

# OmniPicker用户使用手册(适配灵犀X2 Ultra)

## 1. 安全须知

- 了解产品：**在使用该产品之前，请仔细阅读产品手册，了解产品的功能、操作方式以及安全注意事项。
- 遵循使用要求：**确保机器在适宜环境中进行使用，避免过高或过低的温度、潮湿环境等。
- 电源安全：**操作前确保所有电缆、插头、插座完好无损。遇到异常情况(如短路、过热等)时立即断开电源。
- 定期检查：**定期进行维护和检查，确保机器的组件如电机、电源等正常运行，避免因老化或损坏导致的危险。
- 防止误用：**确保机器不会用于任何不当、危险或法律禁止的场景。

## 2. 产品简介

### 2.1 产品概述

智元灵犀OmniPicker是一款自适应的通用夹爪。融合了不同模态的夹爪设计的优点，只用一个主动自由度就可以实现各种不同形状物体的抓取，而且非常轻便，重量只有430g左右。

OmniPicker用途多样，非常适合用于机器人和各种机械臂的抓取，除了人形机器人，工业臂、协作臂都能用，我们提供多种软件通信协议开发接口。此外夹爪还带主动力控，不论是用于数据采集，还是任务作业都有很好的效果。



## 2.2 产品参数

智元OmniPicker关键产品参数典型值如下

参数	典型值
产品重量	0.43kg
最大夹持力	30N
最大推荐负载	1.5kg
最大行程	120mm
打开/闭合时间	0.7s
重复定位精度	±0.05mm
通信协议	CAN、CANFD、RS485、串口
额定工作电压	24V DC

## 3. 产品安装说明

### 3.1 安装OmniPicker到灵犀X2 Ultra

在购买的智元灵犀X2 OmniPicker中包含如下物料

序号	物料名称	数量
1	OmniPicker	1
2	超短头内六角螺栓M3x6	5
3	后盖	1
4	CANFD配线	1
5	转接件	1
6	圆柱头内六角螺丝M4x8	5

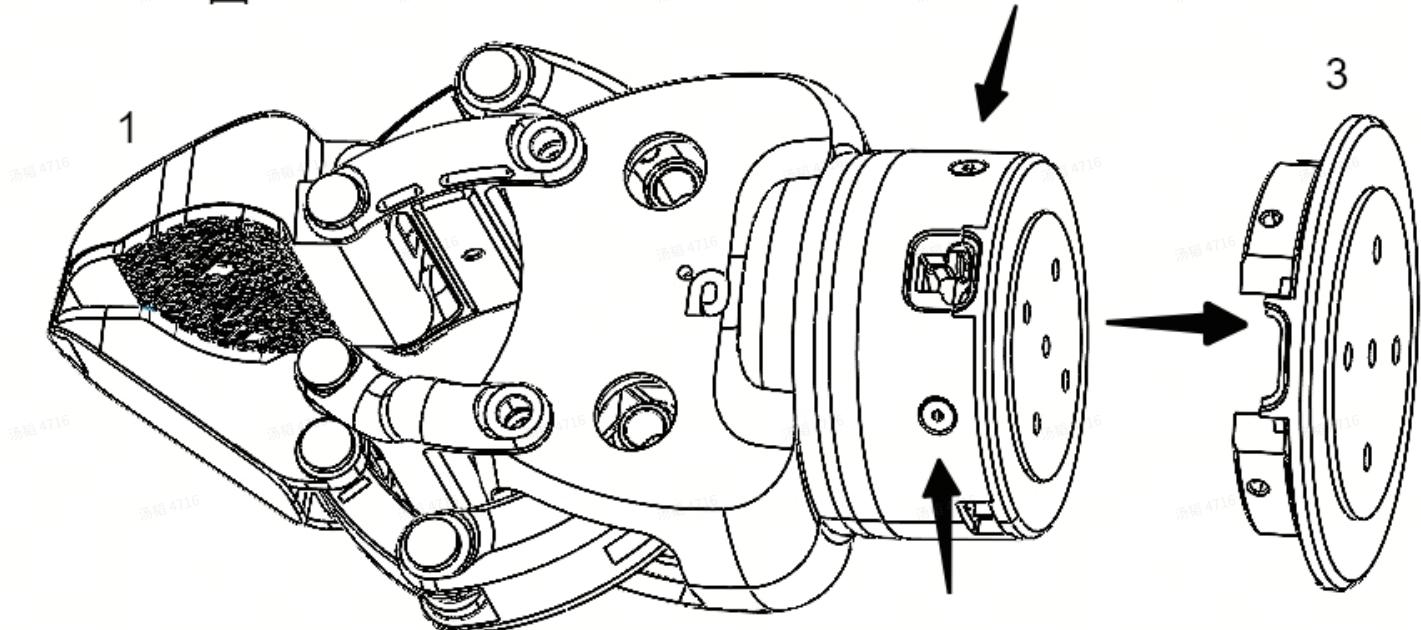
自备如下工具

序号	工具名称	型号/规格
1	内六角扳手	H2
2	内六角扳手	H2.5
3	内六角扳手	H3

按以下步骤完成OmniPicker的组装

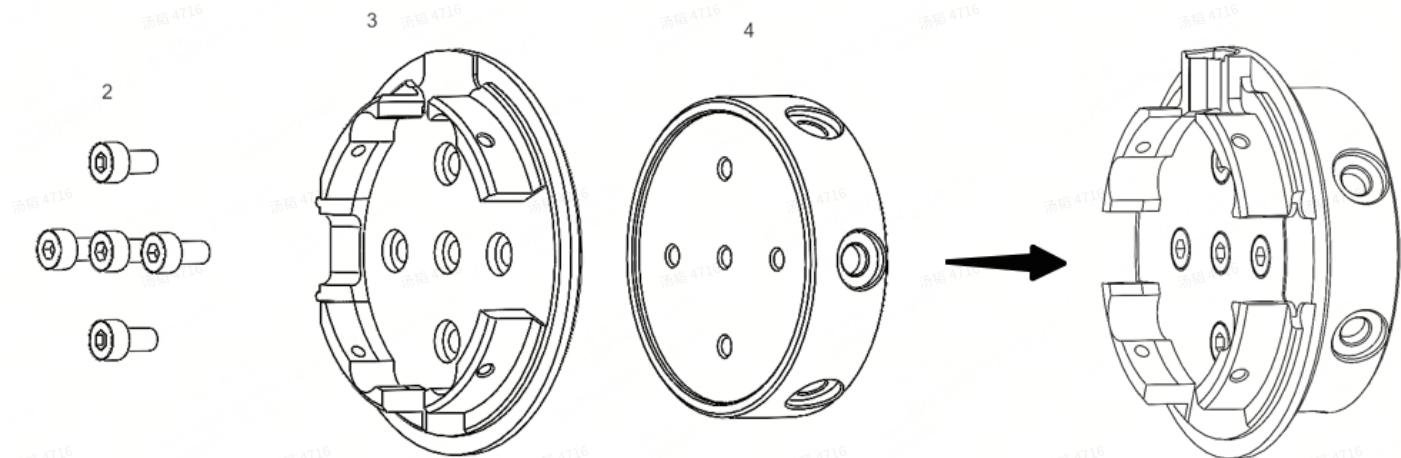
**STEP1:** 如图1，将OmniPicker底部圆周4颗沉头内六角螺丝M2.5x8用H2内六角扳手取下后得到后盖；

