

PiPER ROBOTIC ARM

User Manual

V 1.0

2024.09



AGILE·X

Important Safety Information

This chapter contains important safety information that must be read and understood by any user or organization before using the device for the first time. If there are any questions regarding the usage, please contact us at support@agilex.ai. It is essential to follow and implement all assembly instructions and guidelines in other chapters of this manual. Special attention should be paid to the text associated with warning signs.

Warning icon: ⚠ This refers to situations that may cause danger and, if not avoided, could result in personal injury or serious equipment damage.

Warning⚠: AgileX assumes no responsibility if the robotic arm is damaged, altered, or modified in any way. AgileX is not liable for any damages to the robotic arm or any other equipment caused by program errors or robotic arm malfunctions.

Limitation of Liability: Any safety information contained in this manual shall not be considered as a warranty of the AgileX robotic arm. Even if the robot arm complies with all safety instructions, it may still cause injury or damage.

The information in this manual does not include the design, installation and operation of a complete robotic arm application, nor does it include all peripheral equipment that may affect the safety of this complete system. The design and use of the complete system must comply with the safety requirements established in the standards and regulations of the country where the robot arm is installed.

It is the responsibility of the integrator and end customer of the robotic arm to ensure that relevant regulations and applicable laws and regulations are followed to ensure that there are no major hazards in the complete robotic arm application example.

This includes but is not limited to the following:

1. Effectiveness and Responsibility

- Conduct a risk assessment for the complete robotic arm system.
- Connect additional safety devices for other machinery as defined in the risk assessment.
- Ensure the accurate design and installation of the entire robotic arm system, including both software and hardware.
- The integrator and end customer must follow relevant regulations and applicable legal requirements for safety assessment to ensure that the developed robotic arm has no major hazards or safety risks in actual applications.
- Be aware of any potential safety risks before operating and using the equipment.
- Ensure that users do not modify any safety measures.
- Collect all documents in the technical files, including the risk assessment and this manual.

2. Environment

- Before first use, carefully read this manual to understand basic operations and usage guidelines.
- Choose a relatively open area for use, as the robotic arm does not come with any automatic obstacle-avoidance sensors.
- Use the robotic arm in an environment with a temperature between -20°C and 50°C.
- If the robotic arm is not custom-made with a specific IP protection rating, its water and dust resistance is rated at IP22.

3. Check

- Ensure the robotic arm has no visible abnormalities.
- Ensure proper connection of the wiring harness during use.

4. Operation

- Ensure the surrounding area is relatively open during operation.
- Operate within visual range.
- The maximum payload of the robotic arm is 1.5KG; ensure that the effective load does not exceed 1.5KG during use.
- If the equipment shows abnormalities, stop using it immediately to avoid secondary damage.
- If abnormalities occur, contact relevant technical personnel and do not handle it on your own.
- Use the equipment in an environment that meets the IP protection rating requirements.

5. Warnings⚠:

- Ensure that the robotic arm and tools/end effectors are always correctly and securely fixed in place.
- If you must enter the robotic arm's working space, wear safety goggles to protect yourself.
- Ensure that the robotic arm has enough space for free movement.
- Ensure that safety measures are established as defined in the risk assessment.
- Do not wear loose clothing while operating the robotic arm. Tie back long hair when operating the robotic arm.
- Do not use the robotic arm if it is damaged or showing any abnormalities.
- If the host computer software displays error messages, immediately perform an emergency stop and contact relevant technical personnel.
- Ensure that people keep their heads, faces, or other body parts away from the operating robotic arm or from the area the robotic arm can reach during operation.

- Never modify the robotic arm. Altering the robotic arm may introduce unforeseen dangers for the integrator.
- Do not expose the robotic arm to permanent magnetic fields. Strong magnetic fields can damage the robotic arm.
- The robotic arm generates heat during operation. Do not handle or touch the robotic arm while it is operating or shortly after it has stopped, as prolonged contact may cause discomfort. Power off the system and wait for one hour for the robotic arm to cool down.
- Connecting different machines together may increase risk or introduce new hazards. Always perform a comprehensive risk assessment for the entire installation. Depending on the risk assessment, different functional safety levels may apply; therefore, when different safety and emergency stop performance levels are required, always choose the highest performance level. Always read and understand the manuals for all devices used in the installation.

目录

1. Introduction	- 5 -
1.1. Product list	- 5 -
1.2. Performance Parameters	- 5 -
2. Basic Usage	- 6 -
2.1. Electrical Interface Introduction	- 6 -
2.2. Robotic Arm Teach/Demonstration Mode Instructions	- 7 -
2.2.1 User Coordinate System	- 7 -
2.2.2 Singular Position	- 8 -
2.2.2.1 Zero position	- 10 -
2.3. Robotic Arm Workspace and Payload	- 11 -
2.4. Mechanical Specifications	- 11 -
2.4.1. Base Installation Instructions	- 11 -
2.4.2. End Part Installation Instructions	- 12 -
2.5. Robotic Arm Teach/Demonstration Mode Instructions	- 12 -
3. Host computer instruction	- 14 -
3.1. ArmRobotUA	- 14 -
3.2. Robotic Arm communication	- 15 -
3.3. Vertical menu bar	- 16 -
3.4. Speed percentage setting	- 17 -
3.5. Robotic arm Enable button	- 17 -
3.6. E-stop button	- 18 -
3.7. Window zoom and close buttons	- 18 -
3.8. Operation function area	- 18 -
3.9. Motion control	- 19 -
3.10. Linkage control area	- 20 -
3.11. Gripper control area (valid only with grippers)	- 21 -
3.12. Joint setting area	- 22 -
3.13. Speed setting area	- 23 -
3.14. Collision grade	- 24 -
3.15. Motor setting	- 25 -
3.16. Upgrade	- 26 -
3.17. 3D simulation model	- 27 -
3.18. TraceLib	- 28 -
3.19. Robotic arm joint status button	- 29 -
3.20. Robotic arm status	- 30 -
4. Secondary Development	- 30 -
4.1. PYTHON-SDK	- 30 -
4.1.1. Install	- 30 -
4.1.2. Quick start	- 30 -
4.1.3. Two pairs of robotic arms(four arms)	- 31 -
4.1.4. Note	- 31 -
4.2. ROS1 driver	- 32 -
4.2.1. Install	- 32 -
4.2.2. Quick start	- 32 -
4.2.3. Launch node of single robotic arm	- 32 -
4.2.4. Note	- 33 -
4.3. Simulation	- 33 -
5. CAN protocol	- 34 -
6. Dimensional drawings	- 35 -
6.1. Zero position	- 35 -
6.2. Axonometric drawing of the robotic arm	- 36 -
6.3. Robotic arm with leader gripper	- 37 -

6.4. Robotic arm with follower gripper.....	- 38 -
6.5. Base.....	- 39 -
6.6. End of the arm	- 40 -
7. Appendix 1: End Tool Parameters	- 41 -

1. Introduction

This 6 DOFs robotic arm is specifically designed for the education and research industry, consumer-level applications, and industrial automation. With a payload capacity of 1.5kg, it is suitable for various research and industrial applications, including humanoid robots, automatic assembly, and automated handling. The six rotating joints provide full-range operational flexibility, ensuring high precision and repeatability. The robotic arm features a modular design, making it easy to maintain and upgrade. It offers an intuitive user interface that simplifies programming and operation, allowing even non-professionals to get started quickly. It can be widely applied in fields such as scientific research, education, automotive manufacturing, electronics assembly, food processing, laboratory automation, and medical equipment operation.

1.1. Product list

Name	Quantity
6 DOFs Robotic arm	1
USB toCAN module	1
Power adapter	1
MicroUSB cable	1
Power and communication cable	1
Base mounting screws	4
Base installation wrench	1

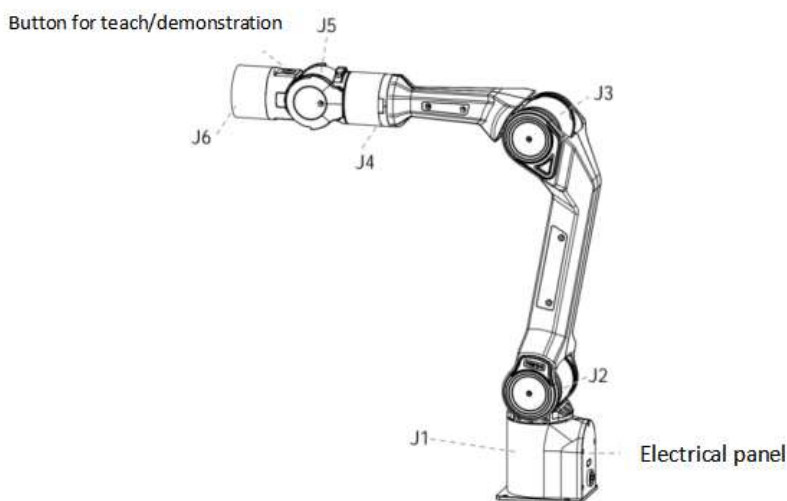
1.2. Performance Parameters

Parameter Type	Item	Specification
Structural Parameters	DOF	6
	Payload	1.5KG
	Weight	4.2KG
	Repeatability	$\pm 0.1\text{mm}$
	Working Radius	626.75mm
	Standard Power Supply Voltage	DC24V (Min. Operating Voltage 24V, Max. Operating Voltage 26V)
	Power Consumption	Max Power Consumption $\leq 120\text{W}$ Average Power Consumption $\leq 40\text{W}$
	Material	Aluminum Alloy Frame Plastic Casing
	Controller	Integrated
	Communication Method	CAN
	Control Method	Drag&Teaching/Offline Trajectory/API/Host PC
	External Interface	1 Power Interface, 1 CAN Interface
	Base Mounting Dimensions	70mm*70mm*M5*4
	Operating Environment	Temperature: -20°C to 50°C Humidity: 25% - 85%, Non-condensing
	Noise	<60dB
Motion Parameters	Installation	Compatible with all products of AgileX Robotics
	Joint Motion Range	J1: $\pm 154^{\circ}$ J2: $0^{\circ}\sim 195^{\circ}$

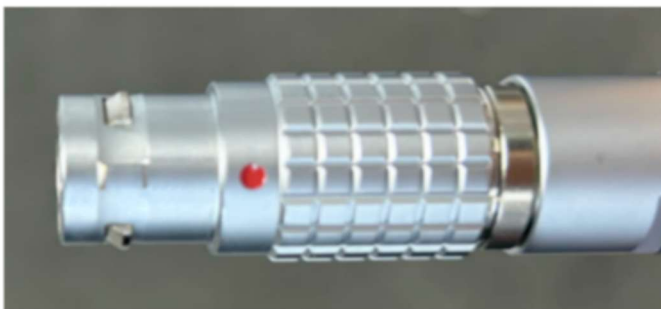
		J3: $-175^{\circ} \sim 0^{\circ}$ J4: $-106^{\circ} \sim 106^{\circ}$ J5: $-75^{\circ} \sim 75^{\circ}$ J6: $\pm 100^{\circ}$
	Maximum joint speed	J1: $180^{\circ}/s$ J2: $195^{\circ}/s$ J3: $180^{\circ}/s$ J4: $225^{\circ}/s$ J5: $225^{\circ}/s$ J6: $225^{\circ}/s$
Note: The above data is the test result of AgileX Robotic Arm in a controlled test environment. The results may vary to varying degrees in different environments and usage methods. Please refer to the actual experience.		

2. Basic Usage

The robotic arm has 6 DOFs and a 1.5kg payload at the end. The six rotating joints provide full-range operational flexibility, ensuring high precision and repeatability. It features a lightweight design, allowing the robotic arm to achieve fast motion capabilities while maintaining a relatively high payload capacity. It can be widely used in embodied intelligence for real-world data collection.



2.1. Electrical Interface Introduction



1: Power and communication port

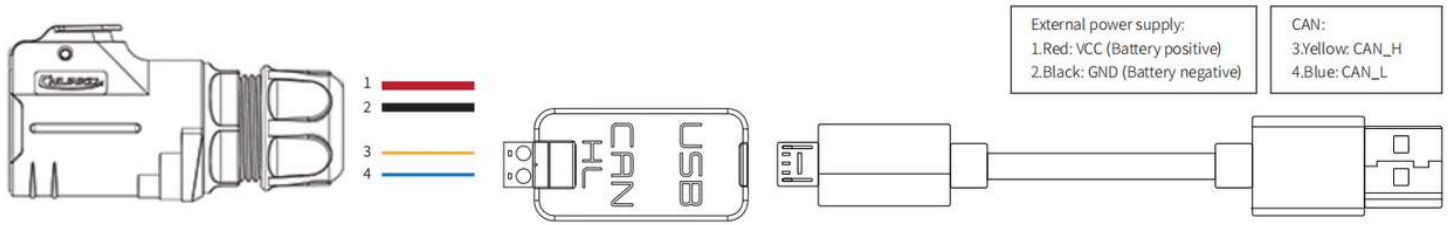
- 2: Status indicator light
- 3: J1J2 connection port
- 4: Power positive
- 5: Power negative
- 6: CAN-H
- 7: CAN-L

Note: Align the red dot with the corresponding red dot on the cable. The textured area of the connector is designed to retract under force. During installation, align the red dot downward with the protruding point and insert it directly. To remove, press down on the textured area and pull it out.

CAN Connection and Preparation

Lead out the CAN cable and connect the CAN_H and CAN_L wires to the CAN_TO_USB adapter.

Connect the CAN_TO_USB adapter to the laptop's USB port. The connection diagram is shown in Figure 3.4.

