产品使用说明书

智元 OmniHand 灵动款 2025灵巧手

版本：1.1

语言：简体中文

目录

1. 重要安全须知

本章节介绍了安装和使用 OmniHand 灵动款 2025 灵巧手时必须遵守的安全要求，OmniHand 灵巧手是一款半成品机械，因此，在首次使用之前，必须仔细阅读本说明书，全面了解相关安全功能及潜在危险。相关人员在每次安装 OmniHand 灵巧手之前，必须进行风险评估，并在使用过程中严格遵守本说明书中所有说明和指导信息。

1.1 使用注意事项

严禁将 OmniHand 灵巧手置于高温、高湿的环境中。

请勿让 OmniHand 灵巧手承受过大的负载和冲击。

严禁在指尖施加过大的力。

严禁用力拉扯 OmniHand 灵巧手手指。

严禁将 OmniHand 灵巧手靠近火源。

严禁将 OmniHand 灵巧手置于易燃、易爆环境中。

严禁在强电磁场环境中使用，如高压电线附近、大功率机器附近等。

严禁用 OmniHand 灵巧手去抓取过重、过热、尖锐、表面粗糙、有腐蚀性的物体。

在没有保护的情况下，严禁让 OmniHand 灵巧手接触液体（酒、水、饮料等）和沙尘，如不慎接触，请立即关闭，并联系售后维修人员。

严禁私自拆解 OmniHand 灵巧手，否则将导致《售后服务承诺书》中保修相关条款失

效。发生任何故障，请及时联系售后维修人员。

严禁用 OmniHand 灵巧手操作危险机械。由于此类行为导致的人身伤害及财产损失，我司不承担任何责任。

1.2 风险评估

潜在危害清单

机械危害：因接触 OmniHand 灵巧手或其夹持的尖锐物体而导致压伤、擦伤等。

电气危害：因接触带电零件而导致电烧伤或触电。

高温危害：因长时间接触 OmniHand 灵巧手的高温表面而导致皮肤烧伤等。

噪音危害：无。

振动危害：无。

辐射危害：无。

材料/物质危害：无。

人体工程学危害：无。

环境危害：无。

1.3 售后服务承诺书

OmniHand 灵动款 2025 灵巧手具有 6 个月有限保修期。若在投入使用后，出现因制造或材料不良所致的缺陷，公司提供必要的备用部件予以更换或维修。若设备缺陷是由处理不当或未遵循产品使用说明书中所述的相关信息或私自拆卸产品所致，则本产品质量保证即告失效。在不违背本产品质量保证的原则下，若产品已经超出保修期，公司保留向客户收取更换或维修费用的权利。

在保修期外，如果设备呈现缺陷，公司不承担由此引起的任何损害或损失，包括但不仅限于生产损失或对其他生产设备造成的损坏。

1.4 免责声明

公司致力于不断提高产品可靠性和性能，并因此保留升级产品的权利，恕不另行通知。公司力求确保本手册内容的准确性和可靠性，但不对其中的任何错误或遗漏信息负责。以下情况导致的故障不在本保修范围内：

1）未按用户手册要求安装、接线、连接其他控制设备；

2）未经允许，私自拆装灵巧手；

3）使用时超出用户手册所示规格或标准；

4）由于运输不当导致的产品损坏；

5）事故或碰撞导致的损坏；

6）火灾、地震、海啸、雷击、大风和洪水等自然灾害。

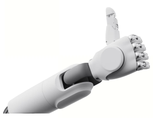
本公司对于因客户违反本章节内的免责声明而造成的任何损失或伤害不承担任何责任。请客户在购买和使用本产品前，仔细阅读并同意本免责声明。

2. 产品简介

2.1 产品概述

OmniHand 灵动款 2025 是一款紧凑型高自由度交互灵巧手，小身形，真百搭。

OmniHand 灵动款 2025 用途多样，支持手势交互、触摸交互、轻作业等场景应用，支持多种常见交互手势，覆盖手心、手背和五指达 400+ 点位触觉感应。采用手感漆面，防夹手安全设计。轻巧灵动，适合小尺寸人形机器人(身高 110~130cm)。





2.2 主要特性

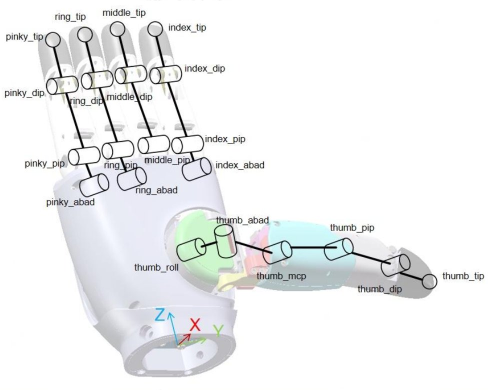
- **紧凑轻巧：**总长 180 mm、轻至 550g，尺寸百搭，各种型号人形机器人理想双手
- **灵动交互：**16 自由度，手势交互更拟人，常见交互手势全覆盖，独创手背触摸交互
- **安全亲和：**全手 400+ 触点感应力控，防夹手设计，交互更安全

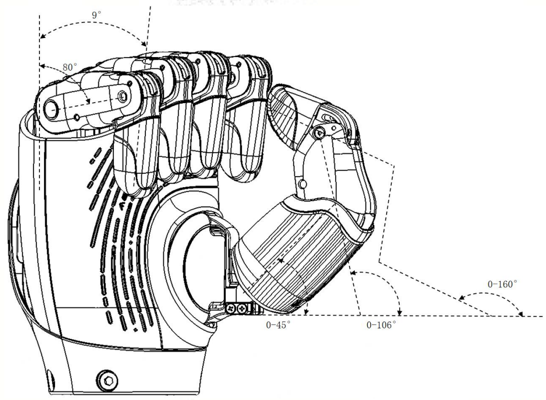
2.3 产品参数

		OmniHand 灵动款 2025	OmniHand 灵动触觉款 2025
本体参数	重量（不含手腕板护盖及螺钉）	≤550g	≤570g
	尺寸	180*85*38.5mm	
	主动自由度	10（大拇指：3；食指：2；中指：1；无名指：2；小指：2）	
	总自由度	16（大拇指：4；食指：3；中指：3；无名指：3；小指：3）	
	最小张开/闭合时间	0.5s / 0.5s	
	指尖重复定位精度	0.3mm	
	负载能力	五指抓握力≥5kg	
	工作电压	18-27V	
	通讯接口	CAN-FD / RS485	
	工作温度范围	-10~45℃	
	在线升级	支持 OTA 在线升级	
触觉传感器	力感知类型	/	指尖+手掌+手背：一维力
	阵列分辨率	/	0.1N
	感知范围	/	0~20N
	最大接受力(不损坏)	/	200N

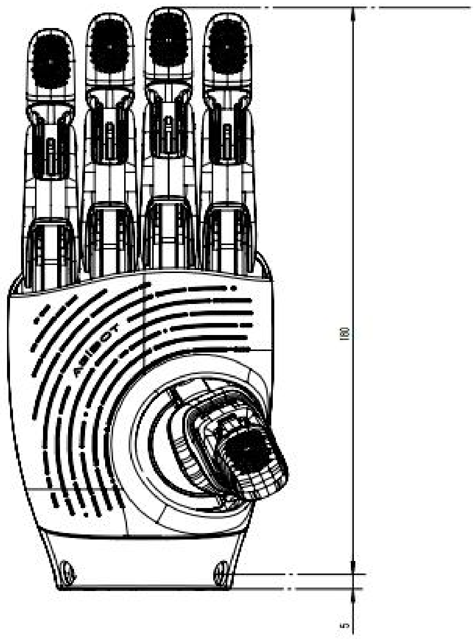
2.4 关节角度范围

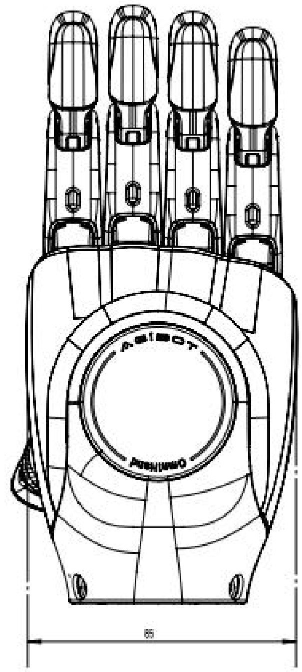
手指	关节	运动方向	角度最小值	角度最大值
四指	DIP	屈伸	0°	171°
	PIP	屈伸	0°	80°
	ABAD	侧摆	0°	10°
大拇指	DIP	屈伸	0°	160°
	PIP	屈伸	0°	106°
	MCP	屈伸	0°	45°
	ABAD	侧摆	0°	90°
	ROLL	自旋	0°	60°

