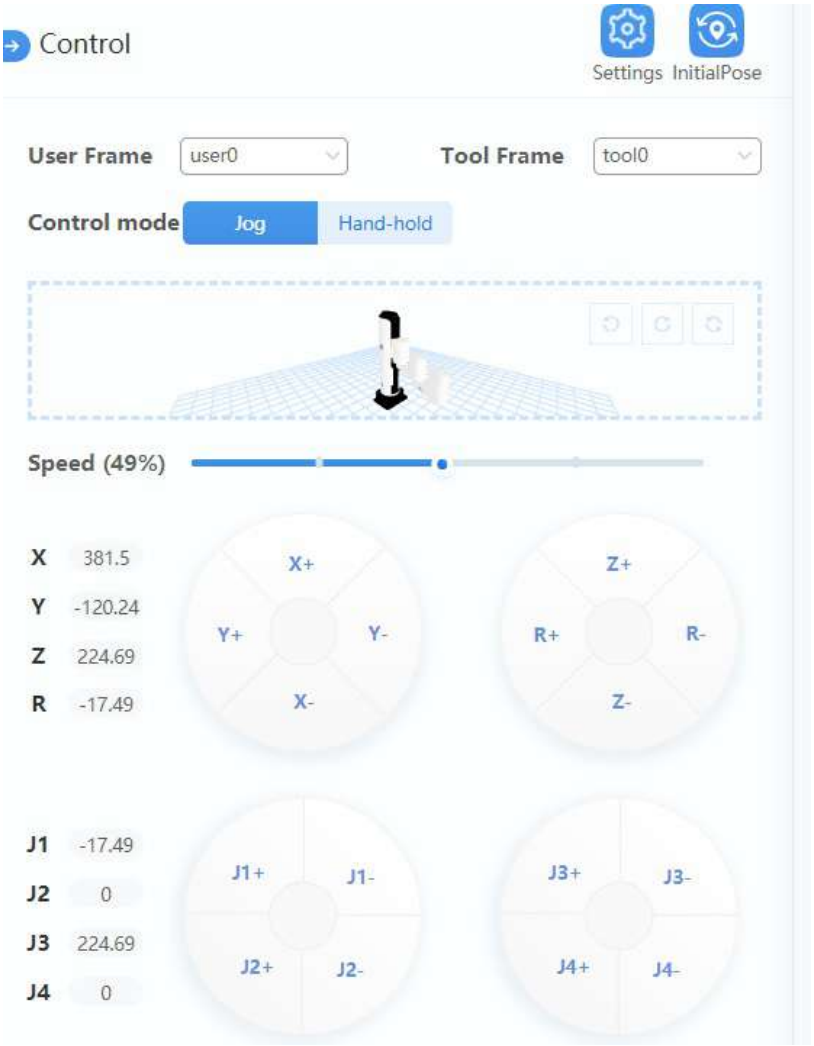
# 3.3.1.3 Motion Mode

The motion modes of Dobot M1 include Jogging, Point to Point (PTP), ARC, and CIRCLE.

- **Jogging Mode**

Jogging mode is to jog Dobot M1 in the Cartesian coordinate system or Joint coordinate system.

> [!NOTE]
>
> This topic describes jogging mode by the GUI operation of DoboStudio2020.




**Cartesian coordinate system mode**:

1. Click **X+**, **X-** and Dobot M1 will move along X-axis in the negative or positive direction.
2. Click **Y+**, **Y-** and Dobot M1 will move along Y-axis in the negative or positive direction.
3. Click **Z+**, **Z-** and Dobot M1 will move along Z-axis in the negative or positive direction.
4. Click **R+**, **R-** and Dobot M1 will rotate along R-axis in the negative or positive direction.

**Joint coordinate system mode**:

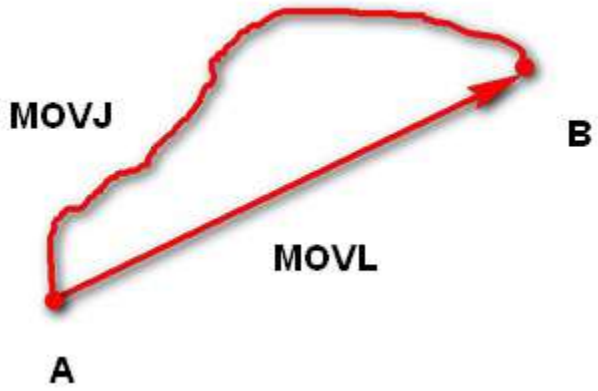
1. Click **Joint1+**, **Joint1-** and control the Rear Arm to rotate in the negative or positive direction.
2. Click **Joint2+**, **Joint2-** and control the Forearm to rotate in the negative or positive direction.

3. Click **Joint3+**, **Joint3-** and control the Z-axis to move in the negative or positive direction.
4. Click **Joint4+**, **Joint4-** and control the R-axis to rotate in the negative or positive direction.

- **Point to Point Mode (PTP)**

PTP mode supports MOVJ, MOVL, and JUMP, which means point to point movement. The trajectory of playback depends on the motion mode.

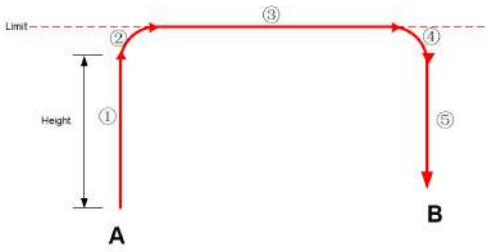
**MOVJ**：Joint movement. From point A to point B, each joint will run from the initial angle to its target angle, regardless of the trajectory, as shown below.




**MOVL**：Rectilinear movement. The joints will perform a straight line trajectory from point A to point B, as shown above.

**JUMP:** From point A to point B, The joints will move in the MOVJ mode, of which the trajectory looks like a door, as shown below.

1. Move up to the lifting Height (**Height**) in the MOVJ mode.
2. Move up to the maximum lifting height (**Limit**).
3. Move horizontally to a point that is above B by height.
4. Move down to a point that is above B by height, which the height of the point is that of point B plus **Height**.
5. Move down to Point B.




In the JUMP mode, if the starting point or the end point is higher than or equal to **Limit**, or the height that the end effector lifts upwards is higher than or equal to **Limit**. Assuming that point A is the starting point, point B is the end point, **Limit** is the maximum lifting height, and **Height** is the lifting height.

- Point A and point B are both higher than **Limit**, but point A is higher than point B.


- Point A and point B are both higher than **Limit**, but point B is higher than point A.


- Point A is higher than **Limit**, but point B is lower than **Limit**.


- The height of point A is the same as that of point B, but both are higher than **Limit**.

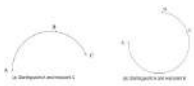

- Point A is lower than **Limit**, but point B is higher than **Limit**.


- The height of point A and point B are both the same as **Limit**.


- Point A and point B are both lower than **Limit**, but point A plus **Height** and point B plus **Height** are higher than **Limit**.


- **ARC**

The trajectory of ARC mode is an arc, which is determined by three points (the current point, any point and the end point on the arc), as shown below.



- **Circle**

The CIRCLE mode is similar to the ARC mode and its trajectory is a circle. In the CIRCLE mode, it is necessary to confirm the starting point with other motion modes.

# 3.3.1.4 Arc Mode Description

Different from PTP, the trajectory of ARC is an arc, you need to save three points to complete the arc trajectory. The method to save points in CIRCLE is the same as that of ARC.

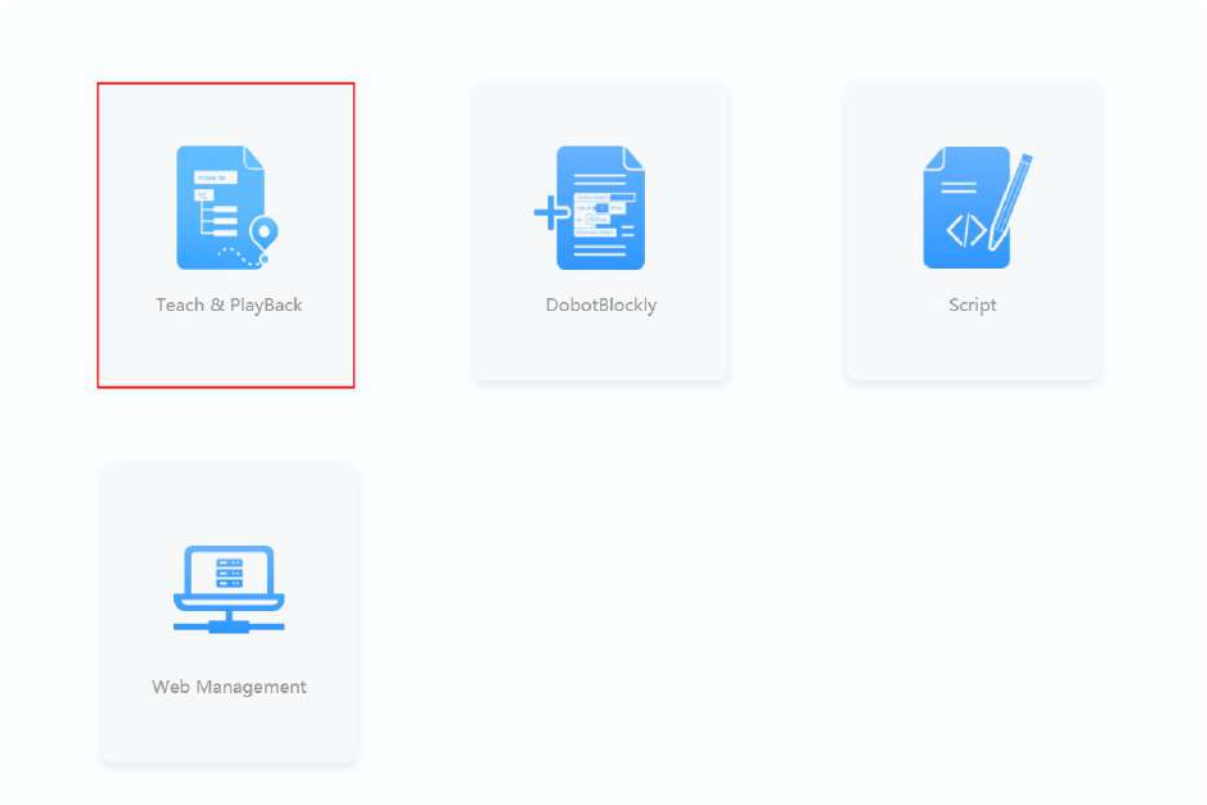
**Prerequisites**

- The Dobot M1 has been powered on.
- The Dobot M1 has been connected to the PC successfully.
- The Dobot M1 has been connected to the emergency stop switch.

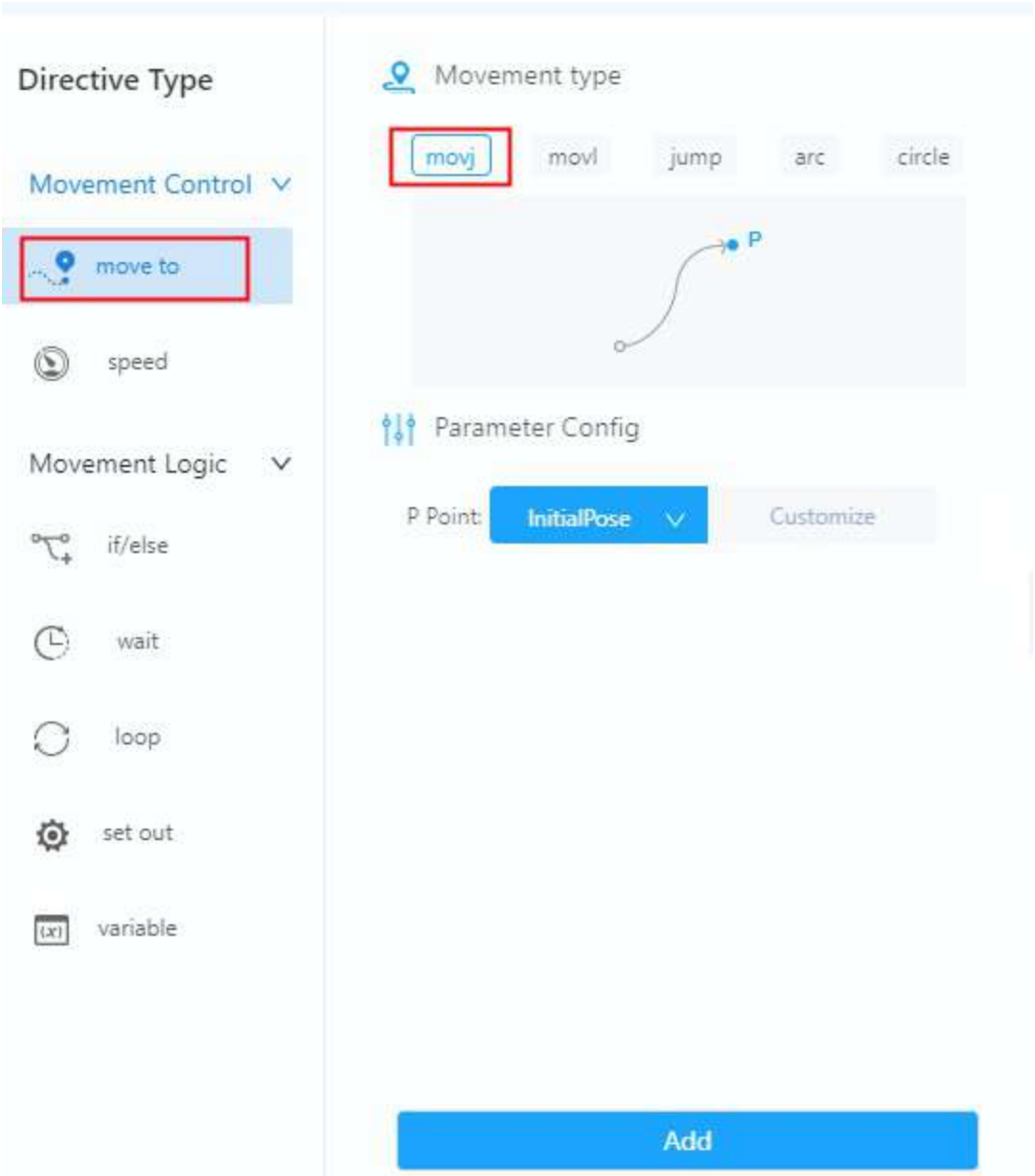
**Procedure**

> [!WARNING]
>
> You need to use other motion modes to confirm the starting point of the arc trajectory because the middle point and the end point only can be confirmed in ARC mode. When saving points in the ARC mode, please pay attention to the following tips to avoid generating an alarm.
>
> - Any two of the three points of the arc cannot coincide.
> - The three points of the arc cannot be in a line.
> - The arc trajectory cannot be out range of the workspace
> - The arm orientations in ARC and other modes used to confirm the starting point should be the same. Otherwise, the Dobot M1 will not work.

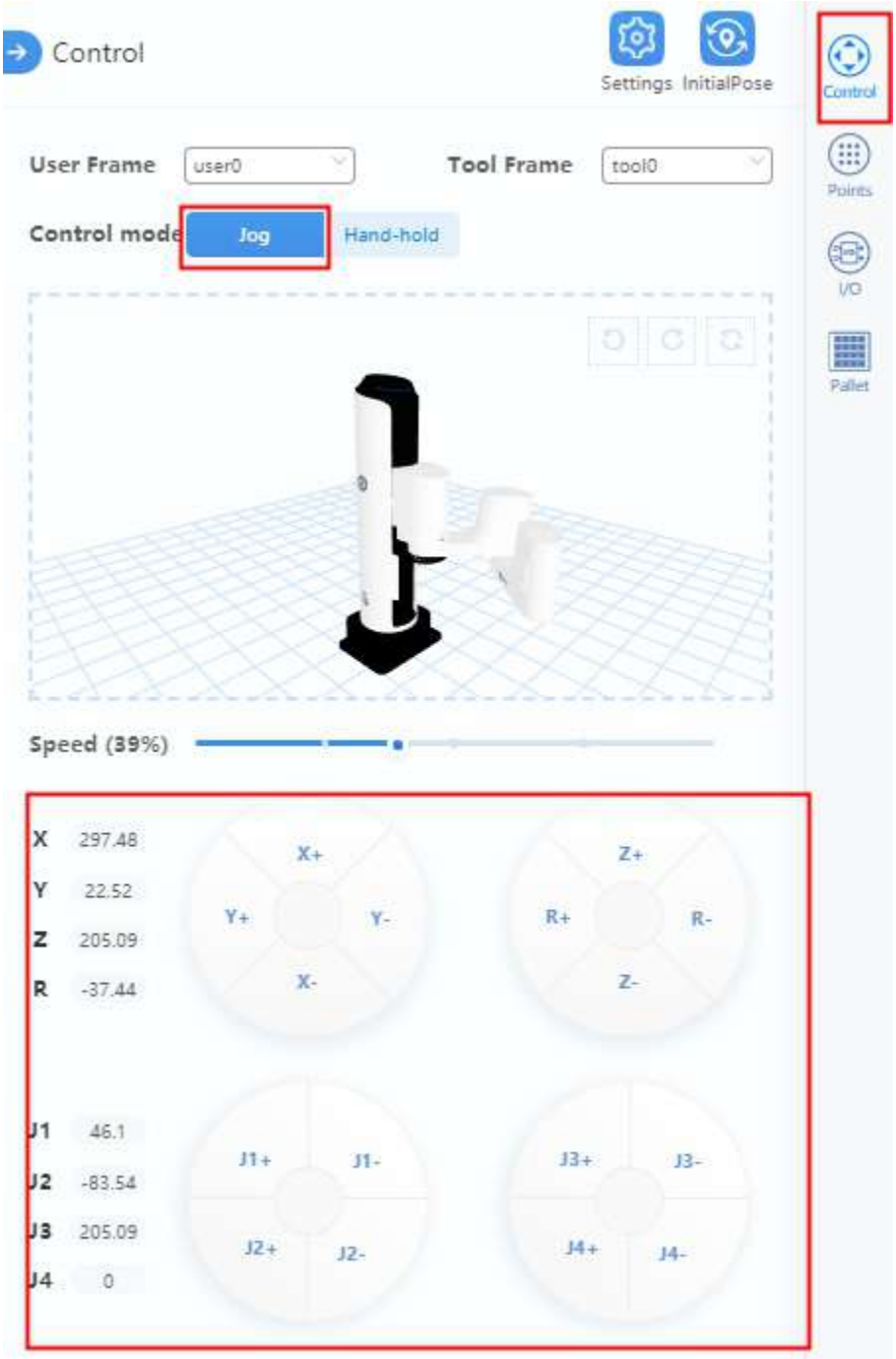
**Step 1** Click **Teach&PlayBack**.



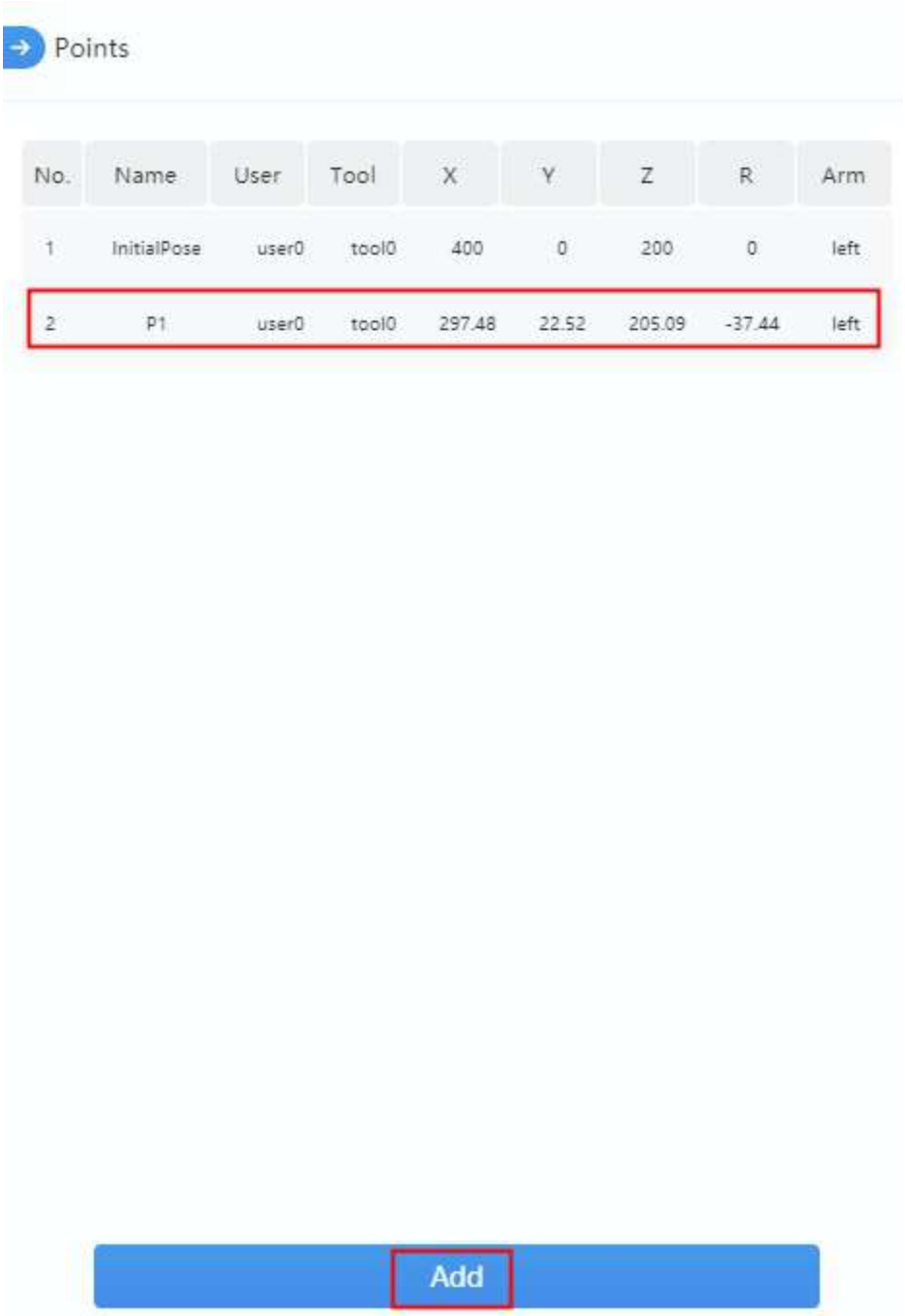

**Step 2** Click **move to** and select **movj**.



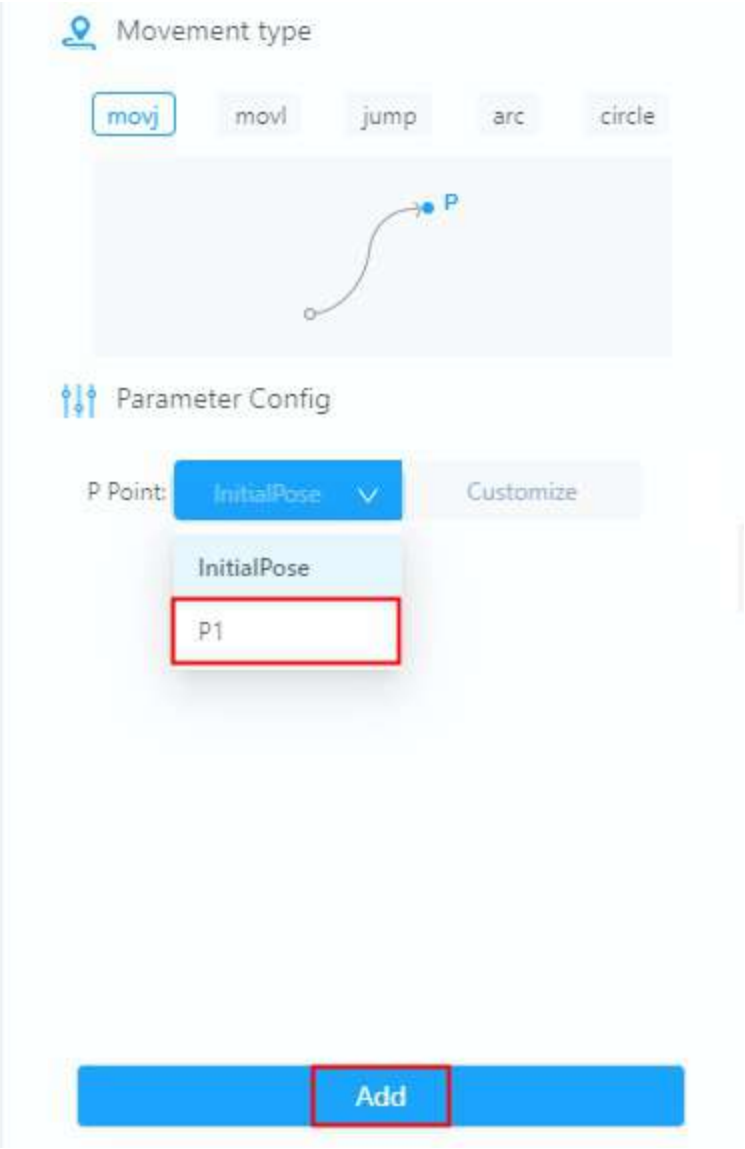

**Step 3** Click **Control** to enter the control page, click **Jog**, and then click the Cartesian coordinate system button to jog the robot arm to a point P1.



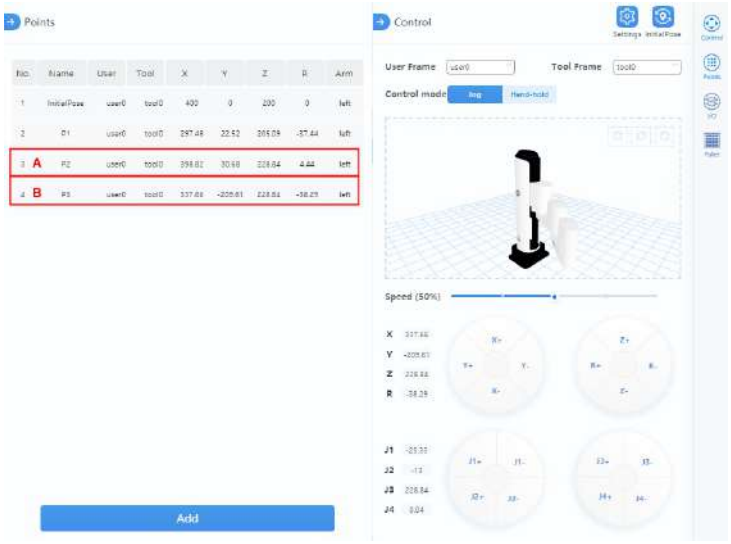

**Step 4** Click **Points** and **ADD** to save point P1.



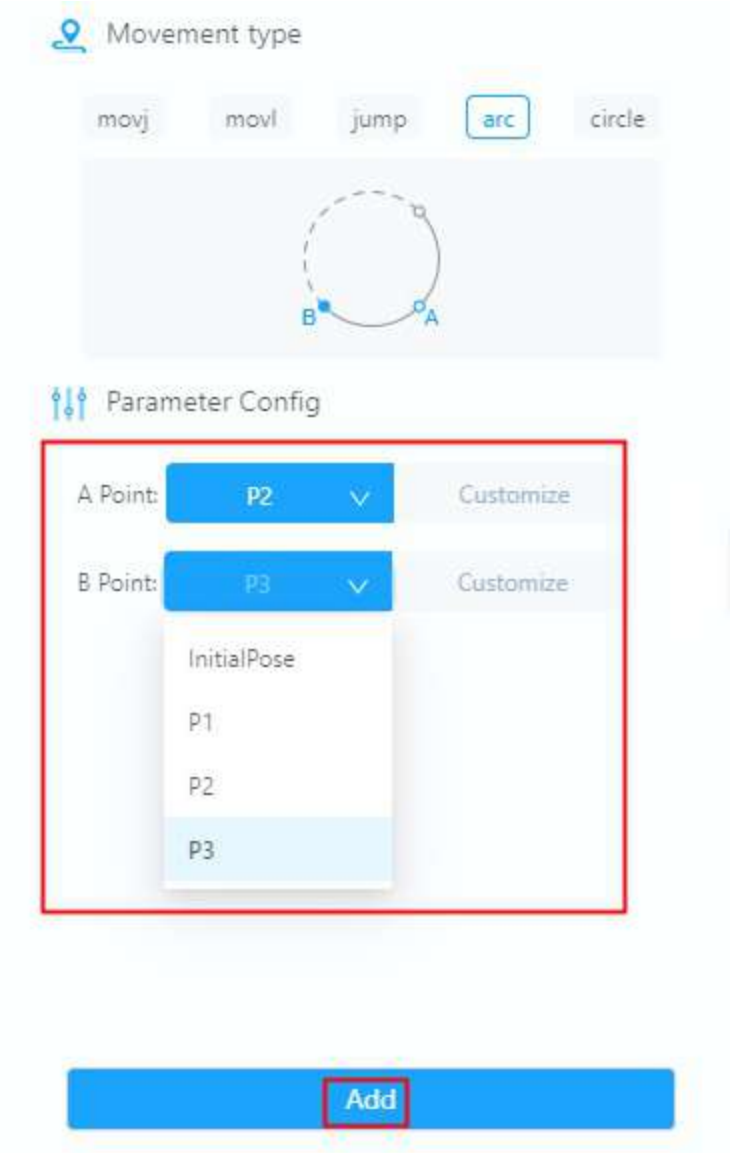

**Step 5** Click the drop-down box to select **P1** point and click **Add** on the **Parameter Config** panel.




**Step 6** Click **Control** to enter the control page, click **Jog**, and then click the Cartesian coordinate system button to jog the robot arm to a point called point A. Click **Points** to enter the saved points list page, click **Add** to save the coordinates of point A. Jog the robot arm again to another point called point B, click **Add** to save the coordinates of point B.



**Step 7** Select the **arc** motion mode, click the drop-down box on the **Parameter Config** panel and select point A and point B respectively, then click **Add**.




**Step 8** Click **Start**，you can see the robot arm moves as an arc.

# 3.3.1.5 Save Point in Jump Mode

From point A to point B in JUMP mode:

- If point A and point B are only different in Z-axis, and the arm orientations of them are the same, Dobot M1 will not work.
- If point A and point B are the same, only different in arm orientations, for example, the arm orientation of point A is left, and that of point B is right, point A moves to point B as the right hand posture, while the terminal coordinate relative to the origin stays constant.

