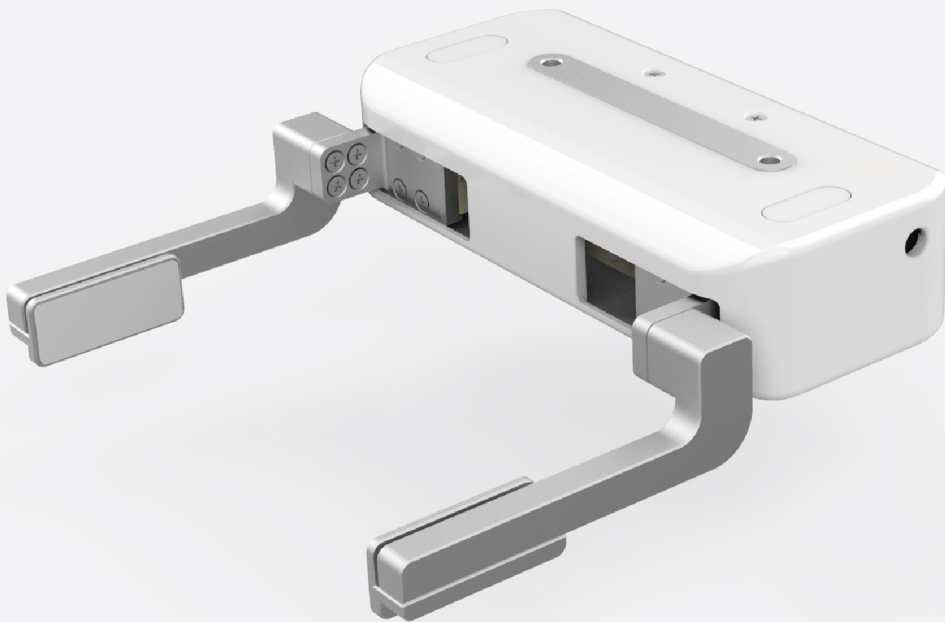




众为创造
UFACTORY

UFACTORY

BIO 机械爪 G2 用户手册



使用前请仔细阅读本手册

V 2.8.0

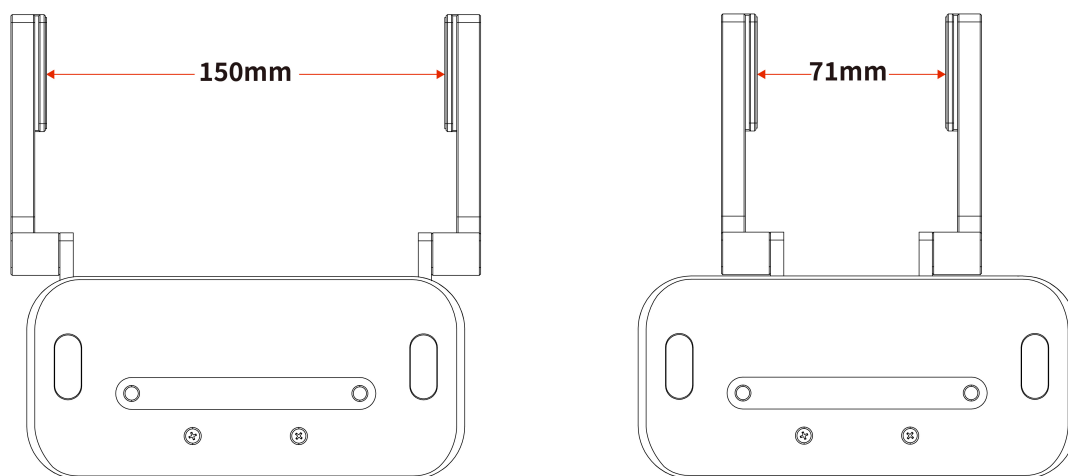
1. 介绍

BIO机械爪G2是专为处理液体板而设计的平行夹爪，具有快速安装以及简单应用的特点。其手指具有出色的抓握能力以及灵活性。

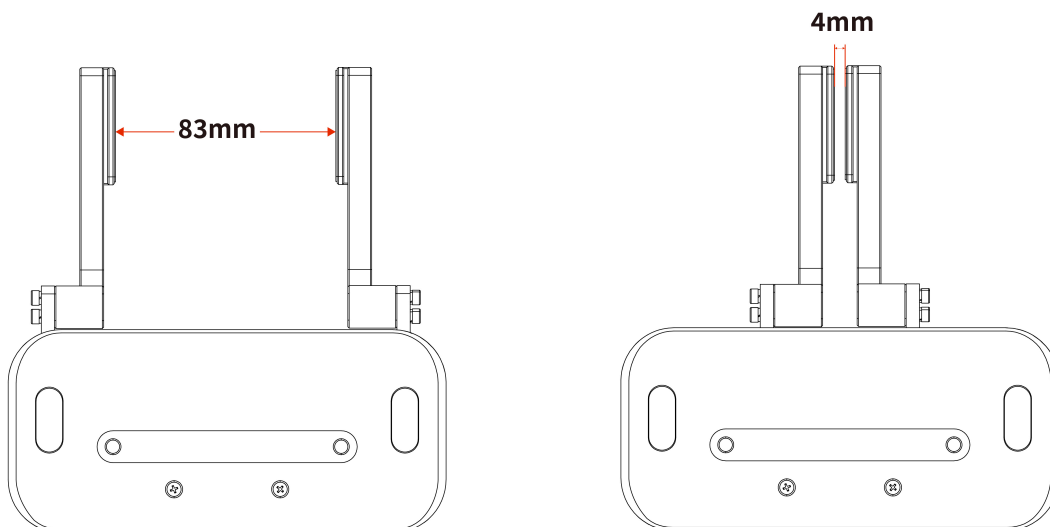
BIO机械爪G2的手指可自由定制，用于处理不同形状的管状和板状的产品。

1.1 工作范围

BIO机械爪G2默认行程：71-150mm



BIO机械爪G2手指反装行程：4-83mm



1.2 设置与控制

BIO机械爪G2通过一条电缆直接供电和控制，该电缆用于24V直流供电和基于RS-485的Modbus RTU通信。

1.3 安全

警告

- 操作员在使用BIO机械爪之前必须已阅读并理解手册中的所有说明。
- 在操作机器人之前，必须正确固定好夹具。
- 请勿安装或操作已损坏或缺少零件的机械爪。
- 切勿为机械爪通交流电。
- 确保所有接线端子稳定连接在机械臂和机械爪两端。
- 请始终使用建议的电气连接。