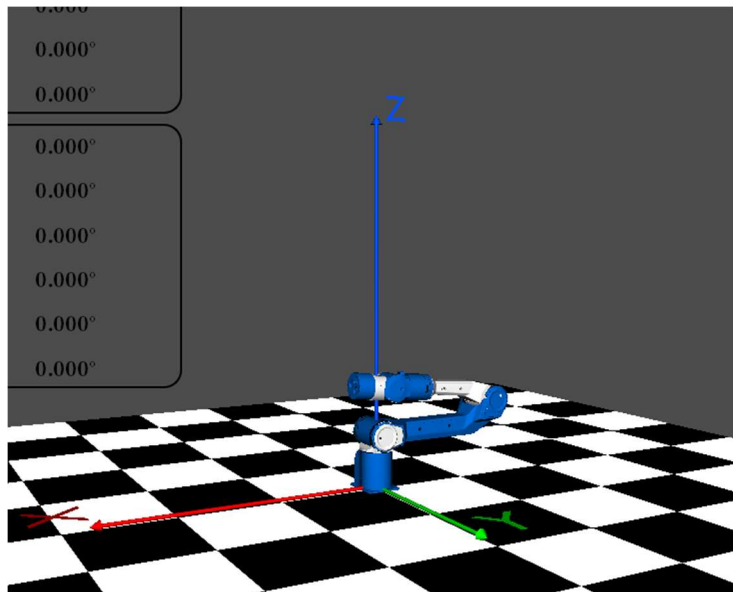


Note: If using a non-standard charger, the power input must not exceed 26V, and the current must be no less than 10A.

2.2. Robotic Arm Teach/Demonstration Mode Instructions

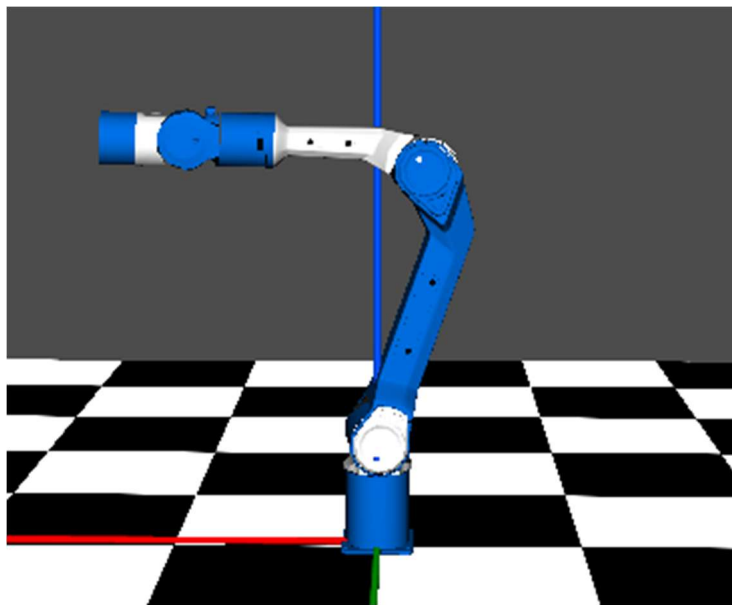
2.2.1 User Coordinate System

The user coordinate system is the robot arm base coordinate system.

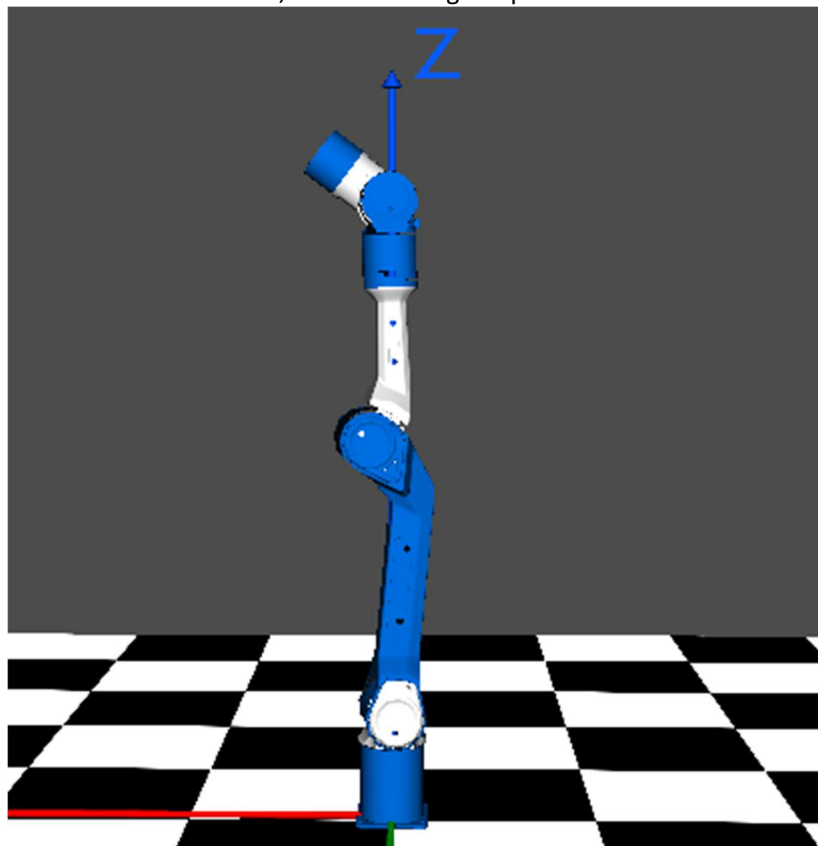


2.2.2 Singular Position

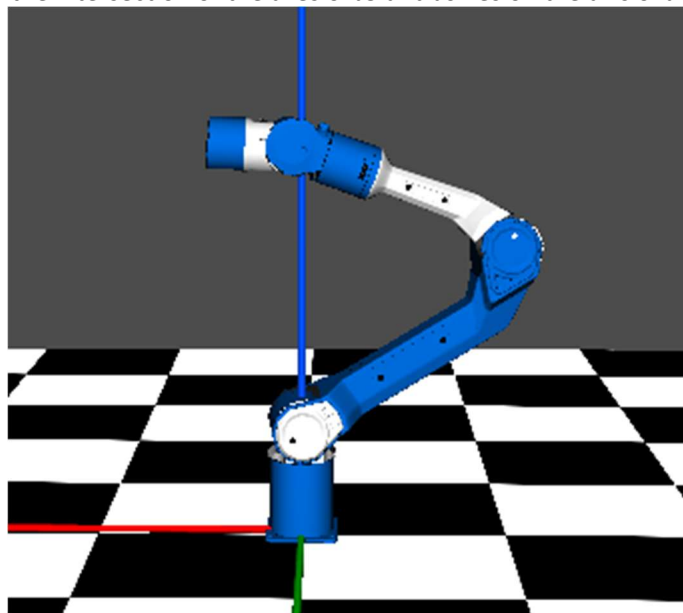
1. Wrist singular position: When J4 and J6 are coaxial, it is a wrist singular position.



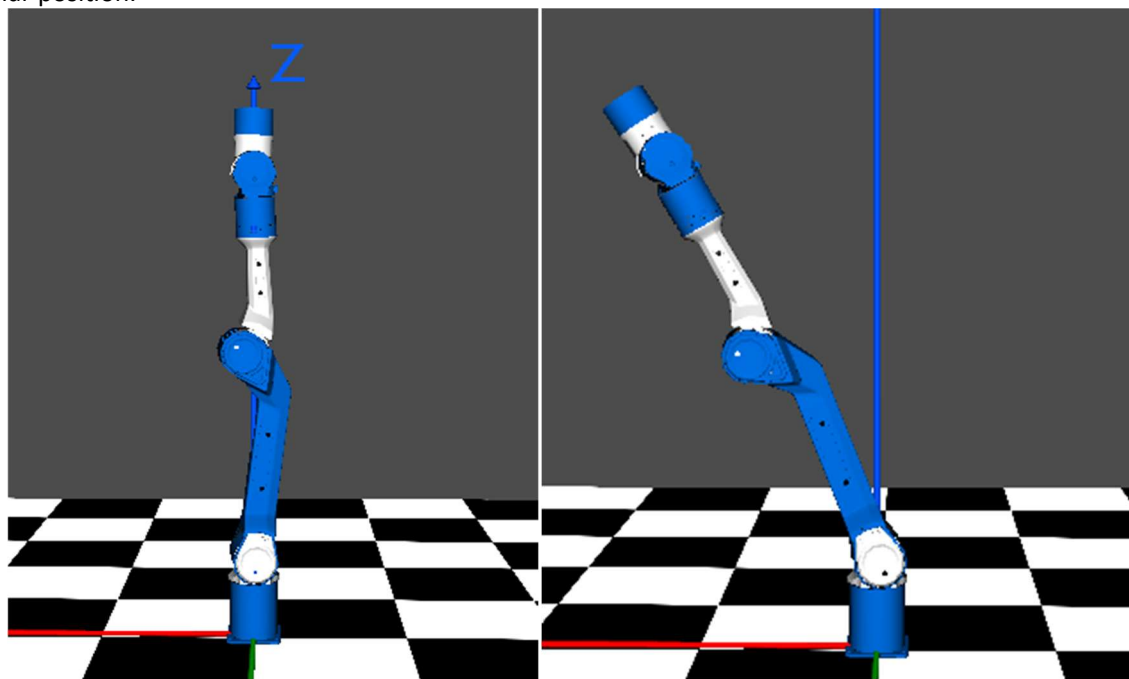
2. Elbow singular position: When J1 and J4 are coaxial, it is elbow singular position.

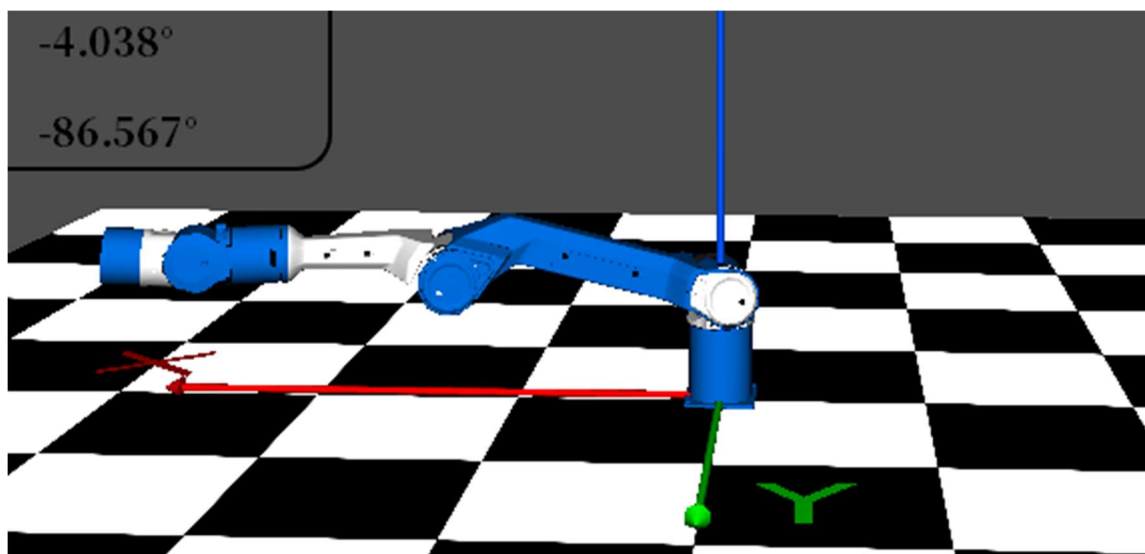


3. Shoulder singular position: when the intersection of the axes of J5 and J6 lies on the axis of J1, it's shoulder singular position.



4. Boundary singular position: It occurs when the end position of the robotic arm is at the maximum working range, resulting in a boundary singular position.