图1



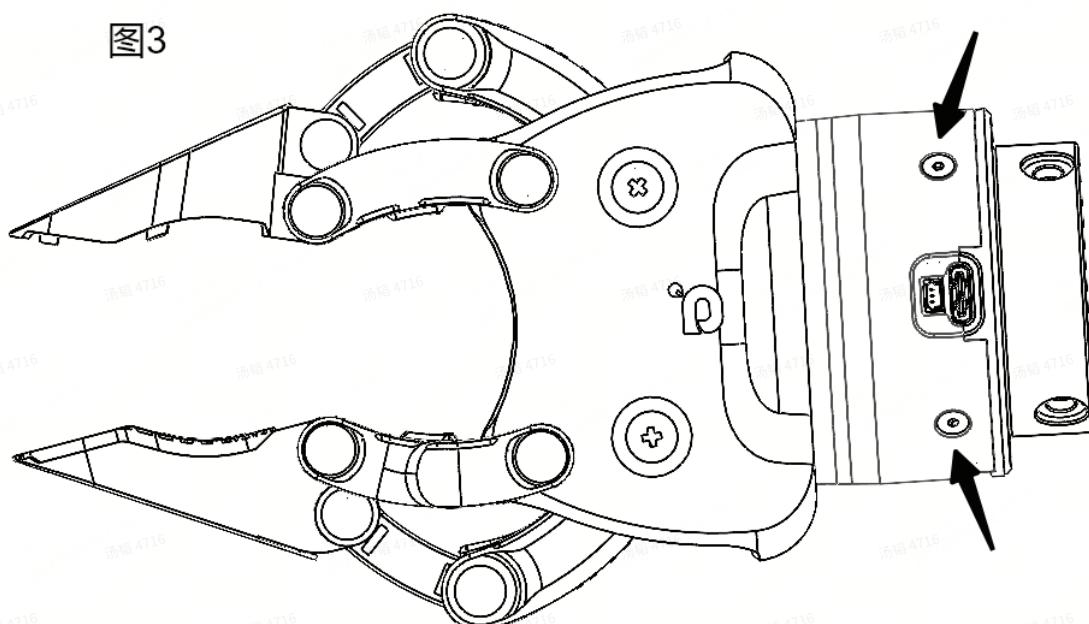
**STEP2:** 如图2，用H2.5内六角扳手将5颗超短头内六角螺丝M3x6拧入后盖，固定在转接件上；

图2



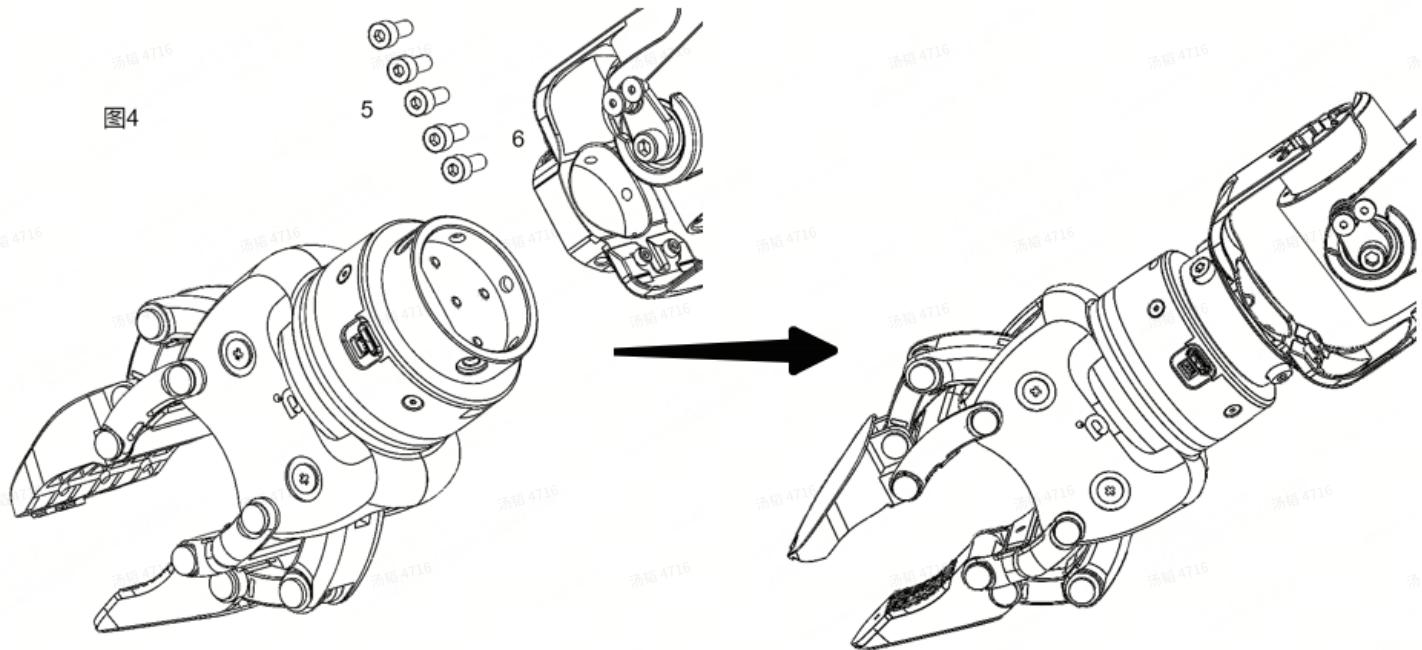
**STEP3：**如图3，用H2内六角扳手将第1步拆下的4颗沉头内六角螺丝M2.5x8装回夹爪上；