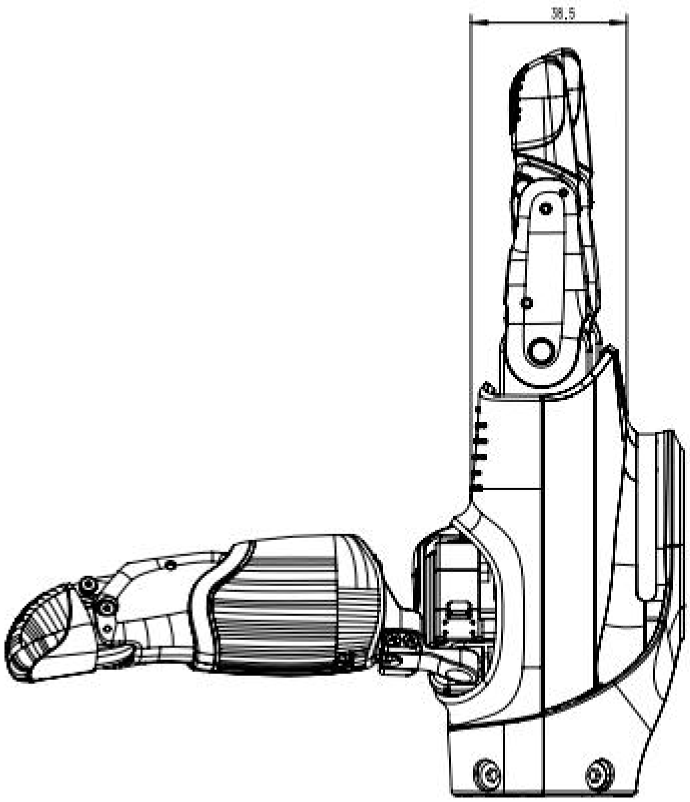




2.5 产品尺寸







3. 安装说明

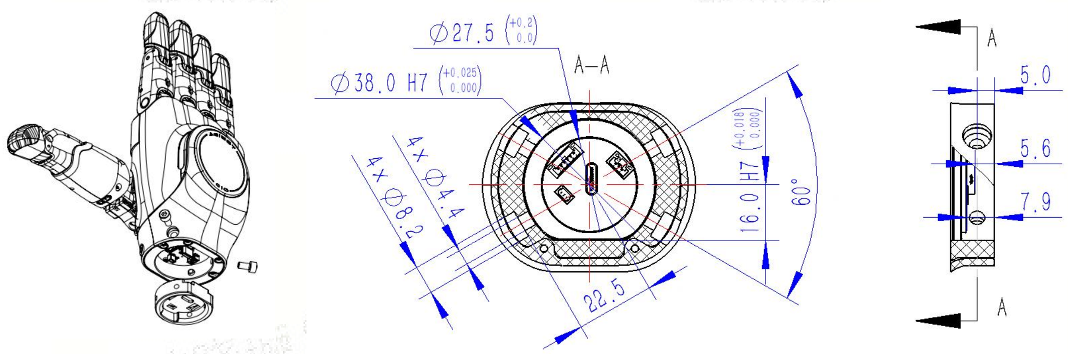
本章节介绍了如何安装 OmniHand 灵巧手，以及电气和通讯接口的相关信息。

3.1 装箱清单

- OmniHand 灵动款 2025 × 1
- (电源+CANFD 通讯线) × 1--电源线端子类型：XT30 母头
- RS485 通讯线 × 1
- USB Type-C 通讯线 × 1
- 产品合格证 × 1

3.2 机械接口

1. 拧掉手腕处 4 颗 M4 螺钉，拆掉手腕后端盖；
2. 连接通讯线和电源线（需要用户自备 24V 电源供电）；
3. 将灵巧手装到机器人手臂处（手臂机械接口设计参考下图），固定 步骤 1 拆下的 M4 螺钉，即完成安装。



3.3 电气和通讯接口

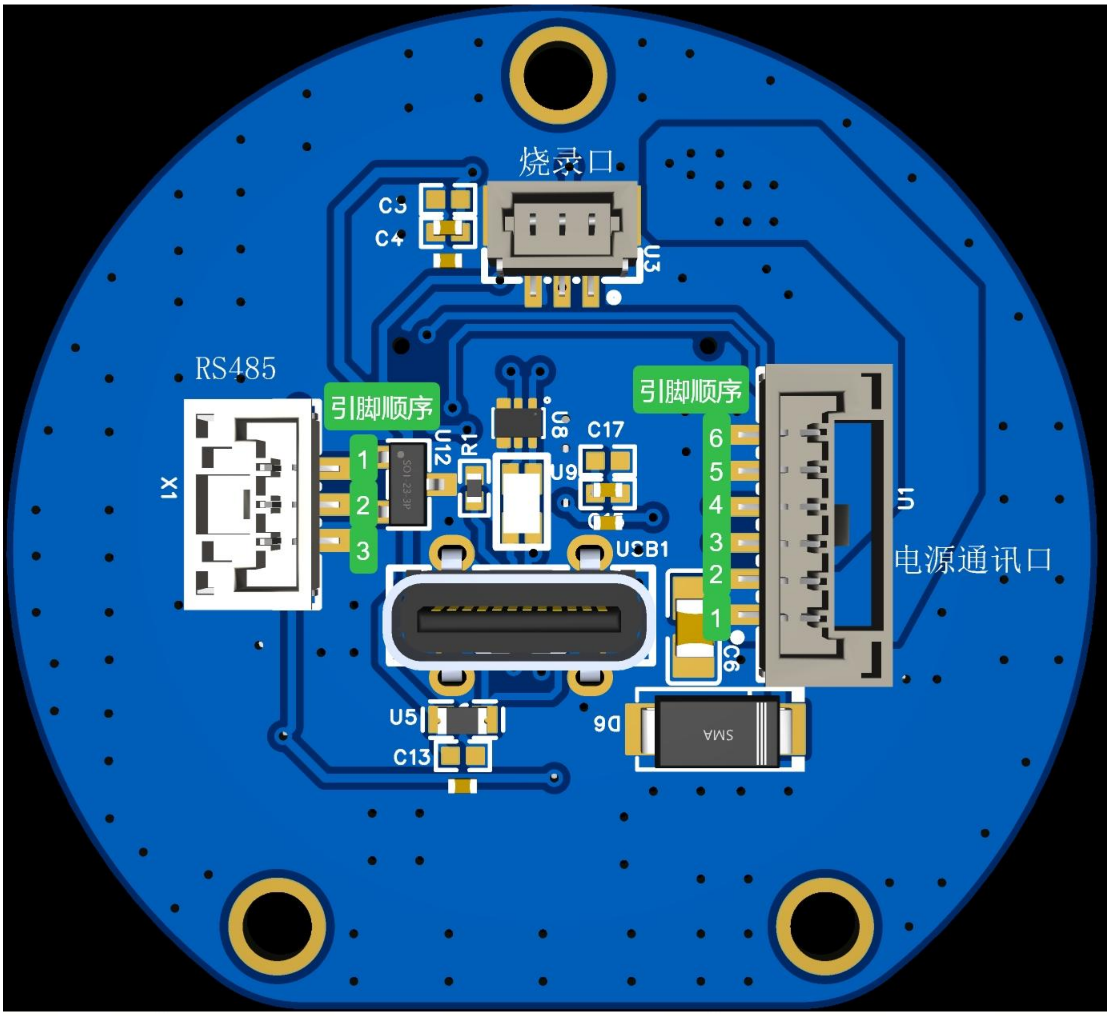
引脚定义

连接器采用 WAFER-GH1.25-6PLB (位号：U1)，支持 CANFD 。具体引脚定义如下：

引脚	引脚定义
1	24V
2	24V
3	CAN_H
4	CAN_L
5	GND
6	GND

连接器采用 GH125-S03CCA-00 (位号：X1)，支持 RS485。具体引脚定义如下：

引脚	引脚定义
1	RS485_A
2	RS485_B
3	GND



通信方式

支持用户通过 USB 通讯方式，连接上位机界面操作灵巧手；

支持基于 CANFD/RS485 通讯方式，进行灵巧手应用二次开发。

4. 通讯协议说明

4.1 CAN-FD 通信帧格式

CAN-FD 通信参数：仲裁域波特率 1M（80%采样点），数据域波特率 5M（75%采样点），不响应经典 CAN。

以下为 CAN-FD 数据域中的通信帧布局：

ID	DLC	D0	D1 - D63
节点 ID	1-64	CMD	数据段

- 节点 ID：0 ~ 0x7FF，标准帧。默认 ID 为 0x09，当节点 ID 未知时，可以利用寻呼地址 **0x7FF** 来查询节点 ID。节点在收到此 ID 数据后依然按照自身 ID 进行回复。
- CMD：控制指令，占据数据域中的首字节，具体定义参见第三章
- 数据段：与控制指令对应的数据，具体定义参见第三章 上下行数据定义，所有数据按照小端排布。