# 3.3.1.6 Basic Operation

**Application Scenario**

If you want to use the Dobot M1 to transport, intelligent sort, write and draw, the teaching and playback function of Dobot M1 can help you to complete. This section uses the suction cup as the end effector to describe how to operate

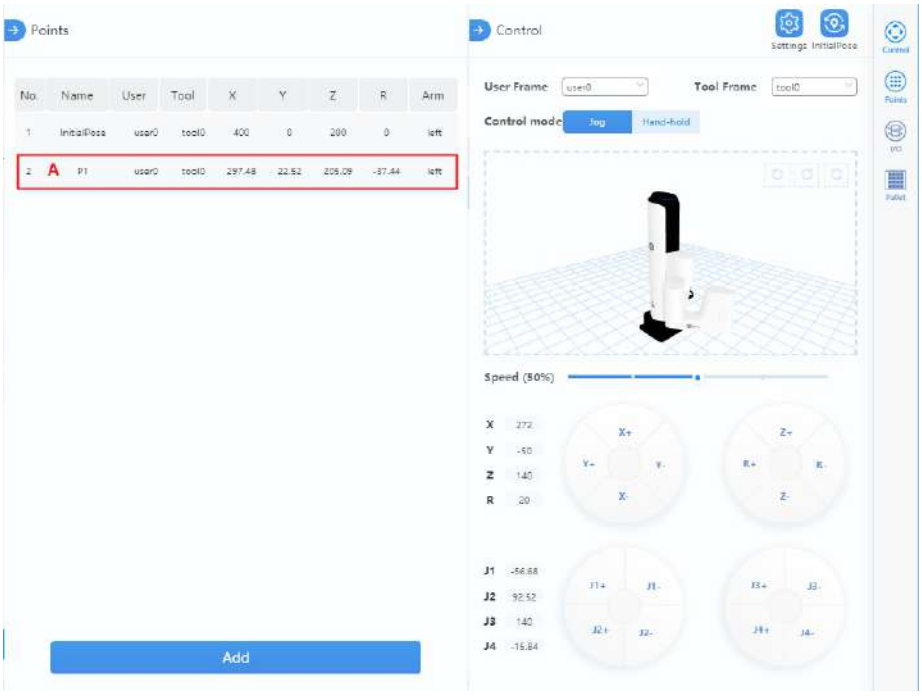
**Prerequisites**

- The Dobot M1 has been powered on and connected to the DobotStudio2020 successfully.
- The Dobot M1 has been connected to the emergency stop switch.
- The air pump and the suction cup need to be installed when you suck up objects over the teaching and playback function. For details, please see ***Dobot M1 User Guide***.

**Procedure**

**Step 1** Place a small object nearby the Dobot M1, choose one of the following three methods to jog the Dobot M1 to the small object, called point A. The distance from the Dobot M1 to the object should be determined based on site requirements.

- click **Jog** on the **Control** page and click the Cartesian coordinate buttons.
- click **Jog** on the **Control** page and click the Joint coordinate buttons.
- click **Hand-hold** on the **Control** page and and jog the Dobot M1 by hand.

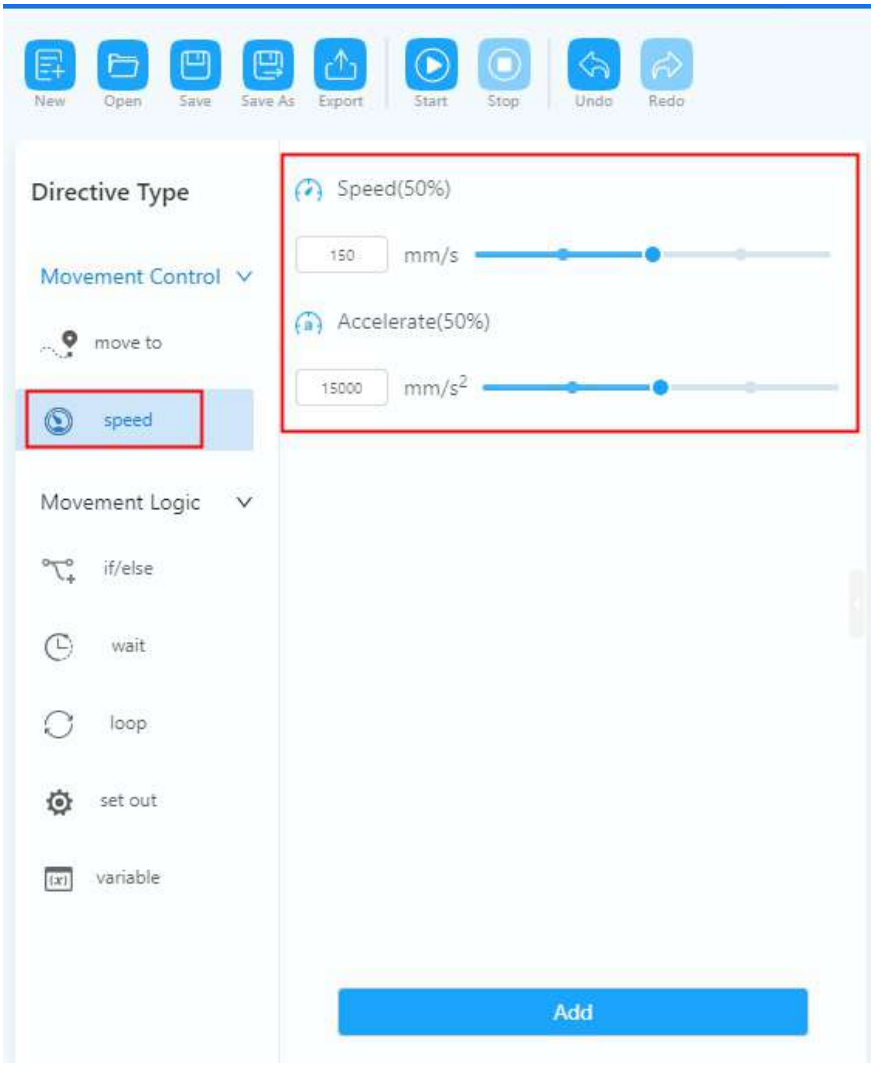



[!NOTE]

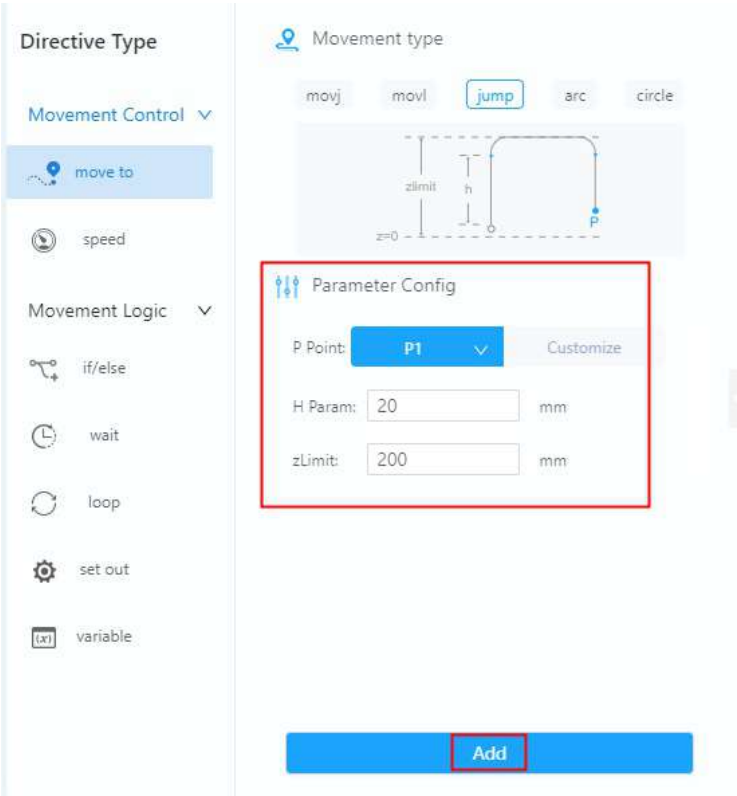
- If you want to jog the Dobot M1 by hand when implementing jogging, please click **Hand-hold**to make the motor of Dobot M1 in the disabled state. If you want to move Dobot M1 by clicking the coordinate buttons , please click **Jog** to make the motor in the enabled state.
- If an axis is limited or a point is at the singular position when implementing jogging, an alarm will be

> generated. If you save a point after an alarm is generated, the saved point is unavailable. You need to jog the Dobot M1 to clear the alarm, and then save the point again. However, if an alarm about singular point is generated when implementing jogging, the saved point is available in JUMP and MOVJ mode.

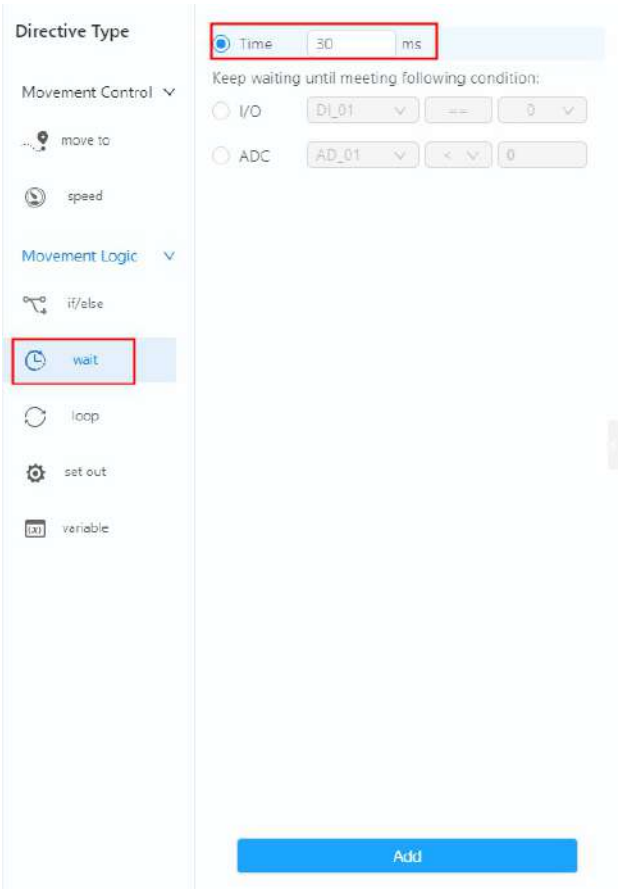
**Step 2** Click **speed** on the **Teaching&PlayBack** page. Set the speed and acceleration ratio to **50%**, click **Add**.




**Step 3** Click the **move to** and select the **jump** motion mode. Click the drop-down box to select point A and set the lifting height (H Param) and the maximum lifting height (zLimit) on the **Parameter Config** page, and click **Add**.




**Step 4** Click **wait** and set **Time** to 30ms, and click **Add**.

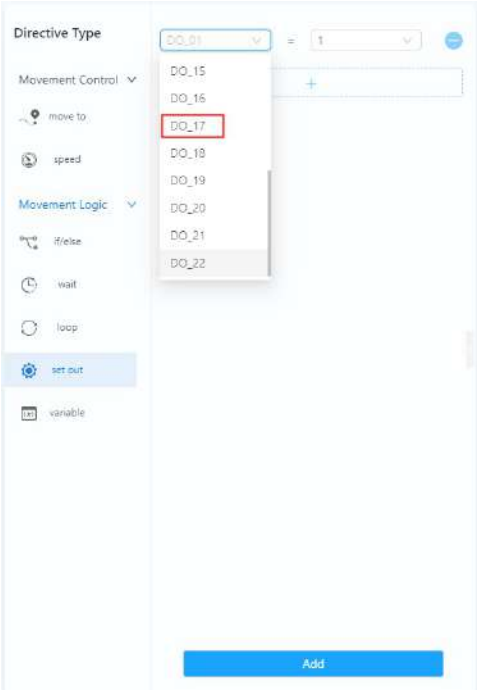



**Step 5** Suck up the small object by the suction cup.

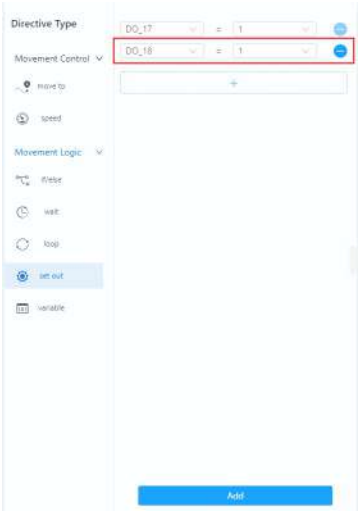
> [!NOTE]

> Supposing that we use DOUT17, DOUT18 on the base I/O interface to control the state of the air pump. DOUT17 controls the intake and outtake of the air pump. DOUT18 controls the startup and shutdown. The description in this topic is for reference only, the outputs depend on the I/O interface used. Please replace the outputs based on site requirements.

1. Click **set out** . I/O configuration information will be displayed.
2. Click the drop-down list to select **DO_17** and set to **1**.

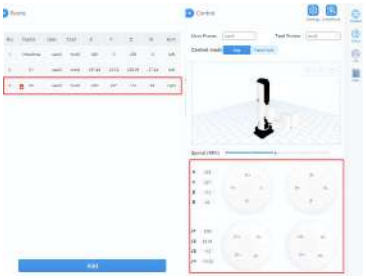



3. Click **+** to add I/O, click the drop-down list to select **DO_18** and set to **1**.




4. Click **Add**.

**Step 7** Click **Z+** to raise the robotic arm, and click other buttons of the Cartesian coordinate system, such as "X+" to move the robotic arm to another point called point B. Click **Add** to save point B.

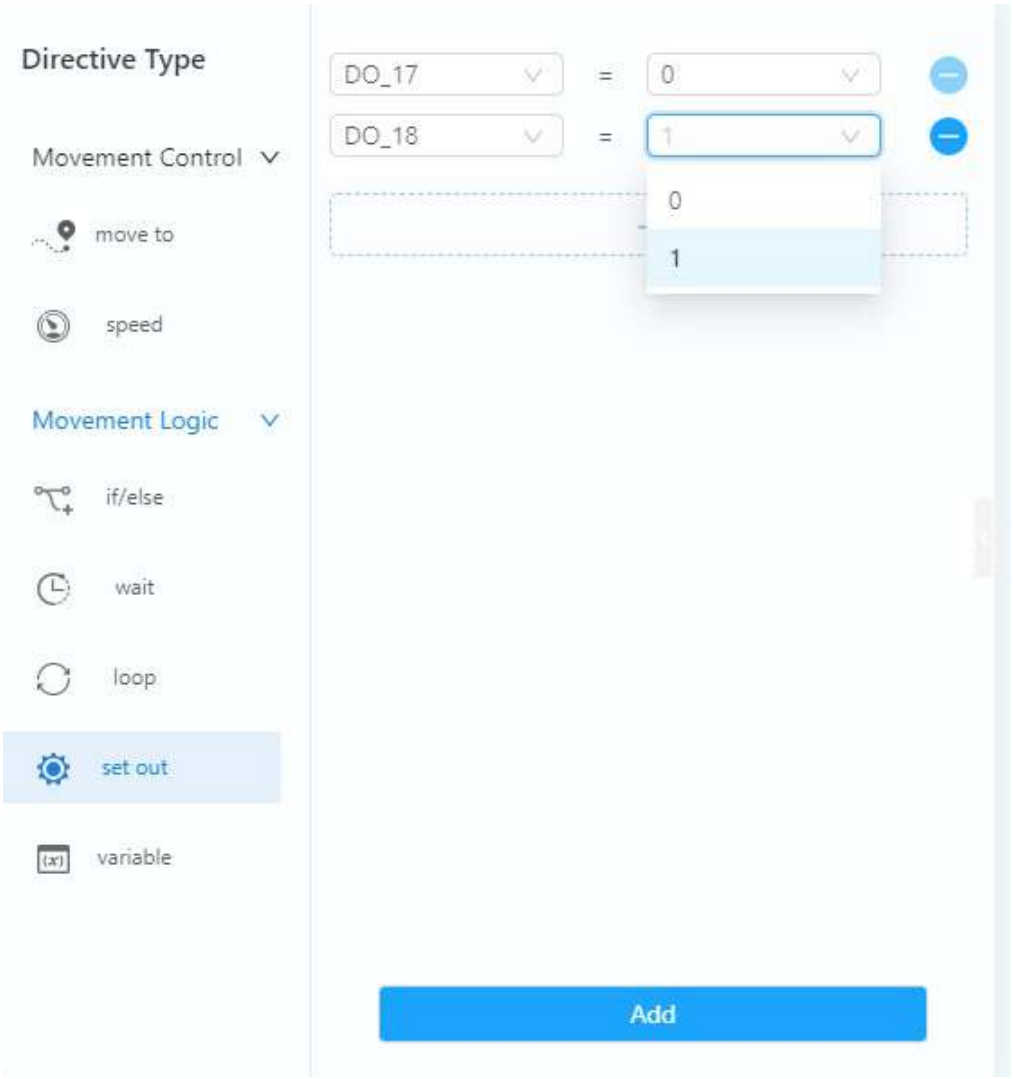


**Step 8** Save Point B. For details, please see Step 3 to Step 4.

**Step 9** Release small object over the suction cup.

1. Click **set out** . I/O configuration information will be displayed.
2. Click the drop-down list to select **DO_17** and set to **0**.

3. Click **+** to add I/O, click the drop-down list to select **DO_18**, and set to **1**.




4. Click **Add**.

**Step 10** Click **Save** and **Start**, the robot arm will teach and playback according to the saved point, and absorb and release the small object.


> [!NOTE]
>
> - This topics only describes a trajectory as an example. You can implement multiple trajectories. For details, please see Step 1 to Step 10.
> - If you need to operate this file in offline mode, you can click **Export** to save it and then upload it on the Web Management page for running. For details, please see 3.2.4.1 Managing Offline File.

# 3.3.2 DobotBlockly

**Prerequisites**

- The Dobot M1 has been connected to DobotStudio2020.
- The Dobot M1 has been connected to an emergency stop switch.

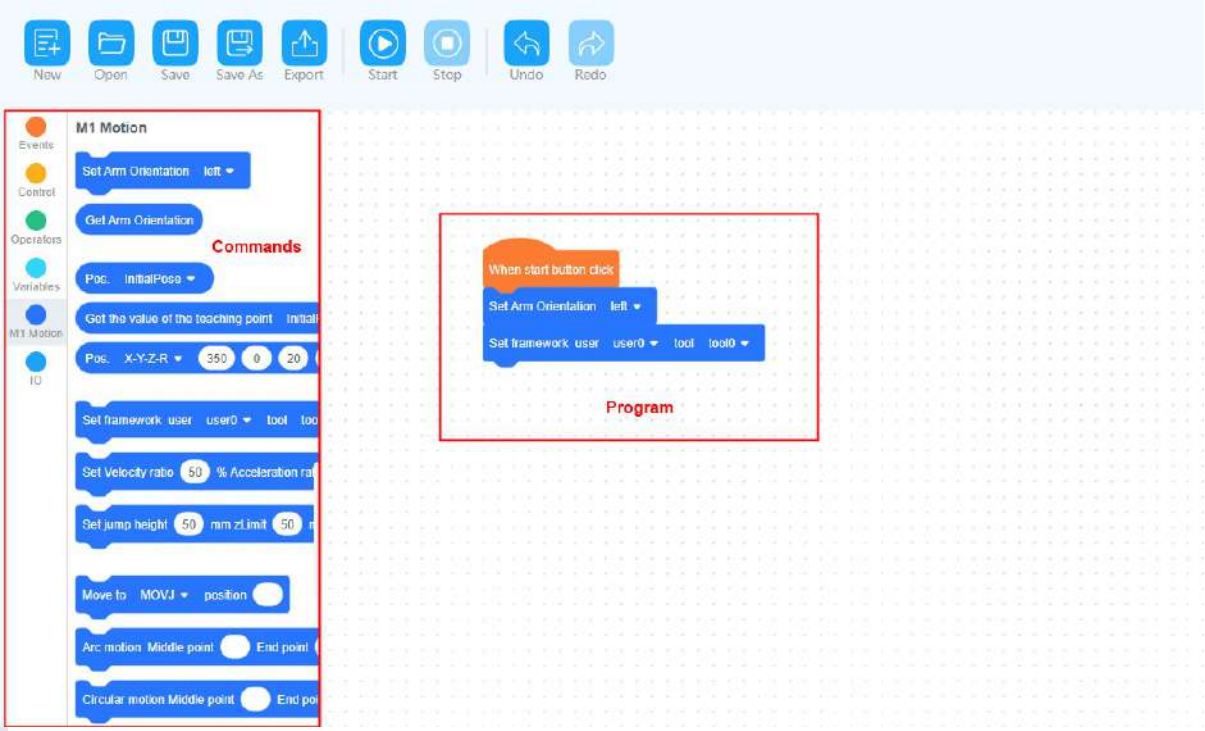
**Application Scenario**

Blockly is a programming platform based on Google Blockly. You can program through the puzzle format, which is straightforward and easy to understand.

**Procedure**

**Step 1** Click **Blockly** to enter Blockly interface.

**Step 2** Drag the blockly module on the left panel of the **Blockly** page to program, as shown below.




> [!NOTE]
>
> If robot moves to a point which is user-defined in the motion command, please add the orientation command before this motion command, which indicates the arm orientation of Dobot M1.

**Step 3** Click **Save** and **Start** to run the program and the robot arm will move as the program.

> [!NOTE]
>
> If you need to operate this file in offline mode, you can click **Export** to save it and then upload it on the Web Management page for running. For details, please see 3.2.4.1 Managing Offline File.

# 3.3.3 Script

**Prerequisites**

- The Dobot M1 has been connected to DobotStudio2020.
- The Dobot M1 has been connected to an emergency stop switch.

**Application Scenario**

You can control a Dobot M1 over scripting. Dobot M1 supports various API, such as velocity/acceleration setting, motion mode setting, and I/O configuration, which uses Python language for secondary development. For details about the Dobot M1 API interface and function description, please see *Dobot API Interface Document.

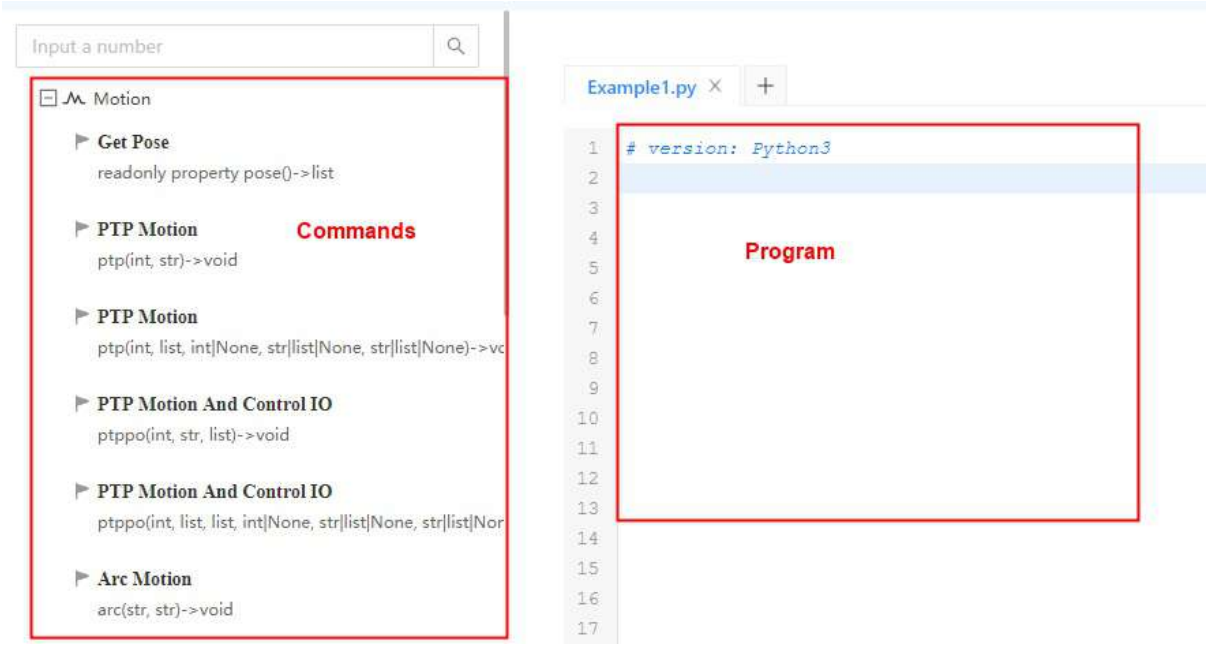
The download path is *www.dobot.cc/downloadcenter/dobot-m1.html#most-download*.

**Procedure**

**Step 1** Click **Script** to enter Script interface.

**Step 2** Write a script.

You can double-click the interface to add it to program area, the corresponding interface will be displayed on the script programming panel, as shown below.



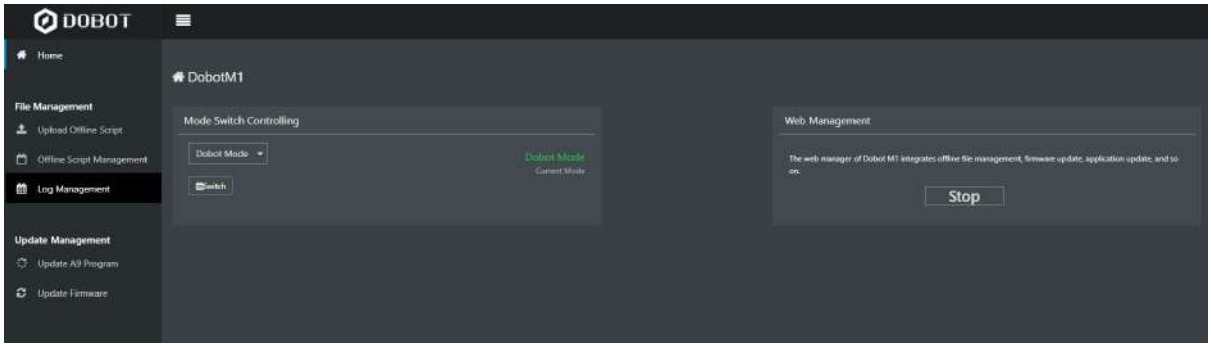

**Step 3** Click **Save** and **Start** on the **Script** page.Dobot M1 will move according to the script file.

> [!NOTE]
>
> If you need to operate this file in offline mode, you can click **Export** to save it and then upload it on the Web Management page for running. For details, please see 3.2.4.1 Managing Offline File.

# 3.3.4 Web Management

# 3.3.4.1 Web Management

The web management of Dobot M1 integrates offline file management, firmware update, and application update, which is used to upload the offline files, make the Dobot M1 in the offline mode, and update the applications. Click **Web Management** to enter the interface as shown below.



# 3.3.4.2 Managing Offline File

You can upload the scripts, the blockly programs, or the saved points lists that have been saved on a local PC to Dobot M1 using the web management, to perform offline operation.

After making the Dobot M1 in the offline mode using the web management, the Dobot M1 will be disconnected from the DobotStudio2020.

> [!NOTE]
>
> If you need to operate a tray program in the offline mode, please make sure that the tray parameters are set on the network condition. Namely, when setting the tray parameters, you must use the network cable to connect the Dobot M1 and the PC. Otherwise, the tray information cannot be loaded into the Dobot M1 system.

**Prerequisites**

- The Dobot M1 has been connected to DobotStudio2020 over a network cable.
- You have saved the scripts, the blockly programs, or the saved points lists.
- The Dobot M1 has been connected to an emergency stop switch.

**Application Scenario**

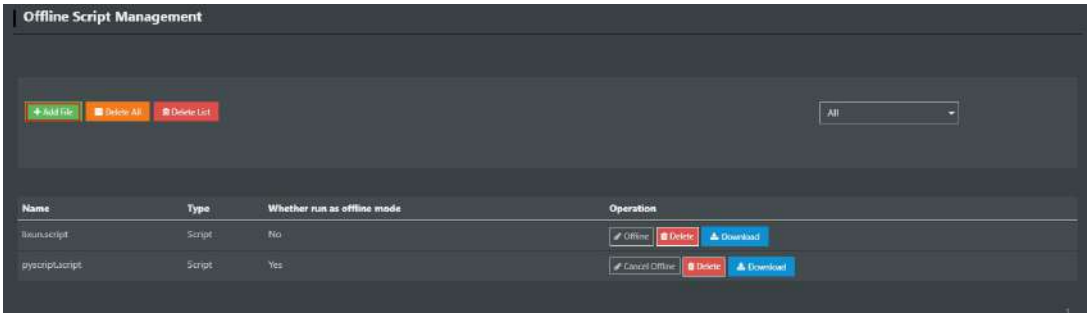
If the Dobot M1 needs to be running in the offline mode, please use the web management.

**Procedure**

**Step 1** Select the right IP address of Dobot M1 on the DobotStudio2020.

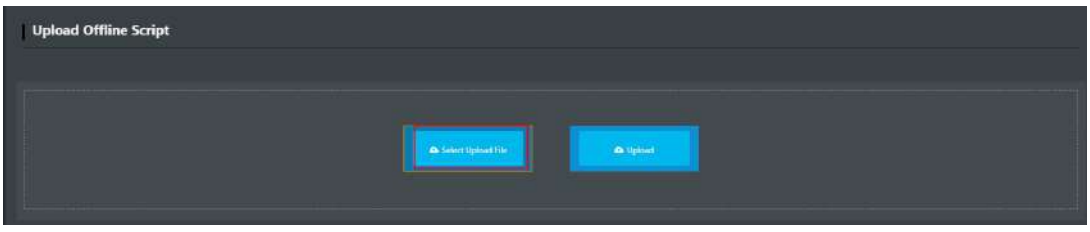
**Step 2** Click **Web Management >Offline Script Management** to enter the Offline Script Management interface .

**Step 3** click **Add File** to enter Upload Offline Script interface.



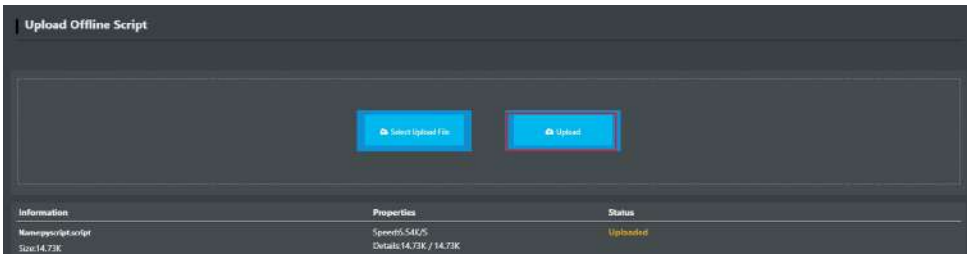

**Step 4** Click **Select Upload File** to select the file to be uploaded.

Only support the files, of which the suffixes are .playback, .blockly, and .script.

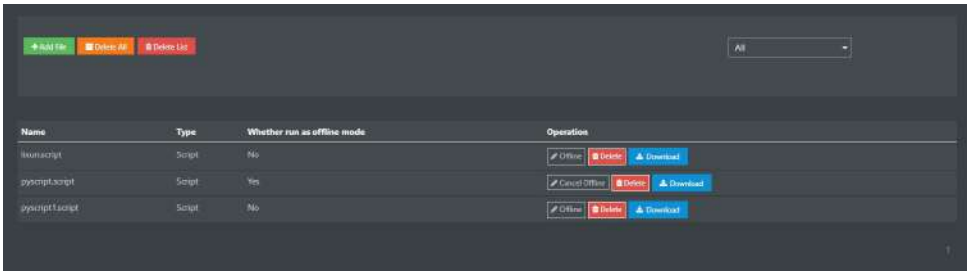



**Step 5**

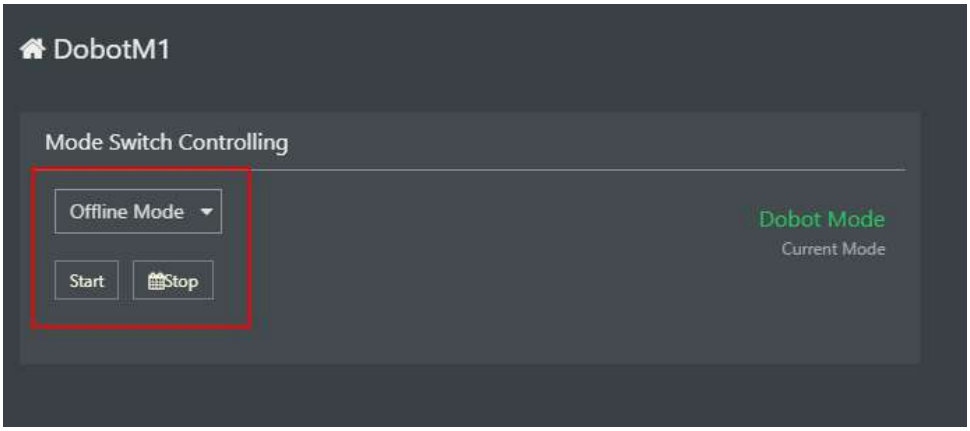
Click **Update**, you can view the update state.