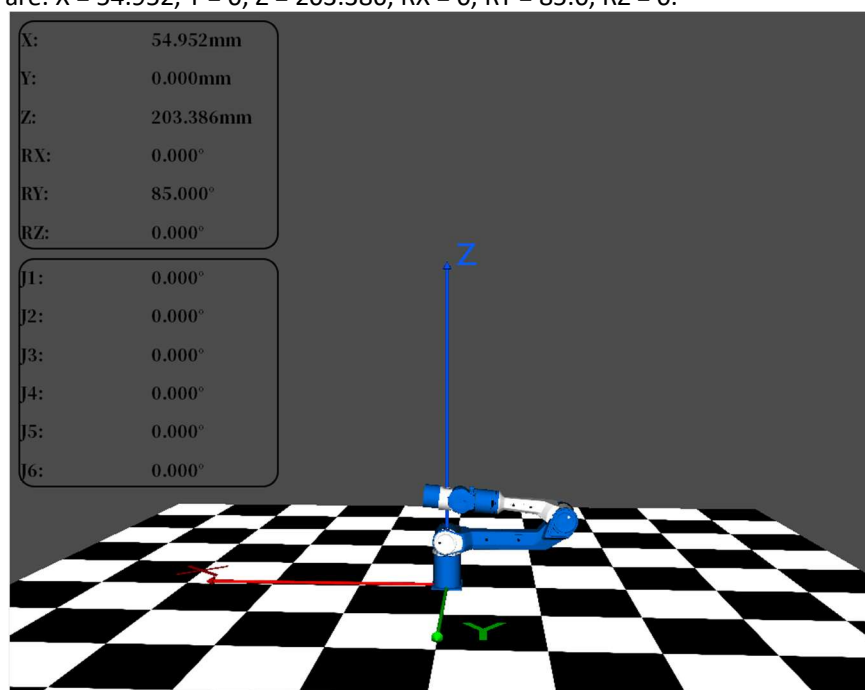




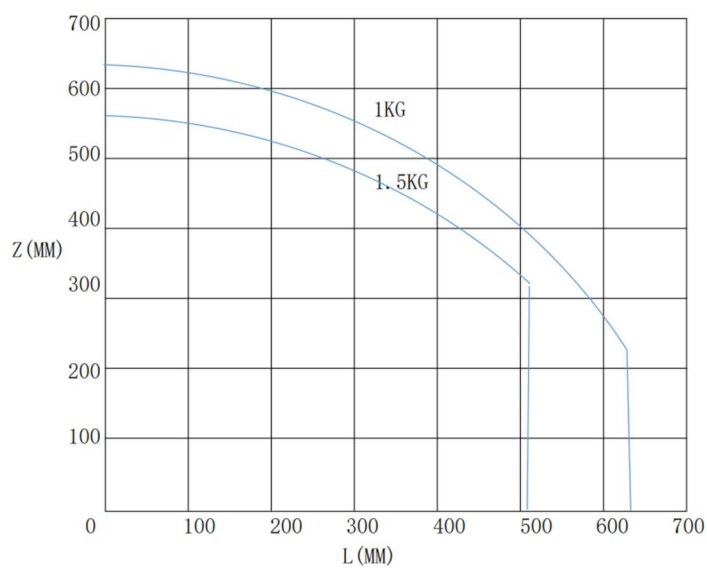
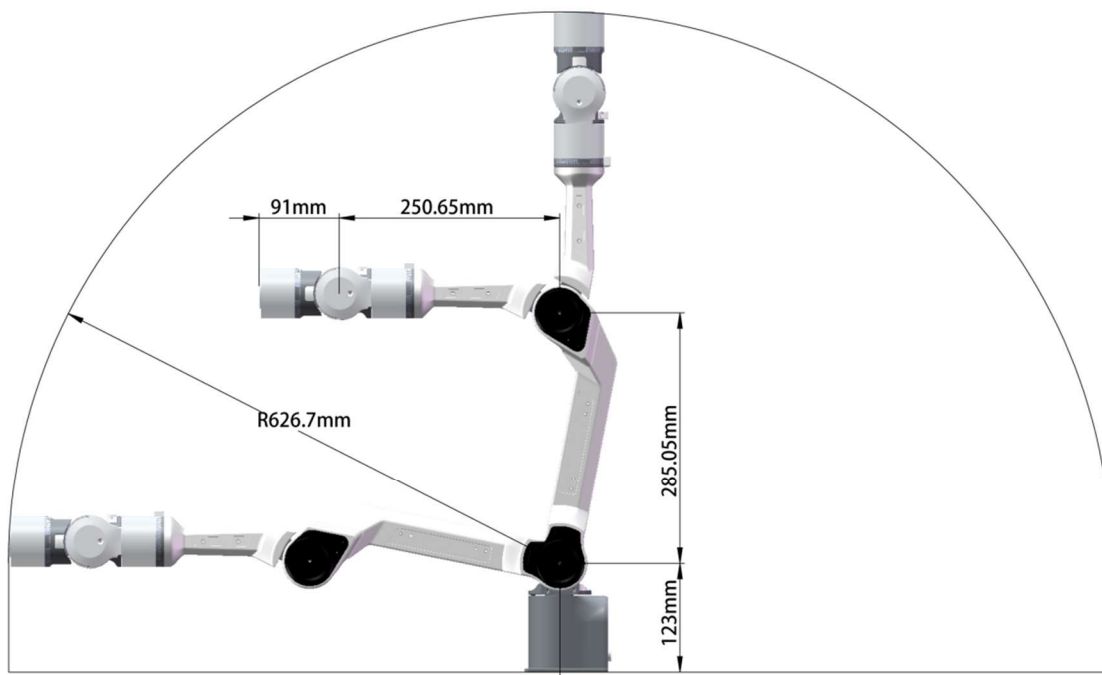
2.2.1. Zero position

The zero position of the robotic arm joints is as shown in the diagram. J1's zero point is aligned with the calibration mark. J2 and J3 reach their zero point when the robotic arm is powered off and naturally hangs down. J5 is at its zero point when the axis of J6 is collinear with J4.

The zero point coordinates are: $X = 54.952$, $Y = 0$, $Z = 203.386$, $RX = 0$, $RY = 85.0$, $RZ = 0$.



2.3. Robotic Arm Workspace and Payload

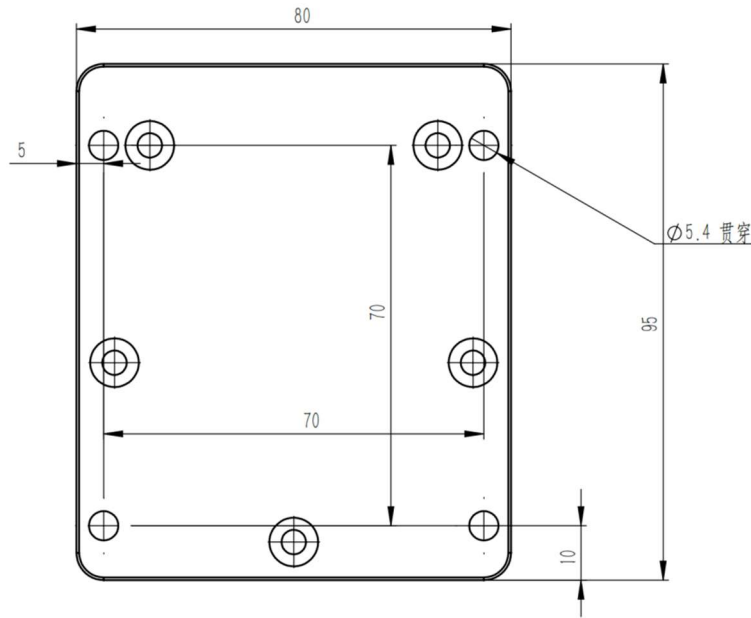


Robotic Arm End-Effector Payload Curve Diagram

2.4. Mechanical Specifications

2.4.1. Base Installation Instructions

The robotic arm base is installed using screws for fixation. The base has four pre-drilled M5 threaded holes. The accessory kit includes four M5 screws, which can be tightened using the provided hex tool. The hole spacing is 70mm. If you need to attach the base to mobile equipment or a fixed surface, you can design the corresponding structure with 70mm hole spacing.



2.4.2. End Part Installation Instructions

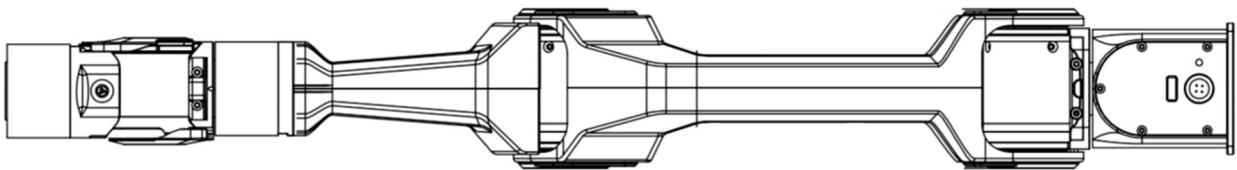
The end can be equipped with other tools via a flange. Optional accessories include a two-finger gripper and a teach pendant. The installation method is shown in the diagram below. Details of the two-finger gripper and teaching device parameters can be found in the technical specifications at the end.

2.5. Robotic Arm Teach/Demonstration Mode Instructions

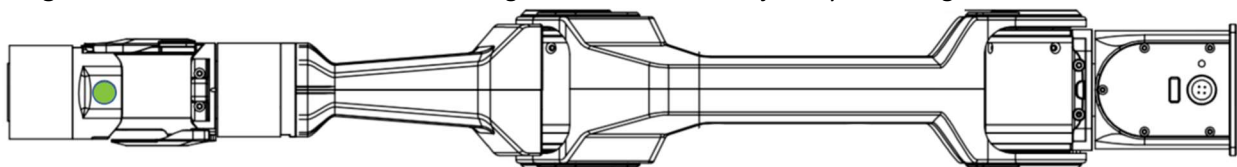
The robotic arm's drag & teach mode status is indicated by the button light between J5 and J6.

There are three types of robotic arm status light displays:

1. **No light display:** Robotic arm's drag & teach mode is stopped, or the drag recording has ended.



2. **Solid green light:** The robotic arm has entered the drag & teach mode for trajectory recording.



3. **Flashing green light:** The robotic arm has entered the drag & teach mode for trajectory playback.

How to switch to drag mode:

1. **Single click button:** Toggle between drag teach trajectory recording and stopping the drag recording.
2. **Double click button:** Activates the drag teach trajectory playback mode.

Instructions:

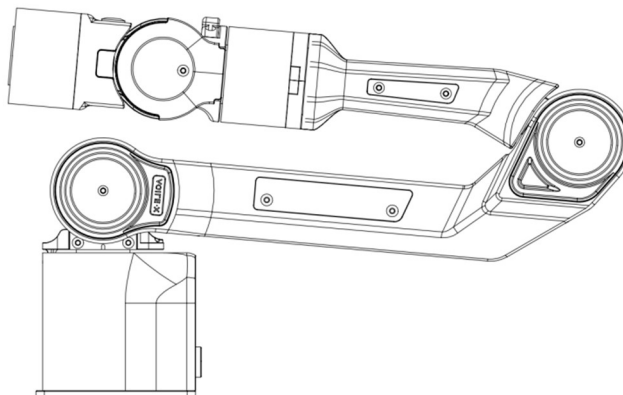
First, observe the indicator light status:

1. If the light is off, click the button once. The green light should turn solid, indicating that the user can drag the robotic arm to start recording the trajectory.
2. If the light is off and a trajectory has been recorded previously, double-click the button. The green light should flash every 500ms, indicating the robotic arm is in playback mode and will reproduce the recorded trajectory.
3. If the light is solid, it indicates trajectory recording is in progress. To stop the recording, click the button once; the light should turn off, indicating that recording has ended. If you want to replay the trajectory, follow step 2.
4. If the light is flashing, the robotic arm is currently in playback mode.

Notes:

1. During trajectory playback, the user must keep a safe distance from the robotic arm to avoid injury.
2. Each time the robotic arm enters teach trajectory recording mode, the previously recorded trajectory is erased. The playback mode will use the most recent recorded trajectory.
3. The maximum trajectory recording time is 3 minutes; any trajectory exceeding this time will be invalid.
4. After finishing drag teaching, ensure the indicator light is off/drag teaching mode is stopped.
5. If you want to switch to host computer control or command control, ensure the indicator light is off/drag teach mode is stopped. Then switch to standby mode via the host computer, and after entering standby mode, switch to CAN mode. The same applies to command control—first switch to standby mode, then switch to CAN control mode.

Note: When switching from link mode and teach by dragging mode to CAN control mode, the robotic arm must be positioned at the zero point before changing modes. The zero point is shown in the figure below:



Robotic Arm Zero Position

3. Host computer instruction

3.1. ArmRobotUA

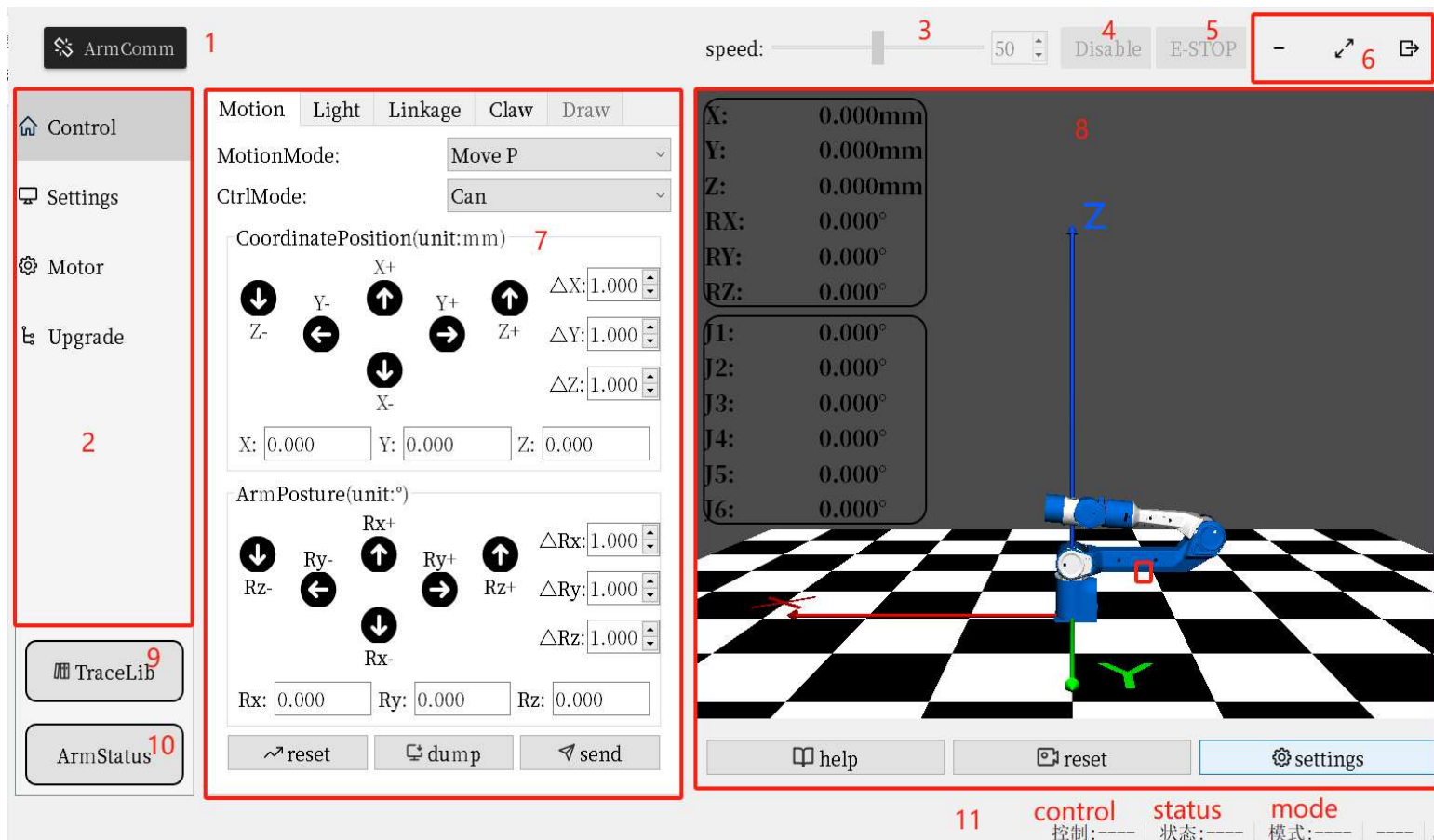
Software Download:

Link: https://drive.google.com/file/d/1g4dJhKQAK9HvbCfGbStzdHnCihfJ_jcn/view?usp=sharing