4.2 USB/串口 通信帧格式

串口通信参数设置：

- 波特率：460800
- 字节总数：8
- 校验位：无
- 停止位：1

使用 USB 连接时，驱动板会虚拟出一个串口，波特率可随意设置，其他参数保持一致。

字段	帧头	ID	数据长度	数据段（CMD+DATA）	CRC
字节占用	2	2	1	N	2

- 帧头：0xAAEE
- ID: 设备节点 ID， 0~0x7FF。默认 ID 为 0x01，当节点 ID 未知时，可以利用寻呼地址 **0x7FF** 来查询节点 ID。节点在收到此 ID 数据后依然按照自身 ID 进行回复。
- 数据长度：数据段字节长度
- 数据段：长度为 1-64，首字节为控制指令，其后跟随对应数据，与 CANFD 中的数据域保持一致，所有数据按照小端排布
- CRC：数据校验，校验范围为 从帧头至数据段，计算方法参考如下文件中的 Crc16 函数。预留 0x5555 为调试专用 CRC，可直接通过校验。

```C++
/*
 * @Author: richie.li
 * @Date: 2023-08-21 21:46:17
 * @LastEditors: richie.li
 * @LastEditTime: 2023-12-26 17:07:10
 */

/* Private includes -----------------------------------------------
------------*/
#include "crc.h"

/* Private types --------------------------------------------------
------------*/
/* Private macro --------------------------------------------------
------------*/
/* Private constants ----------------------------------------------
------------*/
/* Private variables ----------------------------------------------
------------*/
static const uint16_t Crc16Tab[256] = {
    0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50a5, 0x60c6,
0x70e7, 0x8108, 0x9129, 0xa14a, 0xb16b,
    0xc18c, 0xd1ad, 0xe1ce, 0xf1ef, 0x1231, 0x0210, 0x3273,
0x2252, 0x52b5, 0x4294, 0x72f7, 0x62d6,
    0x9339, 0x8318, 0xb37b, 0xa35a, 0xd3bd, 0xc39c, 0xf3ff,
0xe3de, 0x2462, 0x3443, 0x0420, 0x1401,
    0x64e6, 0x74c7, 0x44a4, 0x5485, 0xa56a, 0xb54b, 0x8528,
0x9509, 0xe5ee, 0xf5cf, 0xc5ac, 0xd58d,
    0x3653, 0x2672, 0x1611, 0x0630, 0x76d7, 0x66f6, 0x5695,
0x46b4, 0xb75b, 0xa77a, 0x9719, 0x8738,
    0xf7df, 0xe7fe, 0xd79d, 0xc7bc, 0x48c4, 0x58e5, 0x6886,
0x78a7, 0x0840, 0x1861, 0x2802, 0x3823,
    0xc9cc, 0xd9ed, 0xe98e, 0xf9af, 0x8948, 0x9969, 0xa90a,
0xb92b, 0x5af5, 0x4ad4, 0x7ab7, 0x6a96,
    0x1a71, 0x0a50, 0x3a33, 0x2a12, 0xdbfd, 0xcbdc, 0xfbbf,
0xeb9e, 0x9b79, 0x8b58, 0xbb3b, 0xab1a,
    0x6ca6, 0x7c87, 0x4ce4, 0x5cc5, 0x2c22, 0x3c03, 0x0c60,
0x1c41, 0xedae, 0xfd8f, 0xcdec, 0xddcd,
    0xad2a, 0xbd0b, 0x8d68, 0x9d49, 0x7e97, 0x6eb6, 0x5ed5,
0x4ef4, 0x3e13, 0x2e32, 0x1e51, 0x0e70,
    0xff9f, 0xefbe, 0xdfdd, 0xcffc, 0xbf1b, 0xaf3a, 0x9f59,
0x8f78, 0x9188, 0x81a9, 0xb1ca, 0xa1eb,
    0xd10c, 0xc12d, 0xf14e, 0xe16f, 0x1080, 0x00a1, 0x30c2,
```

```
0x20e3, 0x5004, 0x4025, 0x7046, 0x6067,
    0x83b9, 0x9398, 0xa3fb, 0xb3da, 0xc33d, 0xd31c, 0xe37f,
0xf35e, 0x02b1, 0x1290, 0x22f3, 0x32d2,
    0x4235, 0x5214, 0x6277, 0x7256, 0xb5ea, 0xa5cb, 0x95a8,
0x8589, 0xf56e, 0xe54f, 0xd52c, 0xc50d,
    0x34e2, 0x24c3, 0x14a0, 0x0481, 0x7466, 0x6447, 0x5424,
0x4405, 0xa7db, 0xb7fa, 0x8799, 0x97b8,
    0xe75f, 0xf77e, 0xc71d, 0xd73c, 0x26d3, 0x36f2, 0x0691,
0x16b0, 0x6657, 0x7676, 0x4615, 0x5634,
    0xd94c, 0xc96d, 0xf90e, 0xe92f, 0x99c8, 0x89e9, 0xb98a,
0xa9ab, 0x5844, 0x4865, 0x7806, 0x6827,
    0x18c0, 0x08e1, 0x3882, 0x28a3, 0xcb7d, 0xdb5c, 0xeb3f,
0xfb1e, 0x8bf9, 0x9bd8, 0xabbb, 0xbb9a,
    0x4a75, 0x5a54, 0x6a37, 0x7a16, 0x0af1, 0x1ad0, 0x2ab3,
0x3a92, 0xfd2e, 0xed0f, 0xdd6c, 0xcd4d,
    0xbdaa, 0xad8b, 0x9de8, 0x8dc9, 0x7c26, 0x6c07, 0x5c64,
0x4c45, 0x3ca2, 0x2c83, 0x1ce0, 0x0cc1,
    0xef1f, 0xff3e, 0xcf5d, 0xdf7c, 0xaf9b, 0xbfba, 0x8fd9,
0x9ff8, 0x6e17, 0x7e36, 0x4e55, 0x5e74,
    0x2e93, 0x3eb2, 0x0ed1, 0x1ef0};

/* Private function prototypes -----------------------------------
------------*/
/* Private user code ---------------------------------------------
------------*/
uint16_t Crc16(const uint8_t* buf, uint32_t len) {
  uint16_t crc = 0;
  while (0 < len--) {
    crc = (crc << 8) ^ Crc16Tab[((crc >> 8) ^ *buf++) & 0x00FF];
  }
  return crc;
}
```


韩晓璇 9793

4.3 通信协议内容

手内部的主动轴按照序号进行分布，索引为 1-10（如下图：红色）；触觉传感器序号从拇指开始依次分布，索引为 1-7（如下图：黑色）。

发送和回复的命令字段相同。

命令	定义	下行数据	上行数据	备注
0x01	设备使能或失能	1 字节 00：失能 01：使能 02：校准模式	1 字节 00：失败 01：成功	
0x02	查询当前设备使能状态		1 字节 00：失能 01：使能 02：校准模式	
0x04	设置 ID，CANFD	2 字节，代表 ID 范围 [1, 0x7FF)	1 字节 00：失败 01：成功	

