AGIBOT OmniPicker Product Manual

Release: 2025-02-12 14:15:22

1. Important Safety Precautions

1.1 Safety Instructions

1. Understand the product: Before using the product, please read the product manual carefully to understand the product's functions, operation methods, and safety precautions.

2. Follow the usage requirements: ensure that the machine is used in a suitable environment, avoiding excessively high or low temperatures, humid environments, etc.

3. Power supply safety: Make sure all cables, plugs, and sockets are intact before operation. Disconnect the power supply immediately when encountering abnormal conditions (such as short circuit, overheating, etc.).

4. Regular inspection: Regular maintenance and inspection are performed to ensure that the machine components such as motors and power supplies operate normally to avoid failures caused by aging or

damage.

5. Prevent misuse: Ensure that the machine is not used in any inappropriate, dangerous or legally prohibited scenarios.

1.2 Disclaimer

We are committed to continuously improving the product reliability and performance, and therefore reserve the right to make product improvements or upgrade without further notice. We strive to ensure the accuracy and reliability of this manual but shall not be held liable for any errors or omissions it may contain. Defects or malfunctions caused by the following circumstances are not covered within the scope of warranty:

1. Failure to install, wire or connect other control devices as required by the user manual;

Unauthorized disassembly/assembly of the OmniPicker;

3. Use of OmniPicker beyond the specifications or standards stated in the user manual;

4. Damage resulting from improper transportation;

5. Damage caused by accidents, impacts, or collisions;

6. Damage caused by natural disasters, including but not limited to fire, earthquakes, tsunami, lightning strikes, strong winds and floods.

We shall not be held liable for any loss, damage, injury or expense caused by customers violating the disclaimers in this section. Customers are kindly requested to carefully read and agree to this disclaimer before purchasing and using the product.

2. Introduction

2.1 Product Overview (update：Dec. 23, 2025)

AGIBOT OmniPicker is an adaptive, general-purpose gripper. It combines the advantages of different gripper designs and can grasp objects of various shapes with only one active degree of freedom. The product is very lightweight and easy to use, with a weight of only 430g.

OmniPicker is highly versatile and suitable for a wide range of grasping tasks. It can be integrated with various robot platforms such as humanoid robots, industrial arms and collaborative robots. OmniPicker provides multiple software communication protocol interfaces to help users complete integration quickly. In addition, the gripper features active force control capabilities, enabling tasks such as data collection and light-duty tasks.

2.2 Features

- Cost-effective hardware design
- Adaptive mode & gripping force adjustable
- Supports self-locking gripping force
- Feedforward force control
- Integrated actuator

2.3 Technical Specifications

Parameter	Typical value
Weight	0.43kg

Max gripping force	30N
Payload	1.5kg
Max gripper stroke	120mm
Min opening/closing time (Typ)	0.7s
Positioning repeatability	±0.05mm
Communication Protocols	CAN、CAN-FD、RS485、Serial、Modbus RTU
Operating Voltage	24V DC

3. Installation Instructions

3.1 Packing List

a. OmniPicker *1

b. CAN-FD communication cable *1

3.2 Instructions

The following materials are prepared with AGIBOT X1 Universal Humanoid Package as shown in Chart 1, please prepare the necessary installation tools, as shown in Chart 2.

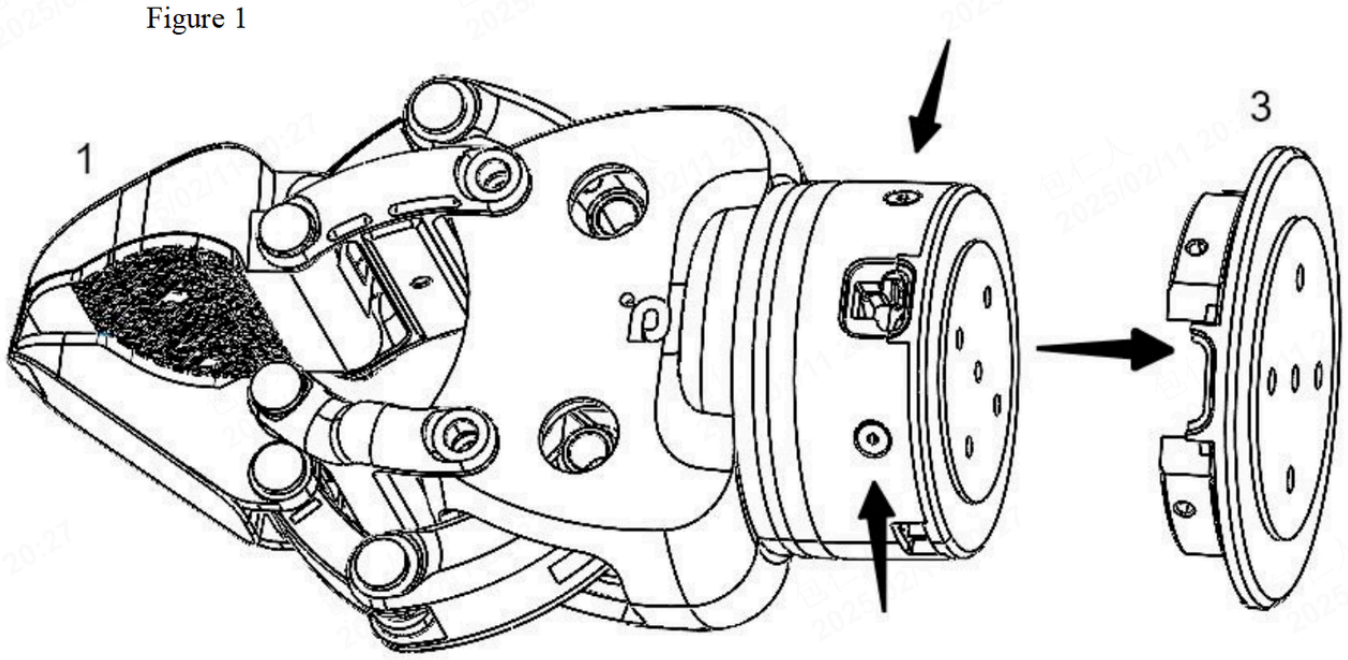
S/N	Material name	Quantity
1	OmniPicker	1
2	Ultra-short hexagon socket screw M3x6	3
3	End cap	1
4	Adapter flange	1
5	Hexagon socket head cap screw M4x8	4
6	Connecting flange	1

S/N	Tool name	Model/specification
1	Hexagon key wrench	H2
2	Hexagon key wrench	H2.5
3	Hexagon key wrench	H3

Complete the installation in accordance with the following steps

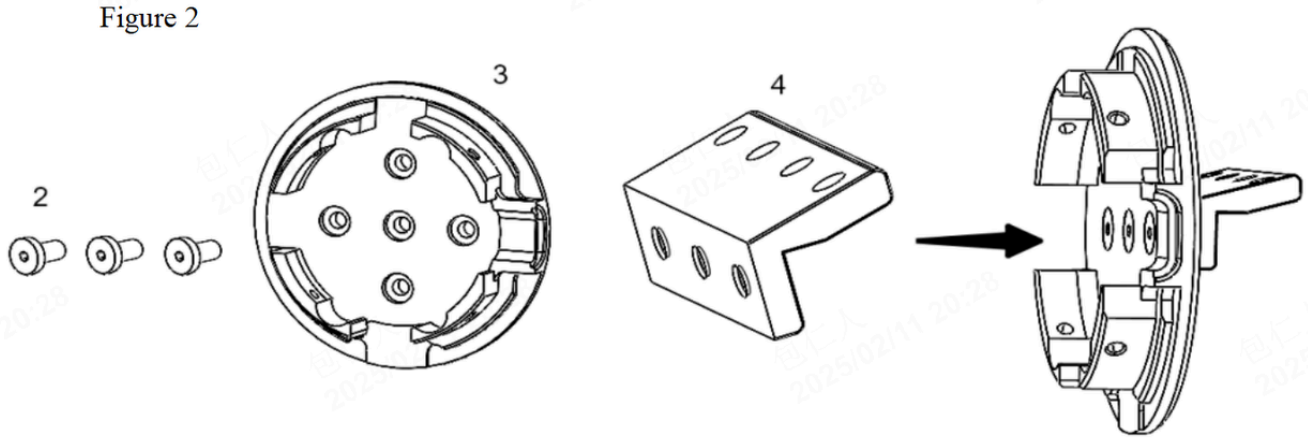
STEP1: As shown in Figure 1, remove the four countersunk head hexagon socket head cap screw M2.5x8 at the bottom of OmniPicker with a H2 hexagon key wrench to get the end cap;