**Step 6** Select the uploaded files that need to be offline executed on the **Offline Script Management** page, and click **Offline** to make the files in the offline mode, as shown below.




**Step 7** Select **Offline Mode** from the drop-down list on the **Mode Switch Controlling** pane of the **Home** page, and click **Start** to make the Dobot M1 in the **Offline Mode** status, as shown below. The Dobot M1 is disconnected from the M1Studio, and Dobot M1 can be running according to the files that are in the offline mode.



# 3.3.4.3 Upgrading Firmware

When the firmware or other applications need to be upgraded, you can use the web management to upgrade the firmware or the application. For example, if you want to use the collision detection or IP address forced configuration function, you can upgrade the Dobot M1 firmware.

> [!DANGER]
>
> When updating a firmware, please do not perform any other operation on the Dobot M1 or power off it, to avoid Dobot M1 in an abnormal condition. Otherwise, it will be vulnerable to injury the device or the person.

**Prerequisites**

- You have connected the Dobot M1 to DobotStudio2020, and the IP addresses of Dobot M1 and the PC must be in the same network segment.
- You have powered on the Dobot M1.
- You have connected the Dobot M1 to an emergency stop switch.
- You have obtained the latest A9 firmware **a9_app-00*xx*.tar**.*xx* indicates the firmware version, please replace it based on site requirements.

Download path：https://cn.dobot.cc/downloadcenter/dobot-m1.html?sub_cat=119#sub-download.

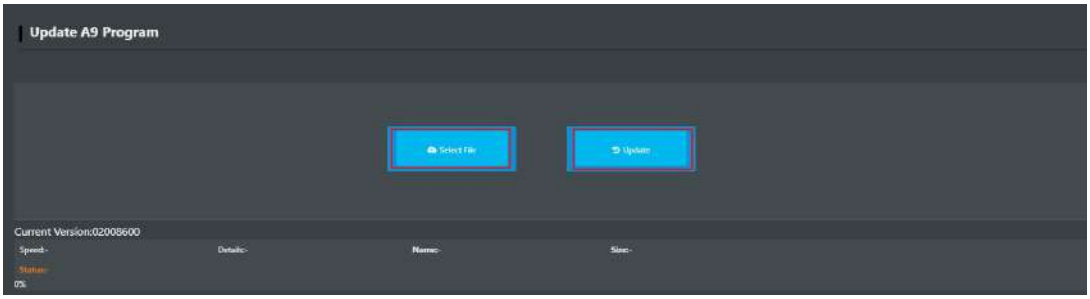
- You have obtained the latest M1Stuido.

  Download path：https://cn.dobot.cc/downloadcenter/dobot-m1.html?sub_cat=119#sub-download.

**Procedure**

**Step 1** Select the right IP address on the DobotStudio2020.

**Step 2** Click **Web Management > Update A9 Program** to enter the interface, as shown below.



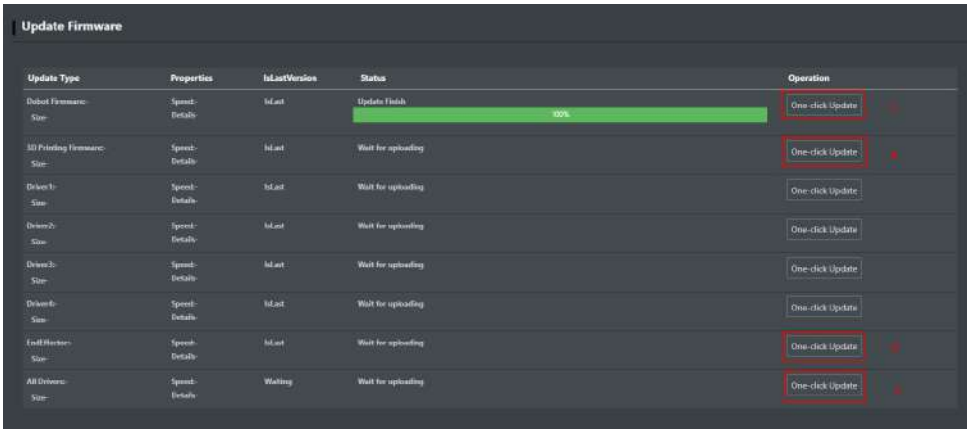

**Step 3** Click **Select File** to upload the obtained firmware and click **Update**.

> [!WARNING]
>
> If the version of the obtained firmware is **0058** or later, please click **Update** again after **Status** is **100%**, to avoid update failure.

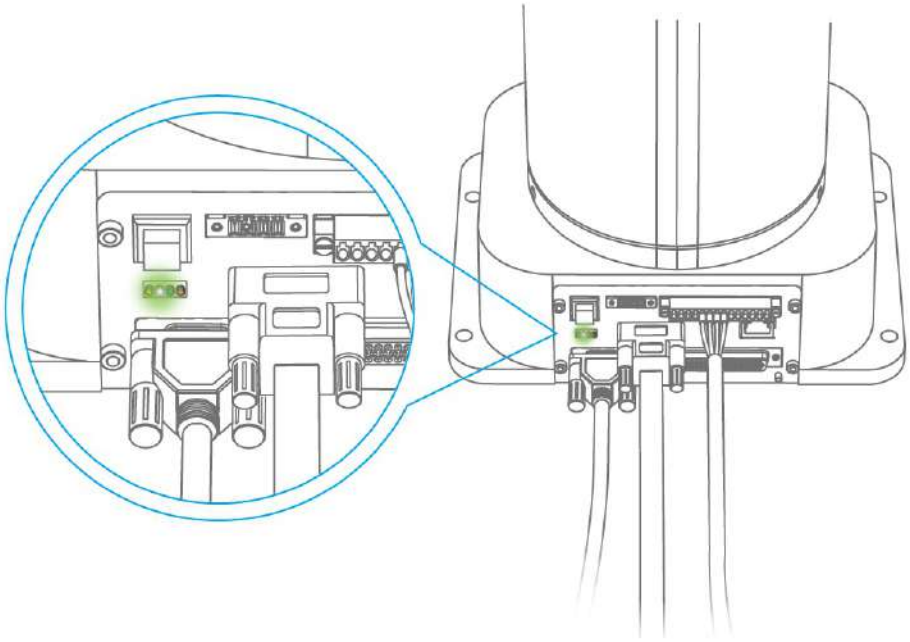
**Step 4** Restart Dobot M1 after the update is finished.

**Step 5** Click **Update Firmware** .

**Step 6** Select firmware exactly as the order shown below and click **One-click Update**.

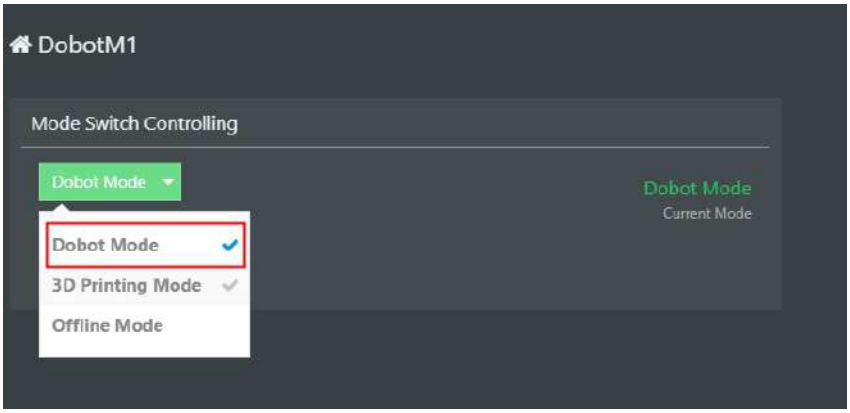



You can view the process of the firmware upgrade. If the progress bar is 100% and the green LED indicator on the base of Dobot M1 is blinking, the update is completed, as shown below. And then the next firmware update can be started.



**Step 7** Select **Dobot Mode** from the drop-down list on the **Mode Switch Controlling** pane of the **Home** page and Click **Switch**.

After upgrading the 3D printing firmware, Dobot M1 will switch to 3D printing mode. You need to switch the **3D Pprinting mode**to **Dobot Mode**, otherwise Dobot M1 cannot be used normally. As shown below.



# 3.3.5 I/O Assistant

**Prerequisites**

- Dobot M1 has been connected to DobotStudio2020.
- Dobot M1 has been connected to emergency stop switch.
- The air pump has been installed (If you use an air pump to debug I/O interface).

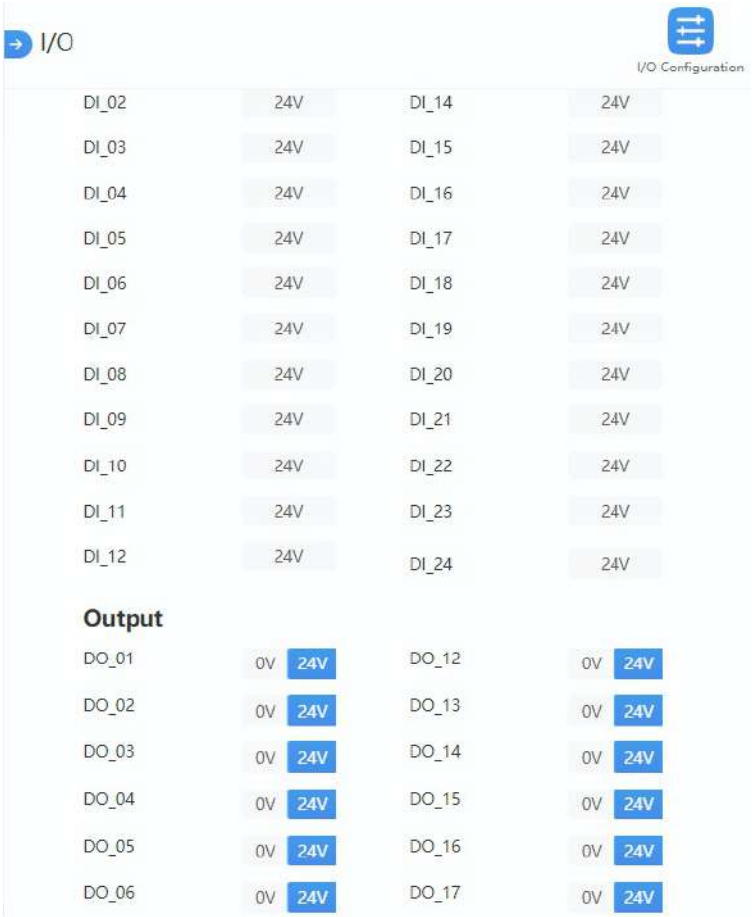
**Application Scenario**

The end effectors such as gripper, suction cup need to work with the air pump. The air pump can be controlled by the I/O interface. You can verify the I/O interface and the air pump over **I/O Assistant**.

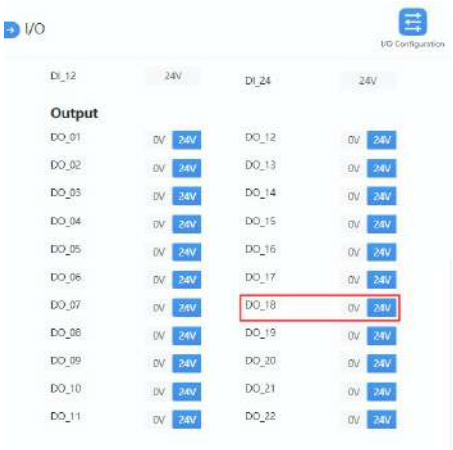
**Procedure**

If the air pump is connected to the base I/O interface, the outputs used are DOUT17, DOUT18 respectively. The DOUT18 output controls the startup and shutoff of the air pump.

**Step 1** Click **Teach&PlayBack > I/O** to enter I/O interface.




**Step 2** Click **24V** of **DO_18** on the **Output** interface, as shown below, The air pump is humming, which indicates that the air pump is enabled. The working state depends on the air pump. Please judge based on site requirements.




**Step 3** Click **0V** of **DO_18** on the **Output** interface. The air pump is not humming, which indicates that the air pump is disabled.

# 3.3.6 Collision Detection

Collision detection is mainly used for reducing the impact on the robot arm, to avoid damage to the robot arm or external equipment. If the collision detection is activated, the robot arm will stop running automatically when the robot arm hit an obstacle.

**Prerequisites**

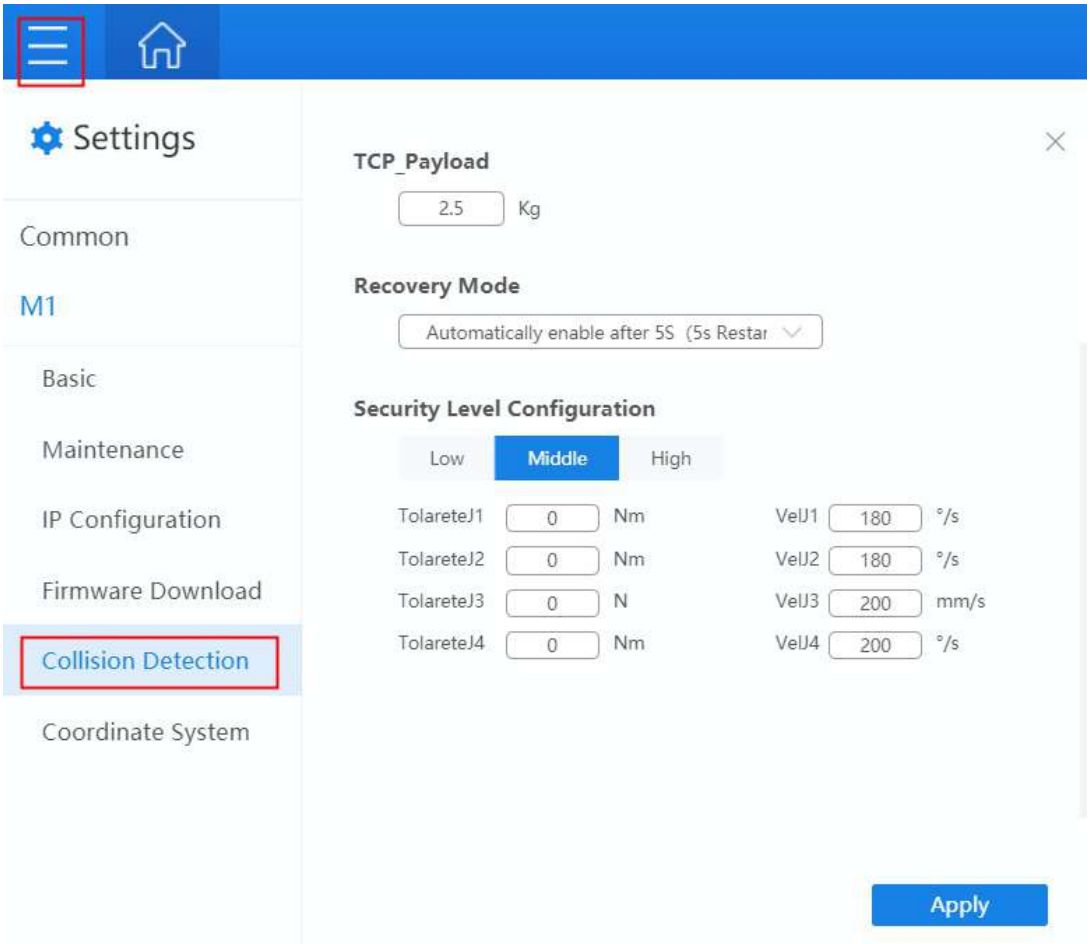
- The Dobot M1 has been powered on and connected to DobotStudio2020 .
- The Dobot M1 has been connected to an emergency stop switch.

**Procedure**

**Step 1** Obtain kinetic parameters of each joint of robot arm, to avoid false collision detection

(1) Remove the fixture on the robot arm and make sure that there are no obstacles within the workspace. If there is no fixture on the robot arm, please skip this step.

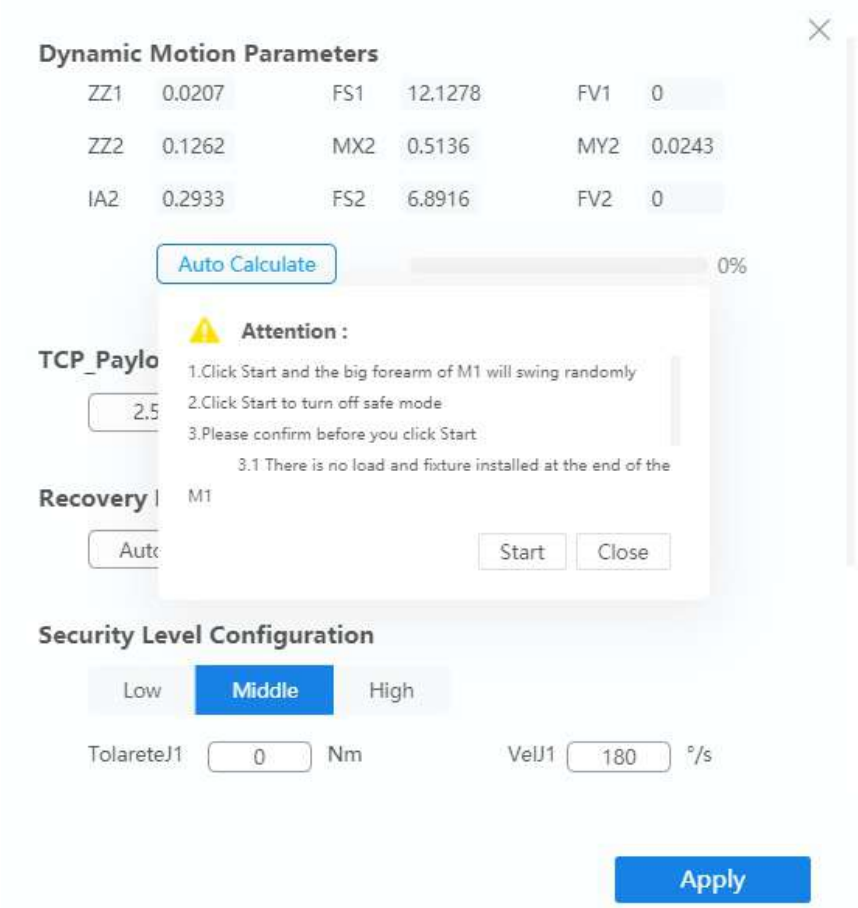
(2) Click **Menu > Settings > M1 > Collision Detection** to enter the collision detection interface.



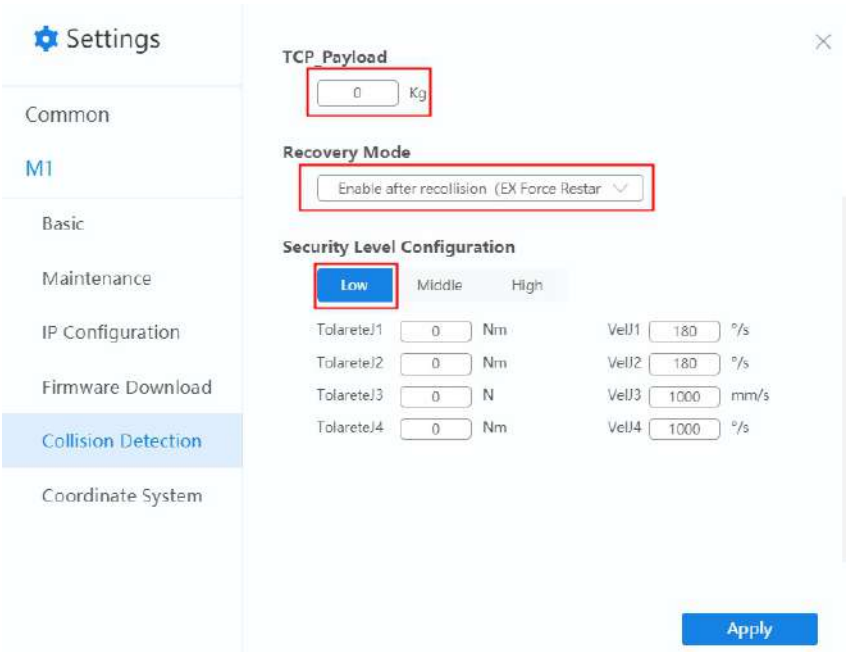

(3) Click **Auto Calculate** to obtain kinetic parameters.

A safety prompt will be shown after clicking **Auto Calculate**, please read the safety precautions and click **Start**.

The robot will move in random directions for a while. After the process bar is 100%, the kinetic parameters will be obtained automatically.




(4) Set **Security Level Configuration** to **Low**, **Tcp_PayLoad** to **0**, select **EX Force Restart** , as shown below.




(5) Click **Apply**, The kinetic parameters are saved in the main controller.

> [!WARNING]
>
> - When obtaining the kinetic parameters, the robot arm will move in random directions for a while. Please make sure that there are no obstacles within the workspace, to avoid obtaining wrong kinetic parameters because of collision.

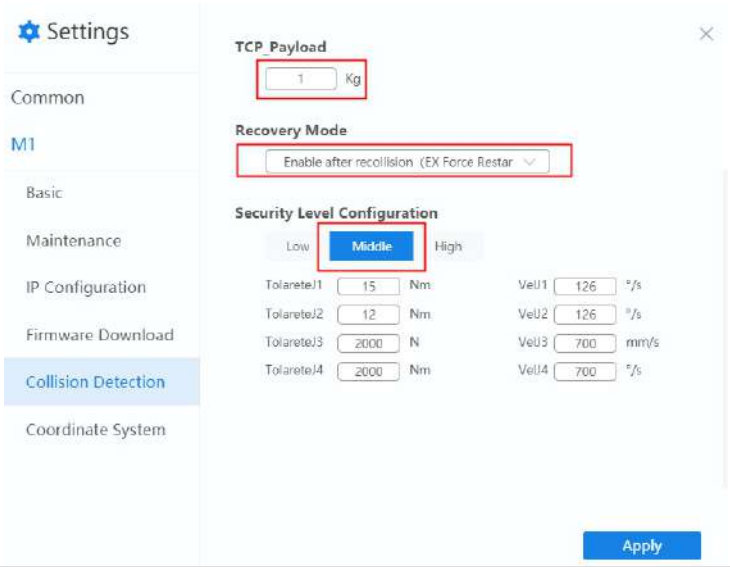
> - During robot arm running, please do not perform any other operation on the Dobot M1 or power off it, to avoid the Dobot M1 in an abnormal condition
> - When the unexpected occurs during robot arm running, please click **Stop** on the **Dynamic Motion Parameters** panel.

**Step 2** Mount the fixture on the Dobot M1 based on site requirements.

**Step 3** Set the payload based on site requirements. In this topic, we set to 1Kg.

**Step 4** Set **Recovery Mode** to **EX Force Restart**.

**Step 5** Set **Security Level Configuration** to **Middle**.




| Level | Description |
| --- | --- |
| Low | No restrict. Namely, deactivate collision detection |
| Middle | Activate collision detection and the speed decreases 30% |
| High | Activate collision detection and the speed decreases 50% |

**Step 5** Click **Apply**.

**Step 6** Restart the Dobot M1 to make the collision detection parameters effective.

**Step 7** Verify whether the collision detection is effective.

When the Dobot M1 hit an obstacle during running, the Dobot M1 stops running automatically. If you touch the Dobot M1, it runs again, indicating that the setting is effective.

# 3.3.7 Example of Safeguard Stop

The safeguard stop function is used to suspend robot movement in a safe mode. Namely, when the safeguard stop function is enabled and the safeguard input signals are triggered, the robot stops immediately and enters into pause state. In Dobot M1 system, the DI7, DI8 and DI9 interfaces on the DB62 expansion board are safeguard inputs, which can connect to safety doors, light curtains, safety floor mats, etc.

> [!WARNING]
>
> - The safeguard input interfaces are available only when the safeguard stop function is enabled. If not enabled, the DI7, DI8 and DI9 on the DB62 expansion board will be used as the common digital input interfaces.
> - In normal use of Dobot M1, please DO NOT enable safeguard stop function at will. Otherwise, the scripts cannot be run.

- **Immediate Recovery Mode**

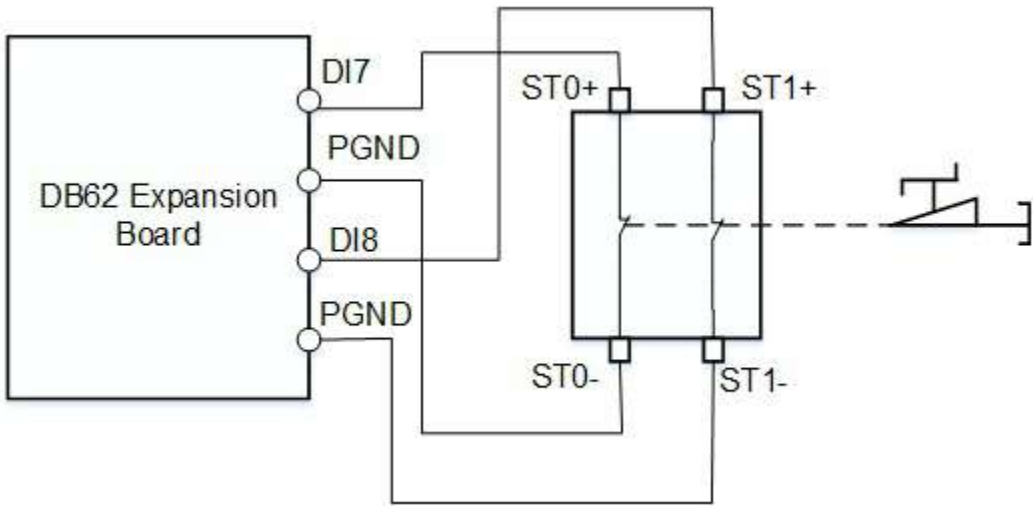
Only DI7 and DI8 are safety interfaces in the immediate recovery mode. The robot stops moving after disconnecting DI7 and DI8, and the robot resumes its movement after reconnecting DI7 and DI8.

**Prerequisites**

- The Dobot M1 has been connected to an emergency stop switch.
- The Dobot M1 has been connected to the DB62 expansion board.

**Procedure**

**Step 1** Connect the DB62 expansion board to the door switch, as shown below.

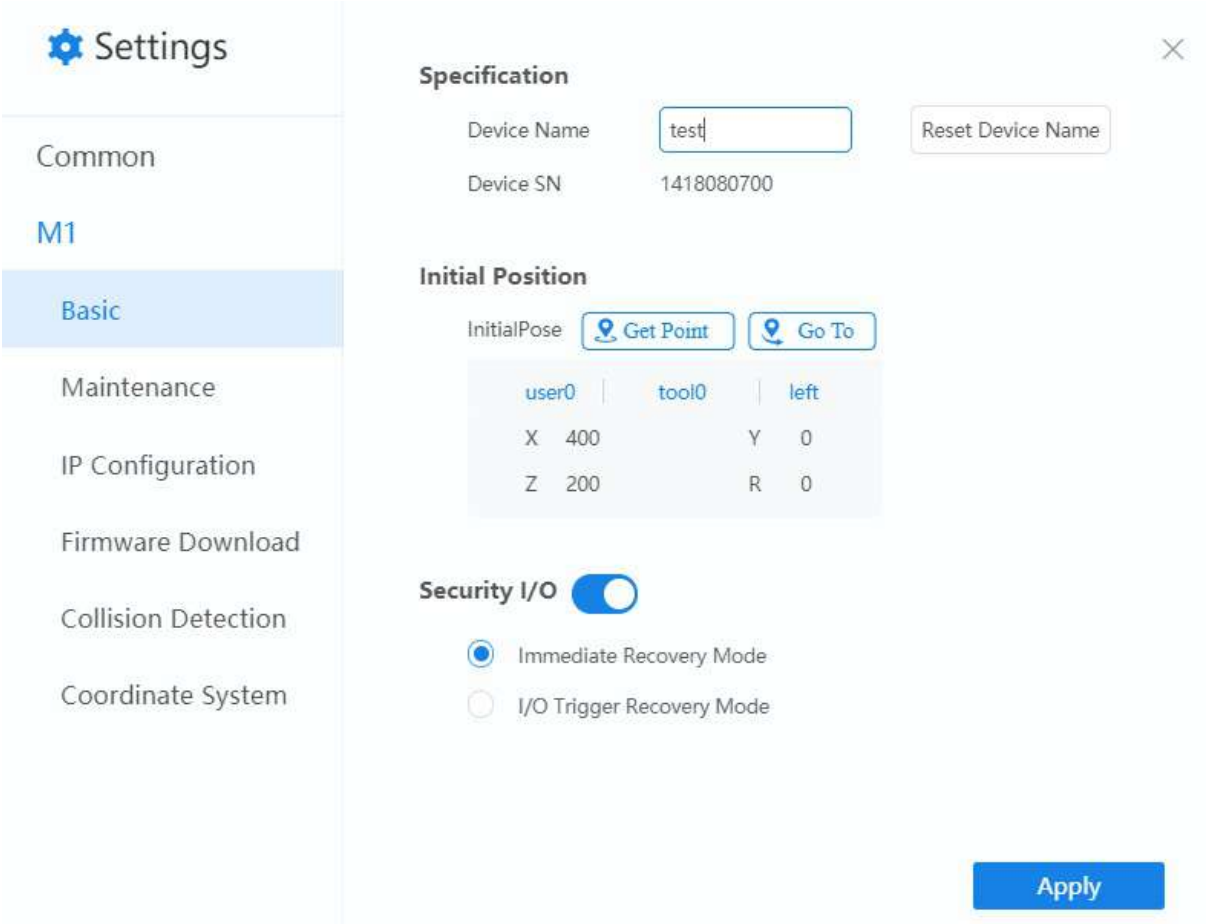



When the safeguard stop function is enabled, either or both of **DI7** and **DI8** interfaces are disconnected from **PGND** interface, the safeguard stop will be triggered and Dobot M1 will stop moving without clearing the queue in the Dobot M1 system. Only when both **DI7** and **DI8** interfaces are connected to **PGND** interface, the Dobot M1 will move again.

**Step 2** Power on Dobot M1 and connect it to DobotStudio2020.

**Step 3** Click **Menu > Settings > M1 > Basic** to enter to the basic setting page.

**Step 4** Select **Immediate Recovery Mode** on the DobotStudio2020, and click **Apply**.




**Step 5** Make the Dobot M1 in the running status, and open the door switch. The Dobot M1 will stop immediately and enter into pause state. If you close the door switch, the Dobot M1 resumes movement.