0x05	回初始位置		1 字节 00：失败 01：成功	阻塞执行
0x06	设置单个轴位置	3 字节 d0: 轴索引 d1-d2: 目标位置，uint16 类型，范围 0 - 4096；将整个角度运行范围分为 4096 份，比如手指运行 alph=0.0°~90.0°，目标位置 = 4096 * alph / 3600，角度精确到 0.1°	3 字节 d0: 轴索引 d1-d2: 当前位置，uint16 类型，范围 0 - 4096；将整个角度运行范围分为 4096 份，比如手指运行 alph=0.0°~90.0°，目标位置 = 4096 * alph / 3600，角度精确到 0.1°	
0x07	读取单个轴位置	1 字节，轴索引 范围 1-12	同 0x06 指令回复	
0x08	设置全部轴位置	20 字节 d0-d19，所有轴目标位置数据 每个轴两字节位置数据，uint16 类型，从 1 号轴开始依次排布，每个轴位置范围 0 - 4096；将整个角度运行范围分为 4096 份，比如手指运行 alph=0.0°~90.0°，目标位置 =	60 字节 d0-d19, 所有轴当前位置数据 每个轴两字节位置数据，uint16 类型，从 1 号轴开始依次排布，每个轴位置范围 0 - 4096 d20-d39：所有轴速度反馈； d40-d49：所有轴力矩反馈； d50-d59：所有轴的故障状态；	

		4096 * alph / 3600，角度精确到 0.1° d20-d23：0，预留自由度；		
0x09	读取全部轴位置数据		20 字节 d0-d19, 所有轴当前位置数据 每个轴两字节位置数据，uint16 类型，从 1 号轴开始依次排布，每个轴位置范围 0 - 4096	
0x0A	读取所有轴电流信息		20 字节 d0-d19, 所有轴当前电流数据，单位：0.01A； 每个轴两字节电流数据，uint16 类型，从 1 号轴开始依次排布；	
0x0B	读取所有轴速度信息		20 字节 d0-d23, 所有轴当前速度数据 每个轴两字节速度数据，uint16 类型，从 1 号轴开始依次排布；	
0x0C	读取全部轴温度数据		10 字节 d0-d9,代表 10 个轴的当前温度数据 每个轴 1 字节温度数据，int8 类型，从 1 号轴开始依次排布，单位为度	

0x0D	读取错误码		2 字节，uint16 类型 0x0000：无错误 1-10：电机 1-10 堵转 21-30：电机 1-10 过温 41-50：电机 1-10 电流异常 61-70：电机 1-10 编码器异常 81-90：电机 1-10 通讯异常 101：通讯异常 102：电机初始化故障 103：触觉传感器初始化故障 104：自动校准故障 105：参数错误	
0x0E	清除错误		1 字节 00：失败 01：成功	
0x0F	播放内置动作	1 字节，动作序号	1 字节 00：失败 01：成功	该指令暂未启用
0x10	读取 10 个轴的位置量程		20 字节 d0-d19, 所有轴当前位置量产数据 每个轴两字节，从 1 号轴开始依次排布，uint16 类型，数据单位为 0.1 度	

0x11	读取单个指尖传感器的数据	1 字节，传感器索引	17 字节（手指）/26 字节（手心手背） d0，代表传感器索引 d1-d16：手指传感器的数据，4*4 点，每点 uint8，按行读取； d1-d25：手心/手背传感器数据，5*5 点，每点 uint8，先按行再按列读取；	
0x12	读取所有指尖传感器的数据 1		48 字节，顺序为： 大拇指 4*4 + 食指 4*4 + 中指 4*4 每点 uint8，每个传感器先按行再按列读取；	
0x13	读取所有指尖传感器的数据 2		32 字节，顺序为： 无名指 4*4 + 小拇指 4*4 每点 uint8，每个传感器先按列再按行读取；	
0x14	读取所有指尖传感器的数据 3		50 字节，顺序为： 手心 5*5 + 手背 5*5 每点 uint8，每个传感器先按行再按列读取；	
0x15	运行模式	2 字节 byte1：电机 ID； byte2：运行模式	1 字节 00：失败 01：成功	
0x1A	读取全部轴负载数据		20 字节 d0-d19, 所有轴当前负载	只适用于 1、

			数据 每个轴两字节负载数据，uint16 类型，当前控制输出驱动电机的电压占空比，单位：0.1%	2、3、5、6、8、10 号电机
0x20	设置电机运行速度	20 字节 d0-d19，所有轴目标速度数据 每个轴两字节速度数据，int16 类型，从 1 号轴开始依次排布，每个轴位置范围 − 4096 ~ 4096	1 字节 00：失败 01：成功	
0x21	设置电机过载扭矩	2 字节 byte1：电机 ID； byte2：过载扭矩；0~1000 代表最大扭矩的 0%~100.0%；单位：0.1%	1 字节 00：失败 01：成功	只适用于 4、7、9 号电机
0x22	设置电机过载时间	2 字节 byte1：电机 ID； byte2：过载时间；达到最大扭矩并保持过载时间后，进入保护扭矩输出阶段；单位：0.01s	1 字节 00：失败 01：成功	
0x23	设置保护扭矩	2 字节 byte1：电机 ID； byte2：保护扭	1 字节 00：失败	

		矩；0~100 代表最大扭矩的 0%~100%；单位：1%	01：成功	
0x24	设置最小启动力	3 字节 d0: 轴索引 HLS3606 舵机: d1: 最小启动力，uint8 类型，堵转扭力的百分比，单位 0.1% SCS2304 舵机: d1-d2: 最小启动力，uint16 类型，堵转扭力的百分比，单位 0.1%	1 字节 00：失败 01：成功	
0x25	设置最大扭矩	3 字节 d0: 轴索引 d1-d2: 最大输出扭矩，0~1000 代表最大扭矩的 0%~100.0%；单位：0.1%	1 字节 00：失败 01：成功	
0x27	读取所有传感器 ID		7 字节 d0-d6,代表 7 个传感器的 ID 数据 每个传感器 1 字节 ID 数据，int8 类型，从 1 号轴开始依次排布	
0x28	设置 PLOT 数据	2 字节 ，单位毫	1 字节	

	上报时间间隔	秒	00：失败 01：成功	
0x29	获取所有轴的 PLOT 数据		60 字节 每个轴占用 6 字节数据，从轴 1 开始排布，至轴 10 6 字节数据中，前 2 字节为位置，中间 2 字节为速度，后 2 字节为电流	
0x31	设置左手/右手	1 字节 0：右手（默认） 1：左手	1 字节 00：失败 01：成功	
0x80	控制源	1 字节，控制源设定： 0：机器人本体控制；（默认） 1：人机界面操作员控制。	返回：原数据	
0x81	控制源查询	0 字节	返回：控制源状态 0：机器人本体控制； 1：人机界面操作员控制。	
0xC1	设置产品系列号	19 字节： d0~d2：3 位供应商暗码 d3~d8：6 位物料暗码 d9~d14：YYMMDD，年月	1 字节 00：失败 01：成功	

		日 d15~d18：4 位流水号		
0xC2	读取产品系列号		19 字节： d0~d2：3 位供应商暗码 d3~d8：6 位物料暗码 d9~d14：YYMMDD，年月日 d15~d18：4 位流水号	
0xCD	查询设备型号以及软硬件版本		8 字节 d0: 设备类型 2：O10 灵巧手 d1-d2：产品状态（ASCII 码，如 T1、T2、P1 等） d3-d5：软件版本（d1.d2.d3） d6-d8：硬件版本（d4.d5.d6） d9: 自由度	兼容 OmniPicker 指令

说明：上下行数据介绍中，d 代表 data，d0 即为第一个字节。

4.4 固件/上位机版本

固件和上位机最新版本可以从官网（https://www.zhiyuan-robot.com/DOCS/OS/Omnihand-O10）上下载。

4.5 控制注意事项

机器人本体控制灵巧手，建议使用位置+力矩的复合模式，使用合适的力矩对物体进行抓握。如果抓握力矩过大，**建议不要频繁抓握，抓握一分钟，放开一分钟为宜。**

4.6 指示灯功能说明

当手背灯显示蓝色，表明整机正在初始化；蓝灯变为绿灯，表明初始化结束；当电压过低或者过高时，手背灯无法从蓝色变为绿色，请注意及时检查电压是否满足 24V+/-10%的要求。