图3



**STEP4：**如图4，用H3内六角扳手将4颗圆柱头内六角螺丝M4x8把夹爪固定在手腕末端上；

图4



**STEP5：**左右夹爪按照同样的方式安装，装完后夹爪宽手指均靠身体外侧；

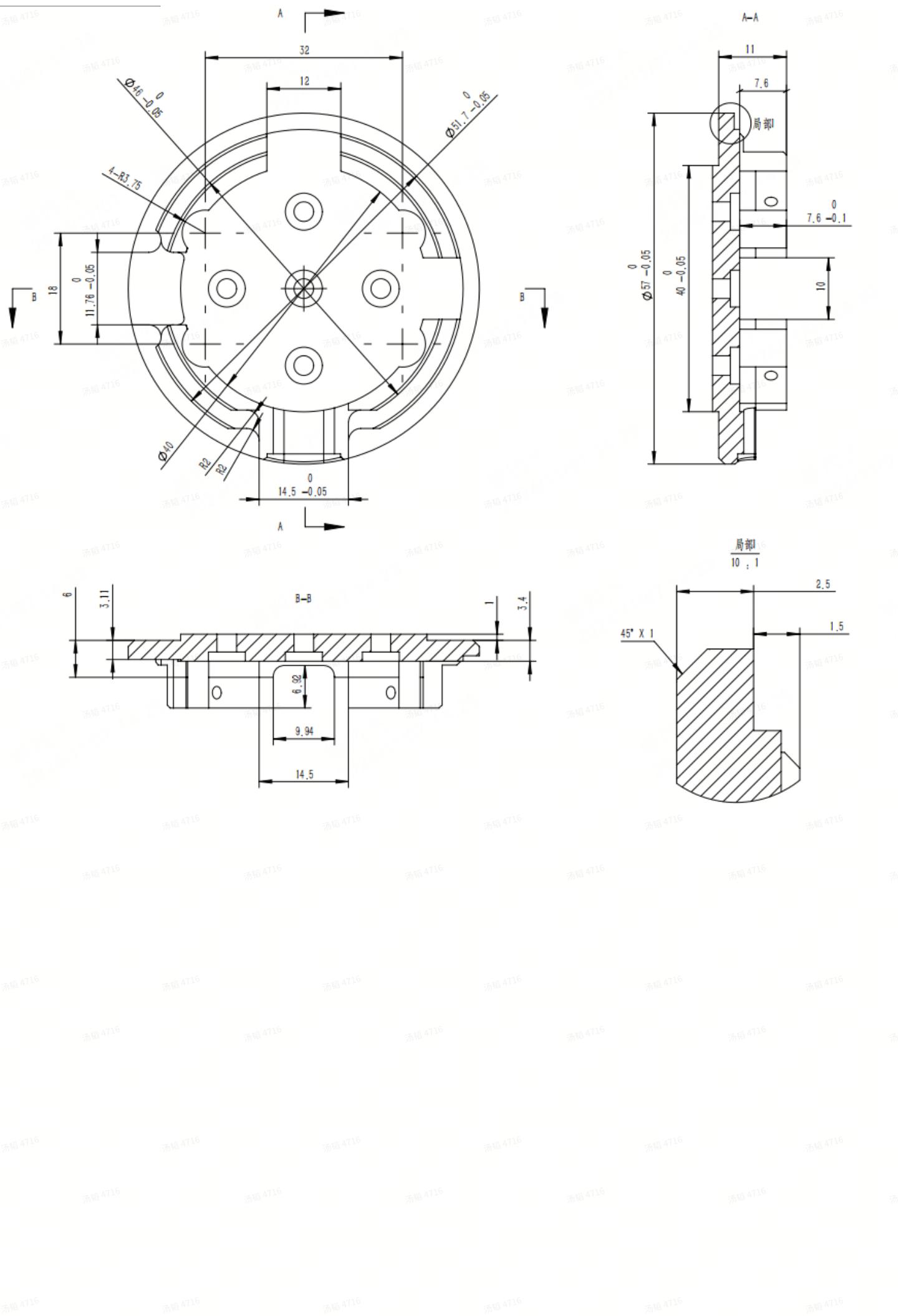
**STEP6：**若拿到的夹爪在**STEP1**拆除螺丝时阻力很大则可能螺丝固定时有螺纹胶，须热风枪加热后拆除；