- **I/O Trigger Recovery Mode**

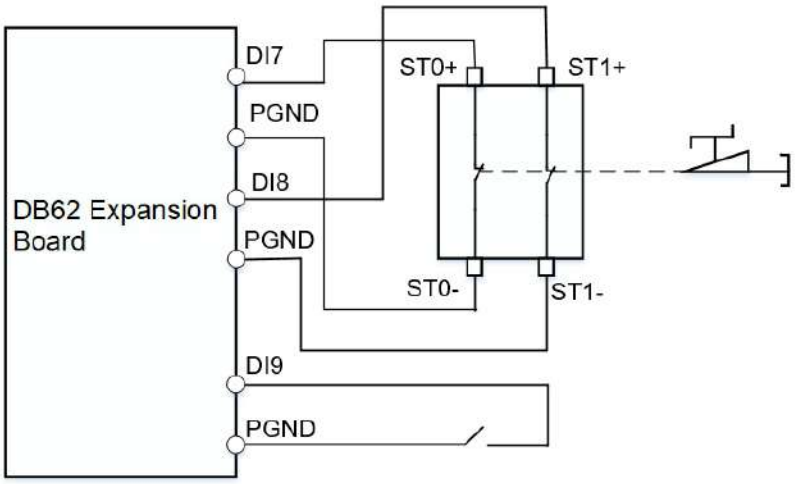
DI7, DI8, and DI9 are safety interfaces in the I/O trigger recovery mode. The robot stops moving after disconnecting DI7 and DI8. After reconnecting DI7 and DI8, the robot will not resume its movement. Only when DI9 is low level, the robot will resume its movement.

**Prerequisites**

- The Dobot M1 has been connected to an emergency stop switch.
- The Dobot M1 has been connected to the DB62 expansion board.

**Procedure**

**Step 1** Connect the DB62 expansion board to the door switch, and connect DI9 to a normally open switch, as shown below.

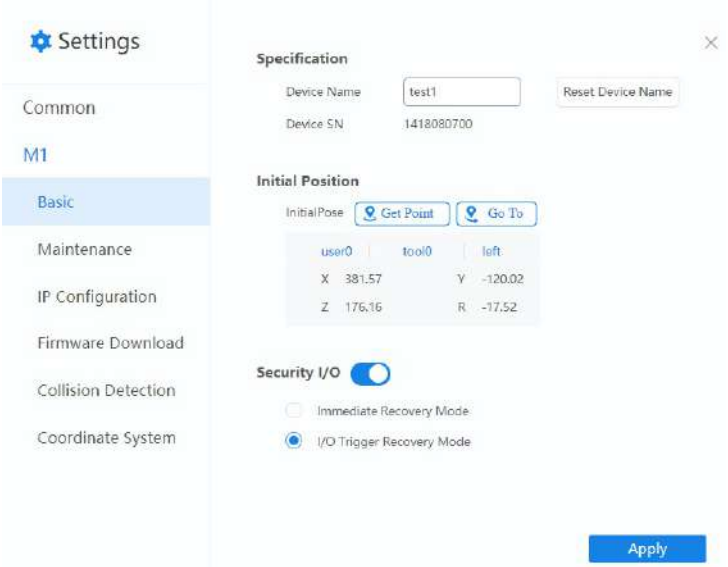



When the safeguard stop function is enabled, either or both of **DI7** and **DI8** interfaces are disconnected from **PGND** interface, the safeguard stop will be triggered and Dobot M1 will stop moving without clearing the queue in the Dobot M1 system. Only when both **DI7** and **DI8** interfaces are connected to **PGND** interface, and DI9 connected to low level, the Dobot M1 will move again.

**Step 2** Power on Dobot M1 and connect it to DobotStudio2020.

**Step 3** Click **Menu > Settings > M1 > Basic** to enter to the basic setting page.

**Step 4** Select **I/O Trigger Recovery Mode** on the DobotStudio2020 , and click **Apply**.




**Step 4** Make the Dobot M1 in the running status, and open the door switch. The Dobot M1 will stop immediately and enter into pause state. If you close the door switch and the normally open switch, the Dobot M1 resumes movement.

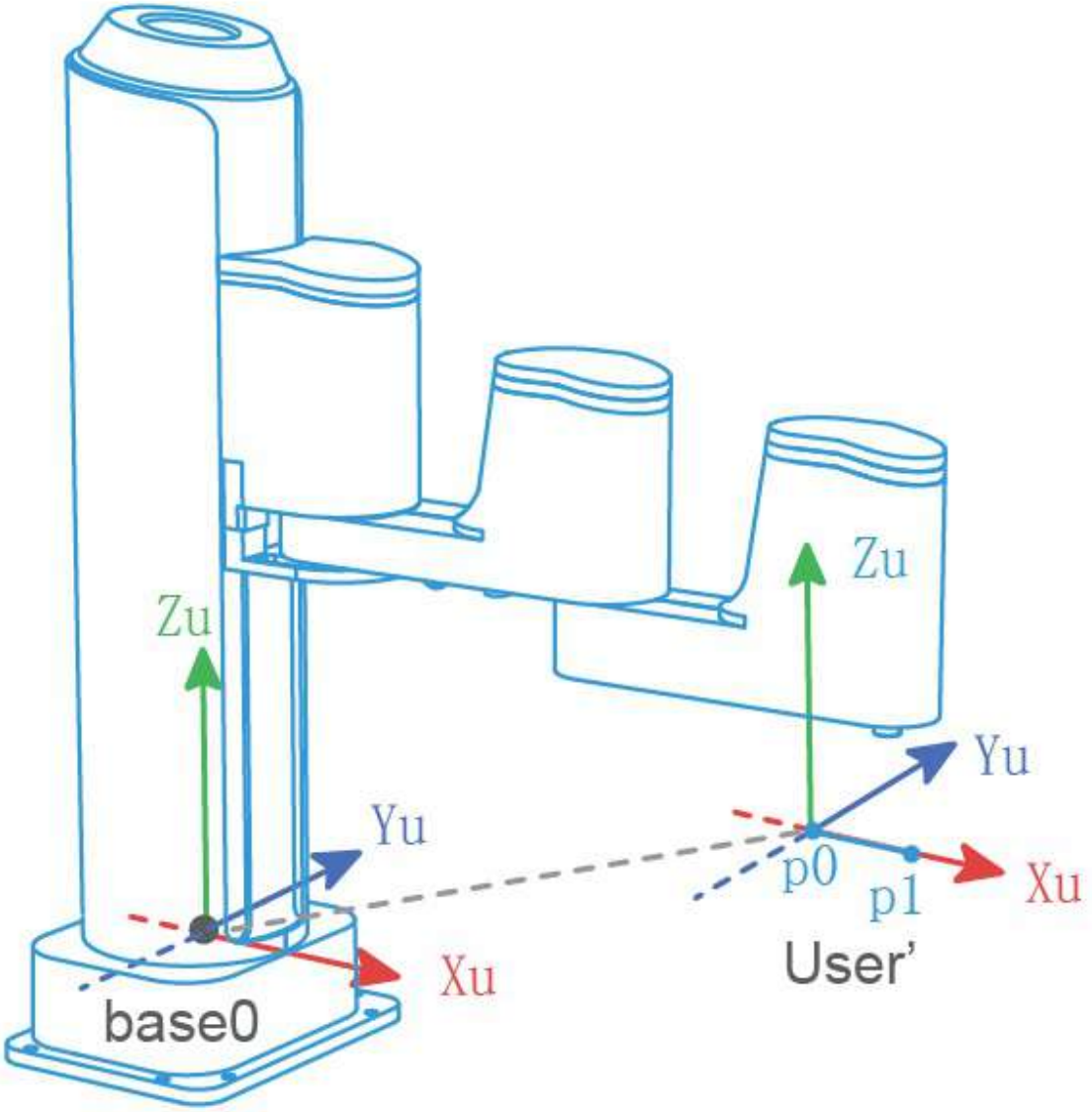
# 3.3.8 Coordinate System

# 3.3.8.1 User Coordinate System

The User coordinate system is a movable coordinate system which is used for representing equipment like fixtures, workbenches. The origin and the orientations of axes can be defined based on site requirements, to measure point data within the workspace and arrange tasks conveniently. When the position of workpiece is changed or a robot program needs to be reused in multiple processing systems of the same type, you can create coordinate systems on the workpiece to simplify programming.

> [!NOTE]
>
> When creating a User coordinate system, please make sure that the reference coordinate system is the base coordinate system. There, User0 is defined as the base coordinate system by default and cannot be changed.

User coordinate system is created by two-point calibration method: Move the robot to two points **p0(x1, y1, z1)** and **p1(x2, y2, z2)**. **p0** is defined as the origin and the line from **p0** to **p1** is defined as the positive direction of X-axis. And then the Y-axis and Z-axis can be defined based on the right-handed rule, as shown below




Take the establishment of User 1 coordinate system as an example.

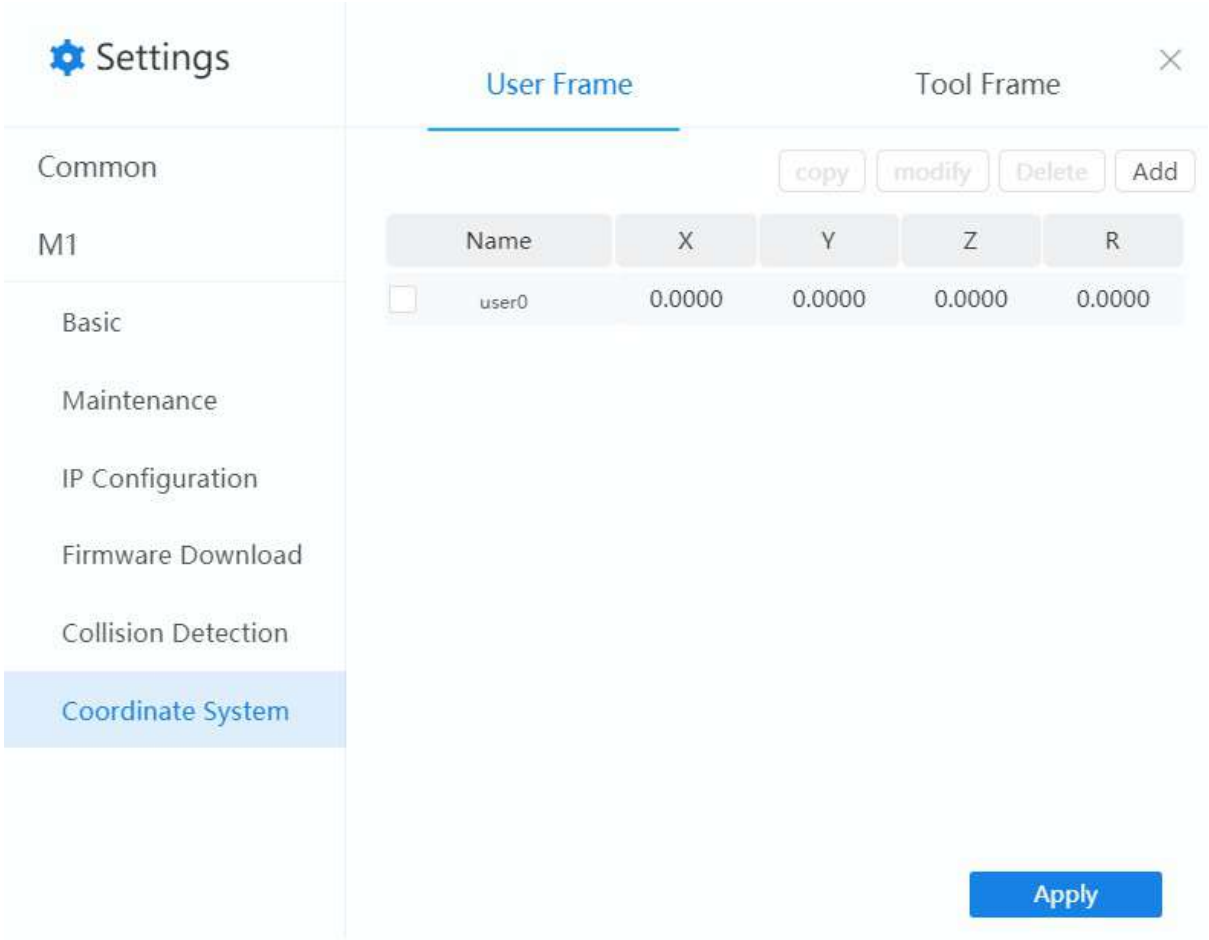
**Prerequisites**

- The Dobot M1 has been connected to DobotStudio2020.
- The Dobot M1 has been connected to an emergency stop switch.

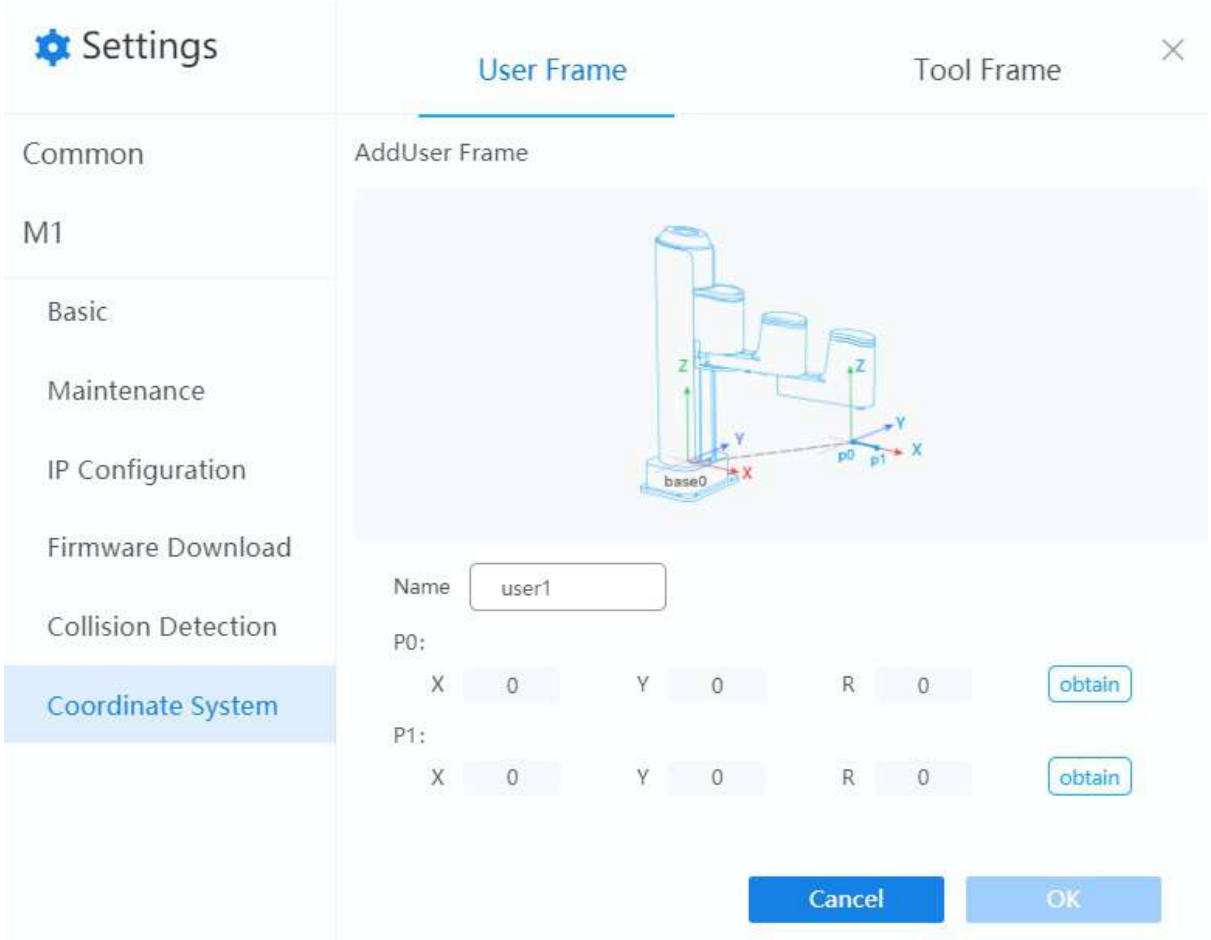
**Procedure**

**Step 1** Click **Menu > Settings > M1 > Coordinate System** on the DobotStudio2020 page.

The coordinate system page is displayed, as shown below.



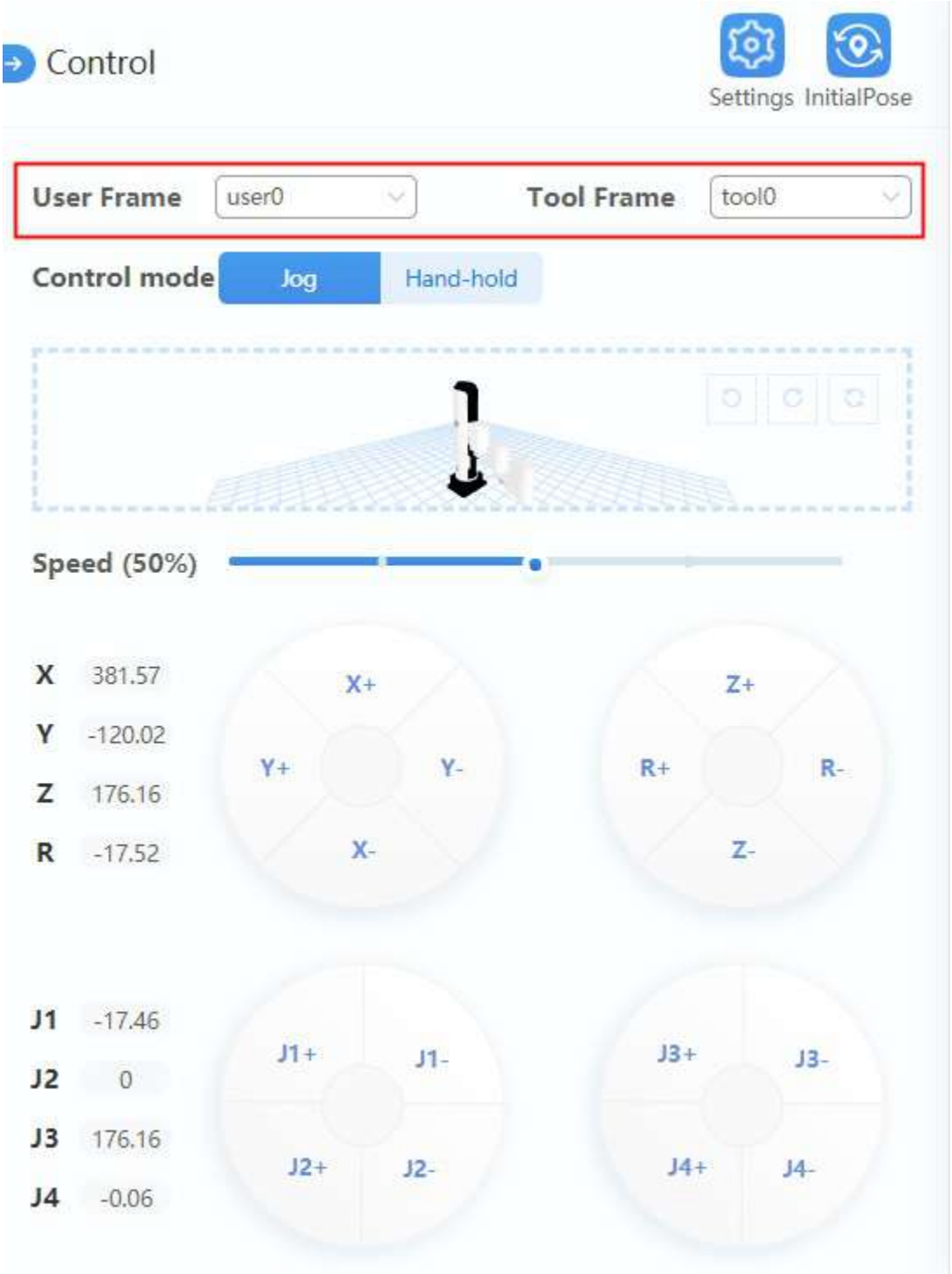

**Step 2** Click **Add** on the User Frame tab to create User coordinate system, as shown below.




**Step 3** Input User coordinate system name on the User Frame tab. In this topic, we set to **user1**.

**Step 4** Jog robot to the point P0 and click **obtain** on the **P0**: panel of the User Frame tab.

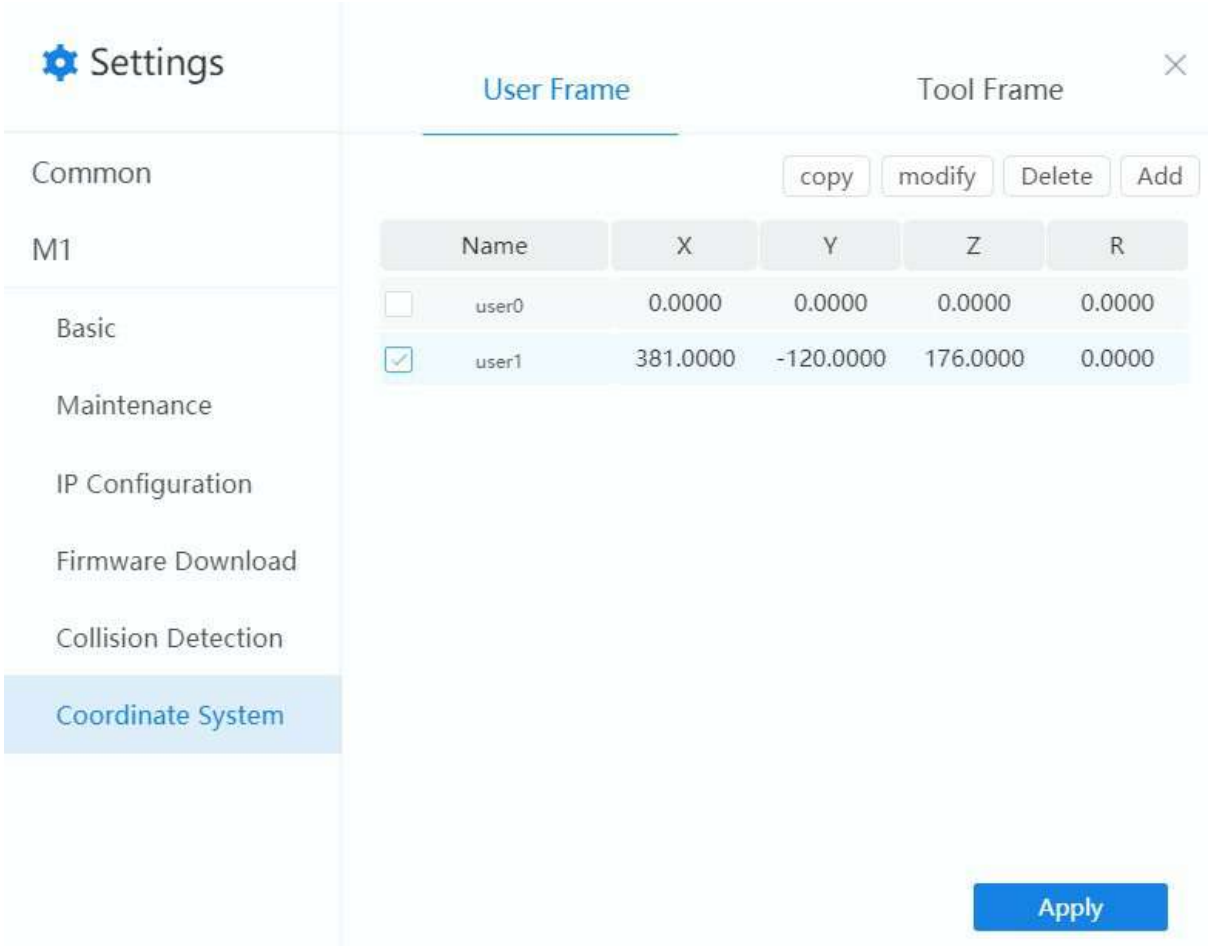
> [!NOTE]
>
> When creating a User coordinate system, please make sure that the reference coordinate system is the base coordinate system. Namely ,the User coordinate system is user0 and the Tool coordinate system is tool0 when jogging robot, as shown below.




**Step 5** Jog robot to the point P1 and click **obtain** on the **P1:** panel of the User Frame tab.

**Step 6** Click **OK**.

The created User coordinate system will display on the User Frame tab, as shown below.



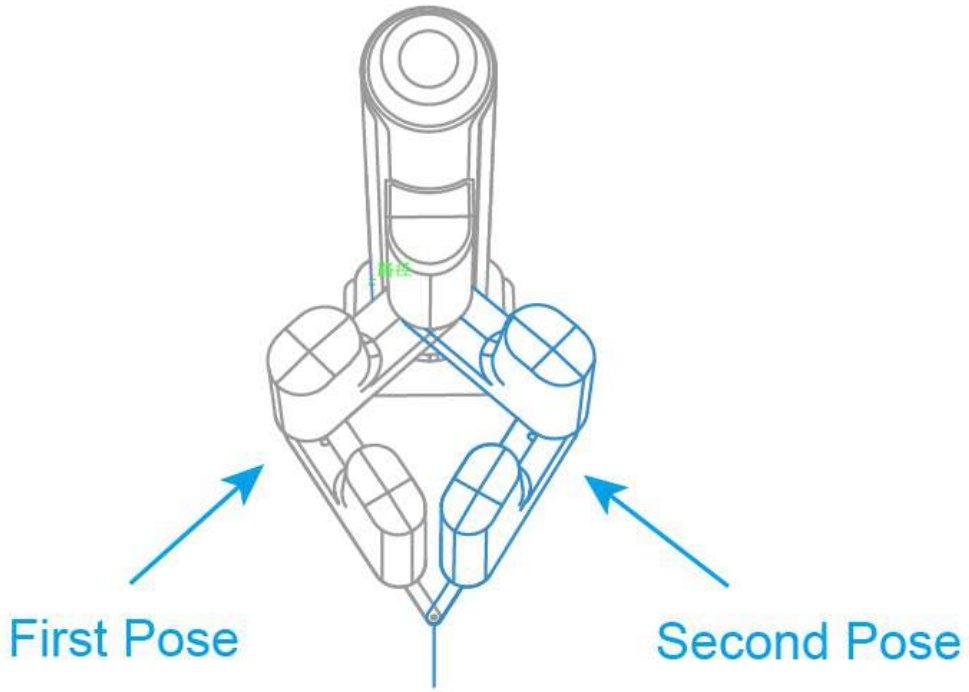

**Step 7** Select **user1** and click **Apply**. Now you can use the User 1 coordinate system for teaching and programming.

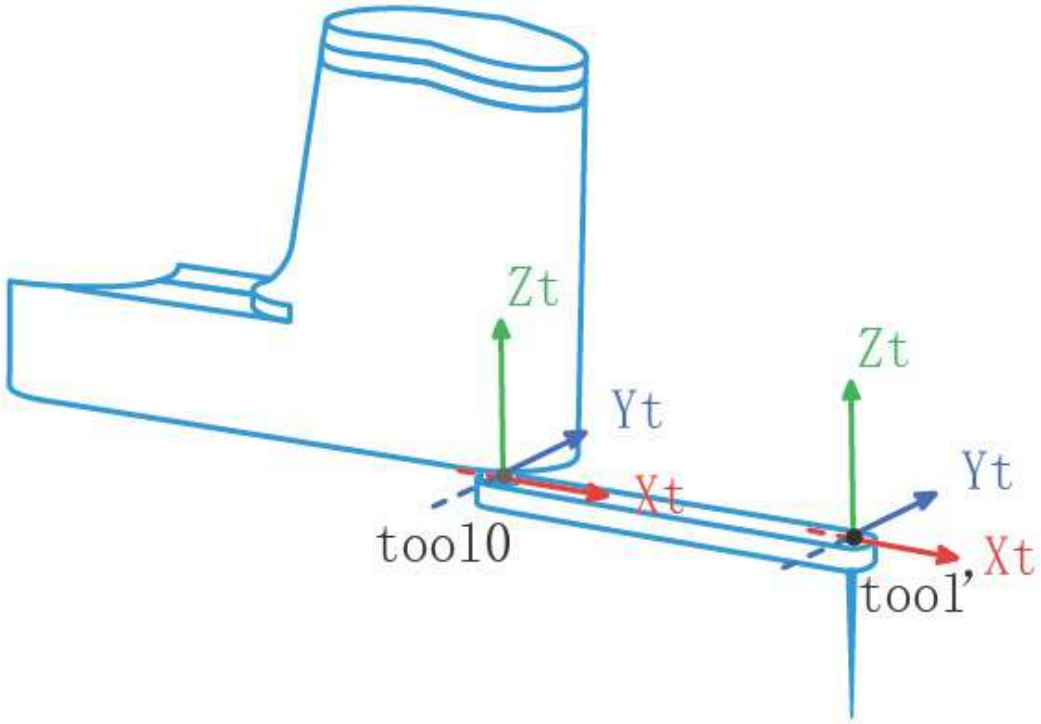
# 3.3.8.2 Tool Coordinate System

Tool coordinate system is the coordinate system that defines the distance and rotation angle of the offset, of which the origin and orientations vary with the position and attitude of the workpiece located at the robot flange. When an end effector such as welding gun, gripper is mounted on the robot, the Tool coordinate system is required for programming and operating a robot. For example, you can use multiple grippers to carry multiple workpieces simultaneously to improve the efficiency by setting each gripper to a Tool coordinate system.

> [!NOTE]
>
> When creating a Tool coordinate system, please make sure that the reference coordinate system is the base coordinate system. There, Tool0 is defined as the base coordinate system by default and cannot be changed.

Tool coordinate system of SCARA robot is created by two-point calibration method: After an end effector is mounted, please adjust the direction of this end effector to make the TCP (Tool Center Point) align with the same point (reference point) in two different directions, for obtaining the position offset to generate a Tool coordinate system, as shown below.






Take the establishment of Tool 1 coordinate system as an example.