4.7 整手不支持热插拔

整手在带电状态下，不支持电源及信号线热插拔功能，请使用者规范使用；如若未按要求进行热插拔出现故障，客户承担相应责任。

5. 上位机使用说明

5.1 运行环境

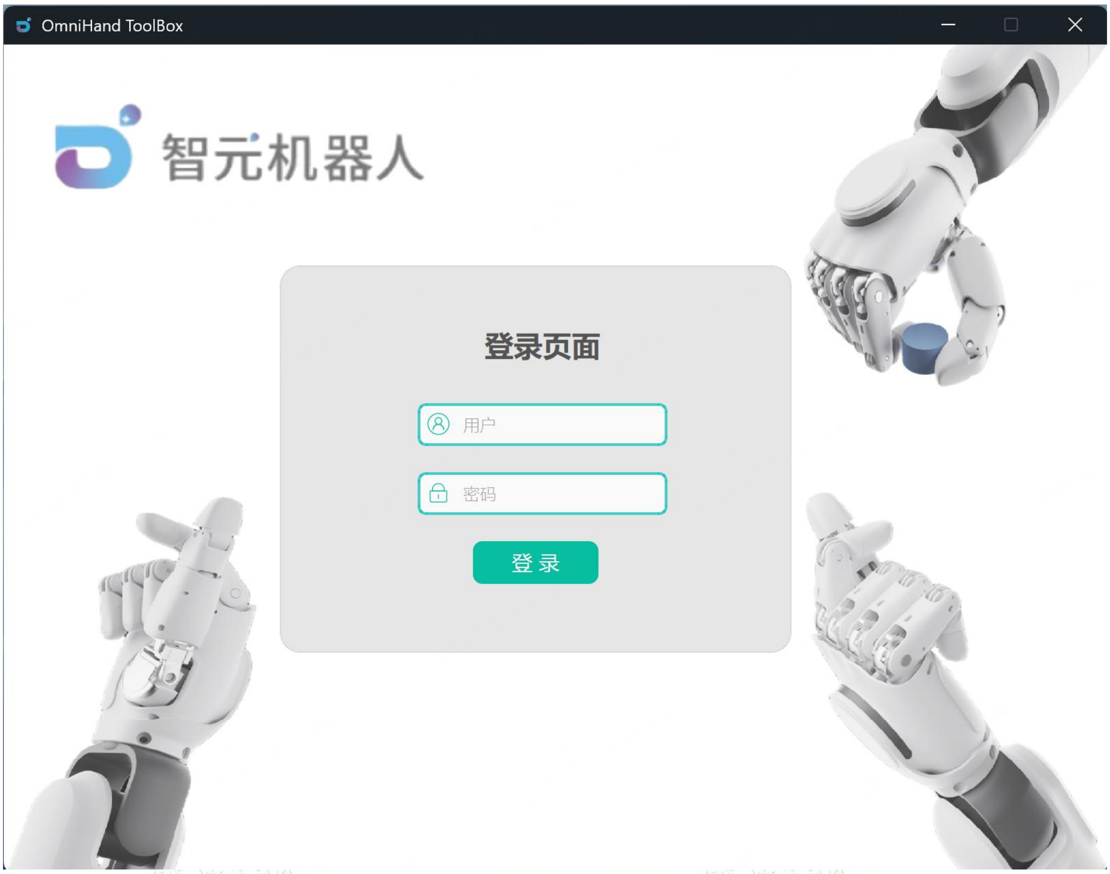
- CPU：Intel Core i5-8 代以上，双核 1.5GHz+
- 内存：8GB+
- 显示：支持 OpenGL 3.3+（集成显卡即可）
- 操作系统：Windows 10/11
- 接口：USB 2.0+
- 软件依赖：VC++ 2019 Redistributable(x86 版本) / https://aka.ms/vc14/vc_redist.x64.exe(x64 版本)，.Net Framework 4.8

5.2 下载安装

灵巧手上位机软件可在官方网站进行下载压缩包，链接见 4.4。

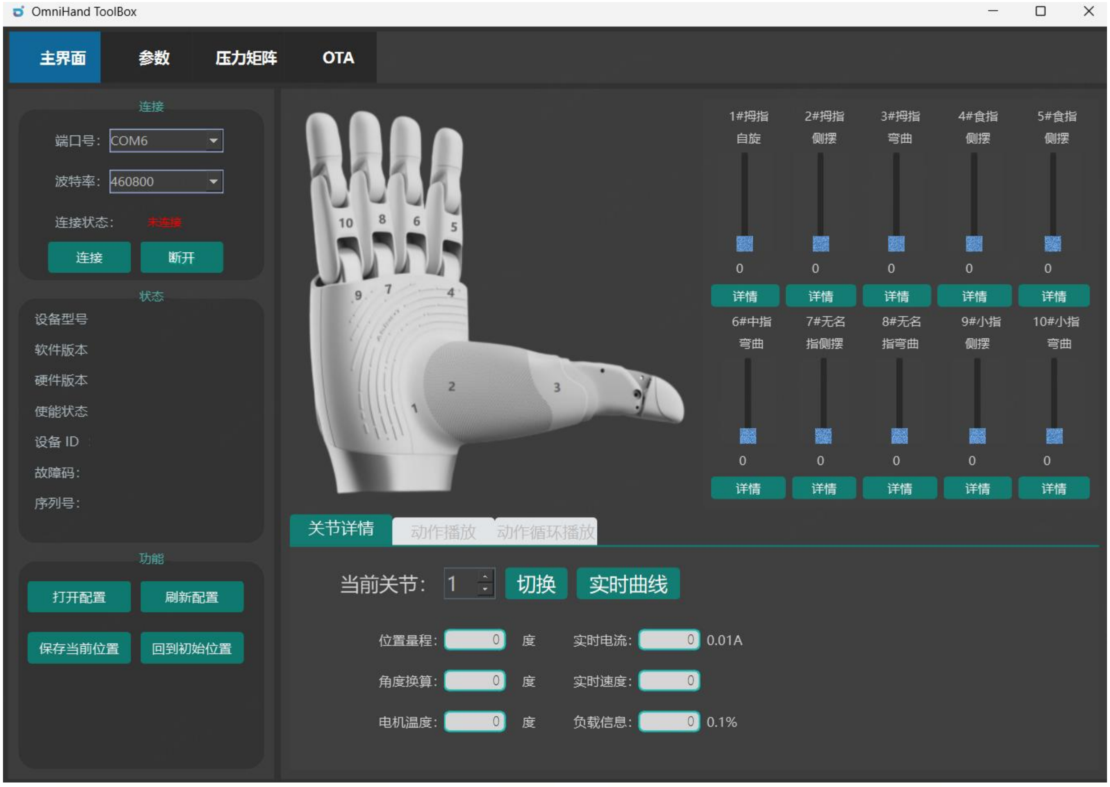
下载完成后，需要进行解压。在打开上位机前，请先安装 VC++redistributable、.NET Framework 软件依赖。安装提示完成后，双击 OmniHand-ToolBox 运行上位机软件。

5.3 登录界面



点击“登录”按钮直接进入用户主界面，无需输入用户和密码。

5.4 上位机主界面

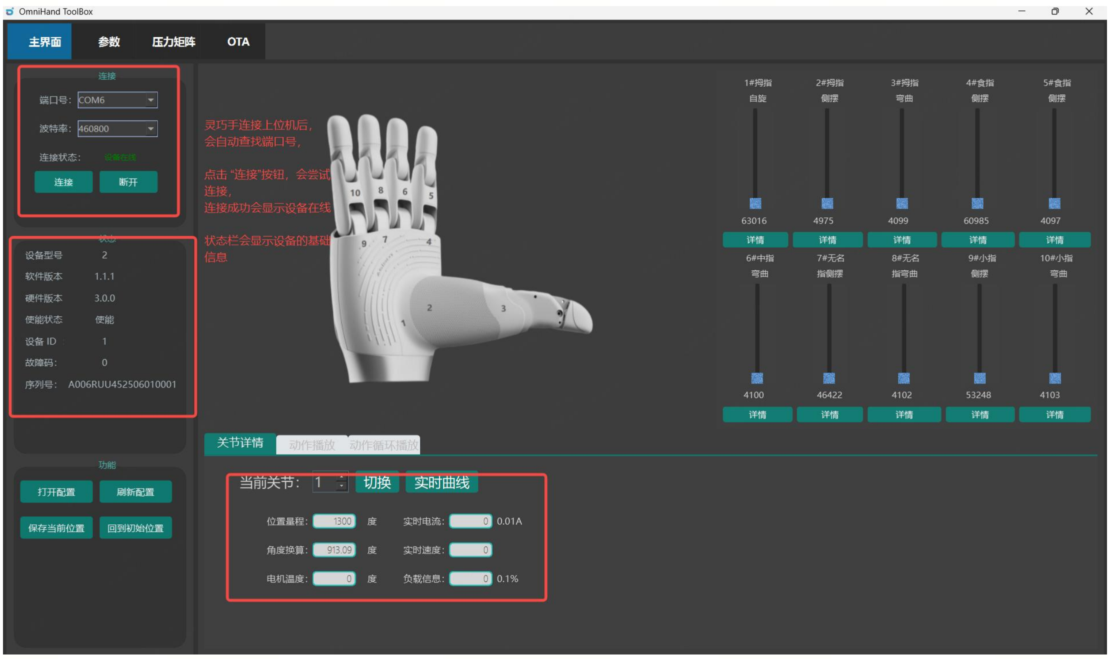


5.5 灵巧手与上位机的连接

OmniHand 灵巧手在接通电源后，使用 USB 线束与主控电脑进行连接，正确接线后，上位机软件会自动搜索端口号，选择正确的端口号并在左边栏的端口号中进行显示。