Using a PC with Windows 7 or higher, double-click to open the host computer software. Through this human-machine interaction software, you can operate the robotic arm and read feedback data from the external network of the robotic arm. The user interface is shown as follows:



Host software



Host Computer Operation Interface

Names of the functional areas in the host computer software panel

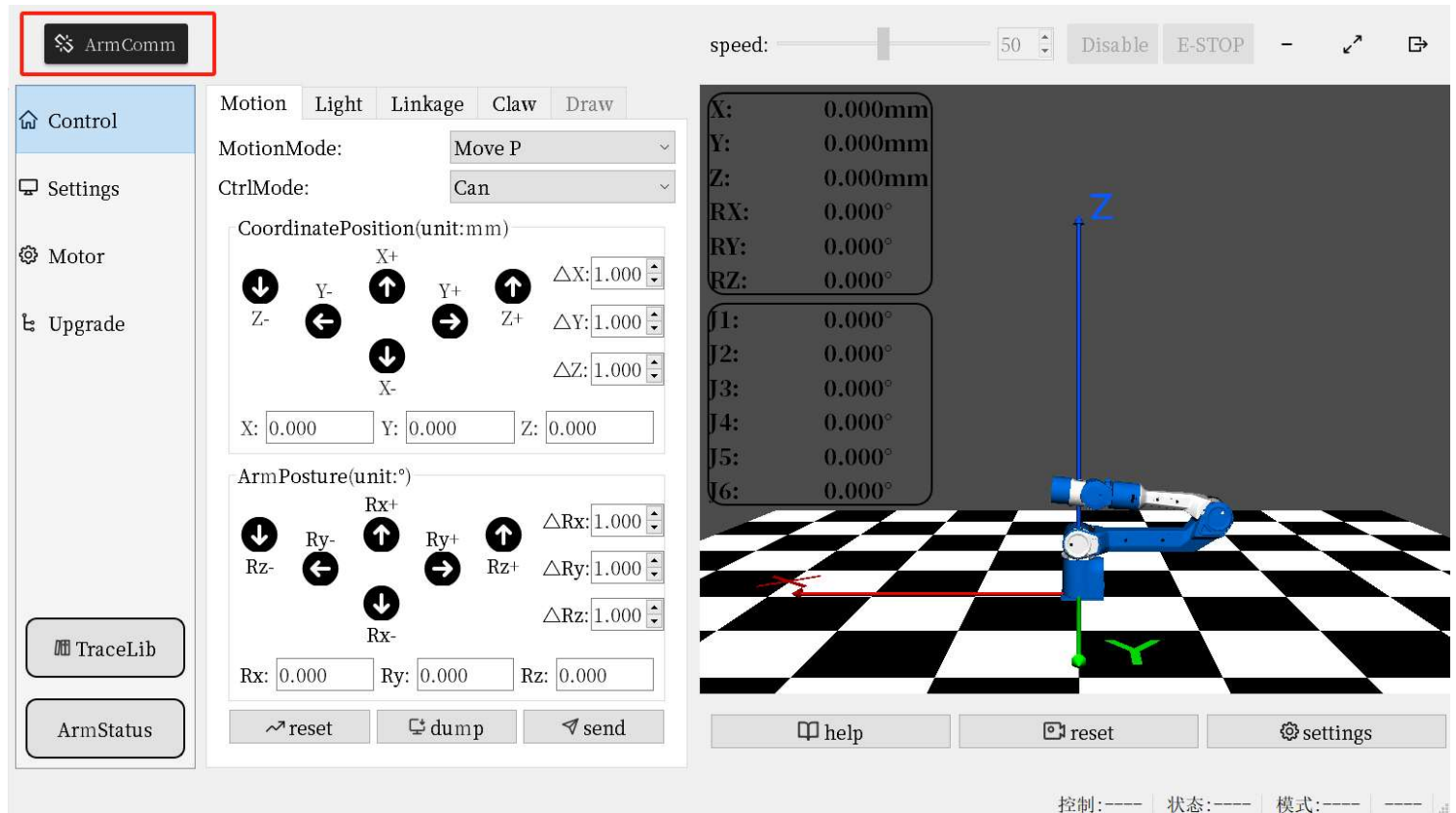
Index	Name
1	Robotic Arm Communication Button
2	Menu Options
3	Speed Percentage Setting
4	Robotic Arm Enable Button
5	Robotic Arm Emergency Stop Button
6	Window Resize/Close Buttons
7	Operation Function Area
8	3D Simulation Model
9	Trajectory Library Function
10	Robotic Arm Joint Status
11	Robotic Arm Status Bar

3.2. Robotic Arm communication

Click setting to switch the language.



Click the robot communication button on the upper left of the window, and a [Connect Robot] window will pop up, which will automatically identify the CAN device connected to the PC device and display it. Users can manually connect and disconnect the selected CAN device. If the connection is successful, the button will monitor and display the current status in real time based on the online status of the connected CAN device.



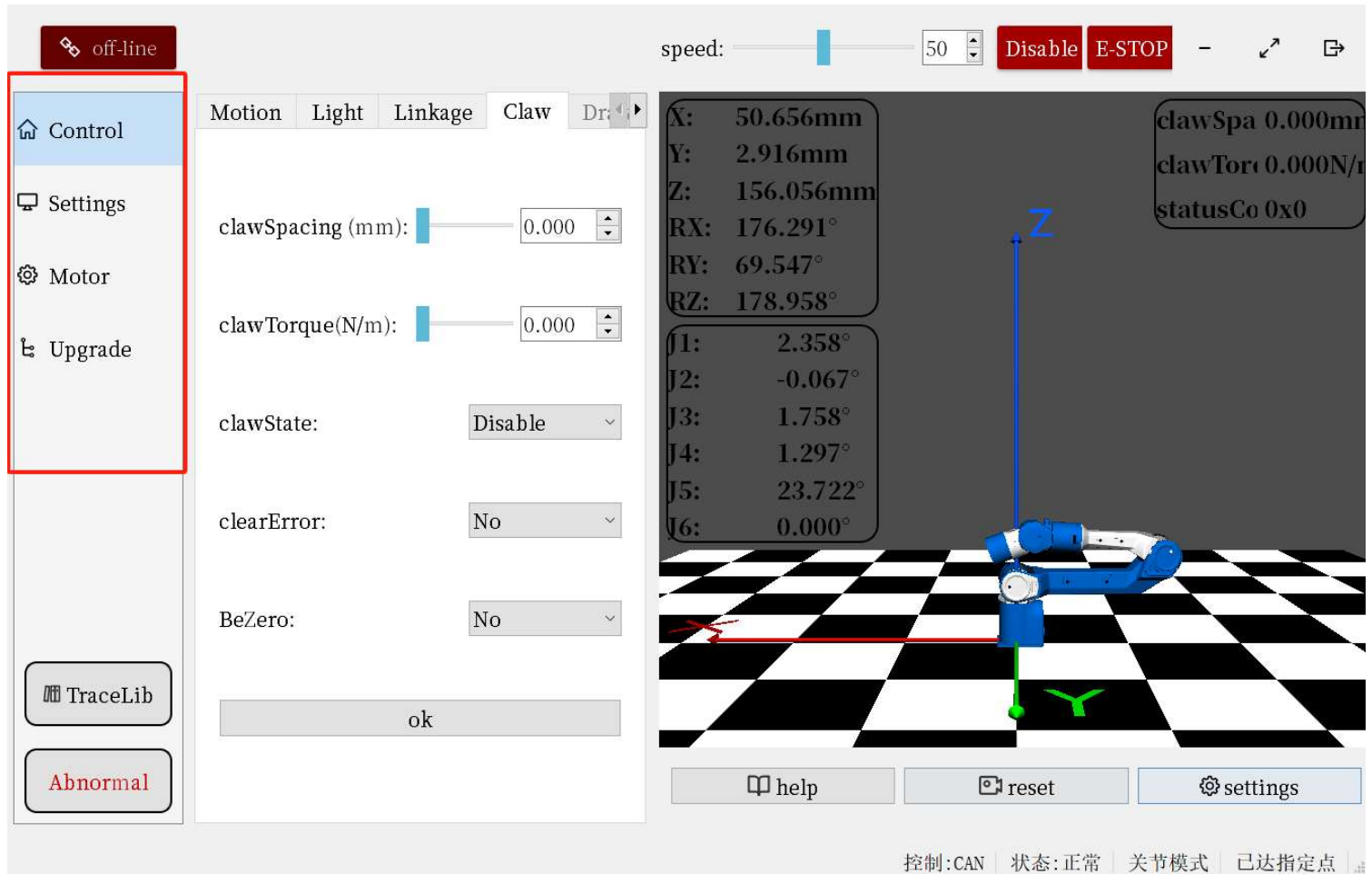
Robotic arm communication button



Robotic arm status display (left: online, right: offline)

3.3. Vertical menu bar

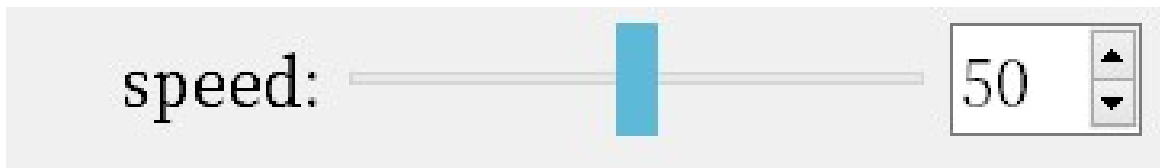
Click the option on the left side of the window to switch to the corresponding panel and select the function you want to operate. The background of the selected item will be light blue.



Vertical menu

3.4. Speed percentage setting

After successfully connecting to the CAN device, the user can change the speed percentage setting at the top of the window to set the movement speed of the robotic arm. The value is the corresponding percentage of the maximum movement speed of the robotic arm.



Speed percentage setting

3.5. Robotic arm Enable button

After successfully connecting to the CAN device, the user can press the robotic arm enable button at the top of the window. The display shows the current status of the six joints. Clicking it will reverse the positions of the six joints at the same time.



Robotic arm Enable button

3.6. E-stop button

After successfully connecting to the CAN device, the user can press the robot arm emergency stop button above the window. After pressing the emergency stop button, the robot arm will immediately stop any movement. Press the button again to resume use. It can be resumed after enabling.



Robotic Arm e-stop button

3.7. Window zoom and close buttons

The three buttons on the upper right are window minimize, window maximize, and window close. Pressing them can realize the corresponding functions of the window.



Window zoom and close button

3.8. Operation function area

The central area of the window is the operation function area, which can be switched synchronously by [2. Vertical menu bar options] to switch to different partitions according to user needs. For example, the figure below is the motion control area, which can realize the actual joint control of the robot arm.

Motion
Light
Linkage
Claw
Draw

MotionMode:
Move J

CtrlMode:
Can

JointDegree:

J1:

—

2.359

+

J2:

—

-0.067

+

J3:

—

1.493

+

J4:

—

1.297

+

J5:

—

23.449

+

J6:

—

0.000

+

↶ reset

✎ edit

Motion control area

3.9. Motion control

The robotic arm has a total of 6 joints, named Joint 1 to Joint 6 from bottom to top, corresponding to the 6 joint settings in the operation bar. Users can edit the joint degrees by themselves or click the plus and minus buttons to realize the joint operation of the robotic arm (unit: $^{\circ}$).

J1:

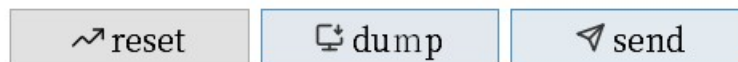
—

0.000

+

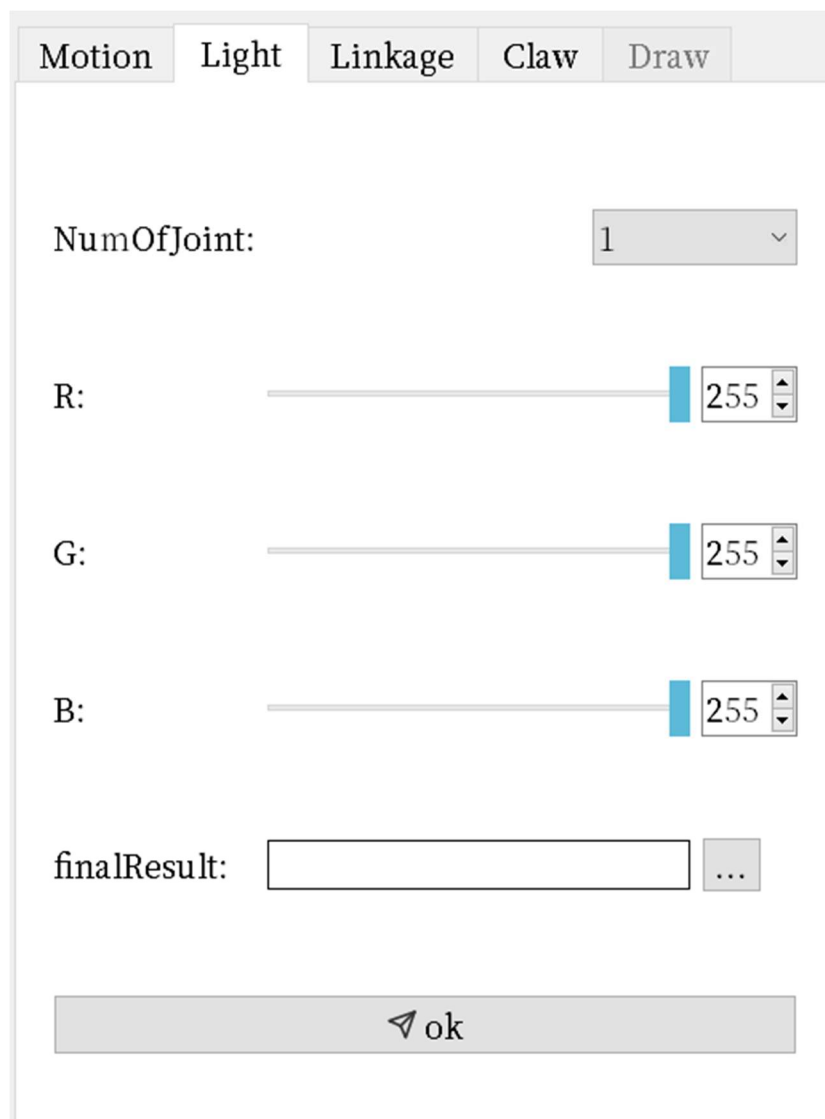
Joint degree

The 3 buttons below are ①Return to zero button, ②Load button, and ③Send button. Button ①: automatically sends the factory initial position angle value of the robot arm, button ②: fills the current degree of the 6 joints of the robot arm into the user edit bar, and button ③: sends the 6 joint degrees in the user edit bar to the robot arm. The specific content is subject to actual experience.