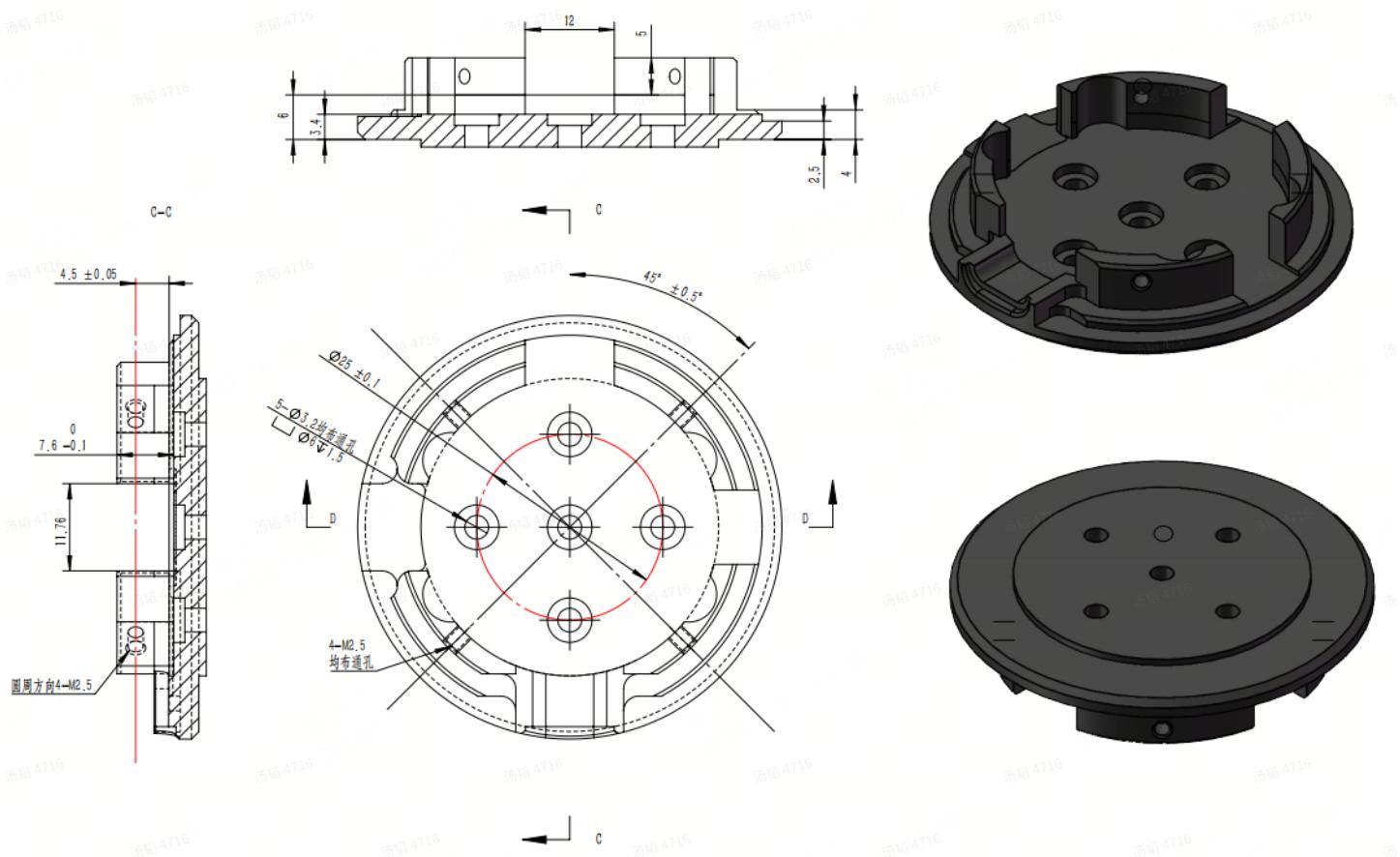
**STEP7：**插入夹爪线缆。

**NOTE：**后盖和转接件固定时注意安装方向

### 3.2 使用其他机械臂

用户如将OmniPicker接到其他机械臂上，可参考如下安装接口图纸自行设计转接件，并按照**3.2**完成组装。

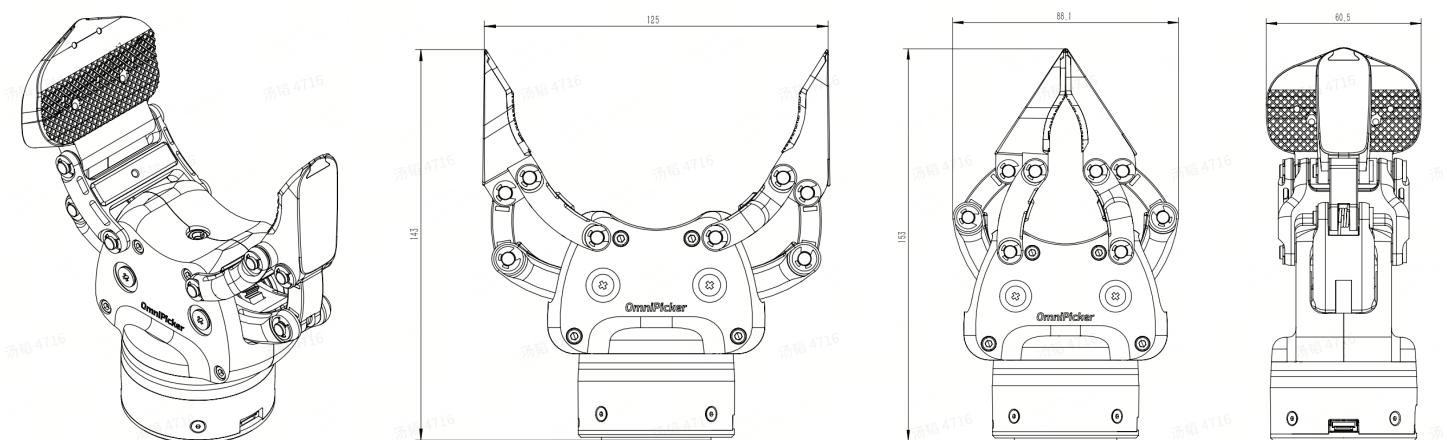




## 4. 产品尺寸信息

### 4.1 外部尺寸图

如下展示了OmniPicker打开和闭合状态下的外部尺寸



## 5. 电气接口互联

### 5.1 6 PIN接口定义

智元灵犀OmniPicker对外的电气接口使用的是间距为1.25mm的6 PIN带锁扣的插座，

## 1. 如夹爪版本号为20，即控制板中红框中为电阻焊盘，其接口引脚定义如下表所示：

序号	PIN定义	功能描述
1	VIN	1、2脚用作电源输入，电压不超过24V DC。
2	VIN	
3	CAN_H	UART2_TX或RS485_B或CAN总线-高。
4	CAN_L	UART2_RX或RS485_A或CAN总线-低。
5	GND	
6	GND	5、6脚用作电源的GND。

## 2. 如夹爪版本号为30，即控制板中红框为拨码开关，其接口引脚定义如下表所示：

序号	PIN定义	功能描述
1	VIN	1、2脚用作电源输入，电压不超过24V DC。
2	VIN	
3	CAN_L	UART2_RX或RS485_B或CAN总线-低。
4	CAN_H	UART2_TX或RS485_A或CAN总线-高。
5	GND	
6	GND	5、6脚用作电源的GND。