灵巧手与主控器的波特率固定为 460800，目前不支持进行波特率的调整。

点击 **连接** 按钮，等待握手成功。当电脑成功与灵巧手连接后，连接状态显示为 “**设备在线**”，“状态栏”会显示设备基础信息。



若显示“设备在线”，但是**无法操作灵巧手的运动**，请检查“端口号”中是否有其他端口，请重新选择端口号后进行连接。

如果发生连接失败，请确认软件左上角的 ”端口号“框内是否能检测到端口：

若**已经显示端口号**，但是连接状态**显示未连接**，请将上位机软件关闭重新尝试。

若**未找到端口号**，请从以下几个方面进行检查：

检查电脑是否安装 USB 设备驱动，是否需要更新 USB 设备驱动，检查设备是否正在正常供电，检查是否在使用拓展坞进行连接（有可能会出现问题，请移除），检查线束是否接触不良，USB 数据线需要支持数据传输功能。

5.6 单关节控制

当上位机软件与灵巧手成功建立连接后，在主界面中可对灵巧手进行操作控制。

在主界面中，#1~#10 分别对应 10 个关节，该款灵巧手支持 10 个关节控制，每个关节对应的位置在页面中间的**灵巧手示意图**中对应找到。其中大拇指有三个自由度，食指、无名指和小指各有有两个自由度（侧摆和弯曲），其中中指只有一个自由度（弯曲，侧摆不支持）。

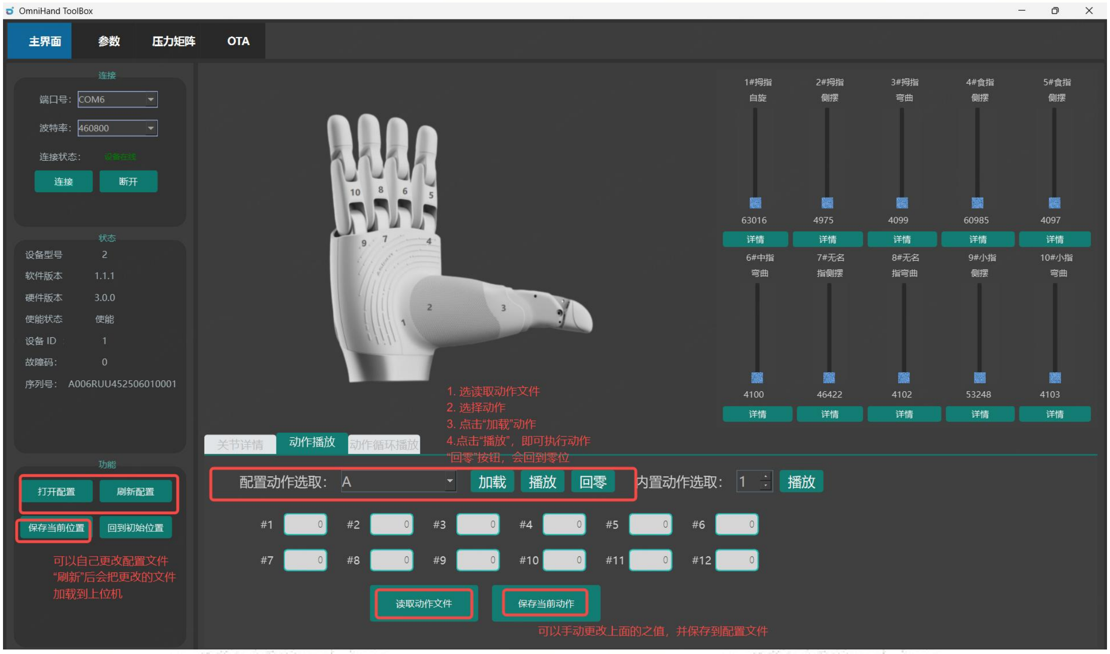
若对手的姿态进行操作，用户可在每个关节的框内，通过拉动滑轨或输入具体数值进行关节控制，数值范围为 0~4096。若使用输入框进行控制，输入数据后，点击 “回车” 键控制生效，同时滑动条下方会显示关节的具体数值。