Figure 1



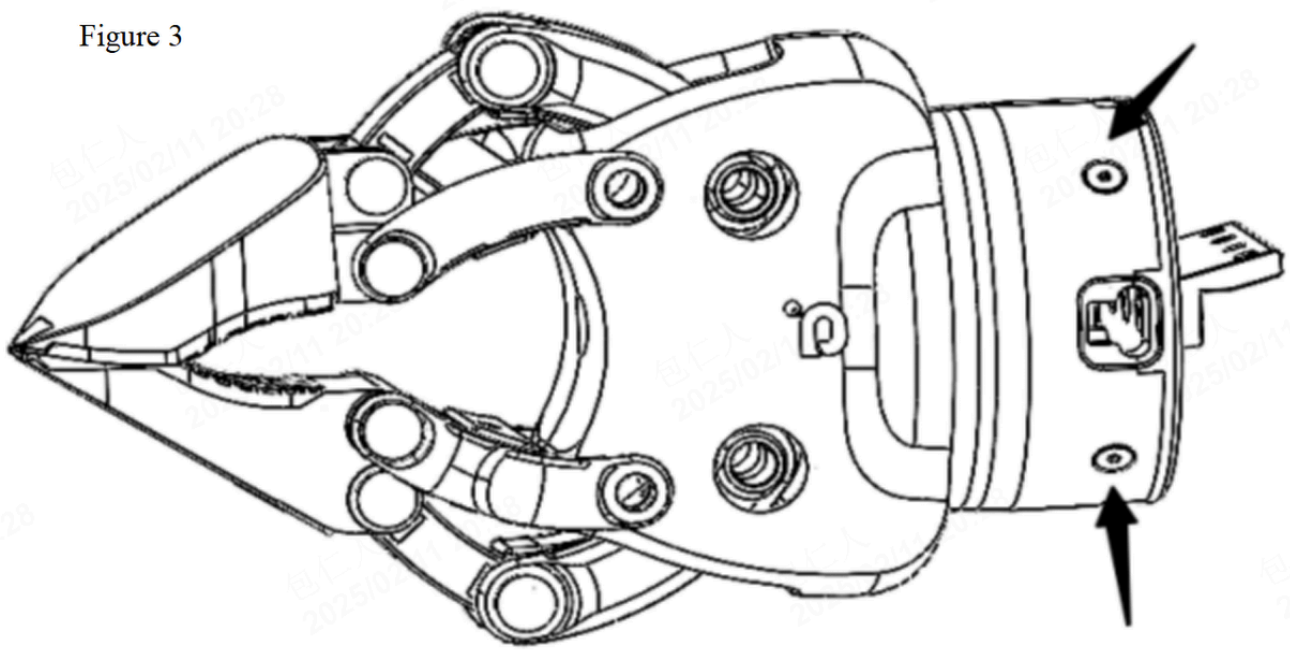
STEP2: As shown in Figure 2, screw three ultra-short hexagon socket head cap screws M3x6 into the end cap with a H2.5 hexagon key wrench and fix on the adapter flange;

Figure 2



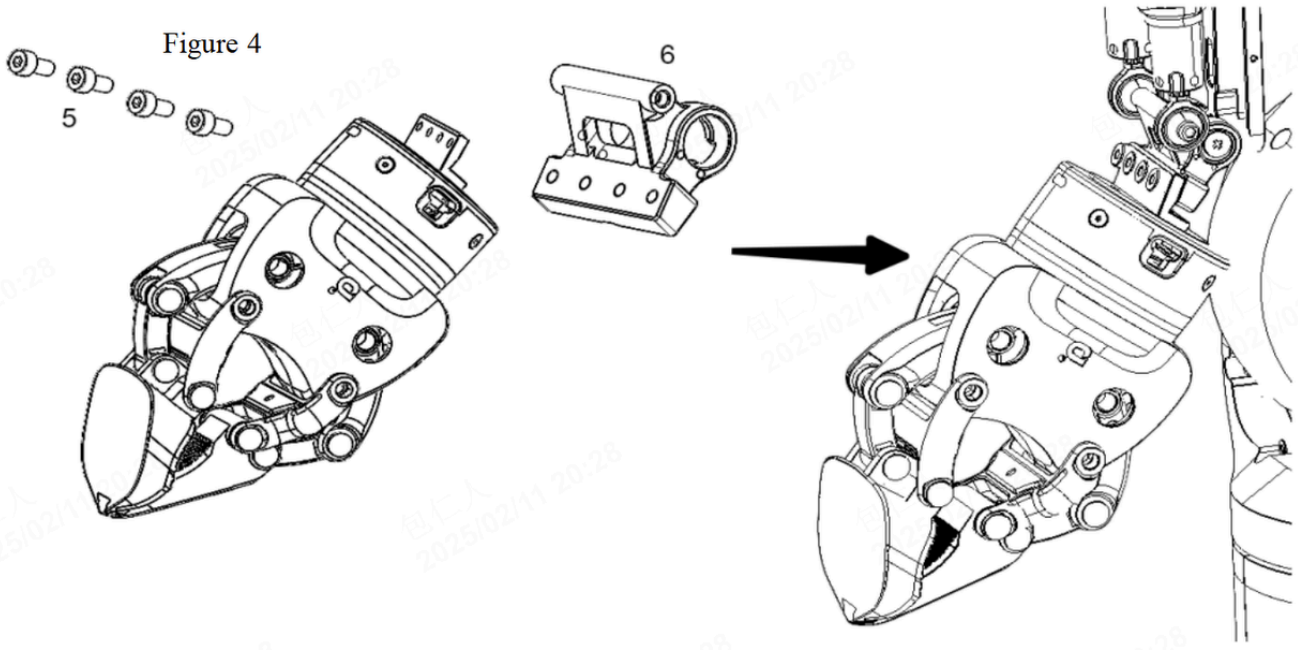
STEP3: As shown in Figure 3, use a H2 hexagon key wrench to replace the four countersunk head hexagon socket head cap screws M2.5x8 removed in Step 1 on the gripper;

Figure 3



STEP4: As shown in Figure 4, use a H3 hexagon key wrench to fix the gripper with four cylindrical hexagon socket head cap screws M4x8 on the connecting flange;

Figure 4



STEP5: The left and right grippers are installed in the same way, and all the wide fingers of the gripper are close to the outside of the body after installation;