！注：可通过夹爪包装或者机身底部的14位SN码区分两种夹爪，区分方式如下，

SN第6、7位为夹爪版本号，有20和30两种，

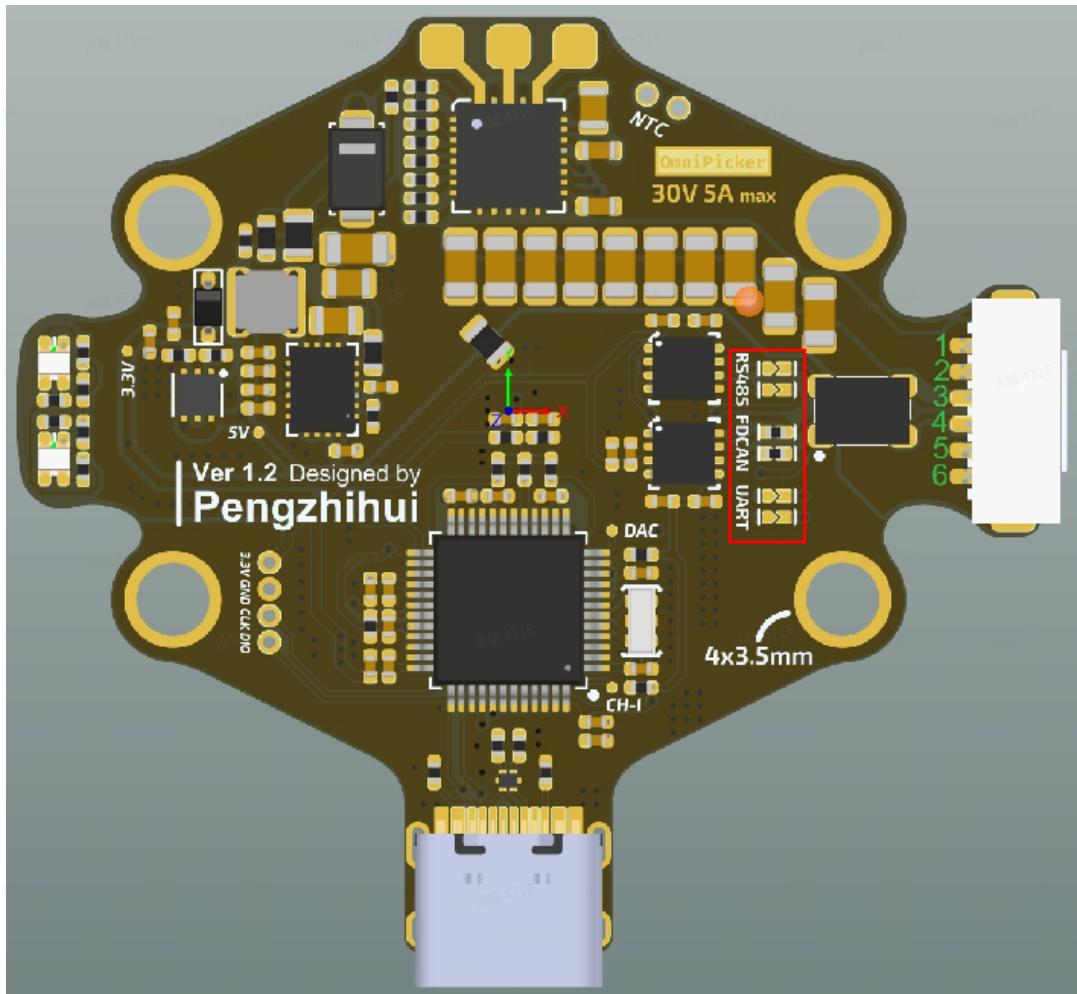
X1 0 06 **20** XXXXXXXX      X1 0 06 **30** XXXXXXXX

两种夹爪差异点为PCBA，具体见 **5.2 通信方式选择**

## 5.2 通信方式选择

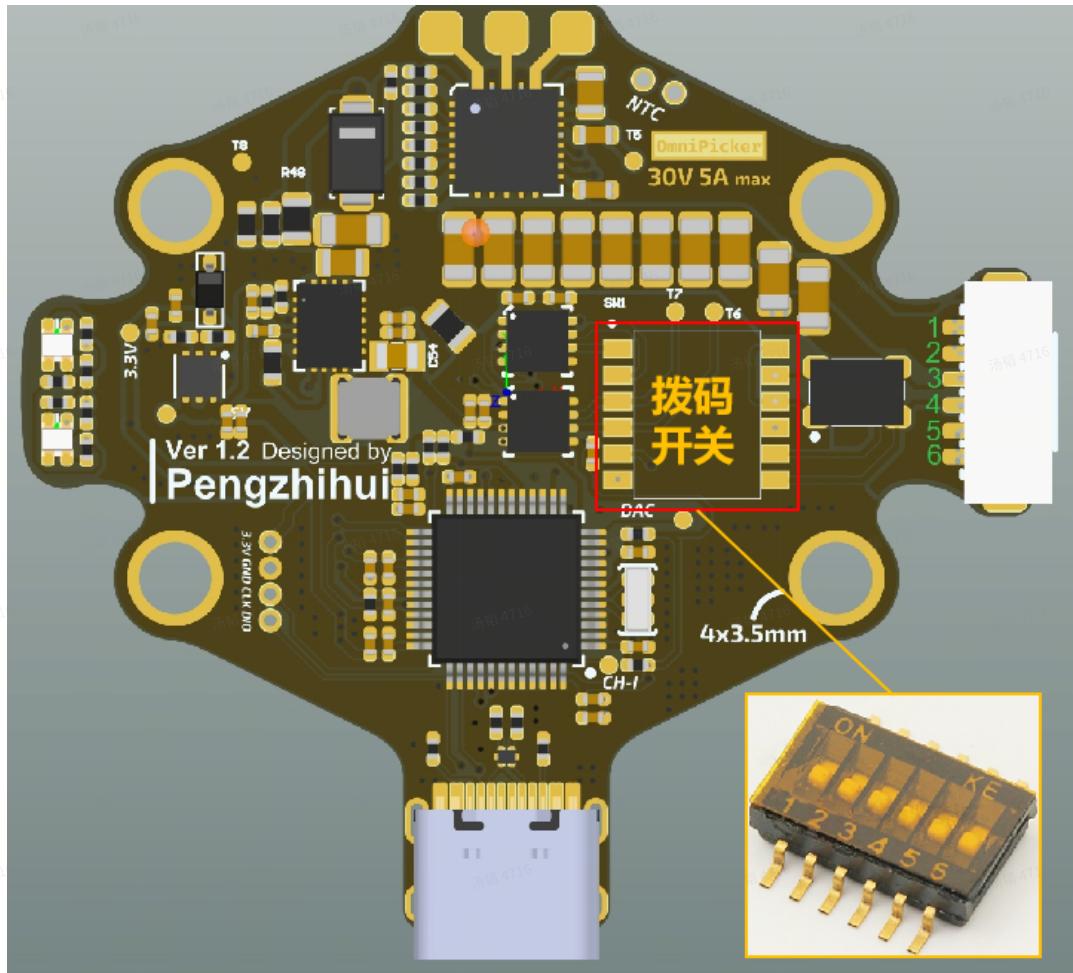
打开机器后盖，可根据下图中红框标出的内容区分两种夹爪PCBA状态，进而采用不同的操作方式。