Joint button

The picture below shows the lighting control area. If the joint light board is installed, the lighting control of the six joints of the robot arm can be realized.

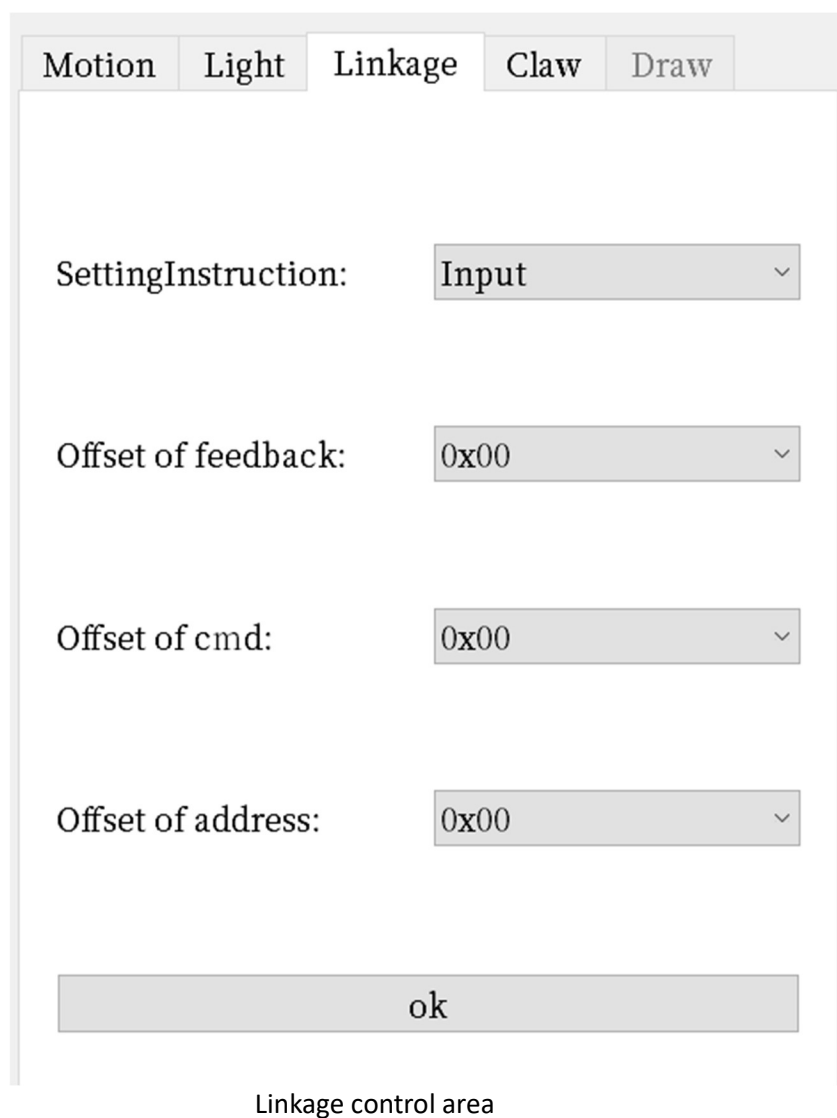


Light control area

The joint number and RGB channel above are the light board that the user needs to operate and the grayscale value of the corresponding channel. The effect bar below is the actual calculated light color. Click the [Confirm] button to operate the light board.

3.10. Linkage control area

The following figure shows the linkage control area. If the user needs to achieve dual-arm linkage, the input and output status and offset value can be configured on this page.



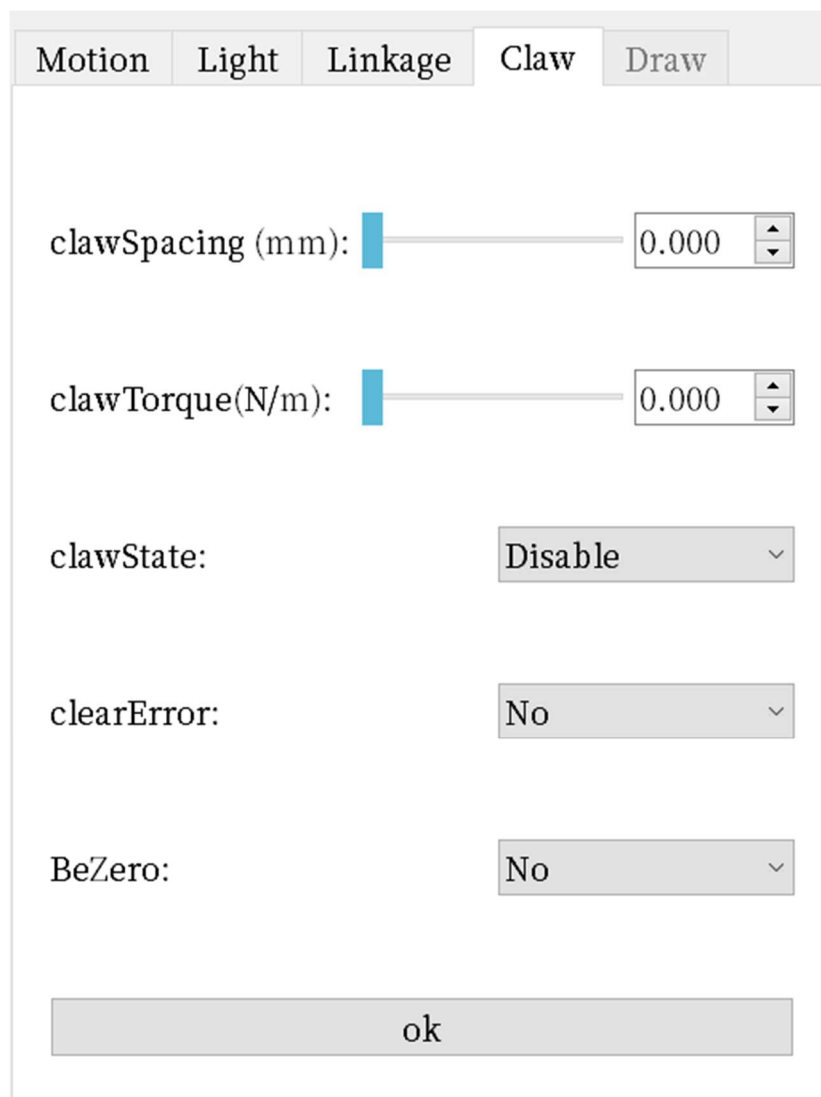
The screenshot shows a software interface with five tabs at the top: Motion, Light, Linkage, Claw, and Draw. The 'Linkage' tab is selected and highlighted. Below the tabs, there are four rows of controls, each consisting of a label and a dropdown menu. The labels are 'SettingInstruction:', 'Offset of feedback:', 'Offset of cmd:', and 'Offset of address:'. The dropdown menus show 'Input', '0x00', '0x00', and '0x00' respectively. At the bottom of the control area is a large 'ok' button. Below the button, the text 'Linkage control area' is centered.

Motion	Light	Linkage	Claw	Draw
SettingInstruction: Input				
Offset of feedback: 0x00				
Offset of cmd: 0x00				
Offset of address: 0x00				
ok				

Linkage control area

3.11. Gripper control area (valid only with grippers)

The following figure shows the gripper control area, where you can set the gripper enable status, torque value, etc. The right area of the host computer will feedback the corresponding gripper value.



The image shows a software interface for configuring a gripper or claw. At the top, there are five tabs: 'Motion', 'Light', 'Linkage', 'Claw', and 'Draw'. The 'Claw' tab is currently selected. Below the tabs, there are five settings:

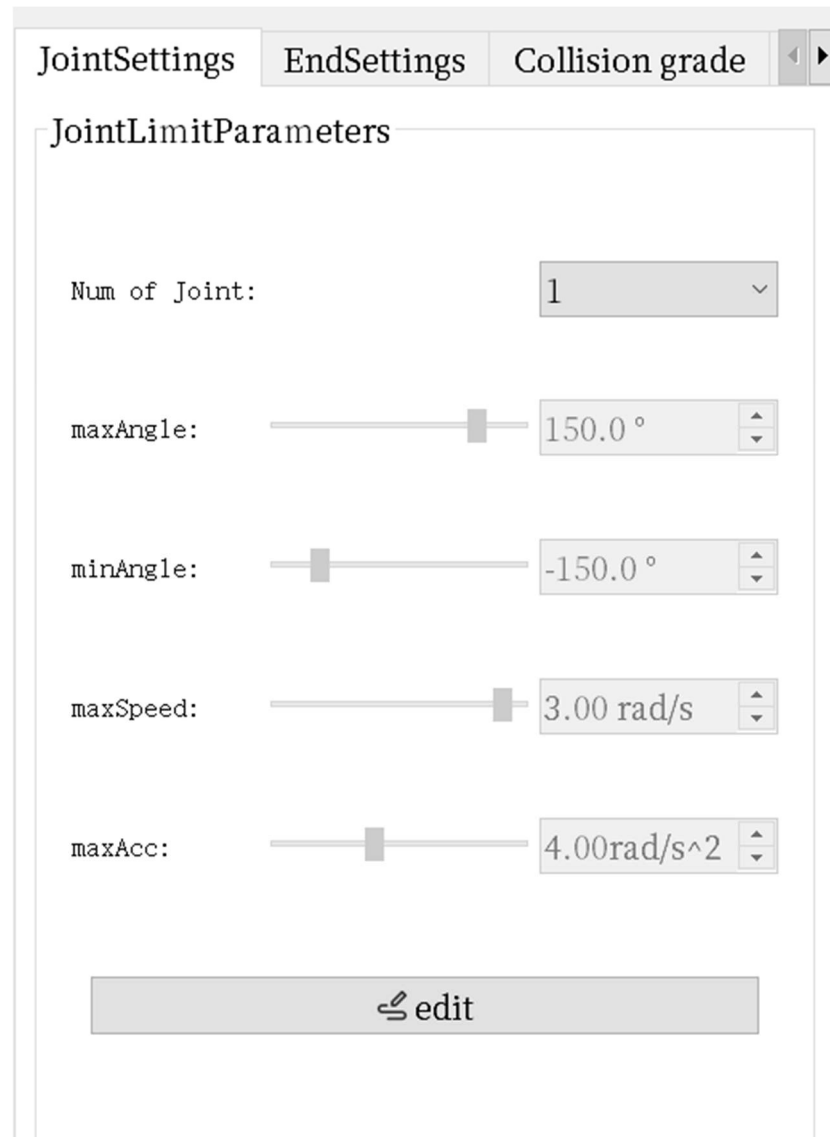
- clawSpacing (mm):** A slider control with a blue bar and a numeric input field showing '0.000'.
- clawTorque(N/m):** A slider control with a blue bar and a numeric input field showing '0.000'.
- clawState:** A dropdown menu with 'Disable' selected.
- clearError:** A dropdown menu with 'No' selected.
- BeZero:** A dropdown menu with 'No' selected.

At the bottom of the settings area is a large 'ok' button.

Gripper/claw setting area

3.12. Joint setting area

The figure below shows the joint setting area, where you can edit the maximum and minimum angles, maximum speed, and maximum acceleration of the six joints of the robot arm. The operation process is [Edit] - [Save].



The screenshot shows a software interface with three tabs: "JointSettings", "EndSettings", and "Collision grade". The "JointSettings" tab is active. Below the tabs is a section titled "JointLimitParameters". Inside this section, there are five parameters, each with a slider and a numeric input field:

- Num of Joint:** A dropdown menu showing the value "1".
- maxAngle:** A slider with a value of "150.0 °".
- minAngle:** A slider with a value of "-150.0 °".
- maxSpeed:** A slider with a value of "3.00 rad/s".
- maxAcc:** A slider with a value of "4.00rad/s^2".

At the bottom of the "JointLimitParameters" section is a button labeled "edit" with a pencil icon.

Joint setting area

3.13. Speed setting area

The following figure shows the speed setting area, where the maximum linear speed and angular speed and the maximum linear acceleration and angular acceleration of the end of the robot arm can be set. The [Initialization button] is used to restore the factory settings.

JointSettings

EndSettings

Collision grade

endLoad

loadValue: Null ok

speedSettings

maximumLinearSpeed: 0.000 m/s

maxAngularVelocity: 0.000 rad/s

maxLinearAcc: 0.000m/s^2

maxAngularAcc: 0.000 rad/s

edit

End speed setting area

reset

Reset area

3.14. Collosion grade

The picture shows the collision level setting. The collision level of level 1 to level 8 can be configured for the 6 joints of the robot arm. The operation process is [Edit] - [Save].

The screenshot shows a software interface for setting collision grades. It features three tabs: 'JointSettings', 'EndSettings', and 'Collision grade'. The 'Collision grade' tab is selected, displaying a 'Level' section with six sliders for joints J1 through J6. Each slider is currently set to 0. Below the sliders are two buttons: 'cancel' and 'save'.

碰撞等级设置区

3.15. Motor setting

The figure below shows the motor settings, which can be used to set the status (enable, disable), zero position setting, and clear errors of the six joint motors of the robot arm.



The screenshot shows a web interface titled "motorSettings". Inside, there is a sub-section also labeled "motorSettings". This section contains three rows of controls:

- motorStatus:** A dropdown menu currently showing "J1", followed by another dropdown menu showing "Disable", and a button with a right-pointing arrow and the text "ok".
- zeroPointSetting:** A dropdown menu currently showing "J1", followed by a button with a right-pointing arrow and the text "ok".
- clearError:** A button with a right-pointing arrow and the text "ok".