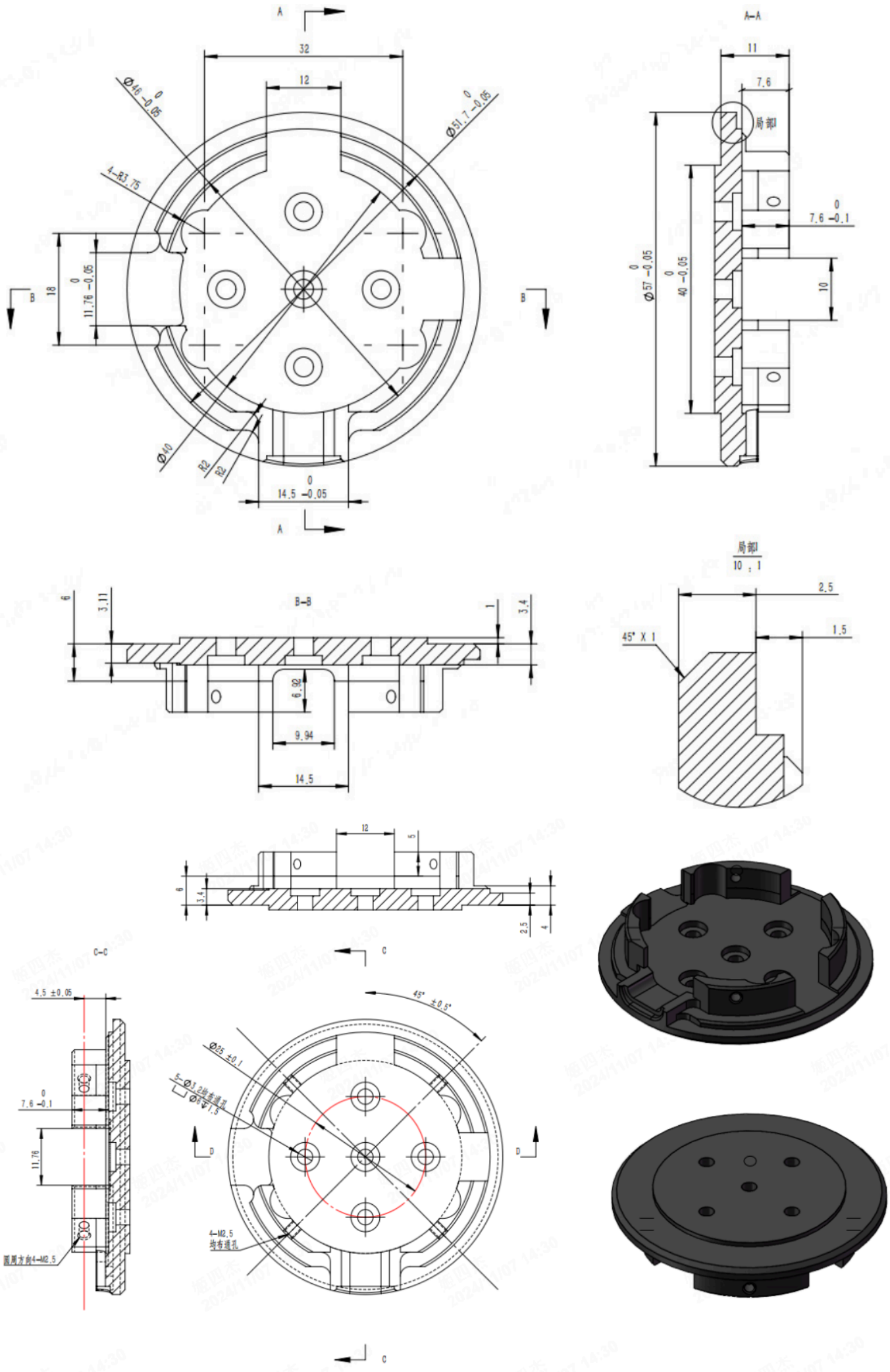
STEP6: If the obtained gripper has great resistance when the screws are removed in STEP1, it may be because threadlocker was applied when the screws were fastened. In this case, it is recommended to use a hot air gun to heat the screws before removal.

STEP7: Insert the cable.

NOTE: Pay attention to the installation direction when the end cap and the adapter flange are installed

3.3 Use of other robots (update: Dec. 23, 2025)

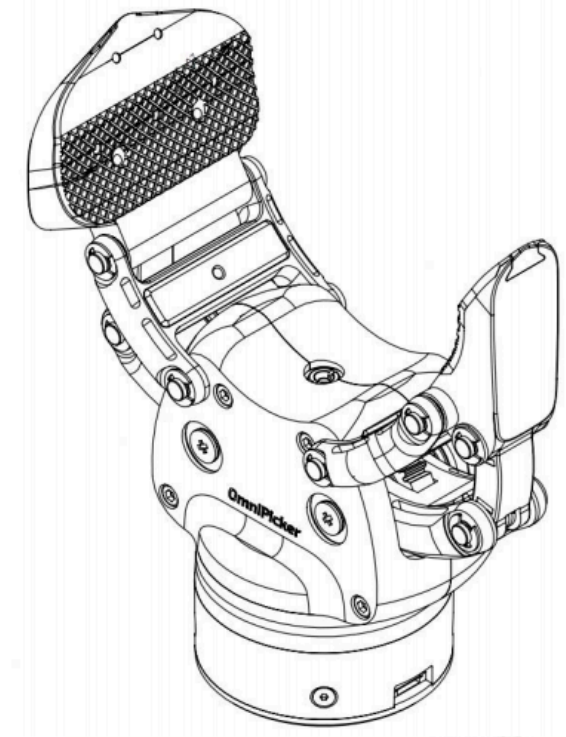
If users connects OmniPicker to other robotic arms, it is allowed to refer to the following installation interface drawings to design the adapter.

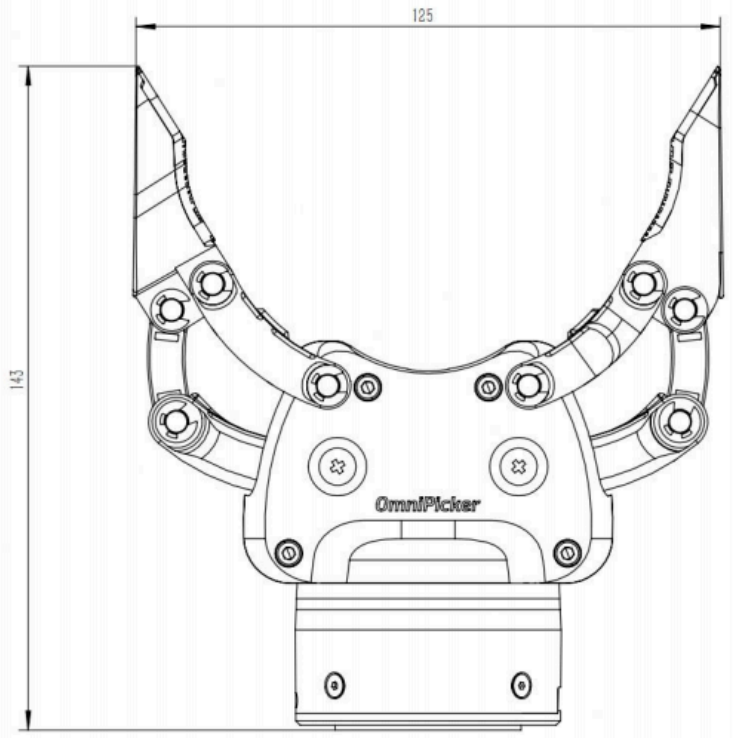


4. Product Dimensions

4.1 Dimensional Drawing

The following shows the external dimensions of OmniPicker in the opening and closing states.





5. Electrical Interface

5.1 6 PIN Interface

The external electrical interface of AGIBOT OmniPicker uses 6PIN socket with lock at a distance of 1.25mm.

1. If the gripper version number is 20, that is, the red box in the control panel is a resistance welding plate, and its interface pins are defined as shown in the following table:

S/N	Definition of PIN	Quantity
1	VIN	Pins 1 and 2 are used as power input, and the voltage does not exceed 24V DC.
2	VIN	
3	CAN_H	UART2_TX or RS485_B or FDCAN2_H.
4	CAN_L	UART2_RX or RS485_A or FDCAN2_L.
5	GND	Pins 5 and 6 are used as GND for power supply.

6	GND

2. If the gripper version number is 30, that is, the red box in the control panel is a dip switch, and its interface pins are defined as shown in the following table:

S/N	Definition of PIN	Quantity
1	VIN	Pins 1 and 2 are used as power input, and the voltage does not exceed 24V DC.
2	VIN	
3	L	UART2_RX or RS485_B or FDCAN2_L.
4	H	UART2_TX or RS485_A or FDCAN2_H.
5	GND	Pins 5 and 6 are used as GND for power supply.
6	GND	

Two kinds of grippers can be distinguished by the gripper package or the 14-digit SN code at the bottom of the fuselage, as follows:

The 6th and 7th digits of SN are gripper version numbers, including 20 and 30.