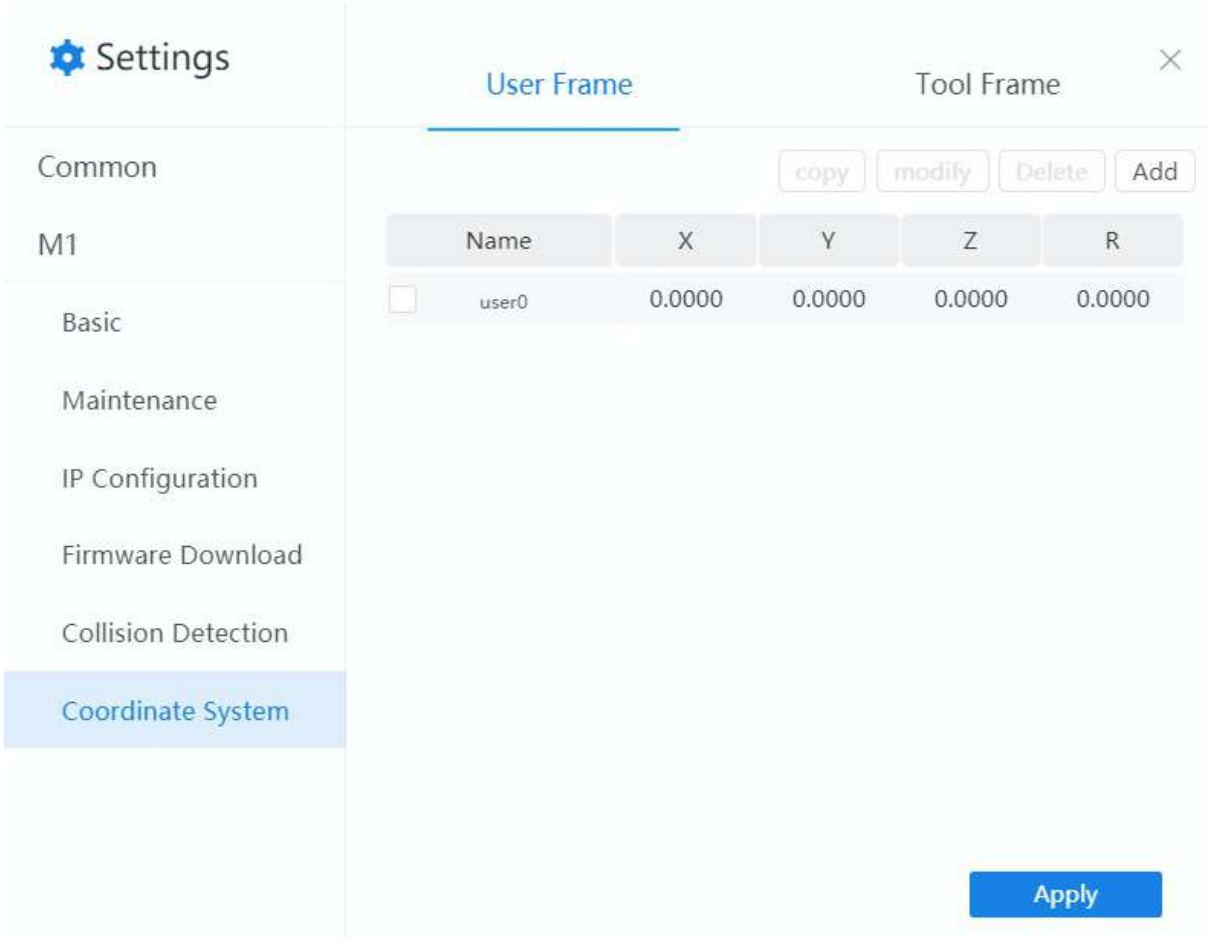
**Prerequisites**

- The Dobot M1 has been connected to DobotStudio2020.
- The Dobot M1 has been connected to an emergency stop switch.
- The end-effetor has been mounted on the Dobot M1.

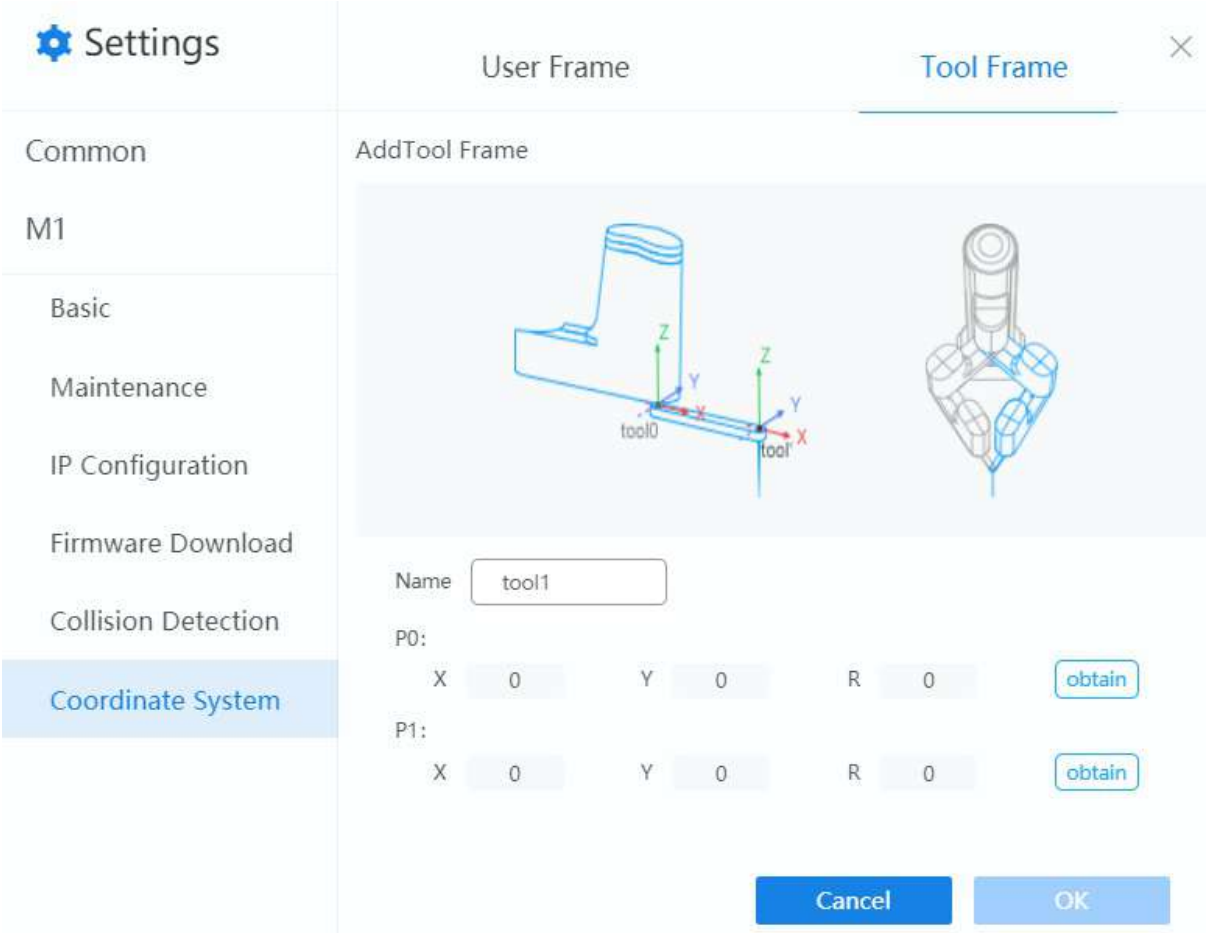
Procedure**

**Step 1** Click **Menu > Settings > M1 > Coordinate System** on the DobotStudio2020 page.

The coordinate system page is displayed, as shown below.



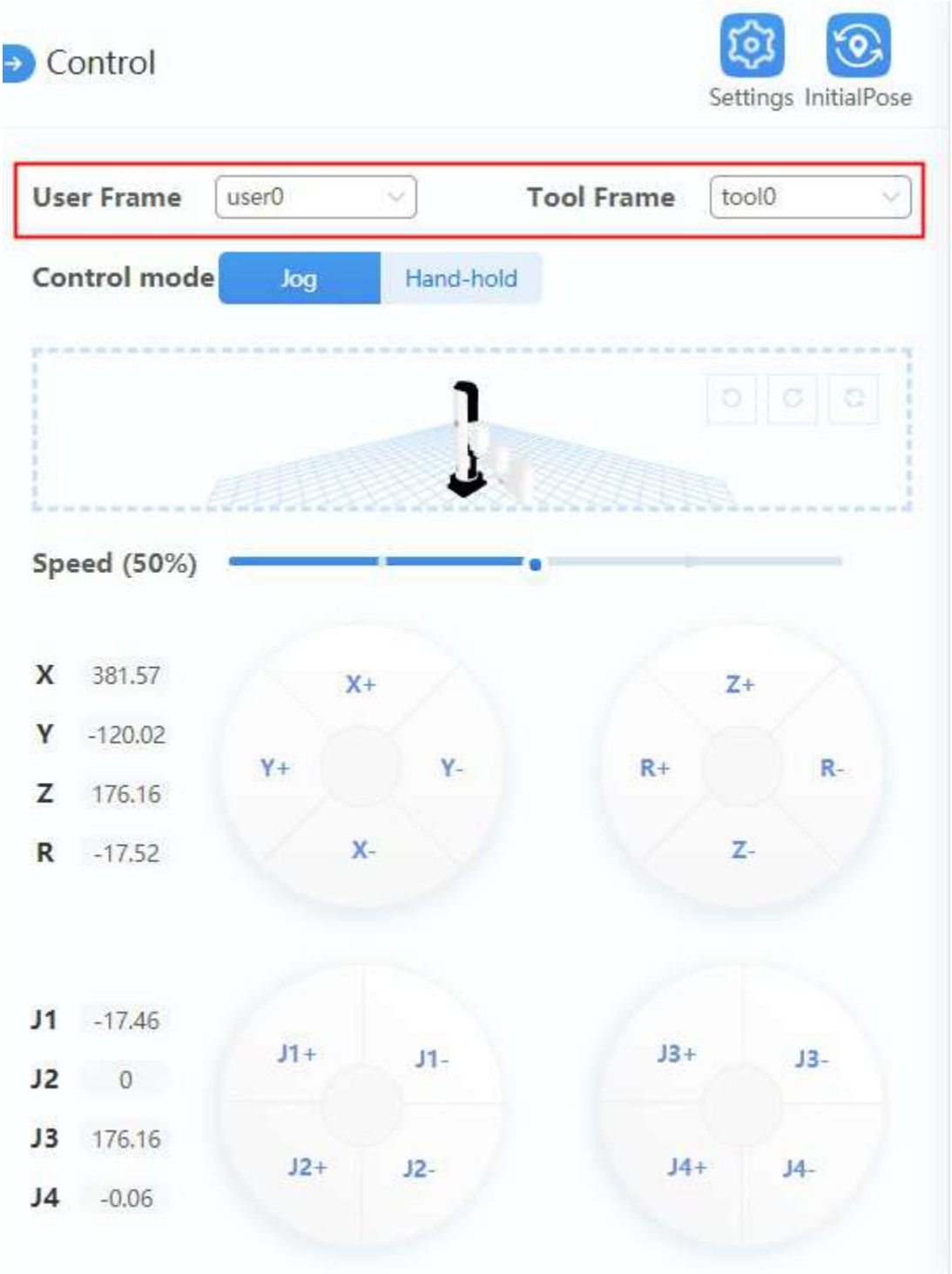

**Step 2** Click Add to create Tool coordinate system on the Tool Frame tab, as shown below.




**Step 3** Input Tool coordinate system name on the Tool Frame tab. In this topic, we set to **tool1**.

**Step 4** Jog robot to the point P0 in the first direction and click obtain on the **P0:** panel of the Tool Frame tab.

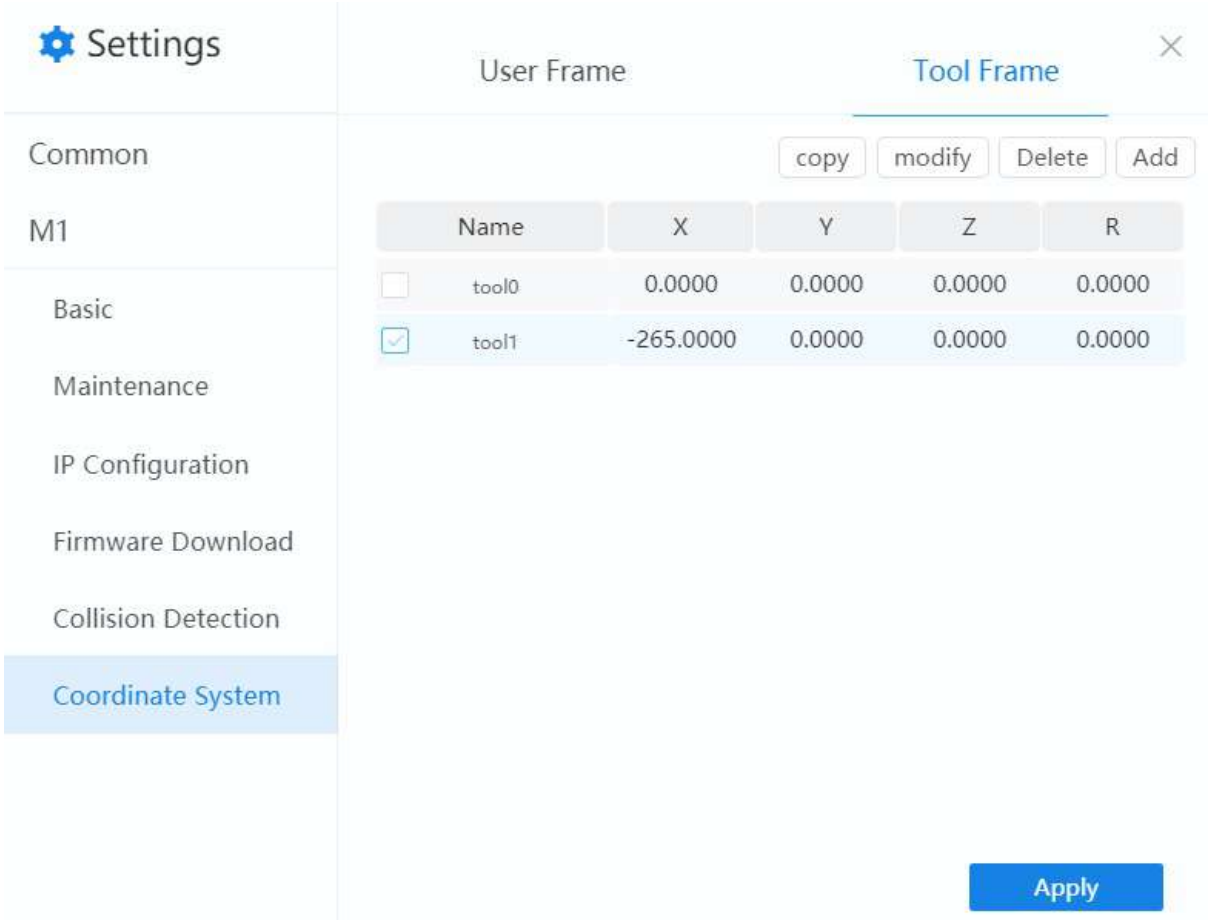
> [!NOTE]
>
> When creating a Tool coordinate system, please make sure that the reference coordinate system is the base coordinate system. Namely ,the User coordinate system is user0 and the Tool coordinate system is tool0 when jogging robot, as shown below.




**Step 5** Jog robot to the point P0 in the second direction and click obtain on the **P1:** panel of the Tool Frame tab.

**Step 6** Click **OK**.

The created Tool coordinate system will display on the Tool Frame tab, as shown below.




**Step 7** Select **tool1** and click **Apply**. Now you can use the Tool 1 coordinate system for teaching and programming.

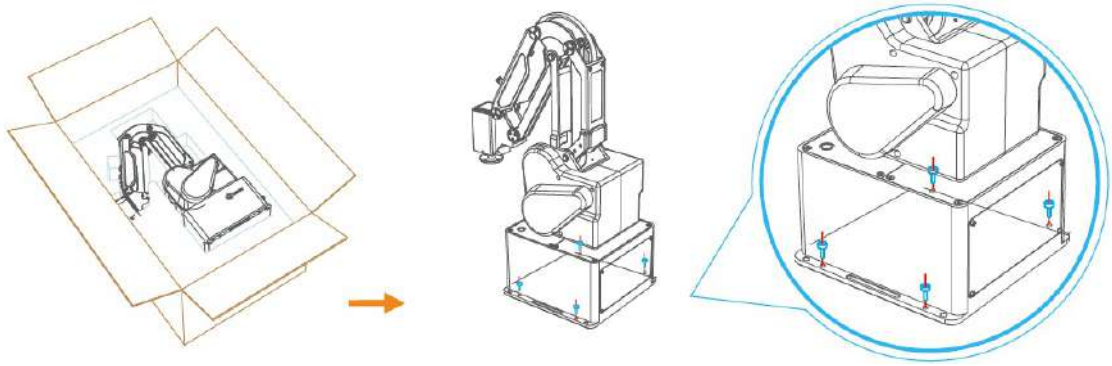
# 4. Dobot MG400

# 4.1 Overview

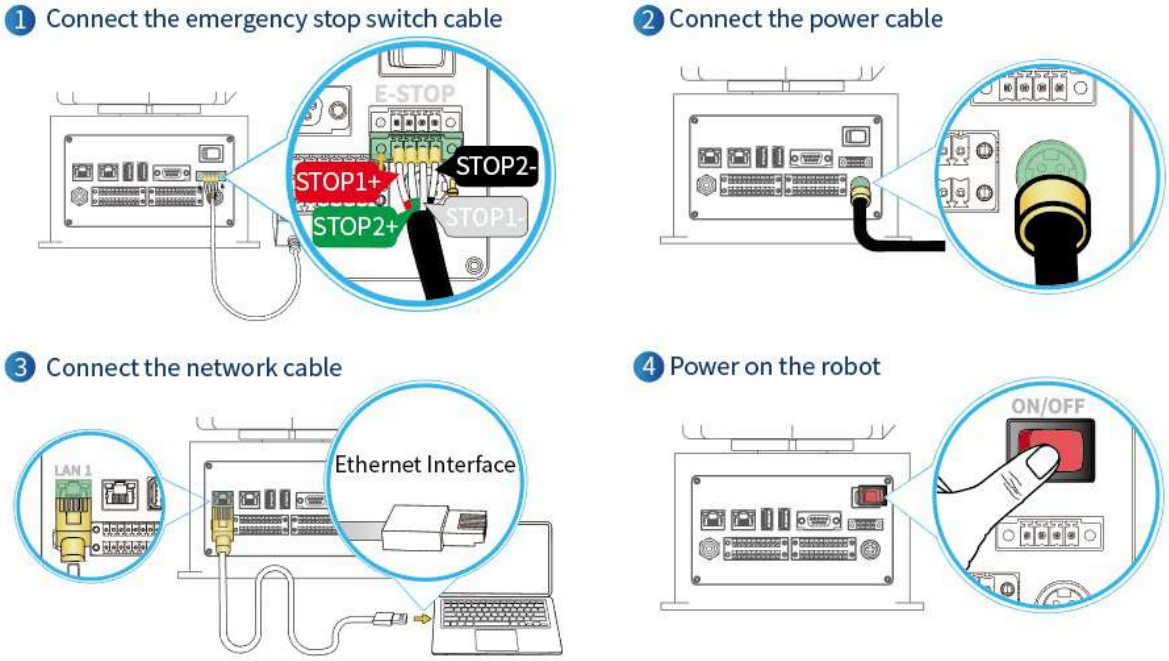
You can control Dobot MG400 through the DobotStudio2020, and perform teaching and playback, Blockly, Script and other operations on the DobotStudio2020.

# 4.2 DobotStudio2020 Connection

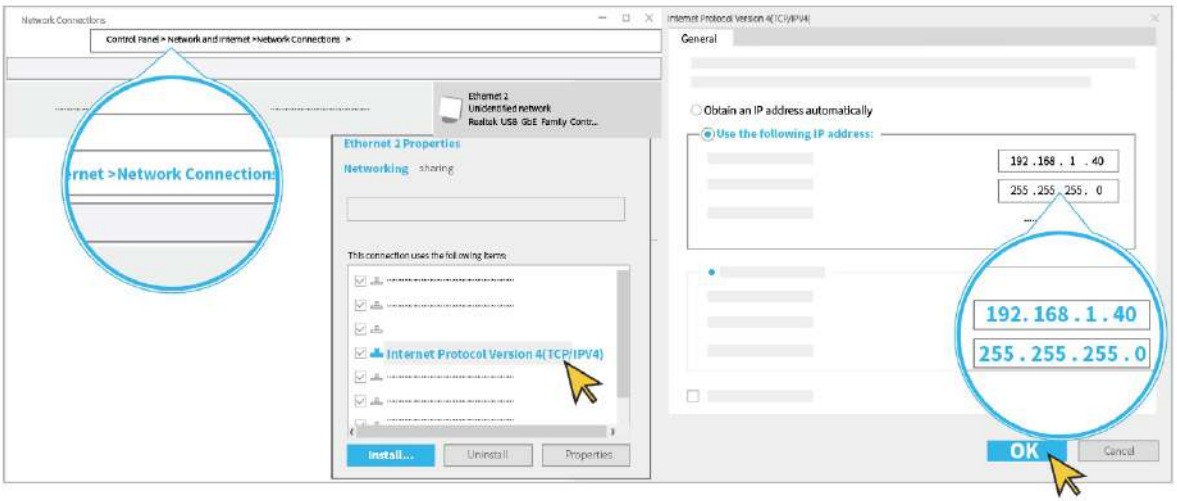
**Step 1** Fix the base of robot on a platform with four M5 screws.



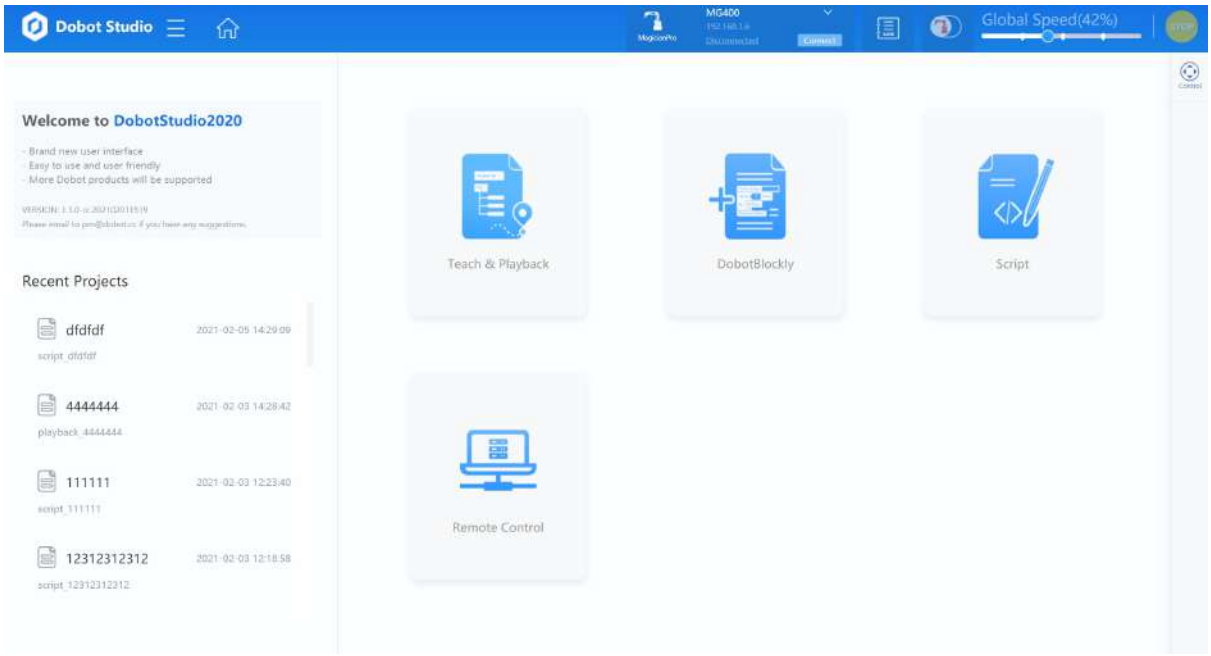
**Step 2** Connect the required cables and power on the robot.




**Step 3** Set the computer’s IP address to communicate with the robot. The robot's IP address is 192. 168.1.6, you need to set the computer's IP address on the same network segment without conflict.



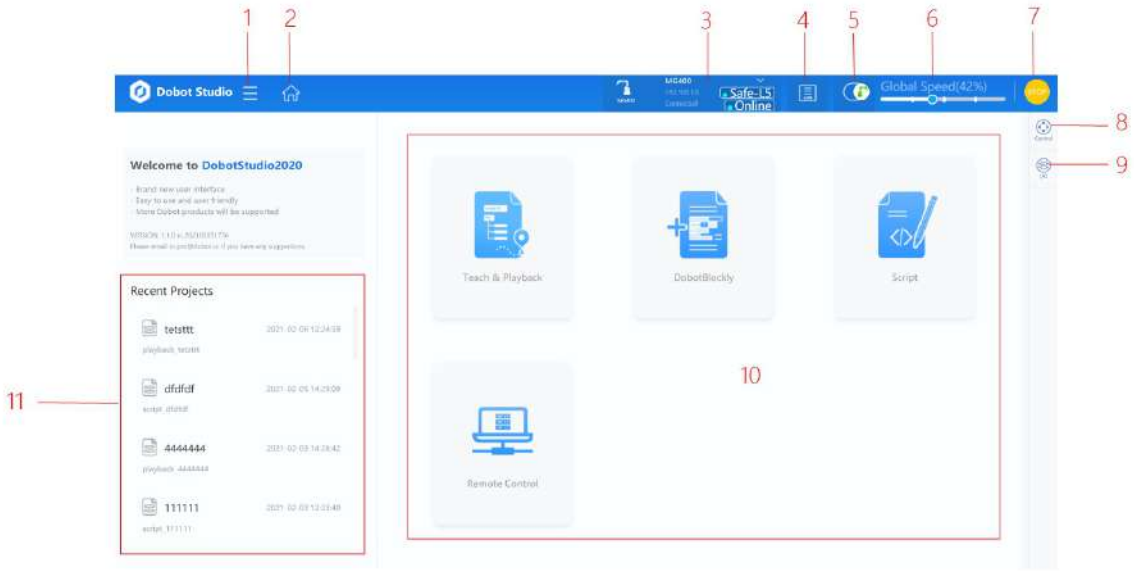

**Step 4** Launch DobotStudio2020, select the right device' IP address and click **Connect**.



# 4.3 Function Description

# 4.3.1 Overview

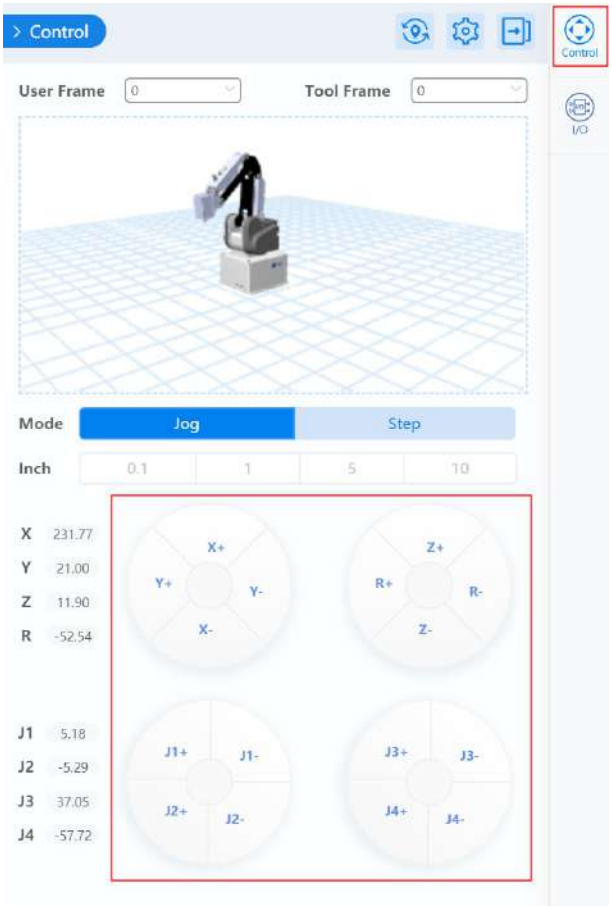
In this section, we will introduce the basic functions in DobotStudio2020.




| No | Description |
| --- | --- |
| 1 | Setting menu, including basic setting, IP setting, Jump parameters setting, etc. |
| 2 | Home page |
| 3 | Status, including connection status, collision detection level, device status |
| 4 | Alarm log. You can click it to check the alarm log. |
| 5 | Enable or disable the MG400 motor |
| 6 | Adjust the jogging speed and the running speed before running a program<br>• Actual jogging speed = the maximum jogging speed * global speed rate<br>• Actual running speed= the maximum running speed * global speed rate * the set velocity rate in the velocity command |
| 7 | Software emergency stop button |
| 8 | Control page , you can jog MG400 by clicking coordinate system buttons on this page |
| 9 | Points page. After jogging the MG400 to a point, you can click **Add** on this page |
| 10 | Main interface. You can select teach&playback, Blockly, or script module to write a program. Also, you can set remote control on the remote control page |
| 11 | Recent projects. You can open the recent project directly in this section |

# 4.3.2 Basic Operation

After connecting to the MG400, you can jog robot by clicking coordinate system buttons on the **Control** page.




You can jog MG400 in Cartesian coordinate system after the MG400 is in the enabled status.

- Click **X+**, **X-** and MG400 will move along X-axis in a negative or positive direction.
- Click **Y+**, **Y-** and MG400 will move along Y-axis in a negative or positive direction.
- Click **Z+**, **Z-** and MG400 will move along Z-axis in a negative or positive direction.
- Click **R+**, **R-** and MG400 will rotate along R-axis in a positive or negative direction.

If you control MG400 in different User or Tool Cartesian coordinate system, you need to set them advanced on the **Settings > Coordinate System** page and select the right coordinate system to control MG400.

Also, you can jog MG400 in Joint Coordinate system after the MG400 is in the enabled status. .

- Click **J1+**, **J1-** and control the base motor to rotate in the negative or positive direction.
- Click **J2+**, **J2-** and control the Rear Arm motor to rotate in the negative or positive direction.
- Click **J3+**, **J3-** and control the Forearm motor to rotate in the negative or positive direction.
- Click **J4+**, **J4-** and control the end-effector to rotate in the negative or positive direction.

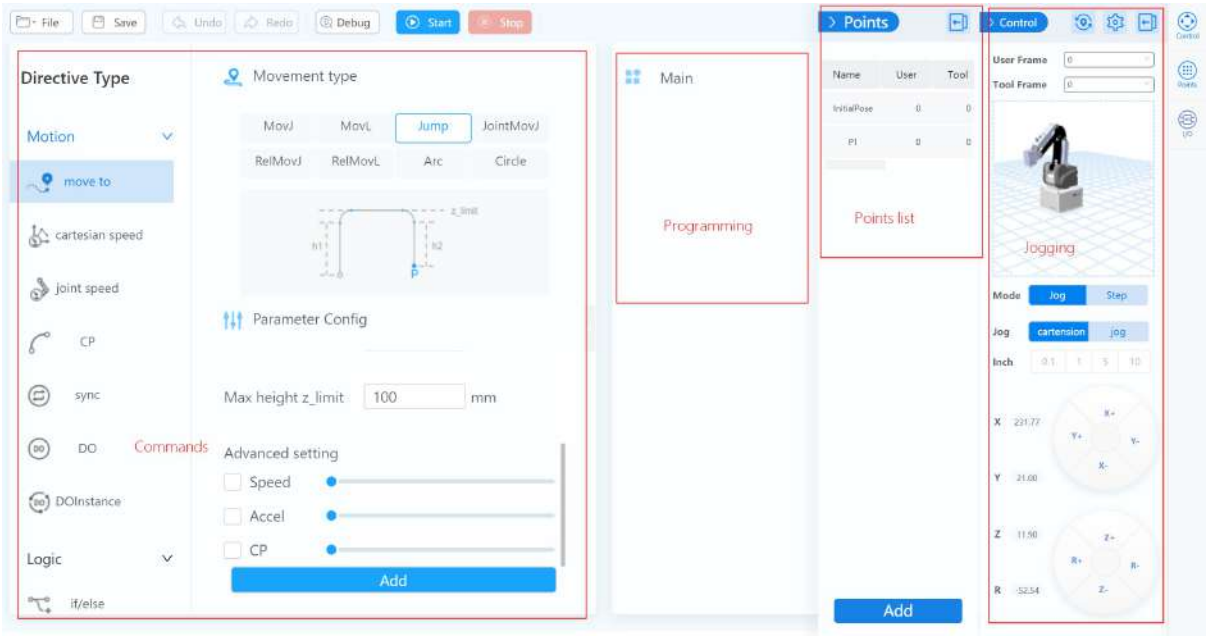
When you want to fine-tune the MG400 by clicking the coordinate system buttons, you can select the right step in the Step mode. The step supports 0.1, 1, 5,and 10. In Cartesian coordinate system, the step unit is mm, and in Joint coordinate system, the step unit is °.