Below the interface, the text "Motor setting" is centered.

3.16. Upgrade

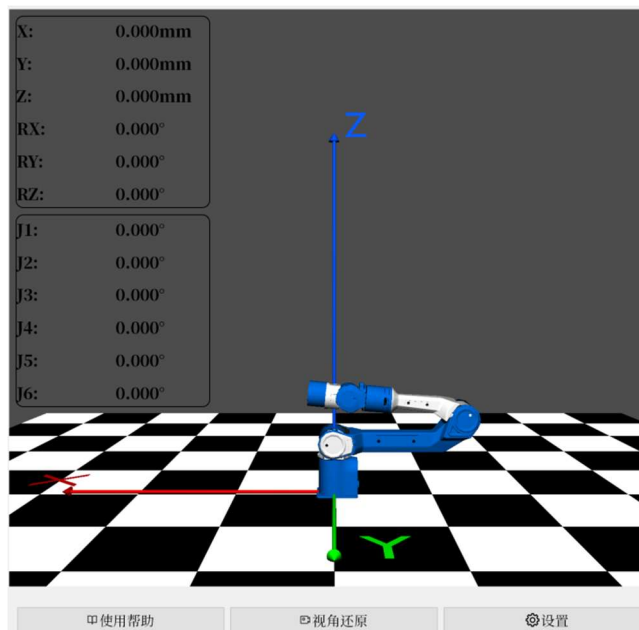
The following figure is the page related to the robot version upgrade. The upgrade steps are as follows:

- ① Check whether there are words such as node information generated in the information display area (area 1). If there are words, it means that the robot has entered the upgrade state normally. If the check is correct, it can be upgraded normally;
- ② Click the [Load Software Package] button (area 5), select the version firmware configured by R&D (suffixed with bin), and after confirmation, the information area (area 4) will display the firmware package information selected by the user;
- ③ Click the [Start Upgrade] button (area 6) and wait for the progress bar (area 2) to complete. The normal time is about 2~3 minutes.



3.17. 3D simulation model

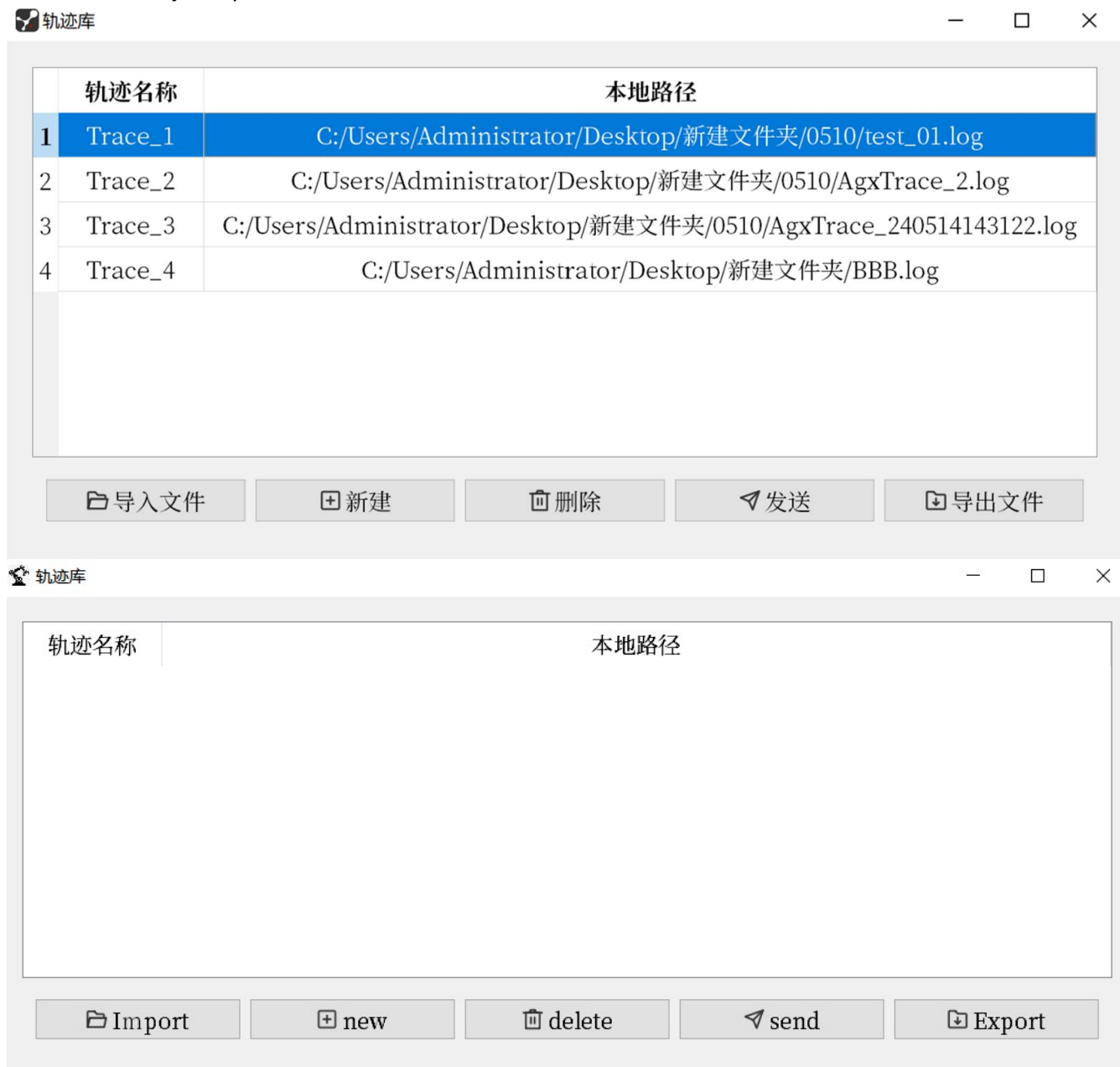
On the right side of the window is the real-time 3D simulation model of the robot arm. The upper left corner shows the real-time feedback coordinate values and 6 joint degrees of the current robot arm. The three buttons below are ① Help, ② View Restore, and ③ Settings. Button ①: Pop-up window displays the usage of the 3D area, button ②: Restore the original 3D display view, and button ③: Set various parameters of the host computer. The specific content is subject to actual experience.



3D simulation model

3.18. TraceLib

Click the [TraceLib] below the vertical menu bar, and a pop-up window will appear. Users can customize and edit a complete robot arm motion trajectory and run it.



轨迹库弹窗

- ① Import: Import the local trajectory library log file.
- ② Create: Create a new motion trajectory.
- ③ Delete: Delete the selected trajectory.
- ④ Send: Send the selected trajectory to make the robot arm run the trajectory once, multiple times, or in a cycle.
- ⑤ Export file: Export the selected trajectory to the local.

At the same time, the user can double-click any trajectory to enter the secondary menu, edit the specific content of the trajectory (as shown below), and right-click to add, delete, modify, and check.



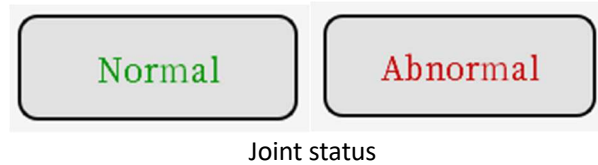
TraceLib secondary menu

3.19. Robotic arm joint status button

Click the robot arm joint status button at the bottom left of the window, and a [Joint Status] window will pop up (as shown below), which will monitor the status information of the 6 joints in real time. The button will monitor and display the current status in real time according to the status of the connected joints (as shown below).

	J1	J2	J3	J4	J5	J6
StatureCode:	0x40	0x40	0x40	0x40	0x40	0x40
motorSpeed(rpm):	-3	-38	0	17	-34	138
Degree(°):	2.258	-0.067	1.493	1.297	23.449	0
Voltage(V):	23	23	23	23	23	23
Current(mA):	49	50	9	4	4	20
motorTemperature(°C):	30	30	30	31	31	31
driverTemperature(°C):	35	34	34	35	36	36
collision:	normal	normal	normal	normal	normal	normal
blockage:	normal	normal	normal	normal	normal	normal
errorState:	normal	normal	normal	normal	normal	normal
EnableState:	Enable	Enable	Enable	Enable	Enable	Enable
Comm:	normal	normal	normal	normal	normal	normal
AngleState:	normal	normal	normal	normal	normal	normal

Joint status window



Joint status

3.20. Robotic arm status

The robot arm status bar at the bottom right of the window will monitor the robot arm status information in real time (as shown below).



4. Secondary Development

Currently, the robotic arm supports secondary development through a Python SDK and ROS1 & ROS2 driver package. For detailed secondary development instructions, please refer to the GitHub link.

SDK:https://github.com/agilexrobotics/piper_sdk

ROS1:https://github.com/agilexrobotics/piper_ros

4.1. PYTHON-SDK

This SDK is used to receive CAN data frames and then process them into custom data types, excluding data offset frames.

Test system environment: ubuntu20.04

Python version: python3.8

4.1.1. Install

Install dependency

```
pip3 install python-can
```

```
pip3 install piper_sdk
```

Install can tool

```
sudo apt update && sudo apt install can-utils ethtool
```

These two tools are used to configure the can module.

If you run the bash script and get ip: command not found, please install the ip command, which is usually

```
sudo apt-get install iproute2
```

4.1.2. Quick start

Enable can module

First set shell script parameters

Single robotic arm

The PC only has one USB to CAN module inserted

The can_activate.sh script is used here

Directly run

```
bash can_activate.sh can0 1000000
```

PC with Multiple USB-to-CAN Modules Inserted

Use the can_activate.sh script here.

Unplug all CAN modules.

Only insert the CAN module connected to the robot into the PC and execute:

```
sudo ethtool -i can0 | grep bus
```

and record the bus-info value, for example, 1-2:1.0.

Note: Generally, the first inserted CAN module defaults to can0. If the CAN interface is not found, use bash find_all_can_port.sh to check the CAN names corresponding to the USB addresses.

Assuming the recorded bus-info value from the above operation is 1-2:1.0.

Then execute the command to check if the CAN device has been successfully activated.

```
bash can_activate.sh can_piper 1000000 "1-2:1.0"
```

Note: This means that the CAN device connected to the USB port with hardware encoding 1-2:1.0 is renamed to can_piper, set to a baud rate of 1,000,000, and activated.

Then run ifconfig to check if can_piper appears. If it does, the CAN module has been successfully configured.

4.1.3. Two pairs of robotic arms (four arms)

For four robotic arms, which means two pairs of master-slave robotic arms:

Use the can_config.sh script here

In the can_config.sh, set the EXPECTED_CAN_COUNT parameter to 2, as four robotic arms use two CAN modules.

Then, insert one of the two CAN modules (usually the one connected to the left arm) into the PC alone and execute the script.

```
sudo ethtool -i can0 | grep bus
```

and record the bus-info value, for example, 1-2:1.0.

```
sudo ethtool -i can1 | grep bus
```

Next, insert the second CAN module, ensuring it is connected to a different USB port than the one used previously, and then execute the script.

Note: Generally, the first inserted CAN module defaults to can0, and the second one to can1. If the CAN interfaces are not found, use bash find_all_can_port.sh to check the CAN names corresponding to the USB addresses.

Assuming the recorded bus-info values are 1-2:1.0 and 1-4:1.0, replace the parameters inside the double quotes in USB_PORTS["1-9:1.0"]="can_left:1000000" with 1-2:1.0.

Similarly, update the other one

```
USB_PORTS["1-5:1.0"]="can_right:1000000" -> USB_PORTS["1-4:1.0"]="can_right:1000000"
```

Note: This means that the CAN device connected to the USB port with hardware encoding 1-2:1.0 is renamed to can_left, set to a baud rate of 1,000,000, and activated.

Then execute bash can_config.sh and check the terminal output to see if the activation was successful.

Afterward, run ifconfig to verify if can_left and can_right appear. If they do, the CAN modules have been successfully configured.

4.1.4. Note

You need to activate the CAN device and set the correct baud rate before you can read the robot arm message or control the robotic arm.

When the `C_PiperInterface` interface class is instantiated, you can pass in the name of the activated CAN route, which can be obtained through `ifconfig`.

Sometimes when executing CAN send, the terminal feedback Message NOT sent, which means that the CAN module has not successfully connected to the device. First check the connection status between the module and the robot arm, then power off and then power on the robot arm, and then try to send CAN again.

4.2. ROS1 driver

Test environment: ubuntu20.04

Ros version: noetic

4.2.1. Install

Install dependency

```
pip3 install python-can
```

```
pip3 install piper_sdk
```

4.2.2. Quick start

Enable can module

First set shell script parameters

Single robotic arm

The PC only has one USB to CAN module inserted

The `can_activate.sh` script is used here

Directly run

```
bash can_activate.sh can0 1000000
```

PC with Multiple USB-to-CAN Modules Inserted

Use the `can_activate.sh` script here.

Unplug all CAN modules.

Only insert the CAN module connected to the robot into the PC and execute:

```
sudo ethtool -i can0 | grep bus
```

and record the bus-info value, for example, 1-2:1.0.

Note: Generally, the first inserted CAN module defaults to `can0`. If the CAN interface is not found, use `bash find_all_can_port.sh` to check the CAN names corresponding to the USB addresses.

Assuming the recorded bus-info value from the above operation is 1-2:1.0.

Then execute the command to check if the CAN device has been successfully activated.

```
bash can_activate.sh can_piper 1000000 "1-2:1.0"
```

Note: This means that the CAN device connected to the USB port with hardware encoding 1-2:1.0 is renamed to `can_piper`, set to a baud rate of 1,000,000, and activated.