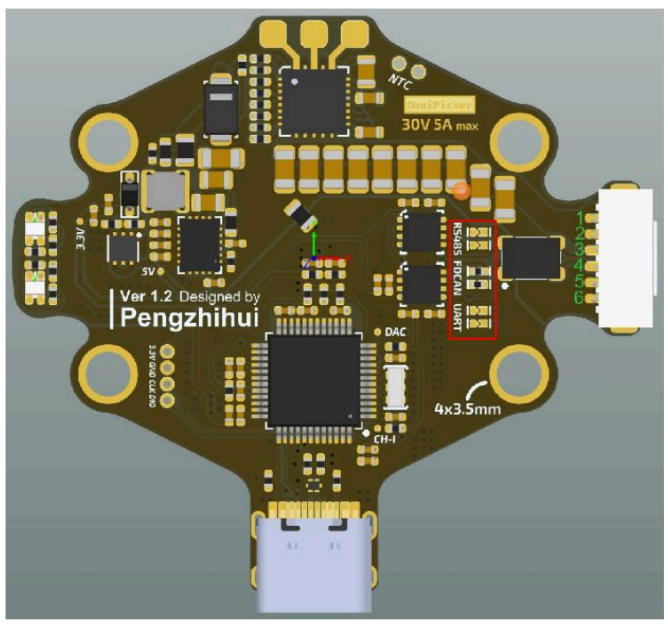
X1 0 06 20 XXXXXXX X1 0 06 30 XXXXXXX

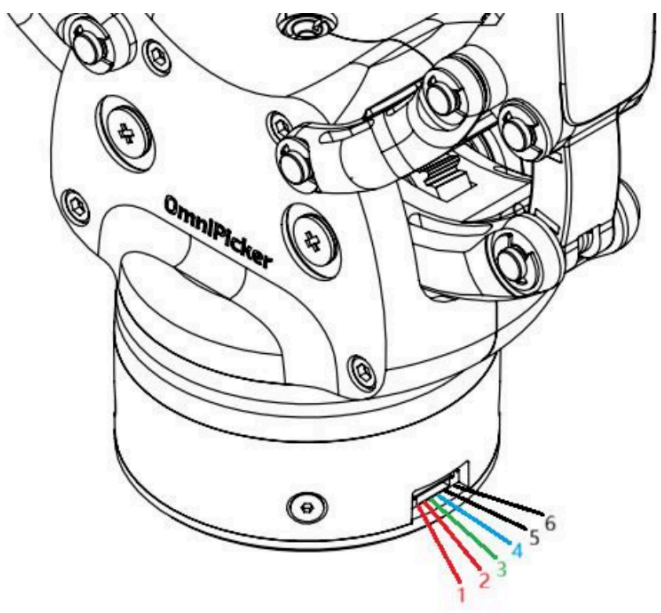
The difference between the two grippers is PCBA. Refer to 5.2 Communication modes for details

5.2 Communication Modes (update: Dec. 23, 2025)

When the end cap of the machine is opened, the state of the gripper PCBA can be distinguished according to the markings shown in the red box in the figure below. Select different operating modes based on this information.

1. If the gripper version number is 20, that is, the red box in the control panel is a resistance welding plate, corresponding communication mode is selected by means of jumper resistance. The jumper resistance is welded on CAN/CAN-FD by default. If it is necessary to switch to other communication modes, it is required to correspond to PCBA screen printing, remove the jumper resistor of the original CAN/CANFD and weld it to the corresponding communication interface resistance welding plate. RS485 is welded to two resistors above the red frame, while UART is welded to two resistors below the red frame.



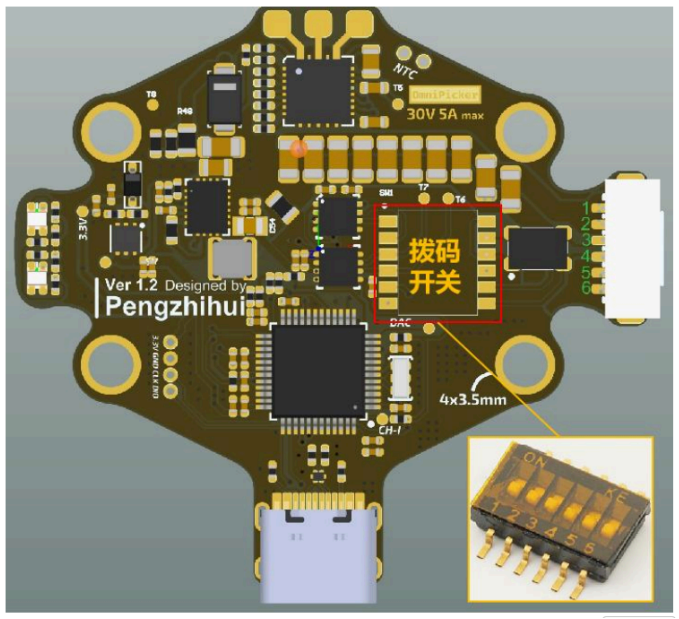


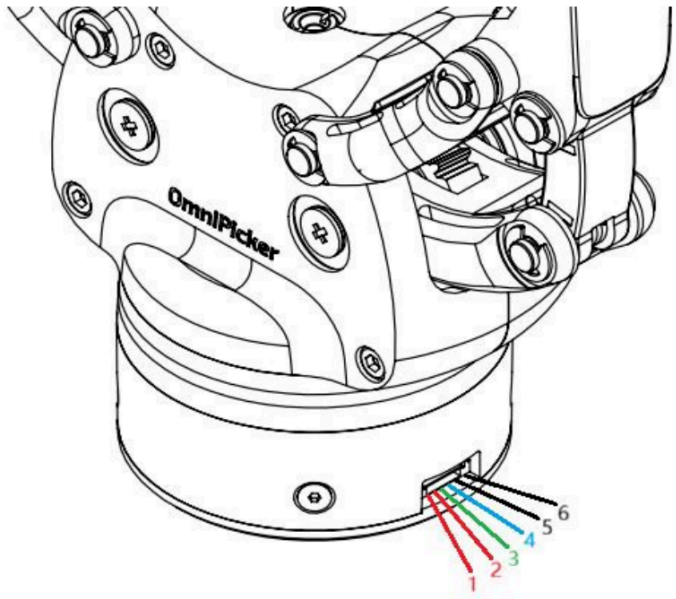
2. If the gripper version number is 30, that is, the red box in the control panel is dip switch, operate as below.

a. The default is CAN/CAN-FD, pins 3 and 4 are in the ON state, and the other pins are in the OFF state.

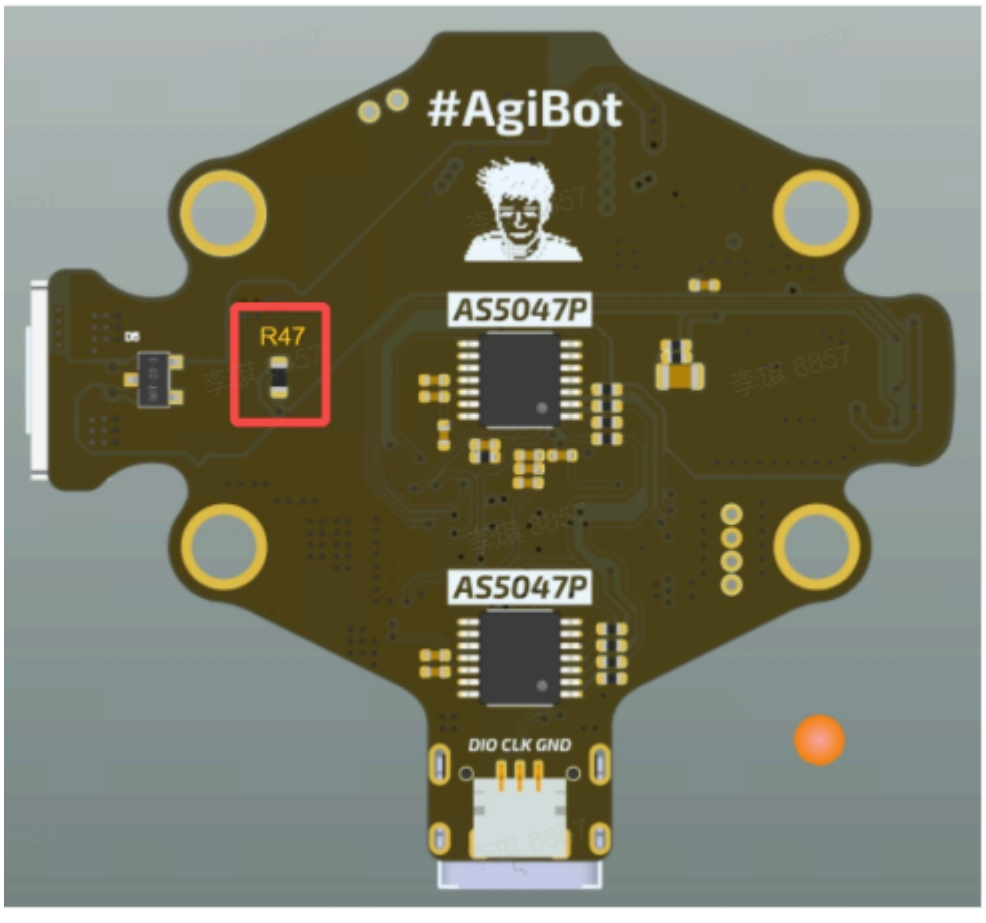
b. If it is necessary to switch to RS485, pull pins 1 and 2 to ON, and recover other pins to OFF.

c. If it is necessary to switch to UART, pull pins 5 6nd 6 to ON, and recover other pins to OFF.