In addition to jogging MG400 on the **Control** page, you can also click  to make MG400 move to the initial pose and click  to enter the **Settings** page.

There is another way to make robot jog: Press the hand-teaching button on the forearm and drag the forearm to a point and then press the button again.

# 4.3.3 Teaching and Playback

The teach and Playback function supports tree programming teaching, and users can perform teach and Playback through tree programming. According to different program instructions, the interface displays different parameter settings. Currently, Teaching and playback only supports single thread. The instruction description is shown in the table below.




| Instruction | Description | Setting |
|---|---|---|
| move to | Motion instructions. Move to a certain point or follow a certain trajectory | Choose different motion mode and set motion parameters. The motion mode supports MovJ, MovL, JointMovJ, Jump, RelMovJ, RelMovL, Arc, Circle<br>All motion modes support advanced settings, including speed, acceleration, CP, indicating that these parameters are valid only in the current mode. In addition, MovJ and MovL motion mode support setting digital output status during moving |
| cartesian speed | Cartesian speed instruction | Set the Cartesian speed acceleration ratio. This command is valid only when the motion mode is MovL, RelMovL, Jump, Arc, Circle |
| joint speed | Joint speed instruction | Set the Joint speed acceleration ratio. This command is valid only when the motion mode is MovJ, JointMovJ, RelMovJ |
| CP | Continuous path instruction | Set Continuous path ratio, the vale range is 0 - 100. This command is invalid when the motion mode is Jump |
| sync | Synchronization instruction. Whether to stop at this point | None |
| DO | I/O instruction | Set the status of digital output port (Queue command) |
| DOInstance | I/O instruction | Set the status of digital output port (Immediate command) |
| | Logical instruction. Set Judgment | |

| | | |
|---|---|---|
| if/else | conditions to trigger robot movement | Logical processing based on digital input or variable setting |
| wait | Waiting instructions. The time can be set to make the robot arm wait | Set the wait time |
| loop | Loop instruction | Set the number of loop |
| set variable | set variable. You can create and set variable values | Create a new variable, and assign an initial value to the variable according to the variable type |

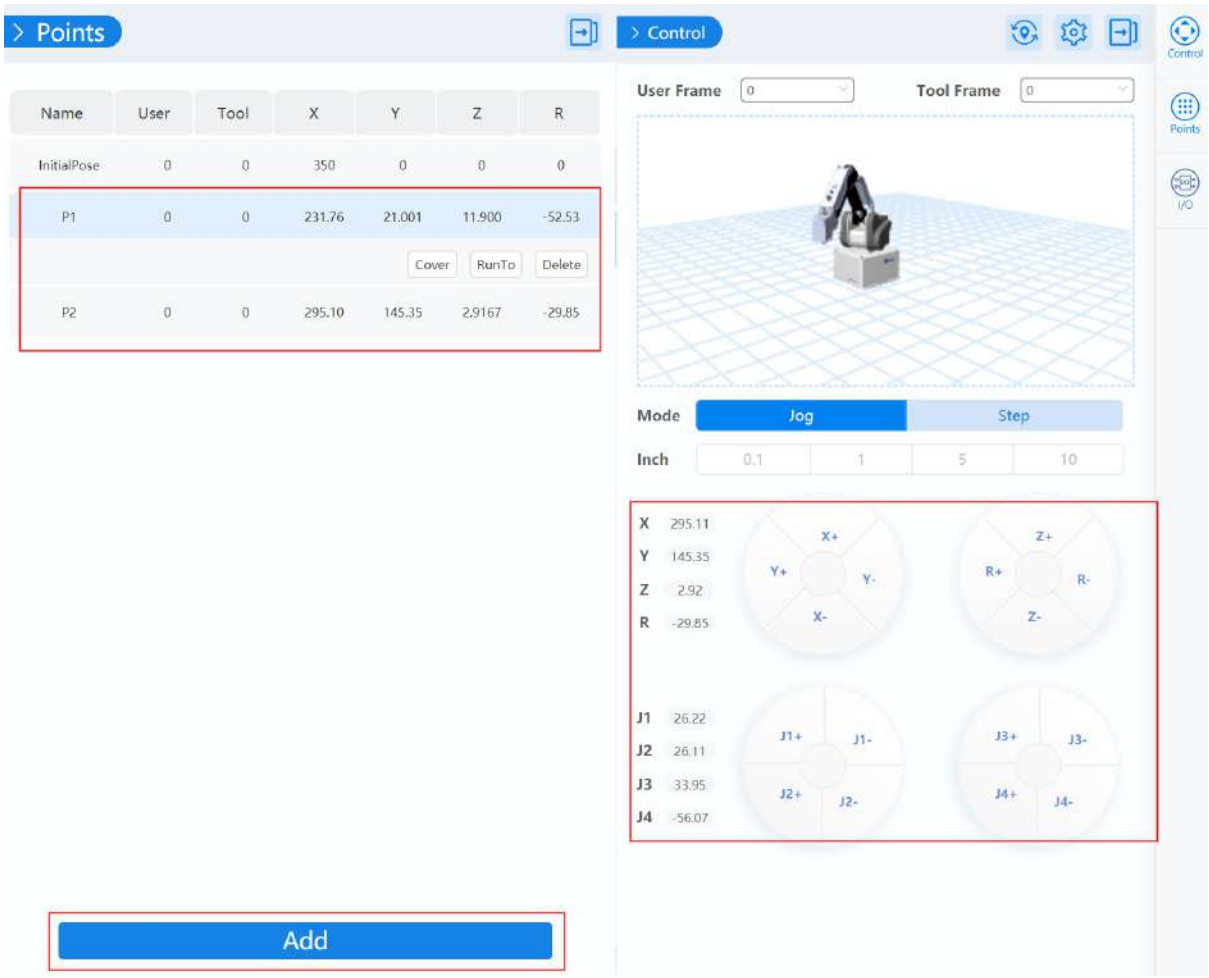
Now, we take an example to describe how to use teach and playback to control MG400 move.

For example, the MG400 moves from point P1 to P2 circularly in MovJ mode under the basic coordinate system.

**Step 1** Make MG400 in the enabled status.

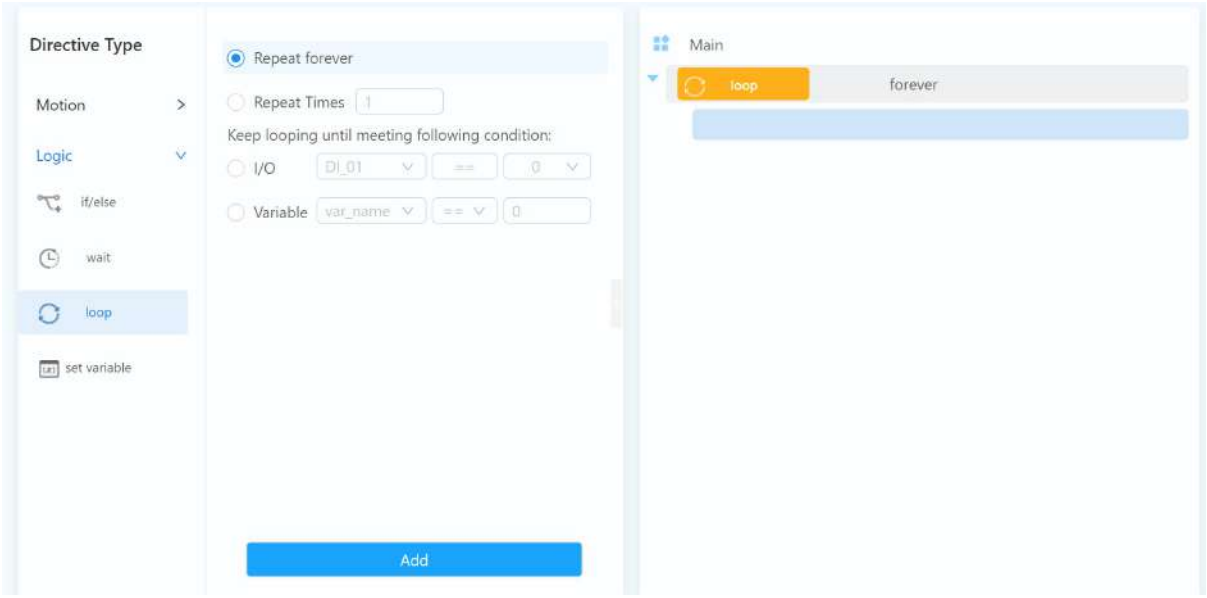

**Step 2** Click the motion buttons on the Control page to make MG400 jog to P1 point and then click **Add** on the **Points** page.

**Step 3** Click the motion buttons on the Control page to make MG400 jog to P2 point and then click **Add** on the **Points** page.

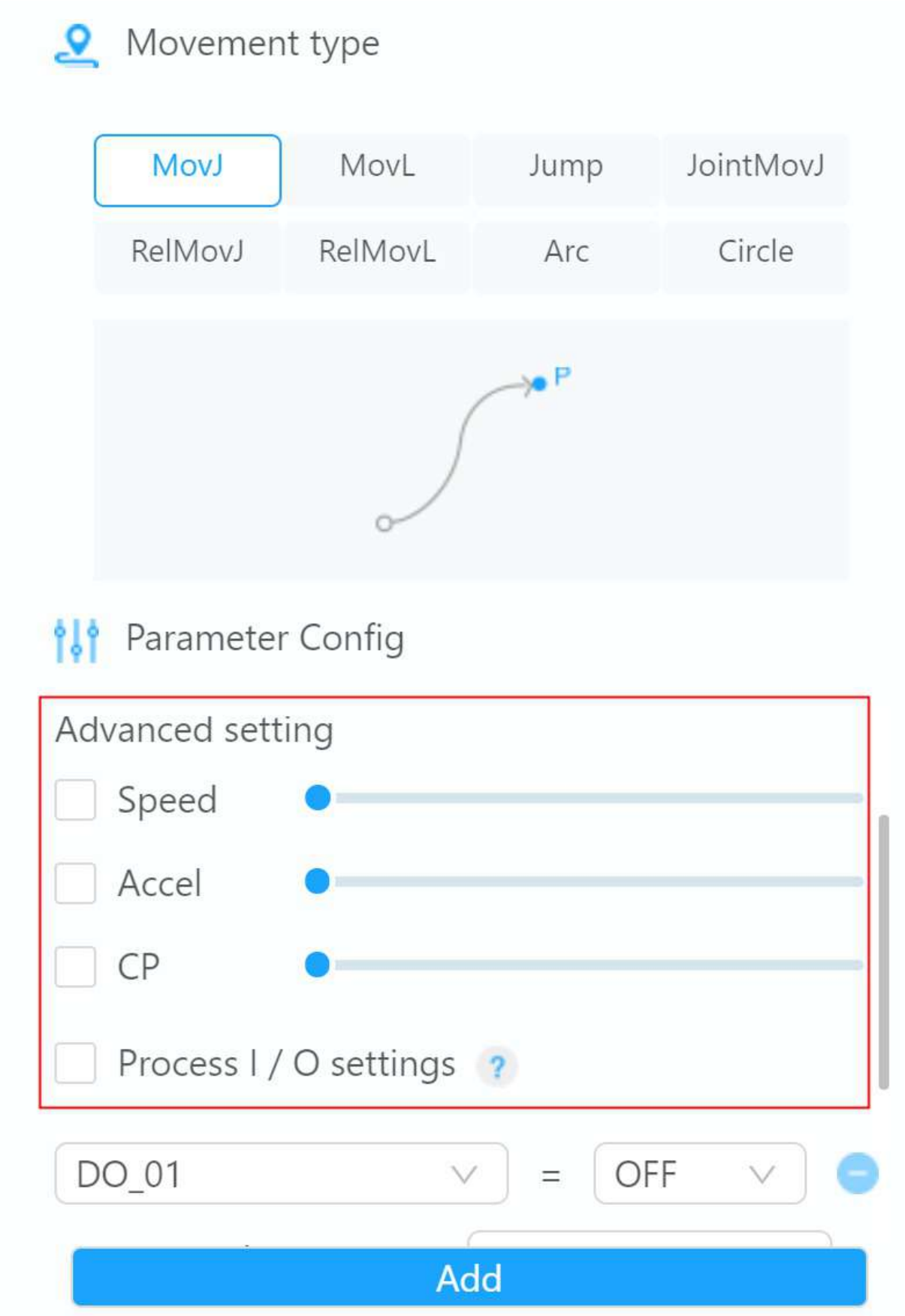



**Step 4** Write a program.

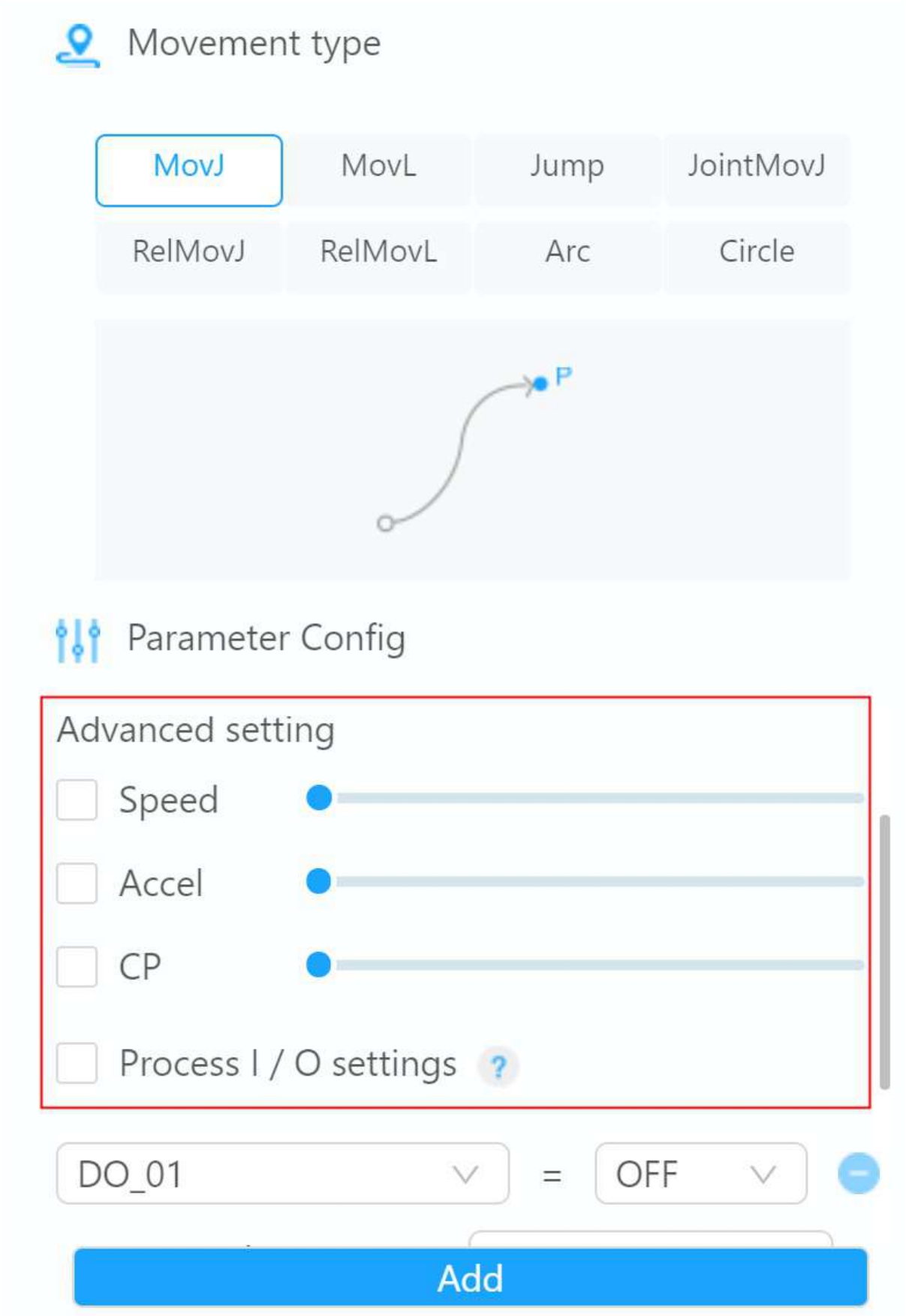
1. Select **Loop** and select **Repeat forever** and then click **Add**.



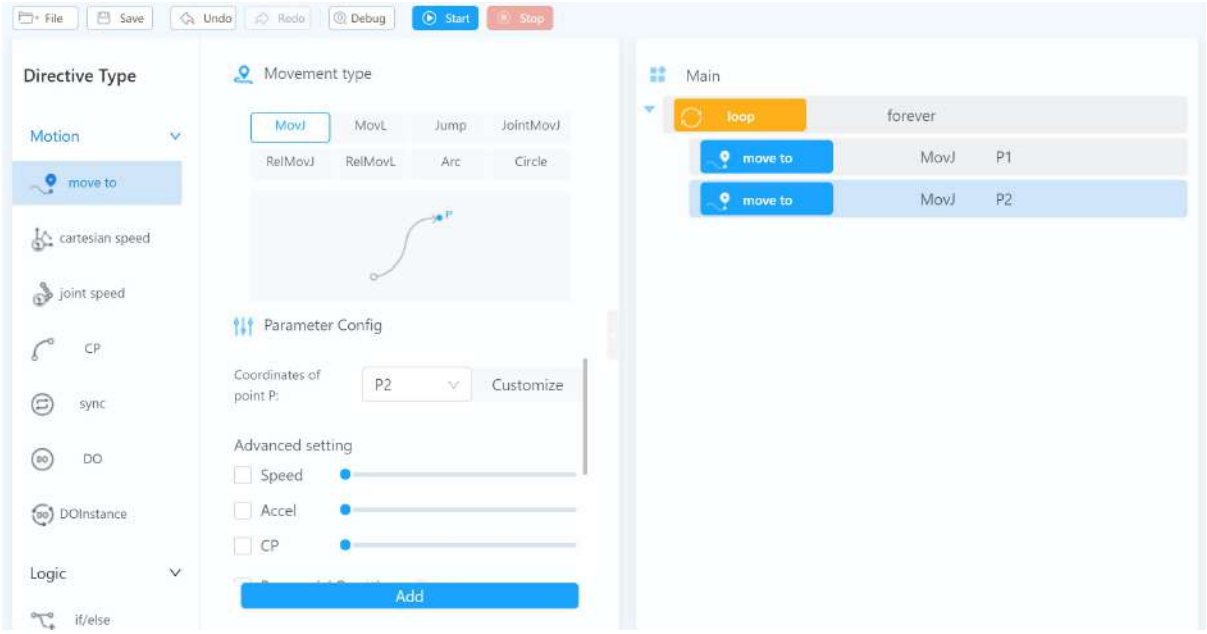

1. Select **MovJ** mode and select **P1**, then click **Add**.




1. Select **MovJ** mode and select **P2**, then click **Add**.




1. Select **MovJ** mode and select **P2**, then click **Add**.



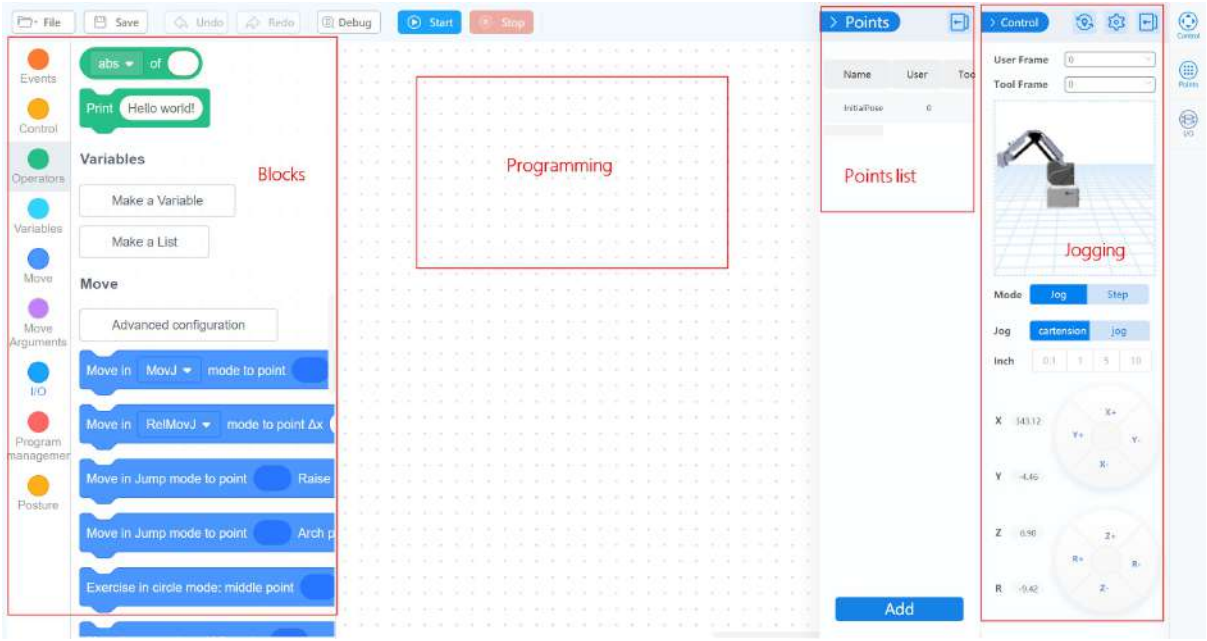

**Step 5** Click Save and input project name, then click **Start**.

The MG400 will moves from P1 to P2 circularly.

In this module, the debug function is not supported.

# 4.3.4 Blockly

Blockly is a programming platform . You can program through the puzzle format, which is straightforward and easy to understand.



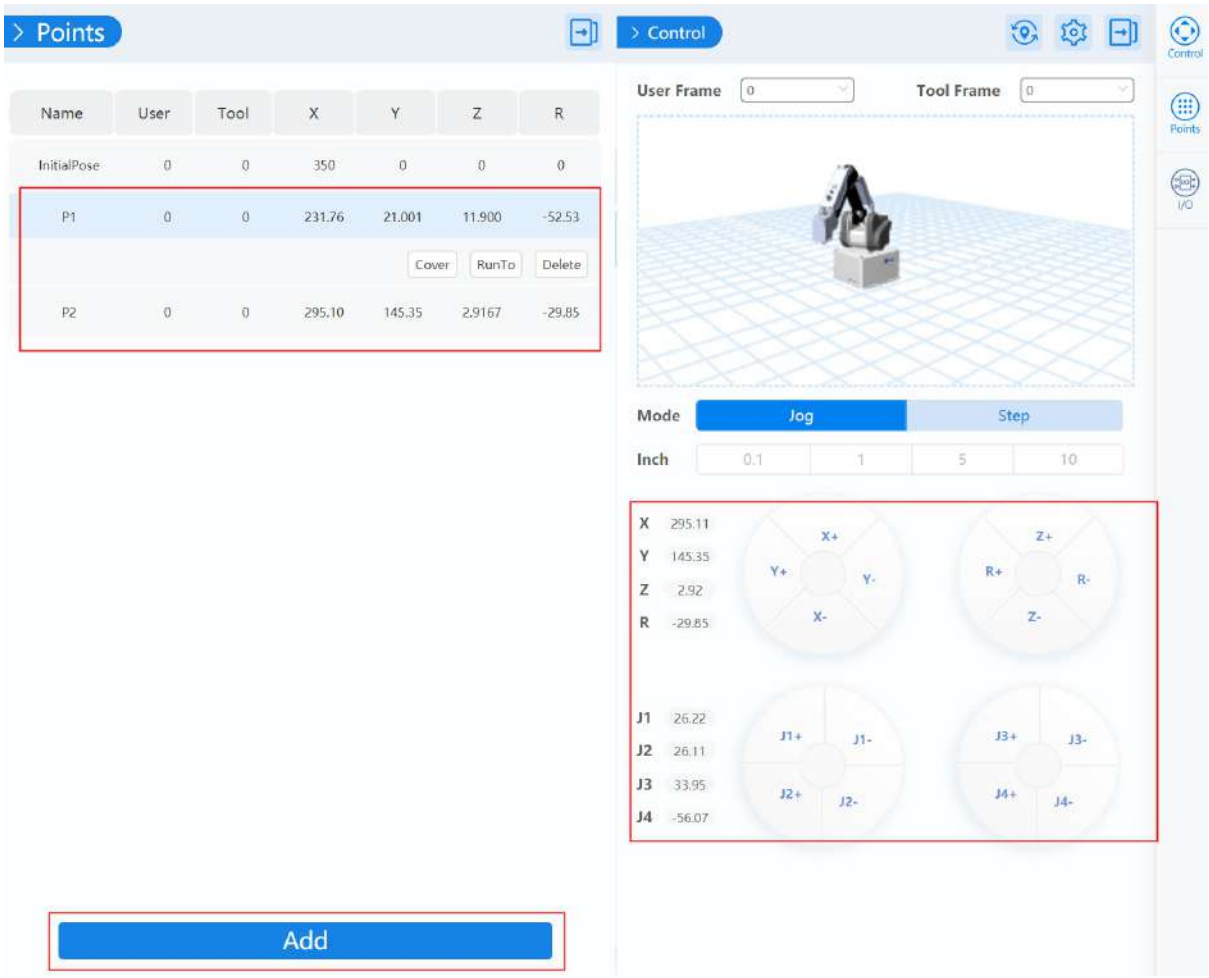

Now, we take an example to describe how to use Blockly to control MG400 move.

For example, the MG400 moves from point P1 to P2 circularly in MovJ mode under the basic coordinate system.

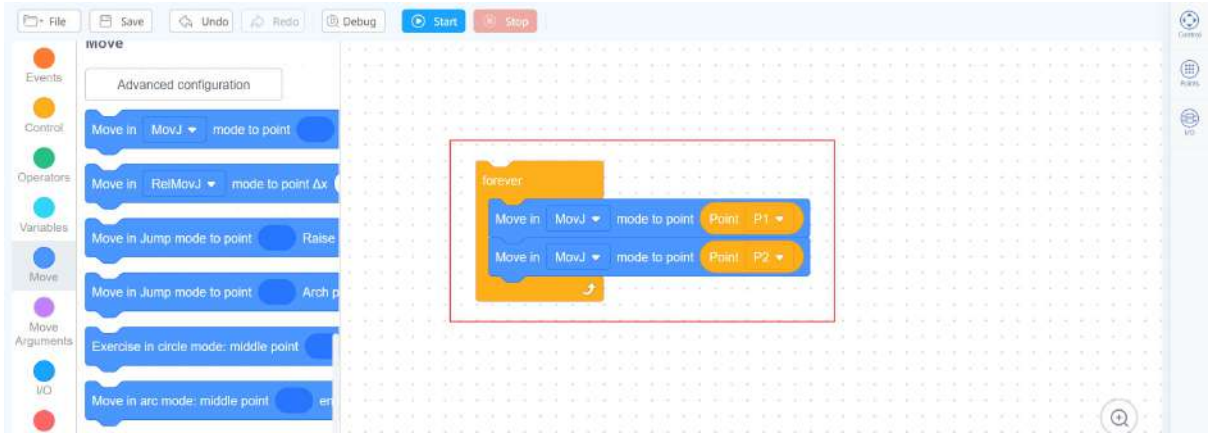
**Step 1** Make MG400 in the enabled status.


**Step 2** Click the motion buttons on the Control page to make MG400 jog to P1 point and then click **Add** on the **Points** page.

**Step 3** Click the motion buttons on the Control page to make MG400 jog to P2 point and then click **Add** on the **Points** page.