点击关节的框内滑轨下方“详情” 按钮，可用于切换下方关节详情中显示所对应关节的

内容，也可以通过下方“关节详情”中的“当前关节”切换按钮实现相同的效果。

点击“实时曲线” 按钮可以查看当前关节的 "实时位置“，”实时速度“，”实时电流“的数据曲线图。

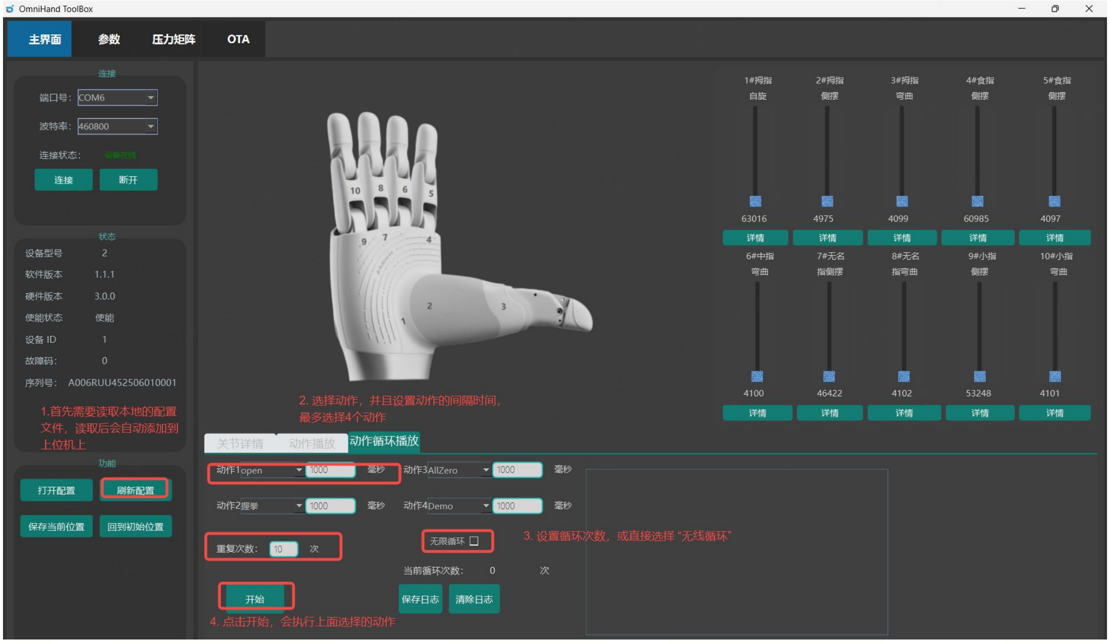
5.7 自定义动作/单动作播放



在本上位机中，可以自定义各种动作，并将当前设置好的动作姿态保存到配置文件中，以便在下次要执行动作的时候，可以一键操作。回零按钮可以让所有关节回到零位。

右下角的功能面板，可以读取/保存的配置文件，并加载到上位机。

5.8 自定义动作循环播放



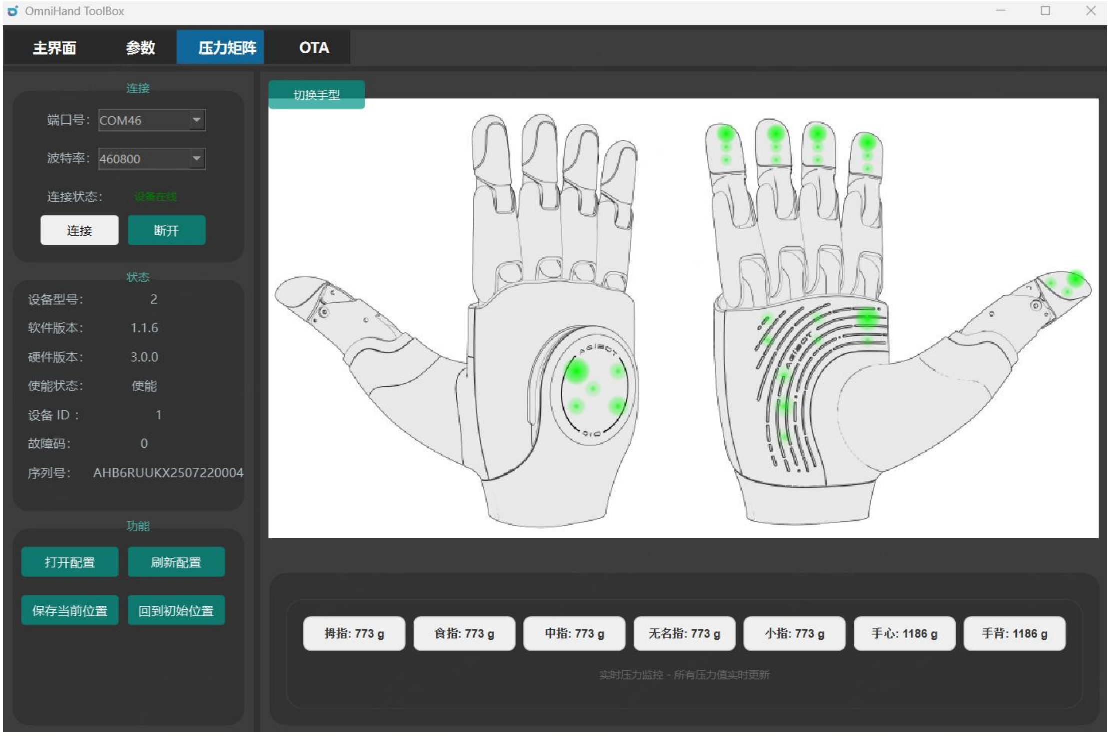
上位机可以一次性选择多个动作，重复的去执行各种动作（目前最多选择 4 个）

5.9 保存和加载自定义手势

可以点击 “打开配置” 查看自定义动作的文件，文件名为 “actions.json”，放在软件同级目录下，也可以点击”保存当前位置” 按钮，会把当前动作保存到文件当中

对于手动添加动作格式，需要添加一个 JSON 对象，每个对象之间用 “,” 分隔，对象内容放在 “{} ”大括号里面。大括号里面需要填写 "name" 字段，为动作名称；”pos_axis_"字段是用户期望关节到达的位置，目前只需要（1-10）即可