3. A 120 Ω termination resistor is soldered on the back side of the PCBA. Whether it should be kept depends on the application scenario. The resistor is located at R47 inside the red box in the figure below.



- When using CAN / CAN-FD / RS485 communication, this resistor must be kept. It is soldered by default at the factory.
- When using UART-TTL communication, this resistor must be manually removed.

6. Communication Interface and Protocols

OmniPicker is equipped with multiple communication interfaces to adapt to different working environments.

Configuration steps are as follows:

a. Select the corresponding communication interface in accordance with the instructions in 5. interconnection of electrical interface. The default is CAN/CAN-FD

b. Configure gripper parameters through USB connection to the Host

c. Connect the gripper into the corresponding hardware systems

6.1 LED indicator lights

LED	Defination
Green: blinking	Disable
Green: steady on	Enabled (enabled automatically by default at power-on)
Red: steady on	Error
Blue: steady on	Motor Calibration in progress (factory-calibrated by default)
Multi-color: blinking	Firmware upgrading

6.2 Host Software

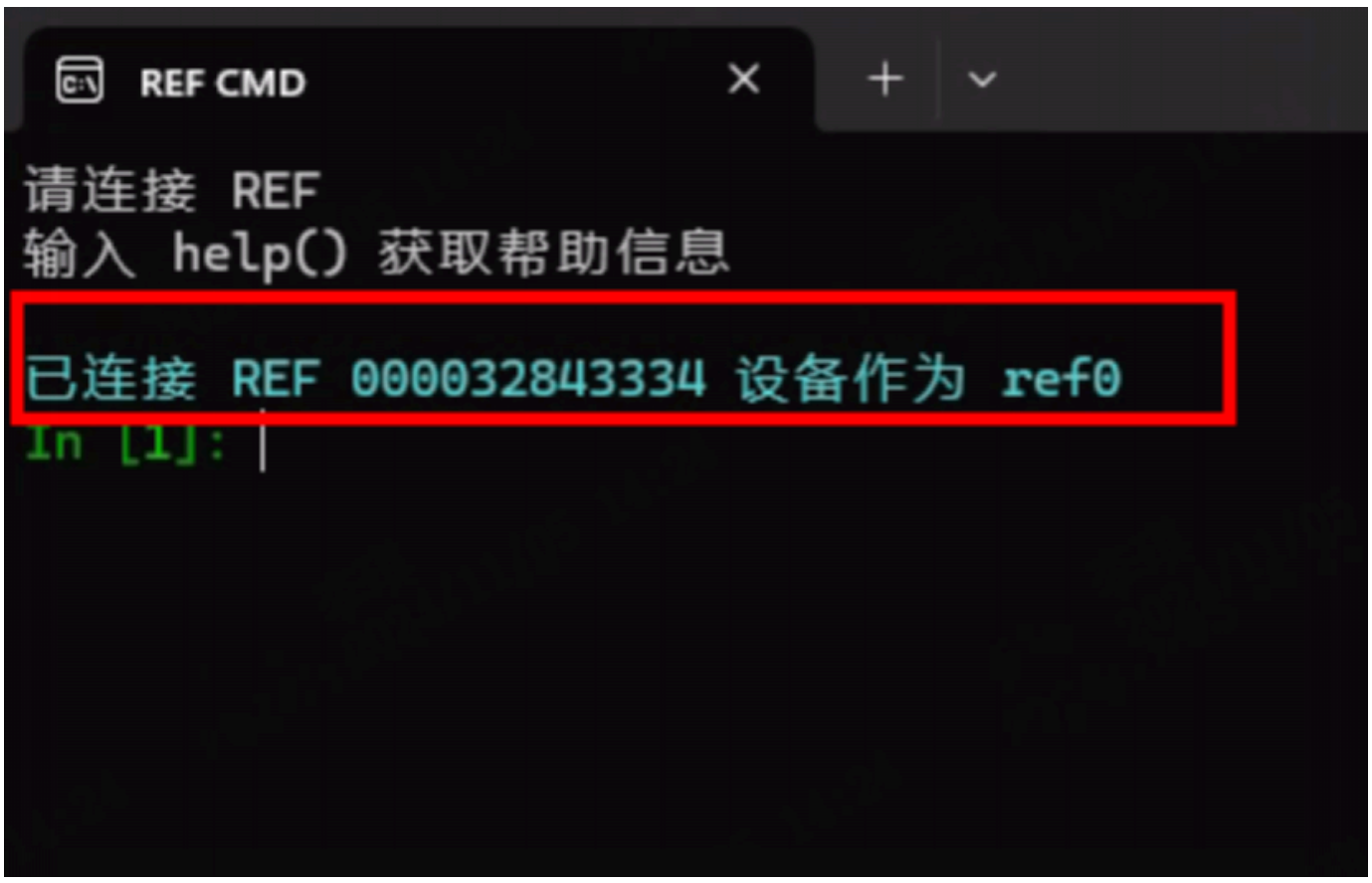
The host can modify parameters via the commands. It is recommended to fully understand the configuration parameters before first use.

6.2.1 How to use

Prepare a Windows computer and download the host software **REF-CLI v1.0.3.exe** .

First, make sure that there are no obstacles between the stroke of the gripper, then power on the gripper. The gripper will automatically close to zero position. In case of failure, it is required to stop and check whether the gripper moves smoothly.