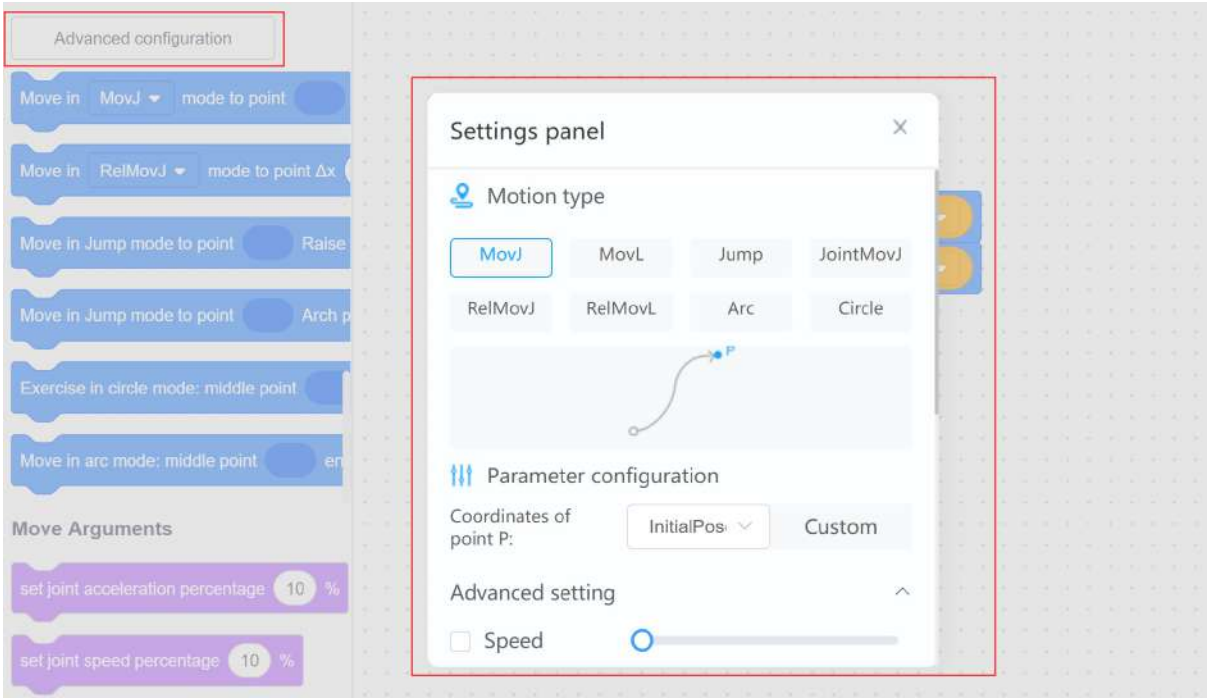



**Step 4** Write a program.

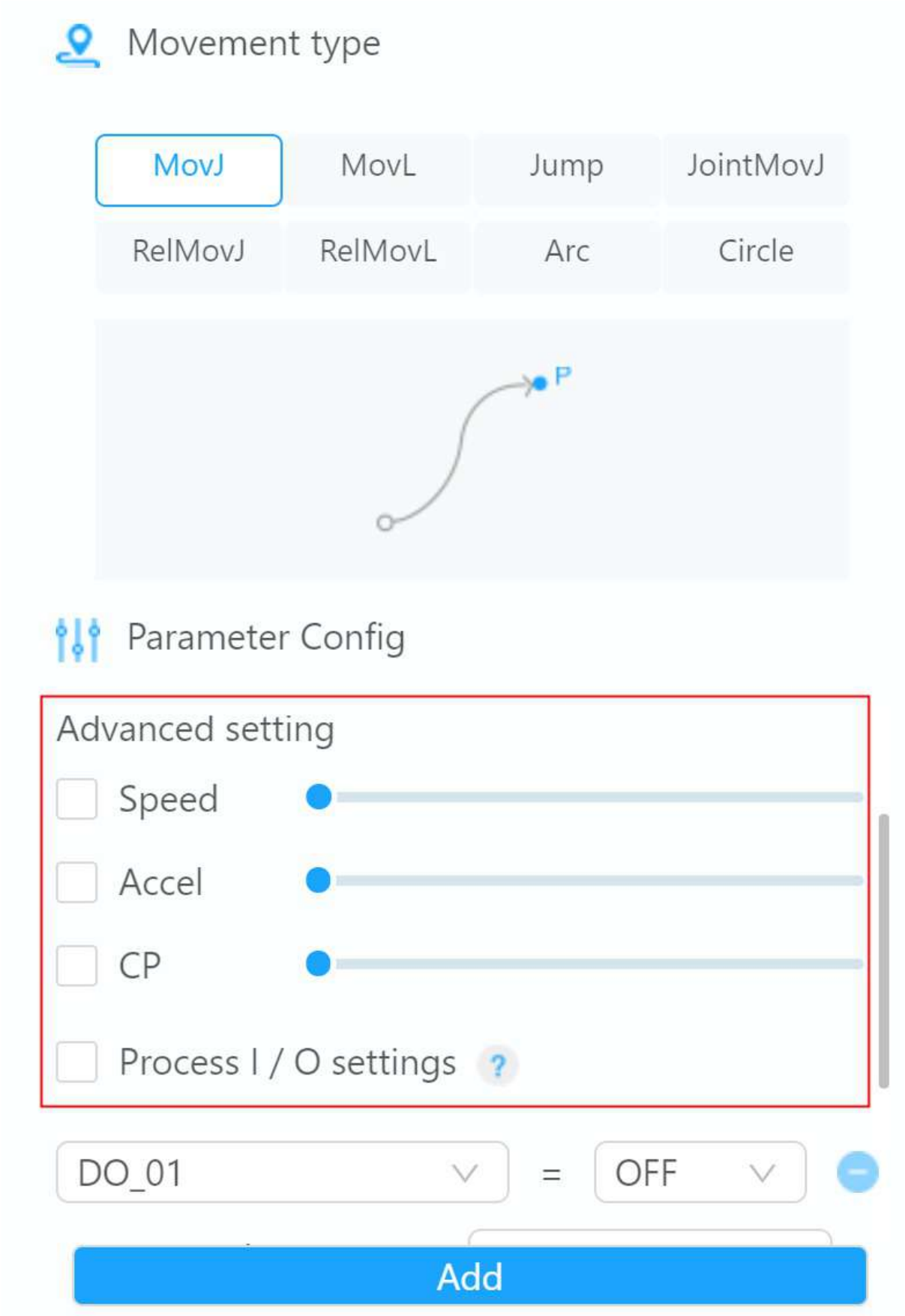



There are two methods to generate commands.

- Drag blocks directly.
- Click Advanced configuration to generate block, only support motion blocks.




In advanced configuration, you can set the speed and acceleration, CP or digital output status directly in this block.



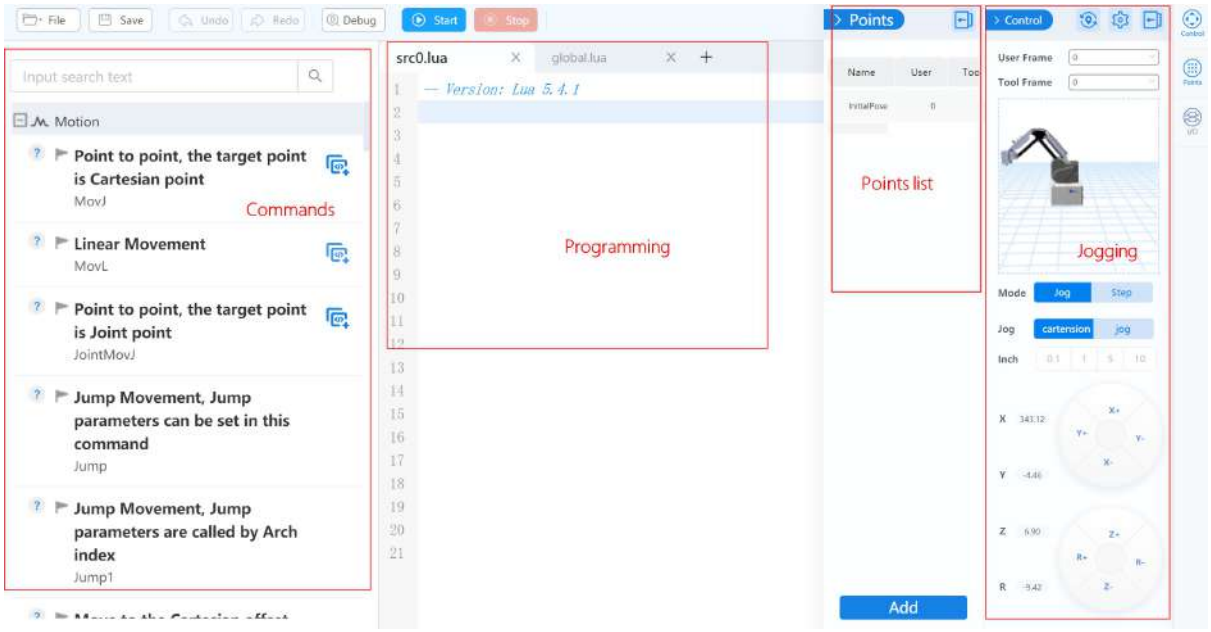

**Step 5** Click Save and input project name, then click **Start**.

The MG400 will moves from P1 to P2 circularly.

In this module, the debug function is not supported.

# 4.3.5 Script

MG400 supports various API, such as velocity/acceleration commands, motion commands, etc., which uses Lua language for secondary development.




In script, multiple threads are supported. Up to 5 threads can be executed simultaneously. **Scr0.lua** is the main thread, Other threads are sub threads, which run program parallel to the main thread, such as I/O control. You can click + to add the sub thread. In the sub threads, the motion commands cannot be called. Only the main thread supports motion commands.

In addition, global variable module (global.lua) is only used to define global variables and module functions. The motion commands cannot be called here.

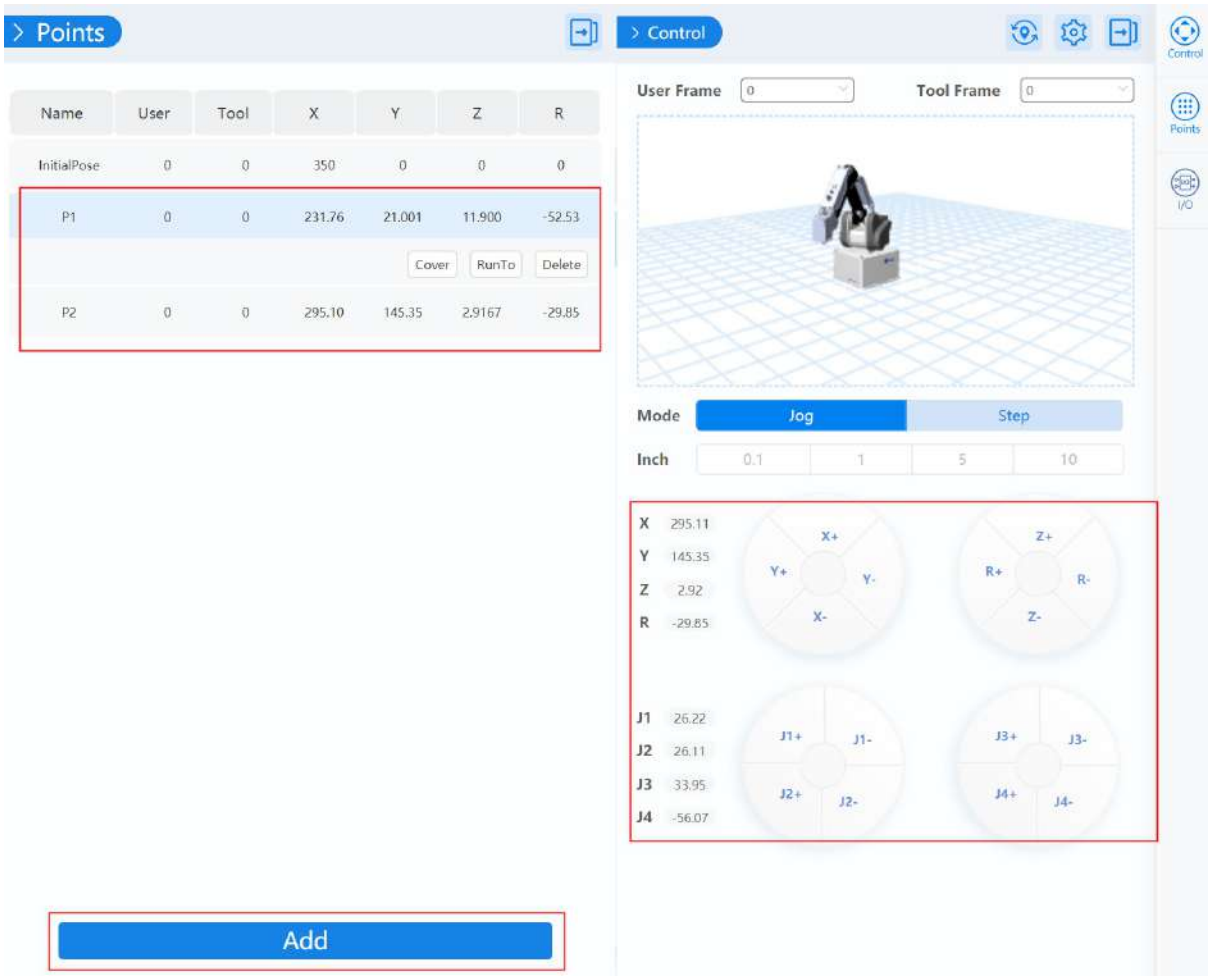
Now, we take an example to describe how to use script to control MG400 move.

For example, the MG400 moves from point P1 to P2 circularly in MovJ mode under the basic coordinate system.

**Step 1** Make MG400 in the enabled status.

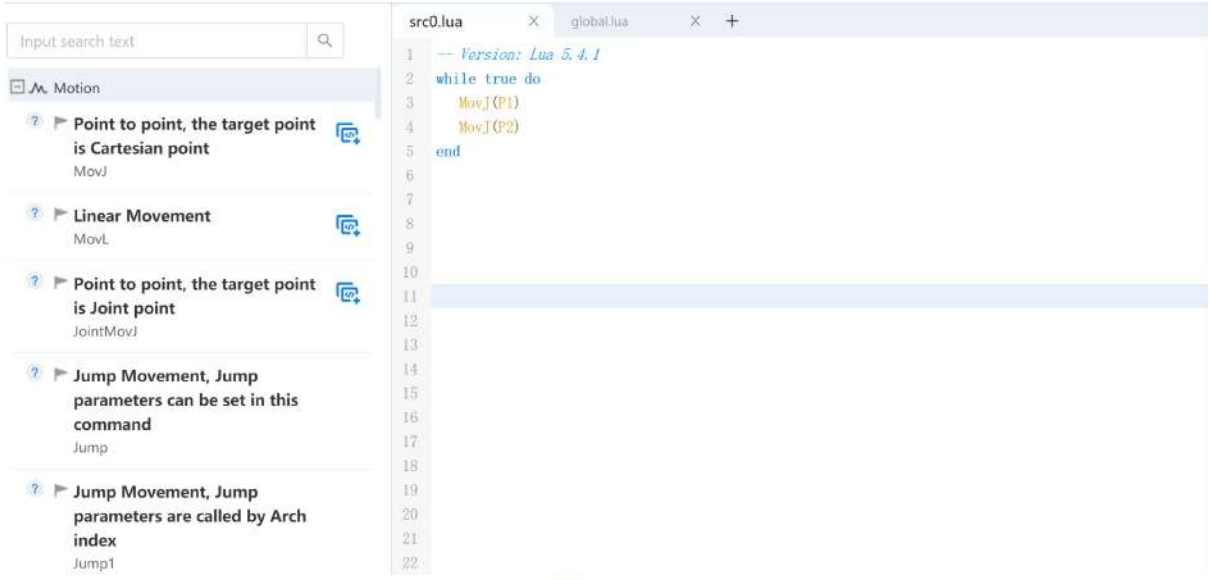

**Step 2** Click the motion buttons on the Control page to make MG400 jog to P1 point and then click **Add** on the **Points** page.

**Step 3** Click the motion buttons on the Control page to make MG400 jog to P2 point and then click **Add** on the **Points** page.

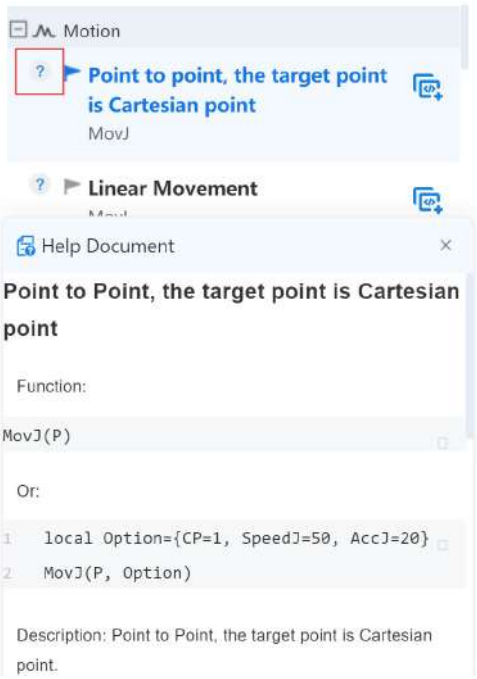



**Step 4** Write a program.

You can double-click the right command to insert it to the programming section or double-click  to insert it with the optional parameters.




When calling a MG400's command, you can click  to view the command's description.




**Step 5** Click Save and input project name, then click **Start**.

The MG400 will moves from P1 to P2 circularly.

If you want to debug this program, you can set the breakpoints and then click Debug to debug.

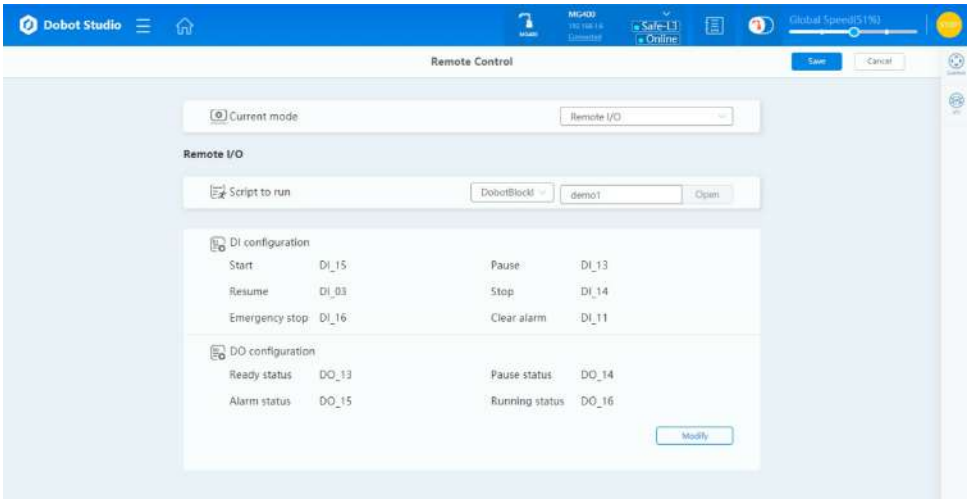
# 4.3.6 Remote Control

# 4.3.6.1 Overview

External equipment can send commands to MG400 by different remote control modes, such as remote I/O mode and remote Modbus mode. The default mode is online mode when the MG400 is shipped out. When you need to set the remote mode, please set it on the **Remote Control** page with the robot motor in the disabled state.

# 4.3.6.2 Remote I/O

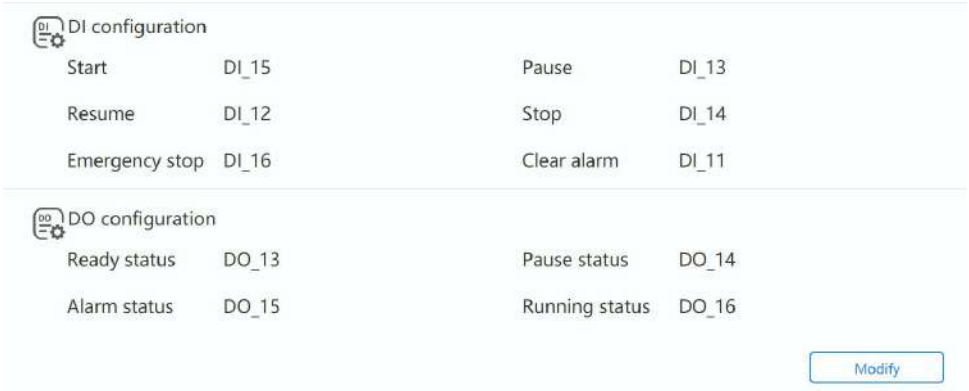
When the remote mode is remote I/O mode, external equipment can control the MG400 in this mode. You can click **Remote Control** to use this module. The remote control page is shown as follows.




In this topic, we only detail how to use remote control mode. The details on how to connect external equipment and use it are not described in this topic.

**Prerequisites**

- The project to be running in the remote mode has been prepared, supporting script, Blockly and teaching and playback projects.
- The external equipment has been connected to the MG400 by the I/O interface. The specific I/O interface description is shown as follows . You can modify the specified I/O on the **Remote Control** page.




- The MG400 has been powered on.

**Procedure**

**Step 1** Make the MG400 in the disabled status and click **Remote Control** on the Home page.

The remote control page is displayed.

**Step 2** Select **Remote I/O** on the **Current Mode** section and select the project with the right program module on the **Script to run** section.

**Step 3** Click **Save**.

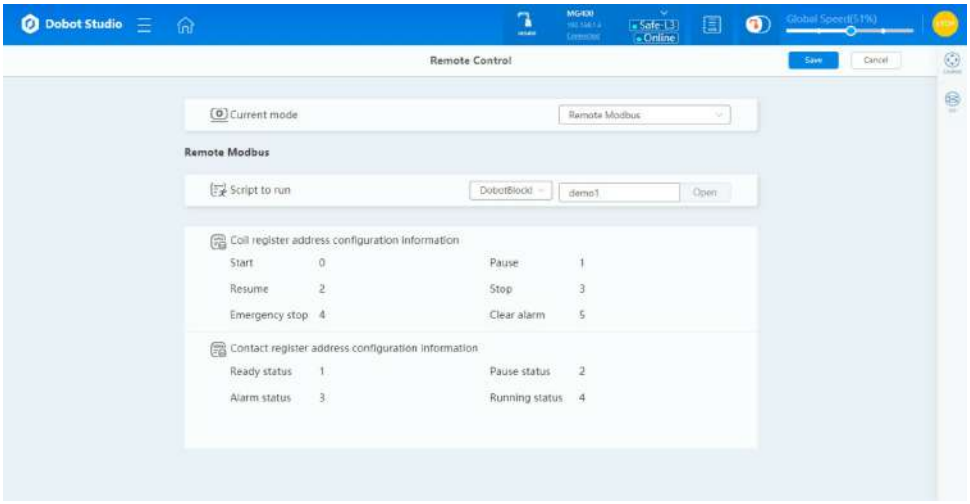
Right now, only the emergency stop button, control page, I/O page are available. The remote status will be displayed on the DobotStudio2020.


**Step 4** Trigger the starting signal on the external equipment.

The MG400 will be in the enabled status and move as the selected project. If the stop signal is triggered, the MG400 will be stopped with the disabled status.

# 4.3.6.3 Remote Modbus

When the remote mode is Remote Modbus mode, external equipment can control the MG400 in this mode. You can click **Remote Control** to use this module. The remote control page is shown as follows.




The specific Modbus register descriptions are shown as follows.

| Register address (Take a PLC as an example) | Register address (Robot system) | Description |
|---|---|---|
| Coil register | | |
| 00001 | 0 | Start running in the remote Modbus mode |
| 00002 | 1 | Pause running in the remote Modbus mode |
| 00003 | 2 | Continue to run |
| 00004 | 3 | Stop to run and exit the remote Modbus mode |
| 00005 | 4 | Emergency stop and exit the remote Modbus mode |
| 00006 | 5 | Clear alarm |
| Discrete input register | | |
| 10001 | 0 | Auto-exit |
| 10002 | 1 | Ready status |
| 10003 | 2 | Pause status |
| 10004 | 3 | Running status |
| 10005 | 4 | Alarm status |

In this topic, we only detail how to use remote control mode. The details on how to connect external equipment and use it are not described in this topic.

**Prerequisites**

- The project to be running in the remote mode has been prepared, supporting script, Blockly and teaching and playback projects.
- The MG400 has been connected to the external equipment with the LAN2 interface. You can connect them directly, please select based on site requirements.

  The IP address of the MG400 and the external equipment must be in the same network segment without conflict. You can modify the MG400's IP address on the **Settings > IP Configuration** page; the default port is **502** and cannot be modified.
- The MG400 has been powered on.

**Procedure**

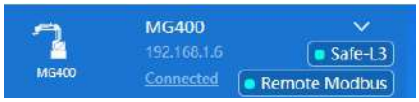
**Step 1** Make the MG400 in the disabled status and click **Remote Control** on the Home page.

The remote control page is displayed.

**Step 2** Select **Remote Modbus** on the **Current Mode** section and select the project with the right program module on the **Script to run** section.

**Step 3** Click **Save**.

Right now, only the emergency stop button, control page, I/O page are available. The remote status will be displayed on the DobotStudio2020.

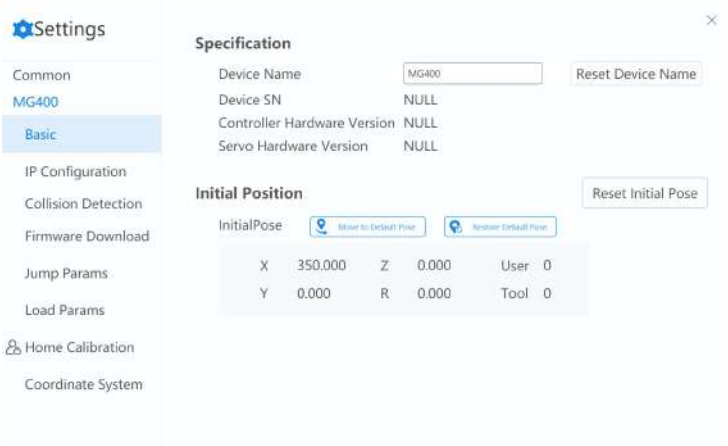



**Step 3** Trigger the starting signal on the external equipment.

The MG400 will be in the enabled status and move as the selected project. If the stop signal is triggered, the MG400 will be stopped with the disabled status.

# 4.3.7 Basic Setting

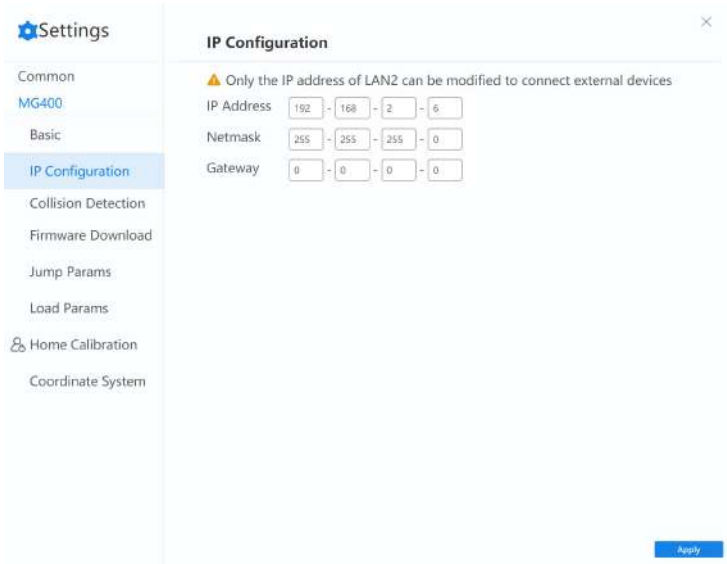
On the Settings > Basic page, you can modify the MG400's name, view the SN number and hardware info of the MG400, and set the initial pose.



# 4.3.8 IP Configuration

The MG400 can be communicated with external equipment by the LAN2 interface which supports TCP, UDP and Modbus protocols. The default IP address is **192.16.2.6**. In real applications, if the TCP or UDP protocol is used, the robot system can be a client or a server based on site requirements; if the Modbus protocol is used, the robot system only can be the Modbus slave, and the external equipment is the master.

You can modify the IP address on the **Settings > IP Configuration** page, as shown below. The IP address of the MG400 must be in the same network segment of the external equipment without conflict.



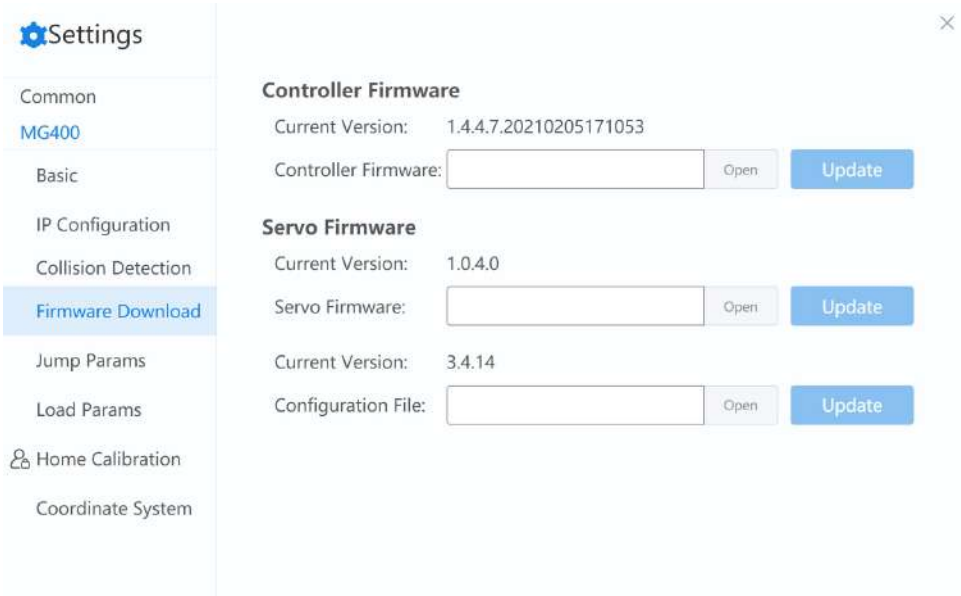
# 4.3.9 Firmware Download

When the controller firmware needs to be updated, you can import the latest firmware on the **Firmware Download** page. Currently, only controller firmware update is supported.

> [!DANGER]
>
> During the updating ,please DO NOT perform any other operation on the MG400 or power off it, to avoid MG400 in an abnormal condition. Otherwise, it will be vulnerable to injury the device or the person.

**Step 1** Click **Settings > Firmware Download** .

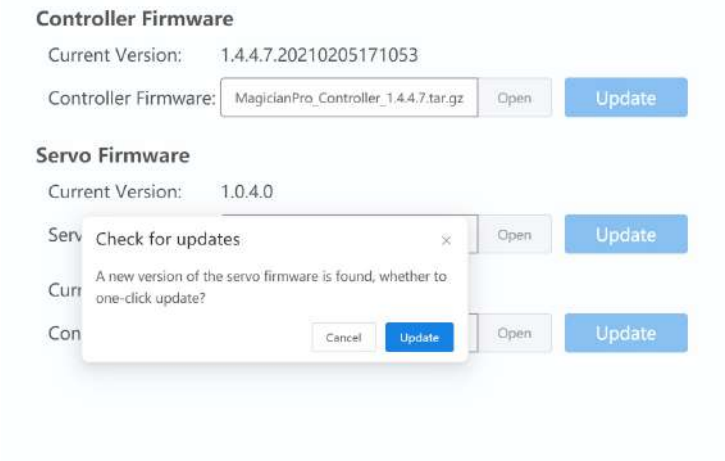
The Firmware download page is displayed.




**Step 2** Open the latest controller firmware from the local and click **Update**.

The MG400's controller will be updated automatically.

**Step 3** After the controller is updated, Click **Update** on the **Check for Update** window to update the servo firmware.




**Step 4** After the servo is updated, Please reboot the MG400.

When rebooting the MG400, the LED indicator on the base delays about 10 seconds before it starts to work.

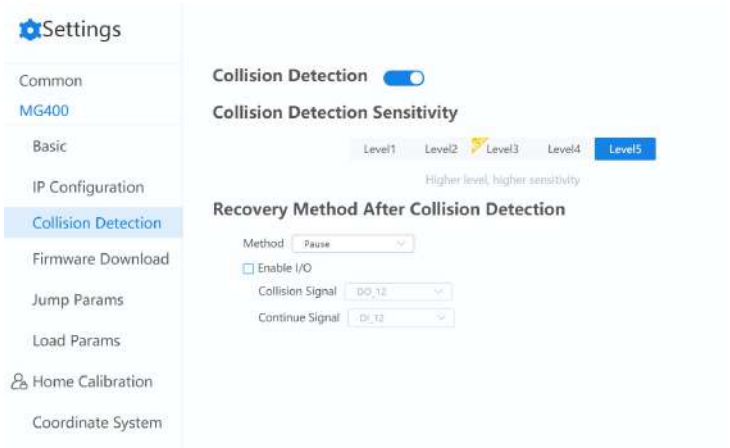
# 4.3.10 Collision Detection

Collision detection is mainly used for reducing the impact on the MG400, to avoid damage to the MG400 or external equipment. If the collision detection is activated, the MG400 will suspend running automatically when the MG400 hits an obstacle.

You can enable collision detection function on the **Settings Collision Detection** page and set the collision level. Meanwhile, you can select **Automatically start dragging after collision**, namely, when the robot arm stops running after hitting an obstacle, you can drag robot to a safe position.

**Step 1** Click **Settings > Collision Detection**.

The collision detection page is displayed.




**Step 2** Enable Collision Detection and select the collision detection sensitivity.

There are five levels to select. The higher level, higher sensitivity.

When you select the level, the DobotStudio2020 will display it.


**Step 3** Select the recovery mode after the collision is occurred.

There are two ways to recover when the MG400 suspends running because of the collision.