5.10 查看关节指尖力



点击 “压力图标”按钮，可以查看五个手指指尖、手心和手背的压力情况，其中在每个手指的指尖的压力测点合成了 3 个点，手心 9 个压力测点，手背 5 个压力测点

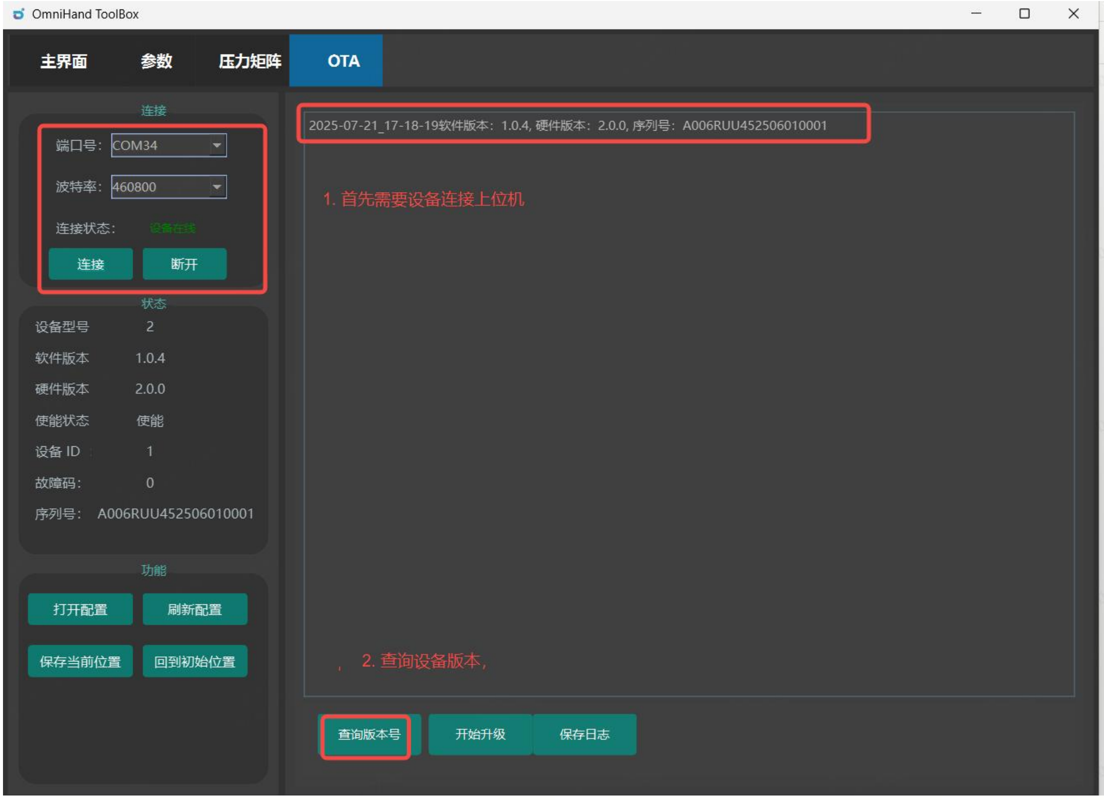
点击 ”切换左右手“ 按钮，可以切换不同的显示效果

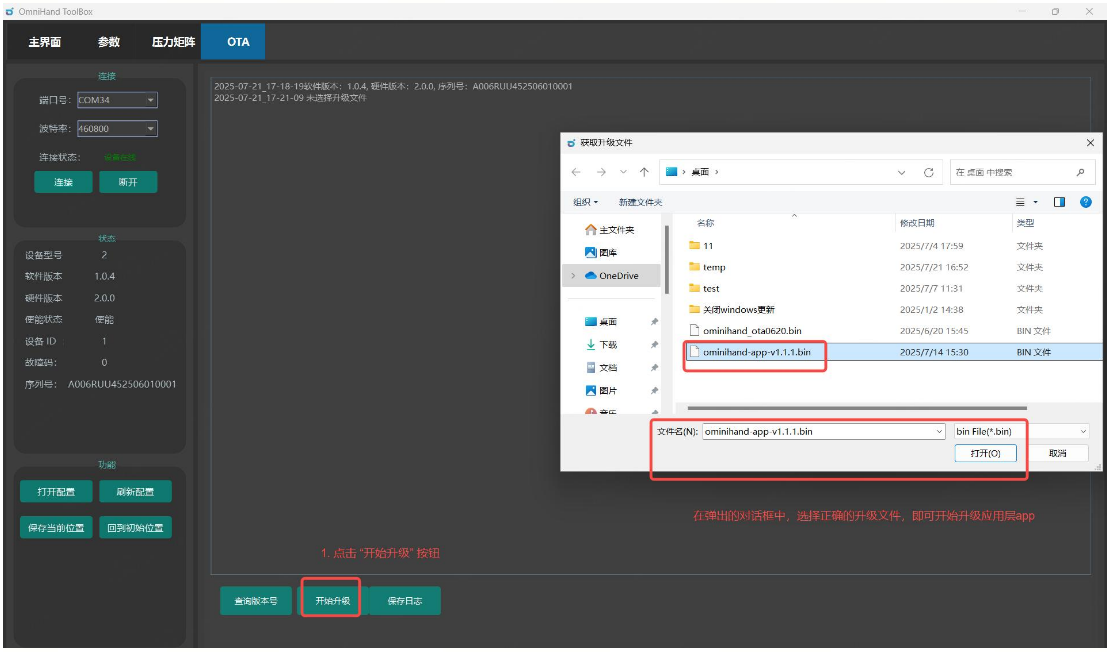
下方显示每个区域的总压力

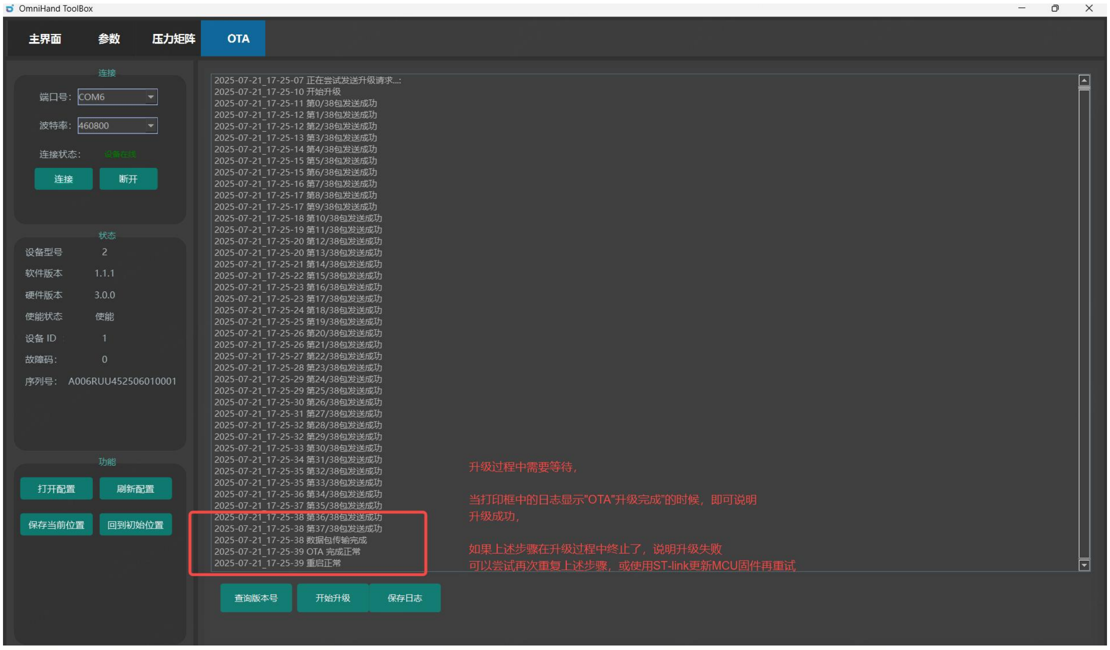
5.11 OTA 升级

在当前 “OTA” 页面，当设备连接成功并显示 “设备在线” 后，点击 “开始升级”，在弹出的选择框中选择要升级的固件 bin 文件，等待升级完成即可，升级过程通过日志打印，通过查看升级过程的日志可以确定升级是否成功，成功后会显示 “OTA 完成正常”，“重启正常” 等字样。

如果显示 “升级失败”，请再次尝试 “开始升级”。若仍然失败，请查看日志中的提示，若出现连接失败，可能需要更新固件。


OmniHand ToolBox
主界面
参数
压力矩阵
OTA
连接
端口号： COM34
波特率： 460800
连接状态： 设备在线
连接
断开
状态
设备型号 2
软件版本 1.0.4
硬件版本 2.0.0
使能状态 使能
设备 ID 1
故障码： 0
序列号： A006RUU452506010001
功能
打开配置
刷新配置
保存当前位置
回到初始位置
2025-07-21_17-18-19软件版本：1.0.4, 硬件版本：2.0.0, 序列号：A006RUU452506010001
1. 首先需要设备连接上位机
2. 查询设备版本，
查询版本本号
开始升级
保存日志


OmniHand ToolBox
主界面
参数
压力矩阵
OTA
连接
端口号： COM34
波特率： 460800
连接状态： 设备在线
连接
断开
状态
设备型号 2
软件版本 1.0.4
硬件版本 2.0.0
使能状态 使能
设备 ID 1
故障码： 0
序列号： A006RUU452506010001
功能
打开配置
刷新配置
保存当前位置
回到初始位置
2025-07-21_17-18-19软件版本：1.0.4, 硬件版本：2.0.0, 序列号：A006RUU452506010001
2025-07-21_17-21-09 未选择升级文件
获取升级文件
桌面
在 桌面 中搜索
组织
新建文件夹
主文件夹
图库
OneDrive
桌面
下载
文档
图片
名称
修改日期
类型
11 2025/7/4 17:59 文件夹
temp 2025/7/21 16:52 文件夹
test 2025/7/7 11:31 文件夹
关闭windows更新 2025/1/2 14:38 文件夹
ominihand_ota0620.bin 2025/6/20 15:45 BIN 文件
ominihand-app-v1.1.1.bin 2025/7/14 15:30 BIN 文件
文件名(N): ominihand-app-v1.1.1.bin
bin File(*.bin)
打开(O)
取消
在弹出的对话框中，选择正确的升级文件，即可开始升级应用层app
1. 点击 "开始升级" 按钮
查询版本号
开始升级
保存日志


OmniHand ToolBox
主界面
参数
压力矩阵
OTA
连接
端口号： COM6
波特率： 460800
连接状态： 设备在线
连接
断开
状态
设备型号 2
软件版本 1.1.1
硬件版本 3.0.0
使能状态 使能
设备 ID 1
故障码： 0
序列号： A006RUU452506010001
功能
打开配置
刷新配置
保存当前位置
回到初始位置
2025-07-21_17-25-07 正在尝试发送升级请求...:
2025-07-21_17-25-10 开始升级
2025-07-21_17-25-11 第0/38包发送成功
2025-07-21_17-25-12 第1/38包发送成功
2025-07-21_17-25-12 第2/38包发送成功
2025-07-21_17-25-13 第3/38包发送成功
2025-07-21_17-25-14 第4/38包发送成功
2025-07-21_17-25-15 第5/38包发送成功
2025-07-21_17-25-15 第6/38包发送成功
2025-07-21_17-25-16 第7/38包发送成功
2025-07-21_17-25-17 第8/38包发送成功
2025-07-21_17-25-17 第9/38包发送成功
2025-07-21_17-25-18 第10/38包发送成功
2025-07-21_17-25-19 第11/38包发送成功
2025-07-21_17-25-20 第12/38包发送成功
2025-07-21_17-25-20 第13/38包发送成功
2025-07-21_17-25-21 第14/38包发送成功
2025-07-21_17-25-22 第15/38包发送成功
2025-07-21_17-25-23 第16/38包发送成功
2025-07-21_17-25-23 第17/38包发送成功
2025-07-21_17-25-24 第18/38包发送成功
2025-07-21_17-25-25 第19/38包发送成功
2025-07-21_17-25-26 第20/38包发送成功
2025-07-21_17-25-26 第21/38包发送成功
2025-07-21_17-25-27 第22/38包发送成功
2025-07-21_17-25-28 第23/38包发送成功
2025-07-21_17-25-29 第24/38包发送成功
2025-07-21_17-25-29 第25/38包发送成功
2025-07-21_17-25-30 第26/38包发送成功
2025-07-21_17-25-31 第27/38包发送成功
2025-07-21_17-25-32 第28/38包发送成功
2025-07-21_17-25-32 第29/38包发送成功
2025-07-21_17-25-33 第30/38包发送成功
2025-07-21_17-25-34 第31/38包发送成功
2025-07-21_17-25-35 第32/38包发送成功
2025-07-21_17-25-35 第33/38包发送成功
2025-07-21_17-25-36 第34/38包发送成功
2025-07-21_17-25-37 第35/38包发送成功
2025-07-21_17-25-38 第36/38包发送成功
2025-07-21_17-25-38 第37/38包发送成功
2025-07-21_17-25-38 数据包传输完成
2025-07-21_17-25-39 OTA 完成正常
2025-07-21_17-25-39 重启正常
升级过程中需要等待，
当打印框中的日志显示"OTA"升级完成"的时候，即可说明
升级成功，
如果上述步骤在升级过程中终止了，说明升级失败
可以尝试再次重复上述步骤，或使用ST-link更新MCU固件再重试
查询版本号
开始升级
保存日志