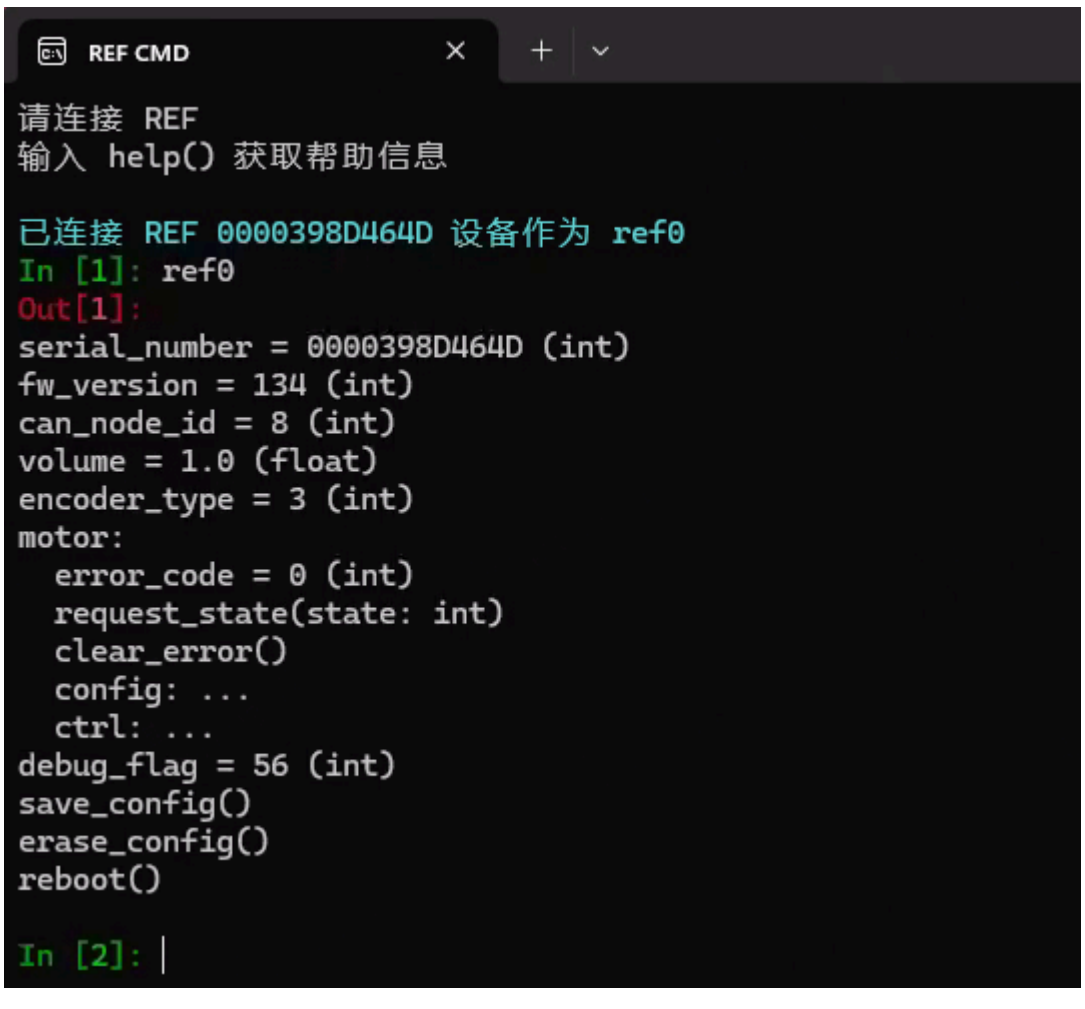
Connect the gripper to the computer with a USB-C cable, and open the host software. Once the host identifies the device successfully, the following display will appear:



At this point, the device with SN 000032843334 has the index name ref0 in the host software.

> Note: If multiple grippers are connected, their index names in the host will be arranged in sequence: the first is ref0, the second is ref1, and so on. Be sure to modify commands according to the corresponding device number. After closing and restarting the host software, indexing will start again from ref0.

In the host software, type ref0 directly and press Enter to show the basic information of the current gripper. As shown in the figure, the current gripper's firmware version is 1.3.4, and the CAN bus ID is 8.



6.2.2 Modify CAN-ID

ref0.can_node_id represents the node ID of CAN bus. It is allowed to view its value directly, or use = to assign values. Execute the following command.

```
Bash
ref0.can_node_id=8    // CAN ID sets to 8
```

```
In [2]: ref0.can_node_id
Out[2]: 8

In [3]: ref0.can_node_id=8
```

All parameter changes need to be saved by calling save_config() method, and this operation needs to be carried out in the Disabled state.

```
Bash
ref0.motor.request_state(0)    // First disable the gripper. 0 - disabled, 1- enabled
ref0.save_config()             // Save all the parameters
```

```
In [10]: ref0.motor.request_state(0)
Out[10]: True

In [11]: ref0.save_config()
Out[11]: True
```

Note: All parameter changes take effect only after saving and power-cycling the device.


```
In [10]: ref0.motor.request_state(0)
Out[10]: True

In [11]: ref0.save_config()
Out[11]: True
```

Note: All parameter changes take effect only after saving and power-cycling the device.

6.2.3 Control the Gripper

1. Power on the gripper again. (Default self-enabled upon power-on)
2. Wait for the gripper to close automatically
3. Connect the host and call set_pos(x) to set its position, where x is the opening and closing percentage, with the range of 0.0-1.0, 0 means closed and 1 means fully open.

```
Bash
ref0.motor.ctrl.set_pos(0.5) // Gripper open 50%
```

6.3 CAN/CAN-FD Communication Protocols (update: Dec. 23, 2025)

When the gripper communicates with CAN-FD, the baud rate is 1M in arbitration domain (80% sampling point) + 5M in data domain (75% sampling point), and it is compatible with CAN protocol with a baud rate of 1M (80% sampling point).

Note: The firmware version of CAN protocol is not less than 3.3.0.

6.3.1 Downstream Protocol

CAN ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7
can_node_id	8	Reserved	Pos Cmd	Vel Cmd	Force Cmd	Acc Cmd	Dec Cmd	Reserved	Reserved

- ID : downstream control ID of the gripper, corresponding to ref0.can_node_id
- Reserved： reserve, write 0

• Pos Cmd： target position, range 0 - FF,0is close, FFis fully open

• Vel Cmd： target velocity, range 0 - FF，FF is max velocity

• Force Cmd： target torque，range 0 - FF，FF is max torque

• Acc Cmd： target acceleration，range 0 - FF，FFis max acceleration

• Dec Cmd： target deceleration，range 0 - FF，FF is max deceleration

E.g. If the gripper is controlled to run to 50% of the stroke, all other parameters can be given the maximum value:

Send message 00 7F FF FF FF FF 00 00

6.3.2 Upstream Protocol

After power-on and successful enabling, every time the gripper receives a control command, the state data of the a frame will be reported immediately. The state frame is defined as follows.

CAN ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7
can_node_id	8	Fault Code	State	Pos	Vel	Force	Reserved	Reserved	Reserved

• ID： upsteam message ID of the gripper, corresponding to ref0.can_node_id

• Fault Code： error code, definitions are as following:

Code	Definition
00	No fault
01	Over heated alarm
02	Speed limit exceeded alarm
03	Initialization fault alarm
04	Limit-exceeding alarm

• State： Current state, definitions are as following:

Code	Definition
00	Gripper reached target position
01	Gripper moving
02	Gripper stalled
03	Object dropped

• Pos : current position, range as Pos Cmd

• Vel： current velocity, range as Vel Cmd

• Force：current torque, range as Force Cmd

Reserved：reserve. Bytes D5–D7 will directly copy the data from the sent packet. They can be used for your own checksum/verification, or simply ignored.

6.4 Serial TTL/485 Protocols

> Note: The serial bus also supports serial chaining, and the node ID reuses can_node_id. The firmware version must be 3.3.2 or higher.

When the gripper communicates with serial port TTL or 485, the baud rate needs to be changed to 115200@8N1 (8bit data bit, 0 check bit, 1 stop bit).

In version 3.3.5 and later, the baud rate is given by the uart_baudrate parameter in the host software. The data length is 8 bits, there is no parity bit, and the number of stop bits is given by the uart_stopbits parameter in the host software.

6.4.1 TTL/485 Switch

Connect Omnipicker to the host via a USB-C cable, open the host , input ref0 and enter, to check the value of rs485_mode .

• If using TTL, please set the value of rs485_mode to 0 and save. Note: remove the resistor on the back of the PCB.

```
Python
ref0.rs485_mode = 0  // turn off 485 mode，enable TTL

ref0.motor.request_state(0)  // Disable
ref0.save_config()           // Save configuration
```

• If using RS495 , please set the value of rs485_mode to 1 and save.

```
Python
ref0.rs485_mode = 1  // turn on 485 mode，turn off TTL

ref0.motor.request_state(0)  // Disable
ref0.save_config()           // Save configuration
```

Take effect after saving and restarting.

6.4.2 TTL/485 Configuration

Firmware version 3.3.5 and above supports configuring TTL/RS-485 communication parameters as well as the communication protocol.

• Baud rate

```
Python
ref0.uart_baudrate = 460800  // baud rate 460800

ref0.motor.request_state(0)  // Disable
ref0.save_config()           // Save configuration
```

Take effect after saving and restarting.

• Stopbits

Code	Definition
0	1 stop bit
1	1.5 stop bit
2	2 stop bit

```
Python
ref0.uart_stopbits = 0       // Set as 1 stop bit(can be replaced with the required
number)

ref0.motor.request_state(0)  // Disable
ref0.save_config()           // Save configuration
```

Take effect after saving and restarting.

• Communication protocol configuration

The communication protocol is determined by the modbus_mode parameter: if it is False, a custom protocol is used; if it is True, Modbus RTU is used.

```
Python
ref0.modbus_mode = True      // Set to Modbus RTU

ref0.motor.request_state(0)  // Disable
ref0.save_config()           // Save configuration
```

Take effect after saving and restarting.