1. 如夹爪版本号为**20**, 即控制板中红框中为电阻焊盘, 则通过跳线电阻的方式选择相应的通信方式。跳线电阻默认焊接在CAN/CANFD上。如需切换到其他通信方式则对应PCBA丝印, 将原CAN/CANFD的跳线电阻拆除后焊接到相应的通信接口电阻焊盘上。RS485焊接红框上方两个电阻, UART则焊接红框下方两个电阻。



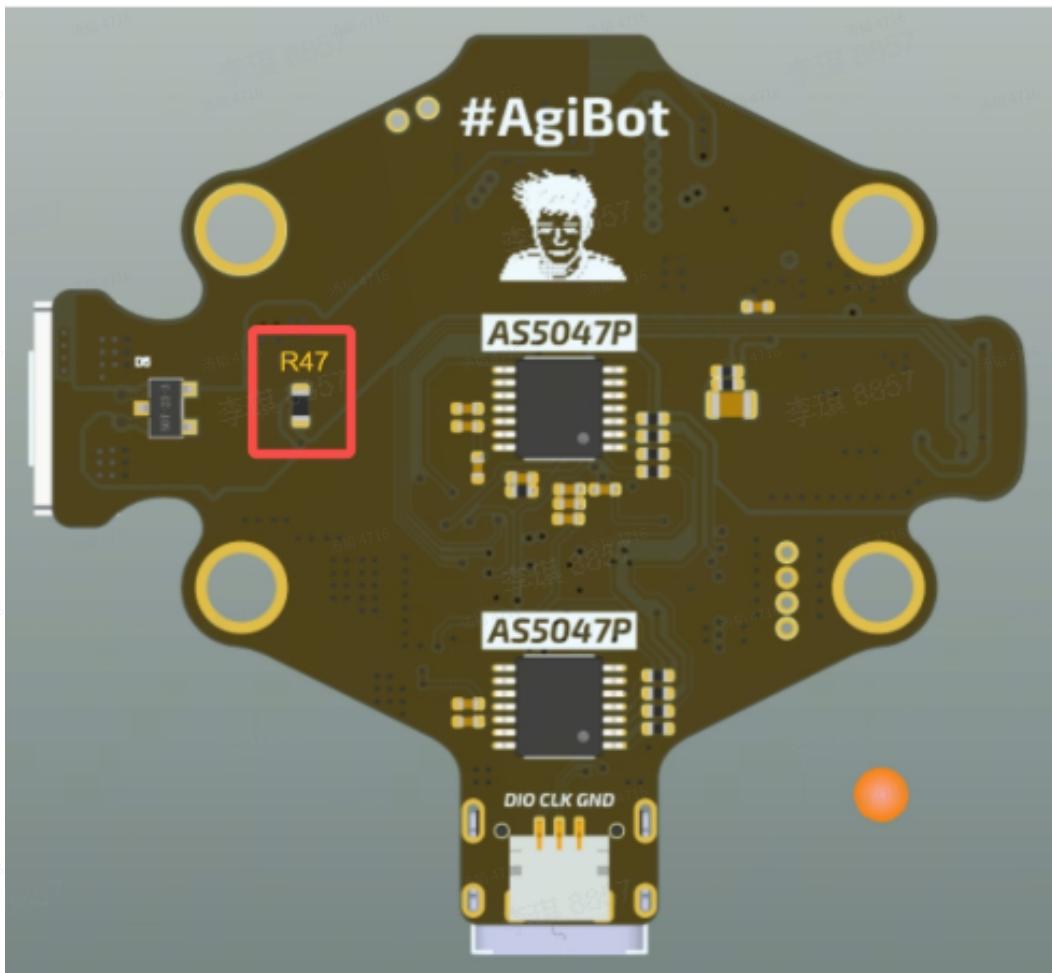
2. 如夹爪版本号为**30**, 即控制板中红框为拨码开关, 则按照如下操作方式。

- 默认为CAN/CANFD, 3和4引脚为ON的状态, 其他引脚处于OFF状态。
- 如果需要切换到RS485, 则1和2引脚拨到ON的状态, 其他引脚恢复OFF状态。
- 如果需要切换到UART, 则5和6引脚拨到ON的状态, 其他引脚恢复OFF状态。



3. 在PCB的背面焊有一颗 $120\Omega$ 的终端电阻，需要根据使用场景来决定是否保留，电阻位置见下图红框内部的R47。

- 当使用 CAN/CANFD/RS485 通信方式时，需要保留该电阻，出厂默认焊接
- 当使用 UART - TTL 通信方式时，需要手动拆除该电阻



## 6. 通信接口及协议

OmniPicker同时配备了多种通信接口，用于适配不同的工作环境。

一般配置步骤如下：

1. 先根据 [5.电气接口互联](#) 中的说明选择对应的通信接口，默认为CAN/CANFD
2. 通过USB连接上位机配置夹爪参数
3. 将夹爪接入对应的硬件系统中