4.2.3. Launch node of single robotic arm

Start node

```
roscore
```

```
roslaunch piper piper_ctrl_single_node.py _can_port:=can0 _mode:=0# Start launch
```

```
roslaunch piper start_single_piper.launch can_port:=can0 auto_enable:=true# Or, it will run with default parameters
```

```
roslaunch piper start_single_piper.launch# You can also use rviz to start control, the parameters that need to be changed are as above
```

```
roslaunch piper start_single_piper_rviz.launch
```

4.2.4. Note

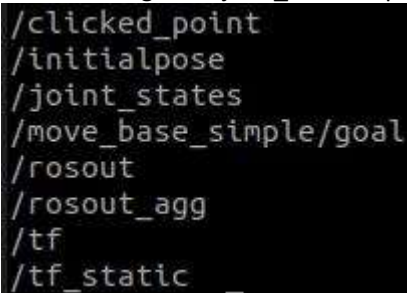
- You need to activate the CAN device and set the correct baud rate before you can read messages from or control the robotic arm.
- To control two slave robotic arms using ROS, you must first disconnect the main arm's connector!!!
- When the node parameter mode is set to 1, the topics `/master/joint_left`, `/master/joint_right`, `/puppet/joint_left`, and `/puppet/joint_right` are data topics only. You only need to read from them. Typically, in this mode, the CAN communication between the master and slave arms is established.
- When the node parameter mode is set to 0, `/master/joint_left` and `/master/joint_right` are topics for controlling the slave arms, while `/puppet/joint_left` and `/puppet/joint_right` are for feedback from the slave arms. To send movement commands to the slave arms via `/master/joint_left` and `/master/joint_right`, you must first send `true` to the `/enable_flag` topic.

4.3. Simulation

`display_xacro.launch` open rviz

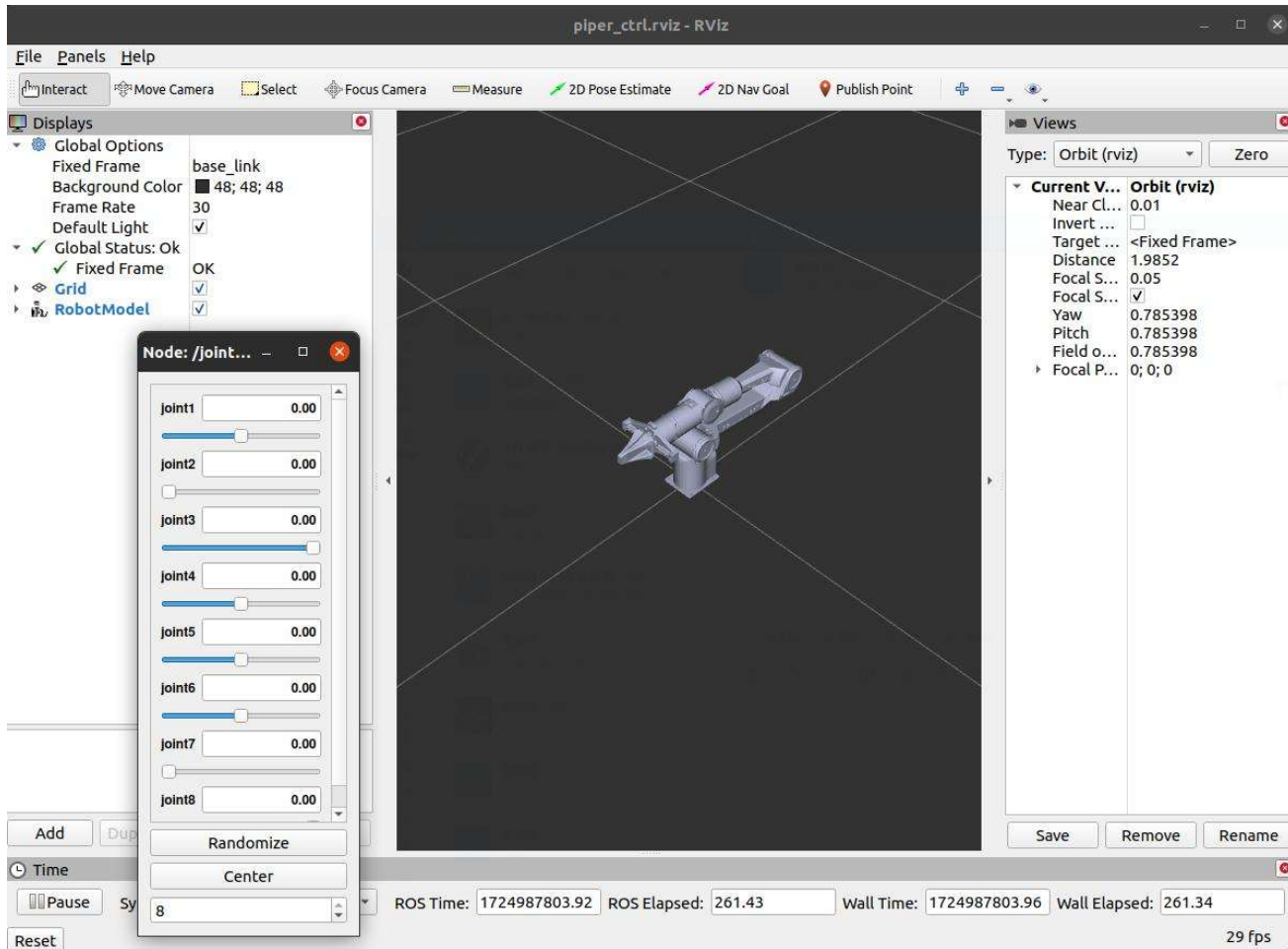
```
roslaunch piper_description display_xacro.launch
```

After running, the `/joint_states` topic will be published. You can view it by `/joint_states`:

A screenshot of a terminal window with a black background and white text. The text lists several ROS topics: `/clicked_point`, `/initialpose`, `/joint_states`, `/move_base_simple/goal`, `/rosout`, `/rosout_agg`, `/tf`, and `/tf_static`. The `/joint_states` topic is highlighted with a light blue background.

```
/clicked_point
/initialpose
/joint_states
/move_base_simple/goal
/rosout
/rosout_agg
/tf
/tf_static
```

Two windows will appear simultaneously as follows. The slider values correspond to the `/joint_states` values. Dragging the sliders will change these values, and the model in rviz will update accordingly.



5. CAN protocol

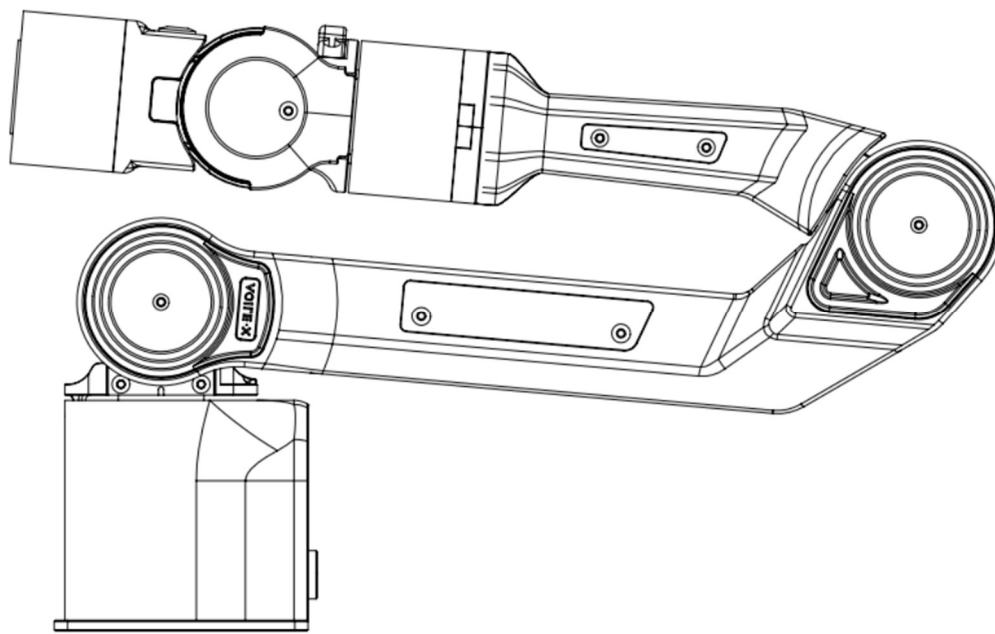
The robot arm provides a CAN interface for user development, through which users can control the robot arm with commands.

The CAN communication standard in the robot arm adopts the CAN2.0B standard, the communication baud rate is 1000K, and the message format adopts the MOTOROLA (MSB) format.

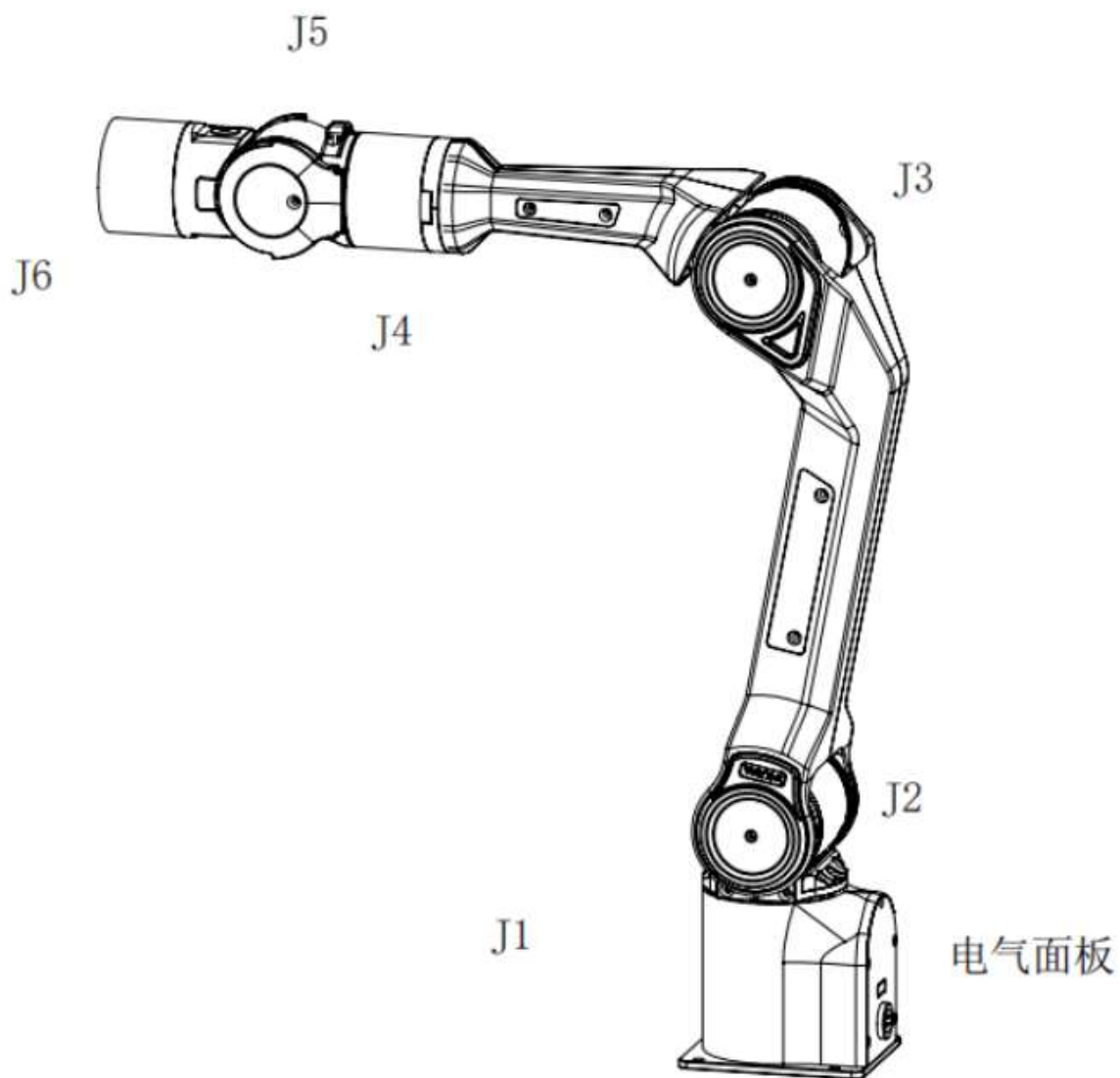
If you need to use CAN command development, please contact Songling Technical Support to obtain the CAN protocol manual.