6.4.3 Custom Protocol

Firmware versions earlier than 3.3.5 use the custom protocol.
For versions 3.3.5 and later, the custom protocol is used when the host parameter modbus_mode is set to false.

6.4.3.1 Downstream Protocol

D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11

Frame Head	ID	Reserved	Pos Cmd	Force Cmd	Vel Cmd	Acc Cmd	Dec Cmd	Reserved	Reserved	CheckSum

- Frame Head：frame head，0x4141，2 bytes
- ID： as can_node_id
- D3 - D10： as D0-D7, CAN bus downstream protocol
- CheckSum： The data and checksum for D2–D10 are calculated as shown in the Python code below.

```python
def checksum(buf) -> int:
    ret = 0
    for i in buf:
        ret += i
    return (~ret) & 0xff
```

E.g. If the gripper is controlled to run to 50% of the stroke, if ID is set to 1, all other parameters can be given the maximum value:

Send message 41 41 01 00 7F FF FF FF FF 00 00 83

6.4.3.2 Upstream Protocol

After power-on and successful enabling, every time the gripper receives a control command, the state data of the a frame will be reported immediately. The state frame is defined as follows.

D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11
Frame Head		ID	Fault Code	State	Pos	Vel	Force	Reserved	Reserved	Reserved	CheckSum

- Frame Head：frame head，0x4141，2 bytes
- ID： as can_node_id
- D3 - D10：as D0-D7, CAN bus downstream protocol
- CheckSum：the calculation method is same as downstream protocol, range is D2-D10

6.4.4 Modbus RTU

For versions 3.3.5 and later, the Modbus RTU protocol is used when the host parameter modbus_mode is set to True.

The gripper’s Modbus slave address is same as can_node_id.
You can control the gripper and obtain its current status feedback by reading and writing Modbus registers.
The Modbus register map is as follows:

Register Address (decimal)	**Definition**	**Details**
0	Firmware verison	
10	Target position	Only the low byte of the register is used. Value range: 0–FF. 0 = fully closed (clamped), FF = fully open
11	Target Velocity	Only the low byte of the register is used. Value range: 0–FF. FF = max velocity
12	Target Torque	Only the low byte of the register is used. Value range: 0–FF. FF = max torque
13	Target Acceleration	Only the low byte of the register is used. Value range: 0–FF. FF = max acceleration

14	Target Deceleration	Only the low byte of the register is used. Value range: 0–FF. FF = max deceleration
15	Motion Trigger flag	Writing a non-zero value triggers the motor to move according to the target position and other parameters in registers 10-14. The flag is automatically cleared when the motor starts executing the motion.
20	Current error code	Refer to error code in CAN-FD protocol
21	Current status	Refer to status code in CAN-FD protocol
22	Current position	Refer to Target position
23	Current velocity	Refer to Target velocity
24	Current Torque	Refer to Target torque

6.4.4.1 Motion Control

The user can control the gripper by writing control parameters to registers 10–15 using the Modbus RTU 0x10 (Write Multiple Registers) command.

For example, to move the gripper to 50% of its stroke, assuming the slave address is 1 and all other parameters are set to their maximum values,

send message: 01 10 00 0A 00 06 0C 00 7F 00 FF 00 FF 00 FF 00 FF 00 01 B9 BA

- Register 10 (target position): 00 7F → 0x7F (≈ 50% stroke)
- Registers 11–14 (speed, torque, etc.): 00 FF 00 FF 00 FF 00 FF → all set to 0xFF (maximum)
- Register 15 (motion trigger flag): 00 01 → non-zero, triggers gripper motion

6.4.4.2 Read Status

The user can read the gripper status by sending a Modbus RTU 0x03 (Read Holding Registers) command to read registers 20–24.

Assuming the slave address is 1, the request frame is:

01 03 00 14 00 05 C5 CD

- Function code: 03 (Read Holding Registers)
- Starting register address: 00 14 → 0x0014 = decimal 20
- Number of registers: 00 05 → read 5 registers (20–24)

6.5 Firmware Upgrade

6.5.1 How to upgrade

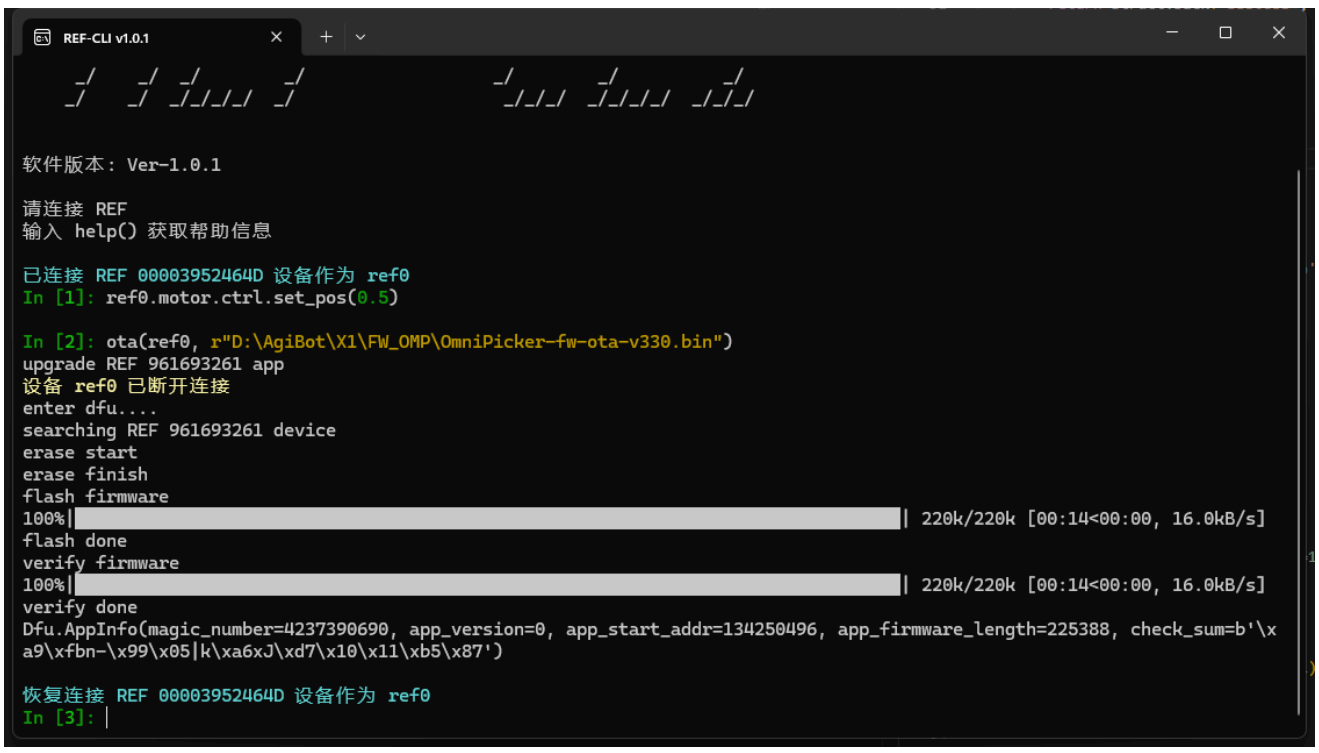
1. Power on the gripper and connect it to PC, and turn on the host to ensure that the equipment is successfully connected and initialized.

2. Control the gripper to 50% of the stroke to prepare for calibration. After the command is executed, observe the gripper and confirm that it has been opened to 50%.

Note: This step is very important. Please follow the instructions strictly

3. Input the upgrade command in the host and wait for the progress to end, where path_to_firmware_file is the absolute path of the firmware in the system. Here, version 3.3.0 is taken as an example.

```
Bash
ota(ref0, r"path_to_firmware_file")
```



Note: Please be careful not to operate or move the gripper during upgrade, so as not to cause abnormal USB communication.

4. After the upgrade is completed, the prompt "Restore Link REF XXX" will reappear in the terminal. Check the LED indicator, wait for the green light to start blinking, and then, enter the calibration command.