### 6.1 LED状态指示

LED	含义
白色闪烁	失能状态
绿色常亮	使能状态（默认上电自使能）
红色常亮	错误状态
蓝色常亮	电机校准中（出厂默认校准完毕）

## 6.2 上位机控制

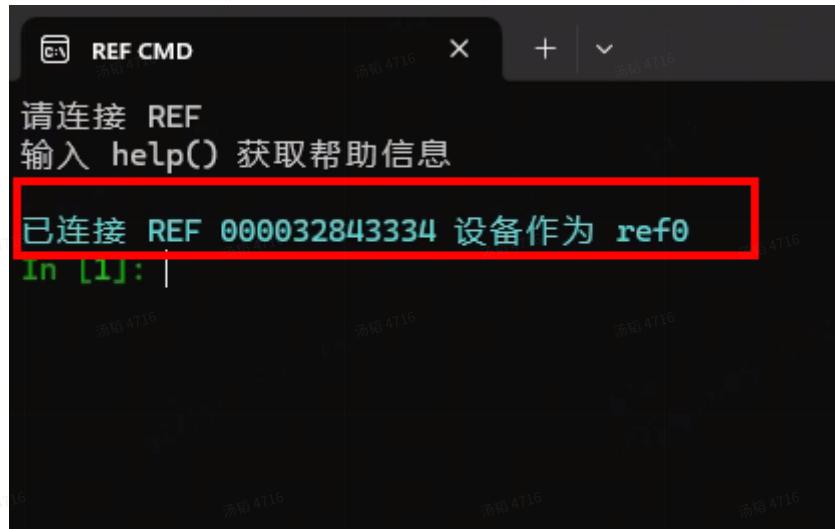
上位机可以通过命令行显式的更改参数，在第一次使用时建议充分了解其配置参数。

### 6.2.1 使用方法

准备一台Windows系统的电脑，并下载上位机软件 [REF-CLI v1.0.3.exe](#)。

确保夹爪的行程中没有障碍物后，将夹爪上电。若一切正常，夹爪会自动闭合以寻找零位，若未能闭合则需停止使用，检查夹爪运动是否流畅。

用USB-C数据线将夹爪连接至电脑，并打开上位机软件。等待上位机识别成功后，会有如下显示：



此时，编号为 000032843334 的设备在上位机中的索引名为 ref0。

在上位机中直接键入 ref0 并敲击回车，会显示当前夹爪的基本信息。根据下图所示，当前夹爪的固件版本为 1.3.4，CAN总线ID为8。

REF CMD

请连接 REF  
输入 help() 获取帮助信息

已连接 REF 0000398D464D 设备作为 ref0

In [1]: ref0

Out[1]:

```
serial_number = 0000398D464D (int)
fw_version = 134 (int)
can_node_id = 8 (int)
volume = 1.0 (float)
encoder_type = 3 (int)

motor:
    error_code = 0 (int)
    request_state(state: int)
    clear_error()
    config: ...
    ctrl: ...
debug_flag = 56 (int)
save_config()
erase_config()
reboot()
```

In [2]: |

## 6.2.2 更改CAN-ID

ref0.can\_node\_id 代表CAN总线的节点ID。可以直接查看其数值，也可以使用 = 进行赋值。执行以下命令。

### 代码块

```
1 ref0.can_node_id=8 // CAN ID设置为8
```

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

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

所有的参数变更，都需要调用 `save_config()` 方法进行保存，而该操作需要在 **失能** 状态下进行。

### 代码块

```
1 ref0.motor.request_state(0) // 先将夹爪 失能，0代表失能，1代表使能
2
3 ref0.save_config() // 保存所有参数
```

```

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

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

```

**注：所有参数变更，都在保存并重新上电后生效。**

### 6.2.3 控制夹爪运动

1. 夹爪重新上电。 (默认上电自使能)
2. 等待夹爪自动闭合。
3. 连接上位机，并调用 set\_pos(x) 方法设置其位置，其中x为开合百分比，范围 0.0 - 1.0，0为闭合，1为全部张开。

#### 代码块

```
1     ref0.motor.ctrl.set_pos(0.5) // 将夹爪运动至行程50%处。
```

## 6.3 CAN/CANFD 通信协议

当夹爪使用CANFD进行通信时，波特率为 仲裁域1M (80%采样点) + 数据域5M (75%采样点)，同时兼容波特率为1M (80%采样点) 的CAN协议。

**注：CAN协议固件版本不低于 3.3.0。**

### 6.3.1 下行控制

控制帧定义如下所示。

CAN ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7
can_no de_id	8	Reserve d	Pos Cmd	Force Cmd	Vel Cmd	Acc Cmd	Dec Cmd	Reserve d	Reserve d

- ID：夹爪的下行控制ID，对应 ref0.can\_node\_id
- Reserved：保留数据，必须写0
- Pos Cmd：目标位置，数值范围 0 - FF，0为夹紧，FF为完全张开
- Force Cmd：目标力矩，数值范围 0 - FF，FF为最大力矩
- Vel Cmd：目标速度，数值范围 0 - FF，FF为最大速度
- Acc Cmd：目标加速度，数值范围 0 - FF，FF为最大加速度

- Dec Cmd： 目标减速度，数值范围 0 - FF，FF为最大减速度

如控制夹爪运行至 50%行程处，其他参数全部给到最大值：

发送报文 00 7F FF FF FF FF 00 00

### 6.3.2 上行状态

上电并使能成功后，每当夹爪收到一次控制指令，会立即上报一帧状态数据。状态帧定义如下所示。

CAN ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7
can_no de_id	8	Fault Code	State	Pos	Vel	Force	Reserve d	Reserve d	Reserve d

- ID：夹爪的上行报文ID，对应 ref0.can\_node\_id
- Fault Code： 错误代码，含义如下

Code	含义
00	无故障
01	过温警报
02	超速警报
03	初始化故障警报
04	超限检测警报

- State： 当前状态，含义如下

Code	含义
00	已到达目标位置
01	夹爪移动中
02	夹爪堵转
03	物体掉落

- Pos： 当前位置，意义同 Pos Cmd

- Vel：当前速度，意义同 Vel Cmd
- Force：当前力矩，意义同 Force Cmd
- Reserved：预留位。D5-D7 三字节数据会直接复制发送的数据包，可自行用作校验，也可无视。

## 6.4 串口 TTL/485 通信协议

当夹爪使用串口 TTL 或 485 进行通信时，波特率需要更改为 115200@8N1（8bit数据位，0校验位，1停止位）。

**注：串口总线也支持串行链接，节点ID 复用 can\_node\_id。固件版本不低于 3.3.2。**

### 6.4.1 TTL/485 模式切换

使用USB-C数据线将夹爪连接至PC，打开上位机，输入ref0并敲击回车，查看 rs485\_mode 字段的值。

- 若使用TTL方式通信，将 rs485\_mode 置为 0 并进行保存。**注：背部终端电阻需要拆除。**

#### 代码块

```
1 ref0.rs485_mode = 0 // 关闭 485模式，使能 TTL
2
3 ref0.motor.request_state(0) // 失能
4 ref0.save_config() // 保存配置
```

- 若使用RS485方式通信，将 rs485\_mode 置为1并进行保存

#### 代码块

```
1 ref0.rs485_mode = 1 // 打开 485模式，关闭 TTL
2
3 ref0.motor.request_state(0) // 失能
4 ref0.save_config() // 保存配置
```

保存并重启后生效。

### 6.4.2 下行控制

控制帧定义如下所示，一帧12字节。

D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11
Frame Head	ID	Reser ved	Pos Cmd	Force Cmd	Vel Cmd	Acc Cmd	Dec Cmd	Reser ved	Reser ved	CheckS um	

- Frame Head: 帧头, 0x4141, 共2字节
- ID: 复用 can\_node\_id
- D3 - D10: 与CAN总线下行协议 D0 - D7 定义相同
- CheckSum: D2 - D10 的数据和校验, 计算方法请参考下方Python代码

#### 代码块

```

1 def checksum(buf) -> int:
2     ret = 0
3     for i in buf:
4         ret += i
5     return (~ret) & 0xff