- Automatically resume after 5s: Namely, after 5s, the MG400 will resume the running automatically.
- Pause: Namely, you need to click 

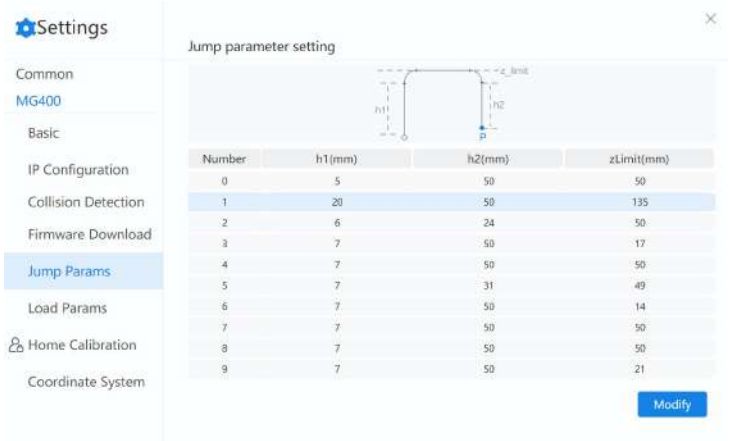
to resume the running.

Also, you can trigger the resume I/O on the external equipment to resume the running after set **Collision Signal** and **Continue Signal**. The Continue Signal is the same to the Resume signal on the Remote I/O mode.

# 4.3.11 Jump Params

If the motion mode is **Jump** when running programs , you need to set **StartHeight**（h1）, **EndHeight**(h2), and **zLimit**.

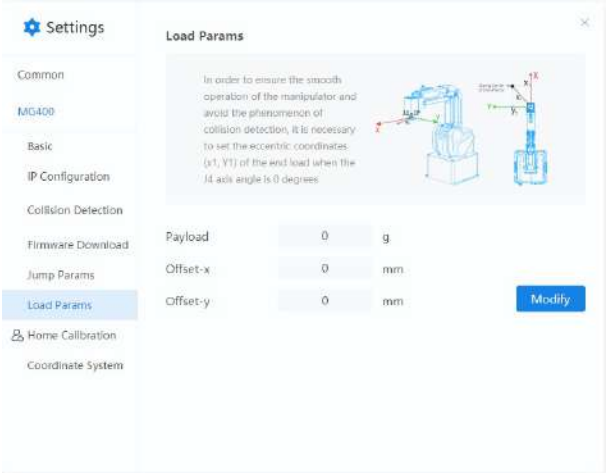
10 sets of Jump parameters are supported. Please select any set of parameters and click **Modify** to set the Jump parameters for calling Jump command with **Arch index** parameter during programming in script or Blockly module, as shown below.



# 4.3.12 Load Params

To ensure optimum robot performance, it is important to make sure the load and eccentric coordinates of the end effector are within the maximum rating for the robot, and that Joint 4 does not become eccentric. Setting load and eccentric coordinates makes the motion of robot optimal, reduces vibration to shorten the operating time. The load is weight of the end effector and work piece, which must not exceed the maximum load(750g). The eccentric coordinates is eccentric coordinates of the end effector and work piece. Please set the right load and eccentric coordinates. Setting a value that is smaller than the actual load may cause errors, excessive shock, insufficient function of the MG400, and shorten the life cycle of parts.

When you launch the DobotStudio2020 and enable the MG400, you need to set the load parameters. At other times, you can set them on the **Settings > Load Params** page.

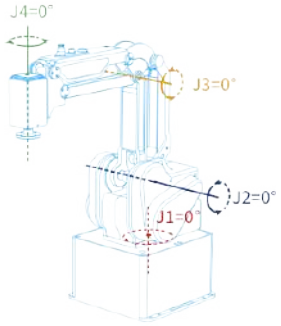



If you call the Load API in script or Blockly program and run it, the values will be displayed on the **Settings > Load Params** page synchronously.

# 4.3.13 Home Calibration

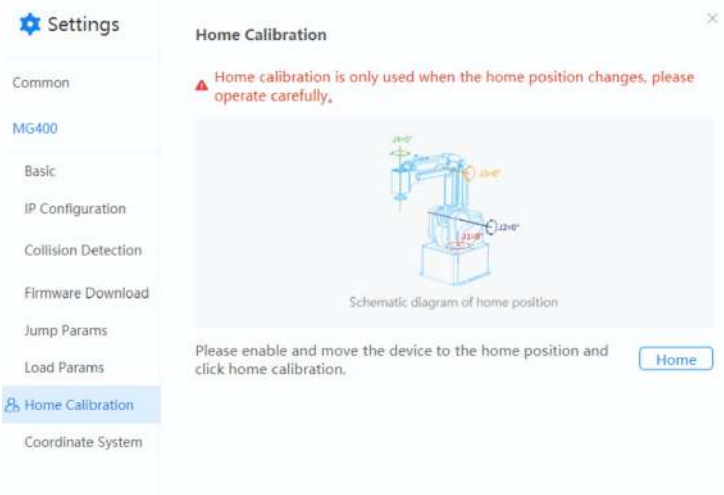
After some parts (motors, reduction gear units) of the MG400 have been replaced or the robot has been hit, the homing point of the robot will be changed. You need to reset the homing point.

**Step 1** Put the robot in the homing position in the hand-teaching mode, as shown below.




**Step 2** Make the MG400 in the enabled status and click **Home**.

> [!DANGER]
>
> Home calibration is only use to calibrate the homing point. Please operate carefully.




After operating the homing procedure, you can check whether the homing calibration is successful, which view the joint coordinates are (0,0,0,0) on the **Control** page.

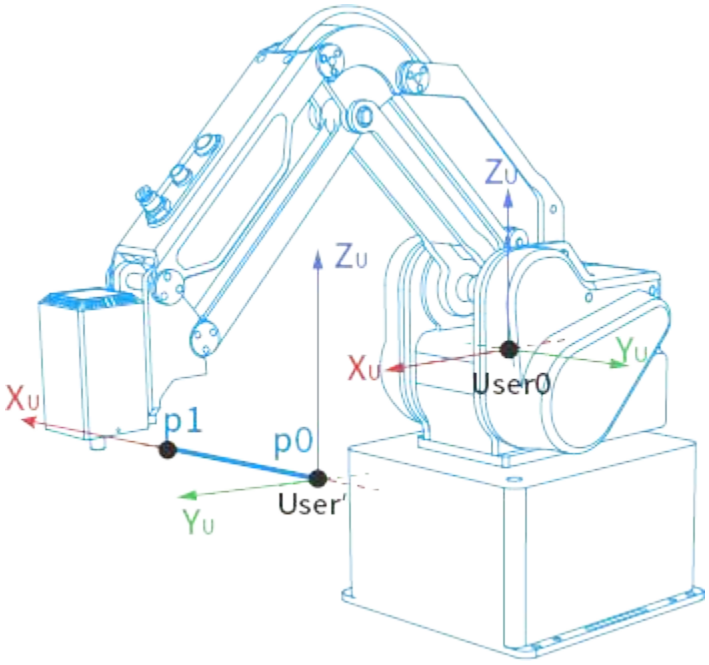
# 4.3.14 Coordinate System

# 4.3.14.1 User Coordinate System

When the position of workpiece is changed or a robot program needs to be reused in multiple processing systems of the same type, you can create coordinate systems on the workpiece to simplify programming. There are totally 10 groups of User coordinate systems, of which the first one is defined as the Base coordinate system by default and cannot be changed. And the others can be customized by users.

> [!NOTE]
>
> When creating a User coordinate system, please make sure that the reference coordinate system is the Base coordinate system.

User coordinate system is created by two-point calibration method. Move the robot to two points **P0(x0, y0, z0)**, **P1(x1, y1, z1)**. Point P0 is defined as the origin and the line from point P0 to Point P1 is defined as the positive direction of X-axis. And then the Y-axis and Z-axis can be defined based on the right-handed rule, as shown below.




Take the establishment of User 1 coordinate system as an example based on two-point calibration method.

**Prerequisites**

- The MG400 has been powered on and enabled.
- The MG400 is in the Base coordinate system.

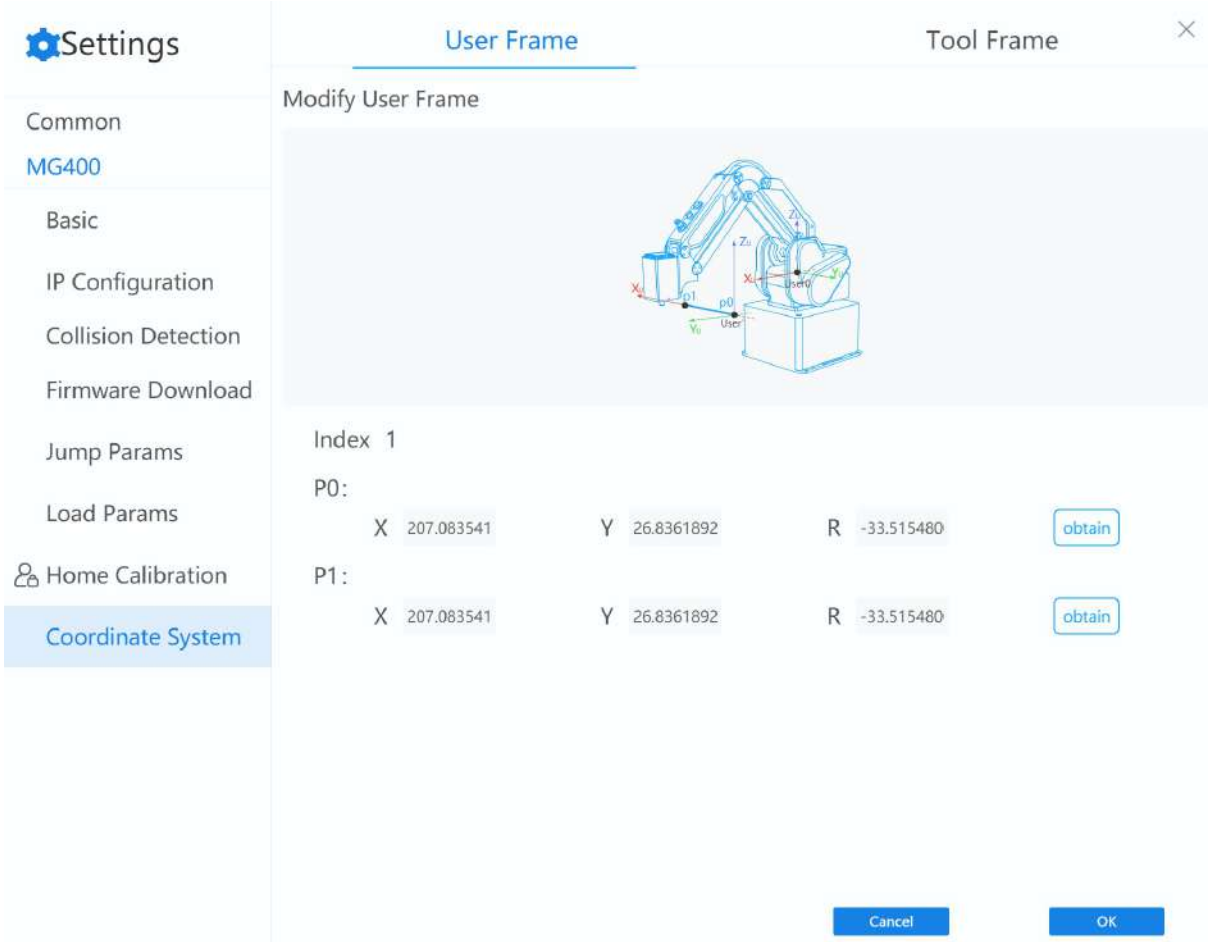
**Procedure**

**Step 1** Click **Setting > Coordinate system**.

The coordinate system page is displayed, as shown below.


**Step 2** Select the right User coordinate system index and click **Modify** on the User **Frame** tab.

The Modify User Frame page is displayed.



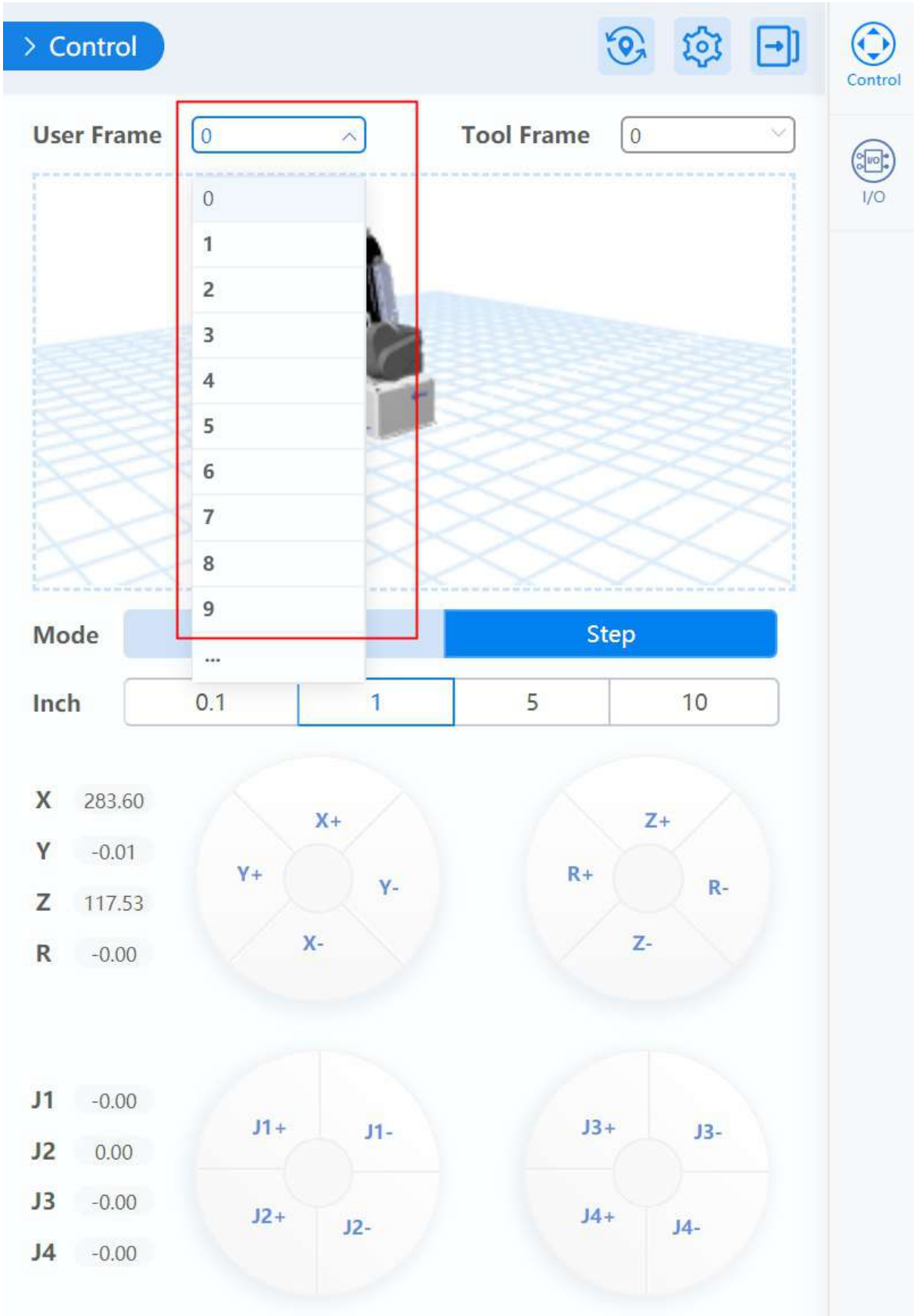

**Step 3** Jog MG400 to the point P0 and click **obtain** on the **P0**: panel of the User Frame tab.

> [!NOTE]
>
> When creating a User coordinate system, please make sure that the reference coordinate system is the base coordinate system. Namely ,the User coordinate system is 0 and the Tool coordinate system is 0 when jogging robot.

**Step 4** Jog MG400 to the point P1 and click **obtain** on the **P1**: panel of the User Frame tab.

**Step 5** Click **OK**, the User coordinate system is modified.

Now, you can select the set User coordinate system and jog MG400.



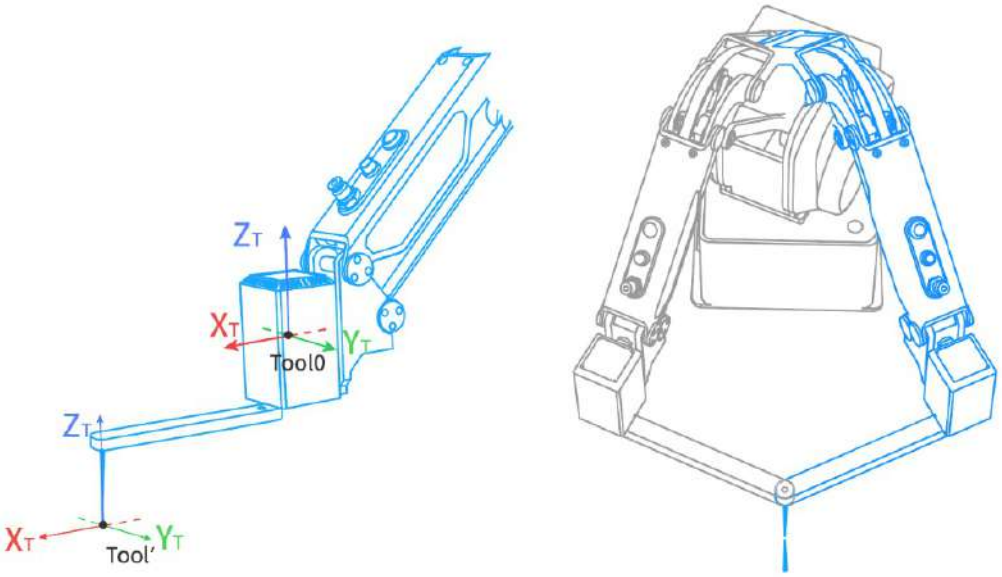
# 4.3.14.2 Tool Coordinate System

When an end effector such as welding gun, gripper is mounted on the robot, the Tool coordinate system is required for programming and operating a robot. For example, you can use multiple grippers to carry multiple workpieces simultaneously to improve the efficiency by setting each gripper to a Tool coordinate system.

There are totally 10 groups of Tool coordinate systems. Tool 0 coordinate system is the predefined Tool coordinate system which is located at the robot flange and cannot be changed.

> [!NOTE]
>
> When creating a Tool coordinate system, please make sure that the reference coordinate system is the Base coordinate system.

Tool coordinate system of robot is created by two-point calibration method: After an end effector is mounted, please adjust the direction of this end effector to make the TCP (Tool Center Point) align with the same point (reference point) in two different directions, for obtaining the position offset to generate a Tool coordinate system, as shown below.




Take the establishment of Tool 1 coordinate system as an example based on two-point calibration method.

**Prerequisites**

- The MG400 has been powered on and enabled.
- The MG400 is in the Base coordinate system.

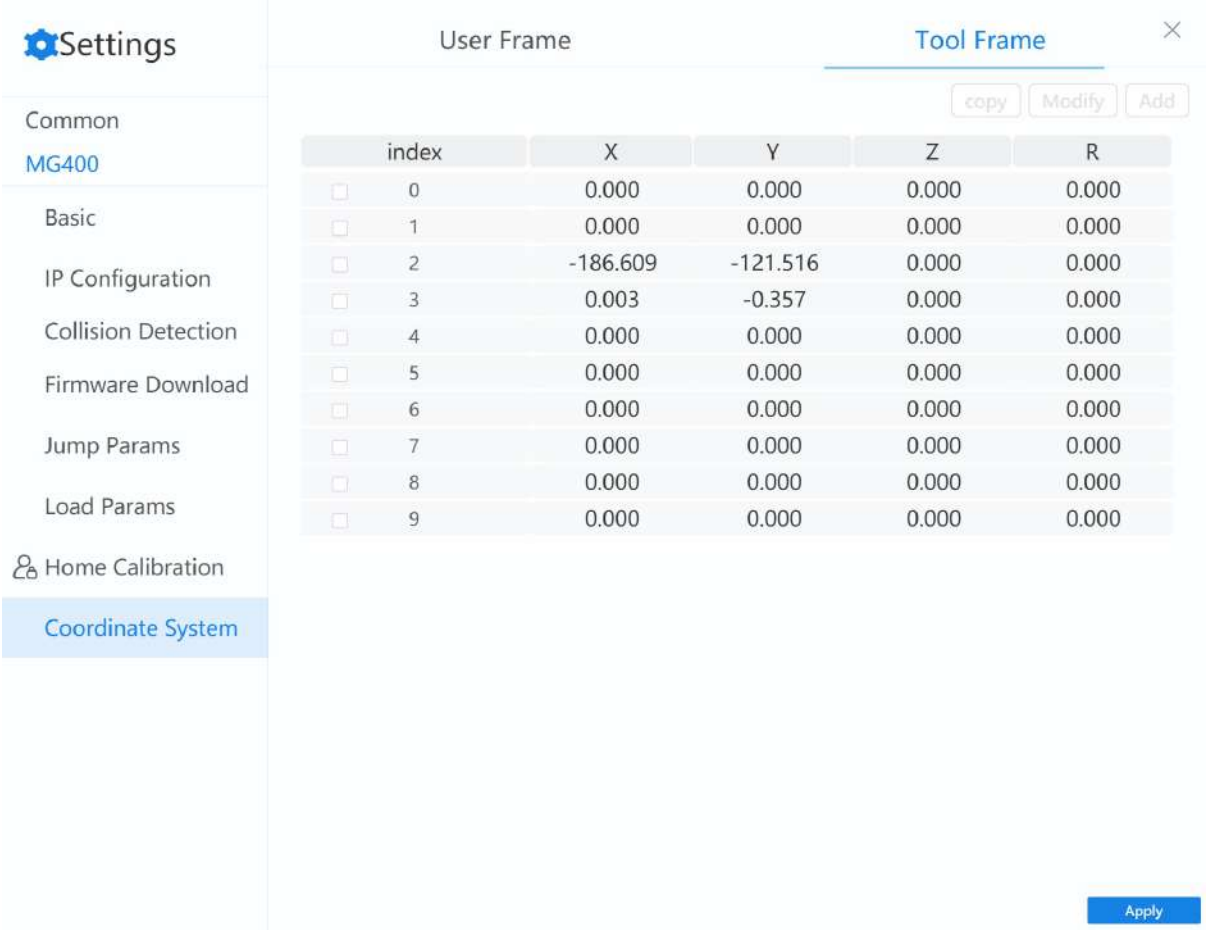
**Procedure**

**Step 1** Mount an eccentric end effector on the robot. The detailed instructions are not described in this topic.

The end effector must be eccentrical. Otherwise, the tool coordinate system cannot be successful.

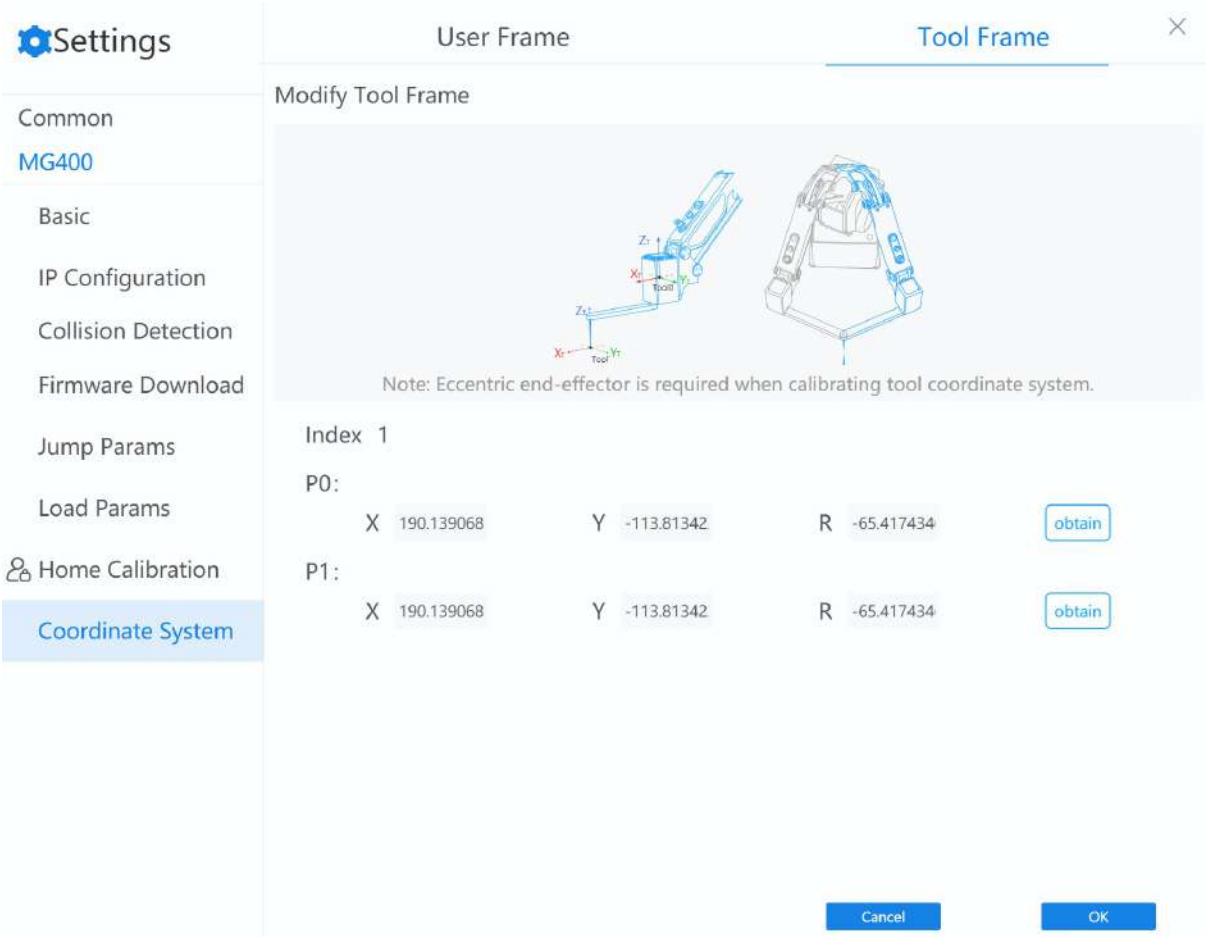
**Step 2** Click **Setting > Coordinate system**.

The coordinate system page is displayed, as shown below.




**Step 3** Select the right Tool coordinate system index and click **Modify** on the Tool **Frame** tab.

The Modify Tool Frame page is displayed.



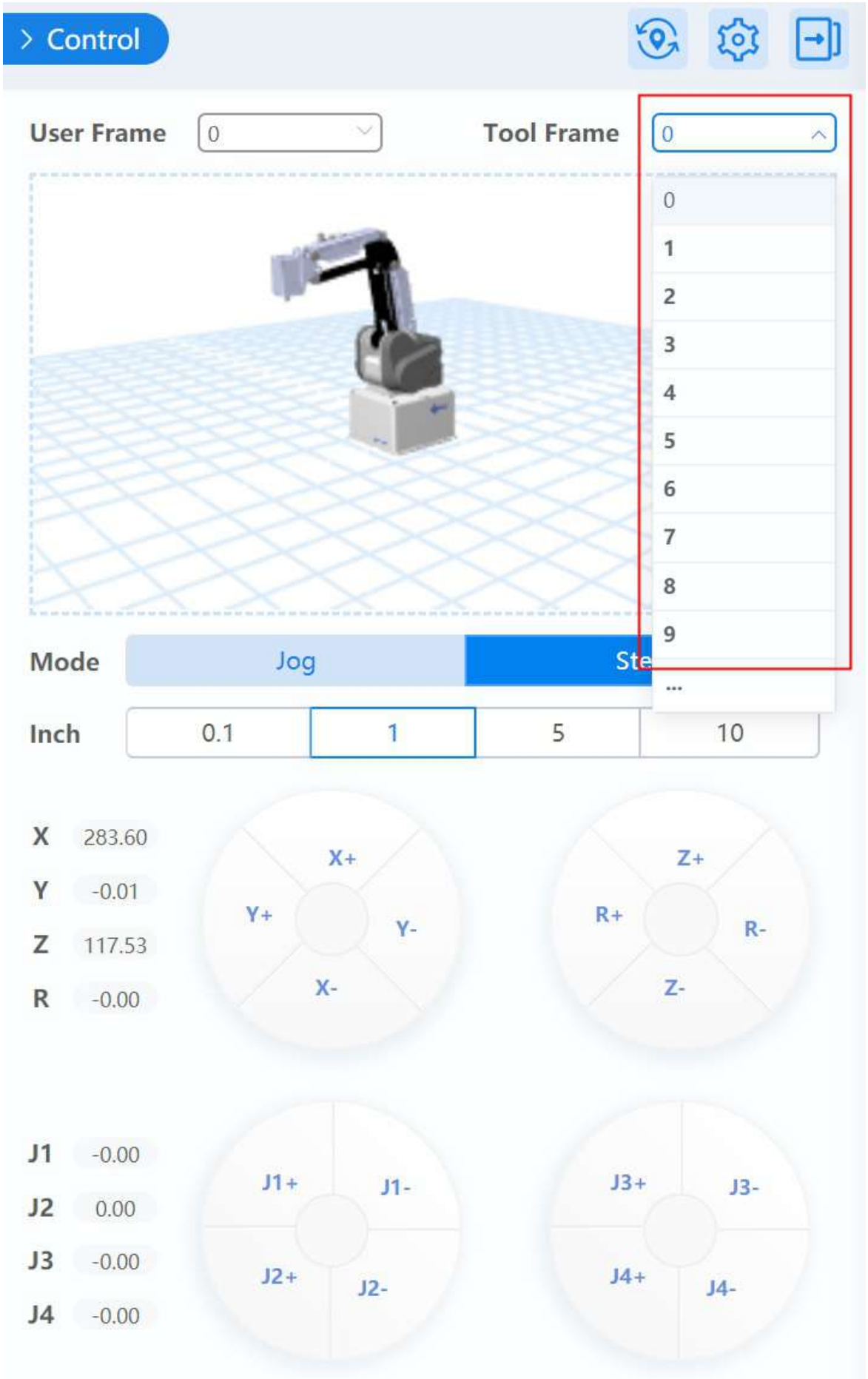

**Step 4** Jog the MG400 to the reference point in the first direction, and click **obtain** on the **P0**: panel of the Tool Frame tab.

> [!NOTE]
>
> When creating a Tool coordinate system, please make sure that the reference coordinate system is the base coordinate system. Namely ,the User coordinate system is 0 and the Tool coordinate system is 0 when jogging robot.

**Step 5** Jog the MG400 to the reference point in the second direction, and click **obtain** on the **P1**: panel of the User Frame tab.

**Step 6** Click **OK**, the Tool coordinate system is modified.

Now, you can select the set Tool coordinate system and jog MG400.


Control
User Frame
0
Tool Frame
0
0
1
2
3
4
5
6
7
8
9
...
Mode
Jog
Ste
Inch
0.1
1
5
10
X
283.60
Y
-0.01
Z
117.53
R
-0.00
X+
Y+
Y-
X-
Z+
R+
R-
Z-
J1
-0.00
J2
0.00
J3
-0.00
J4
-0.00
J1+
J1-
J2+
J2-
J3+
J3-
J4+
J4-

# 4.3.15 I/O

On the I/O page, there are three features.

- Output: Set the output status.
- Monitor: Monitor the status of the input and output when the MG400 is running.
- Set I/O alias: You can click  to set I/O alias.