6. Dimensional drawings

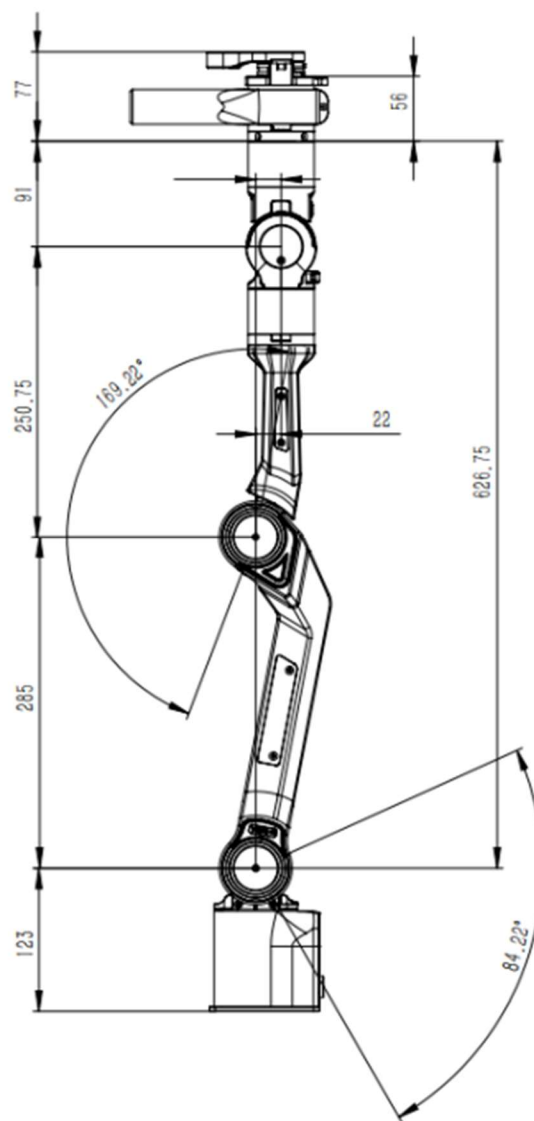
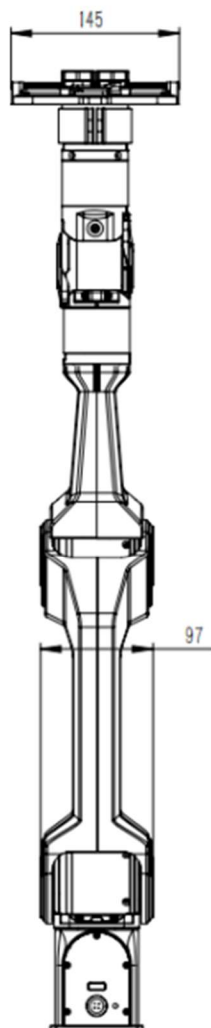
6.1. Zero position



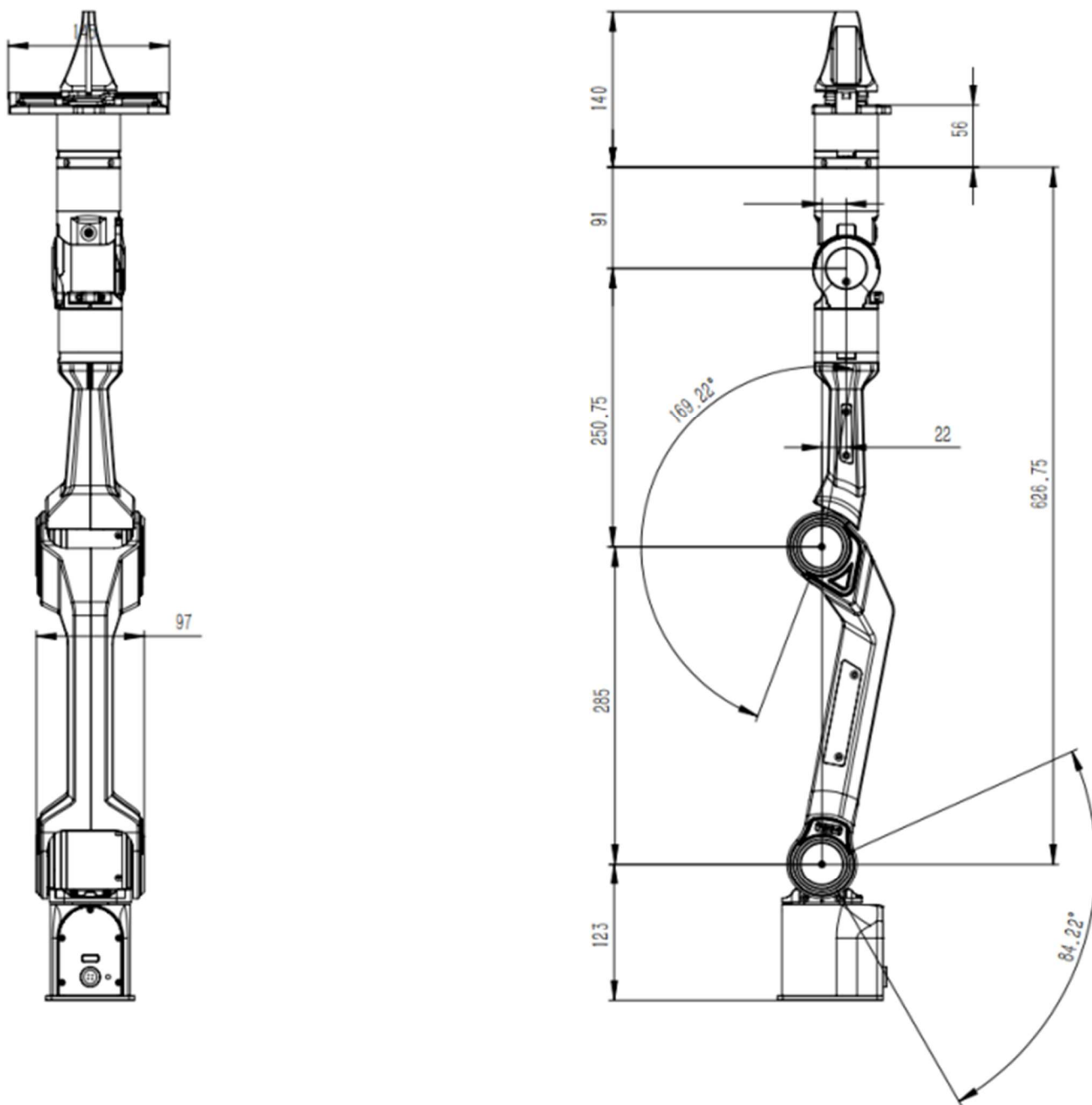
6.2. Axonometric drawing of the robotic arm



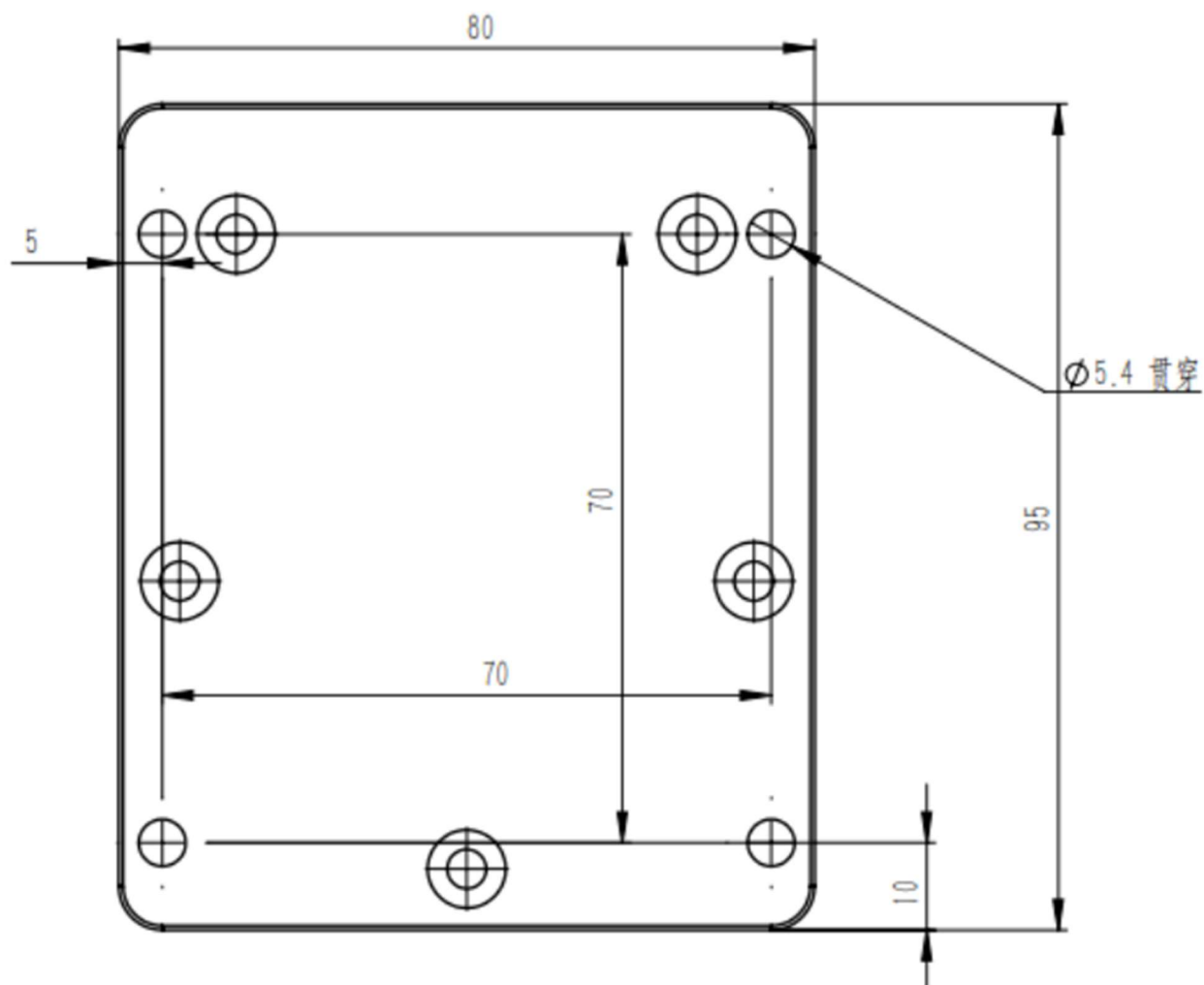
6.3. Robotic arm with leader gripper



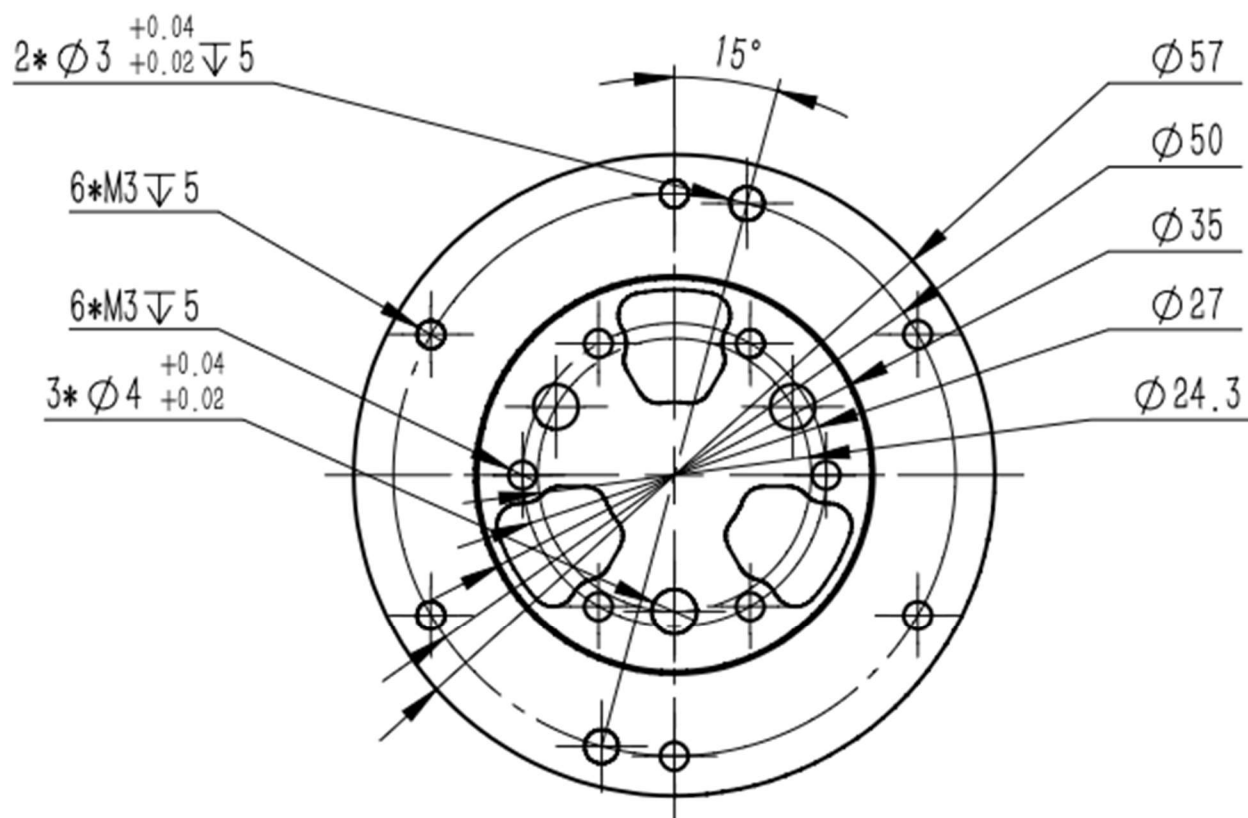
6.4. Robotic arm with follower gripper



6.5. Base



6.6. End of the arm



7. Appendix 1: End Tool Parameters

Optional follower Gripper Specifications:

Two-Finger Gripper Parameters	
Weight	500g
Accuracy	$\pm 0.5\text{mm}$
Opening Distance	0-70mm
Rated Clamping Force	40N
Max Clamping Force	50N
Power Supply Voltage	DC24V
Power Consumption	Max Power $\leq 50\text{W}$, Comprehensive Power $\leq 30\text{W}$
Contact Surface Material	Rubber
Controller	Integrated
Communication Method	CAN
Control Method	Teach by Dragging / Offline Trajectory / API / Host Computer
External Interfaces	Power Interface x1, CAN Interface x1
Installation Method	Flange Mount
Working Environment	Temperature: -20 to 50°C , Humidity: 25%-85%, Non-condensing
Noise	$<60\text{dB}$
Note: The above data are test results of AgileX in a controlled testing environment. Results may vary under different environments and usage methods; actual experience should be considered.	

Optional leader gripper Specifications:

Teaching Device Parameters	
Weight	550g
Accuracy	$\pm 0.5\text{mm}$
Opening Distance	0-70mm
Rated Clamping Force	40N (Force Control, Force Feedback)
Max Clamping Force	50N (Force Control, Force Feedback)
Power Supply Voltage	DC24V
Power Consumption	Max Power $\leq 50\text{W}$, Comprehensive Power $\leq 30\text{W}$
Contact Surface Material	Rubber
Controller	Integrated
Communication Method	CAN
Control Method	Teach by Dragging / Offline Trajectory / API / Host Computer
External Interfaces	Power Interface x1, CAN Interface x1
Installation Method	Flange Mount
Working Environment	Temperature: -20 to 50°C , Humidity: 25%-85%, Non-condensing
Noise	$<60\text{dB}$
Note: The above data are test results of AgileX in a controlled testing environment. Results may vary under different environments and usage methods; actual experience should be considered.	