```

如控制夹爪运行至 50%行程处, 假设ID为1, 其他参数全部给到最大值:

发送数据 41 41 01 00 7F FF FF FF FF 00 00 83

### 6.4.3 上行状态

上电并使能成功后, 每当夹爪收到一次控制指令, 会立即上报一帧状态数据。状态帧定义如下所示, 一帧12字节。

D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11
Frame Head	ID	Fault Code	State	Pos	Vel	Force	Reser ved	Reser ved	Reser ved	CheckS um	

- Frame Head: 帧头, 0x4141, 共2字节
- ID: 复用 can\_node\_id
- D3 - D10: 与CAN总线上行协议 D0 - D7 定义相同
- CheckSum: 计算方法同下行协议, 校验范围为 D2 - D10

## 6.5 固件升级

### 6.5.1 升级方法

1. 夹爪上电连接PC, 并打开 REF-CLI 上位机, 确保设备已正常连接并初始化。

2. 将夹爪控制至0.5行程，为校准做好准备。执行命令后，观察夹爪，确认已经张开至50%。

```
已连接 REF 00003952464D 设备作为 ref0
In [1]: ref0.motor.ctrl.set_pos(0.5)
In [2]: |
```

**注：此步骤非常重要，请严格根据说明执行。**

3. 上位机中输入升级指令并等待进度结束，其中 path\_to\_firmware\_file 是固件在系统中的绝对路径，此处以3.3.0版本为例。

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

```
REF-CLI v1.0.1
软件版本: Ver-1.0.1
请连接 REF
输入 help() 获取帮助信息
已连接 REF 00003952464D 设备作为 ref0
In [1]: ref0.motor.ctrl.set_pos(0.5)
In [2]: ota(ref0, r"D:\Agibot\X1\FW_OMP\OmniPicker-fw-ota-v330.bin")
upgrade REF 961693261 app
设备 ref0 已断开连接
enter dfu...
searching REF 961693261 device
erase start
erase finish
flash firmware
100%|██████████| 220k/220k [00:14<00:00, 16.0kB/s]
flash done
verify firmware
100%|██████████| 220k/220k [00:14<00:00, 16.0kB/s]
verify done
Dfu.AppInfo(magic_number=4237390690, app_version=0, app_start_addr=134250496, app_firmware_length=225388, check_sum=b'\xa9\xfbn-\x99\x05\k\x a6xJ\xd7\x10\x11\xb5\x87')

恢复连接 REF 00003952464D 设备作为 ref0
In [3]:
```

注：升级过程中请注意不要操作或者挪动夹爪，以免导致USB通信异常。

4. 升级完成后，终端中会重新出现“恢复链接 REF XXX”的提示。

注：若升级完成后，夹爪自动闭合，请跳过此步骤，直接前往5

观察LED指示灯，等待白灯开始闪烁后，输入校准命令。

#### 代码块

```
In [1]: ref0.motor.request_state(2)
```

```
In [5]: ref0.motor.request_state(2)
Out[5]: True
```

此时LED由白色闪烁变为蓝色常亮，并且夹爪会开始慢速运动，此时不要操作夹爪，等待校准完成。

当校准成功后，夹爪会发出声音，并且LED由蓝色常亮再次变回白色闪烁。

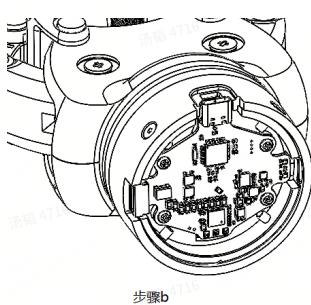
**！注：在执行校准之前，若夹爪未能张开或已经无法控制，请按以下步骤手动张开夹爪。**

- a. 拔掉电源以及所有连接线；
- b. 手动拆掉后盖，如下图；
- c. 手动拆掉控制板上的4个十字螺丝，如下图；
- d. 将控制板掀起，露出磁铁，注意掀起过程中控制板的3根焊线不能断开，如下图；

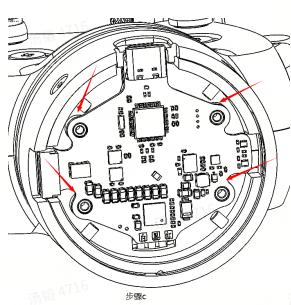
e. 手动逆时针旋转磁铁支架，直至OmniPicker张开约50%的行程，如下图；

f. 复将控制板装回，锁紧4个十字螺丝；

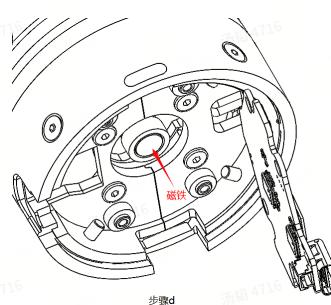
g. 盖上后盖，锁紧螺丝。



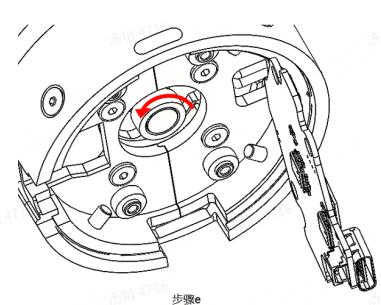
步骤b



步骤c



步骤d



步骤e

5. 夹爪重新上电，自动开机且闭合，升级完成。

```
In [3]: ref0
Out[3]:
device_type = 1 (int)
serial_number = 00003952464D (int)
fw_version = 330 (int)          版本已更新
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()
```

6. 若升级失败，可以使用 flash\_app命令重新写入文件。

#### 代码块

```
1  flash_app(r"PATH") // PATH是升级文件在系统中的绝对路径
```

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

注：使用该命令的REF-CLI 版本不低于1.0.3。

## 6.5.2 固件发布

版本	日期	固件	备注
3.3.0	2024.12.2 6	 OmniPicker-fw-ota-v330.bin 220.11KB	<ol style="list-style-type: none"><li>新增 串口通信支持 (485)</li><li>新增 CAN协议报文支持</li></ol>
3.3.3	2024.1.10	 OmniPicker-fw-ota-v333.bin 220.48KB	<ol style="list-style-type: none"><li>新增 串口通信支持 (TTL)</li></ol>

## 7. 其他

本手册基于OmniPicker 硬件 Ver1.2，软件 Ver3.2.8 编写。

其他产品手册中未注明事项，请查阅智元官网，或联系智元服务人员获取信息。

## 8. 附录

请使用ROS2 Humble加载夹爪的URDF文件。