Note: If, after the upgrade is completed, the gripper closes automatically, skip this step and go directly to step 5.

```
Bash
ref0.motor.request_state(2)
```

At this point, the LED changes from green blinking to blue steady-on, and the gripper will start to move slowly. Do not operate the gripper during this time and wait for the calibration to complete. When the calibration is successful, the gripper will make a sound, and the LED changes back from blue steady - on to green blinking.

Note: Before performing the calibration, if the gripper fails to open or can no longer be controlled, manually open the gripper according to the following steps.

a. Unplug the power supply and all connection cables.

b. Manually remove the rear cover as shown in the figure below.

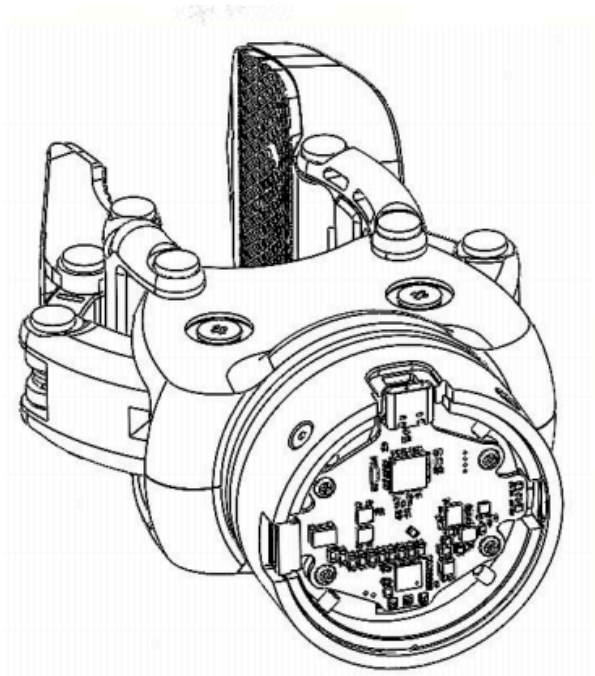
c. Manually remove the 4 cross-head screws on the control board as shown in the figure below.

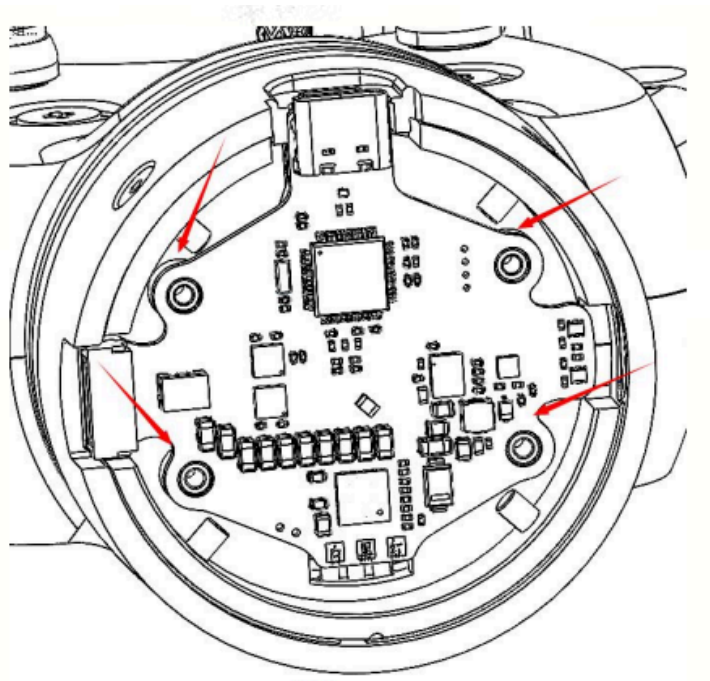
d. Lift the control board to expose the magnet. Note that the 3 welded wires of the control board should not be disconnected during the lifting process, as shown in the figure below.

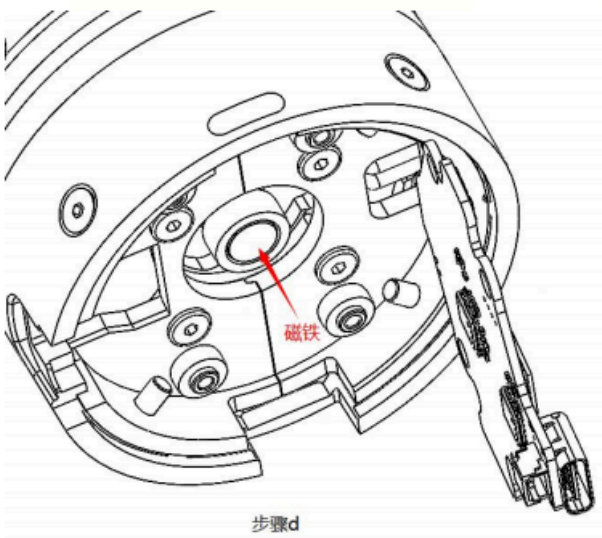
e. Manually rotate the magnet bracket counter-clockwise until the OmniPicker opens about 50% of its stroke, as shown in the figure below.

f. Re-install the control board and tighten the 4 cross-head screws.

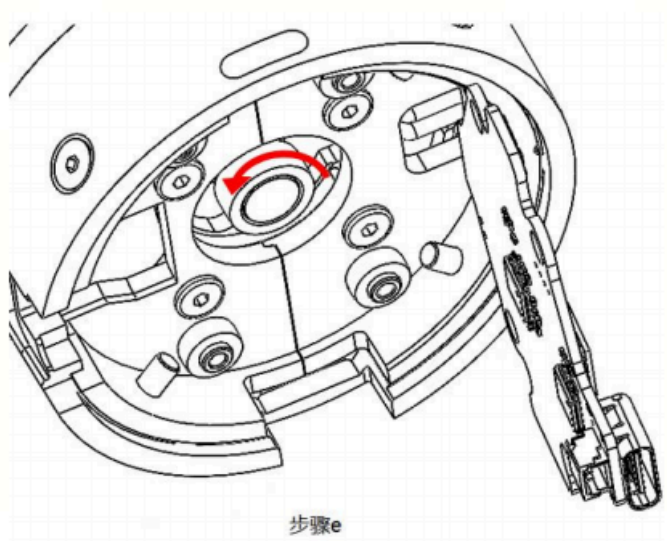
g. Put on the rear cover and tighten the screws.


步骤b


步骤c



步骤d


步骤e

5. The gripper is powered on again, automatically turns on and closes, and the upgrade is complete.

```
In [3]: ref0
Out[3]:
device_type = 1 (int)
serial_number = 00003952464D (int)
fw_version = 330 (int)
can_node_id = 1 (int)
volume = 1.0 (float)
encoder_type = 3 (int)
motor:
  error_code = 0 (int)
  request_state(state: int)
  clear_error()
  apply_user_offset()
  calc_current_gain()
  config: ...
  ctrl: ...
debug_flag = 56 (int)
save_config()
erase_config()
reboot()
enter_dfu()
```

版本已更新

6. If the upgrade fails, use the flash_app command to rewrite the file.

Bash

```
flash_app(r"PATH")  // PATH is the absolute path of the firmware in the system
```

```
已连接 REF 000032893334 设备作为 ref0
In [1]: flash_app(r"C:\Users\admin\OmniPicker-fw-ota-v333.bin")
```

NOTE: the version of REF-CLI has to be 1.0.3 or above

6.5.2 Firmware release

Version	Date	Firmware	Remarks
3.3.0	2024.12.26	[OmniPicker-fw-ota-v330.bin]	1. Serial communication support is added (TTL/485) 2. CAN protocol message support is added
3.3.3	2025.1.10	**[OmniPicker-fw-ota-v333.bin]**	1. Serial port communication support（TTL）
3.3.5	2025.12.23	**[OmniPicker-fw-ota-v335.bin]**	1. Supports Modbus RTU communication

7. Others (update: Dec. 23, 2025)

This manual is based on OmniPicker hardware Ver1.2 and software Ver3.3.0.

For other matters not specified herein, please refer to AGIBOT official website or contact AGIBOT service personnel for information.

8. URDF

Please use ROS2 Humble to load the URDF file of the gripper.