- 在初始化机械臂程序之前，请确保没有人在机械臂和机械爪路径中。
- 始终不要超过机械爪的有效载荷。
- 根据您的应用情况，相应设置机械爪的速度。
- 接通电源时，手指和衣服应远离机械爪。

- 请勿在人或动物身上使用机械爪。
- 机械爪不适用于对物体或表面施加力。

注意 术语“操作员”是指负责在BIO机械爪G2上进行以下任何操作的任何人：

- 安装
- 控制
- 维护

本文档说明了BIO机械爪G2从安装到运行再到使用的整个生命周期的一般操作。

本文档中的图形和照片是代表性的示例，它们与交付的产品之间可能存在差异。

BIO机械爪G2用于工业机器人，最终应用中使用的机器人、抓取器和任何其他设备必须进行风险评估。机器人集成商的责任是确保遵守所有本地安全措施和规定。根据不同的应用，可能存在需要采取额外保护/安全措施的风险，例如，机械爪操作的工件可能对操作员具有固有的危险。

始终遵守有关自动化安全和通用机器安全的本地和国家法律，法规和指令。本设备只能在其技术数据范围内使用。产品的任何其他使用均被视为不当和意外使用。对于因任何不当使用或不当使用引起的任何损坏，UFACTORY将不承担任何责任。

2. 安装

警告

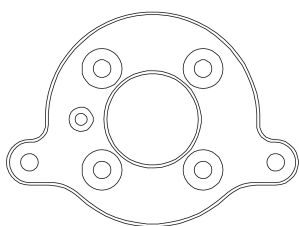
安装之前： 阅读并理解与BIO机械爪有关的安全说明。 根据发货清单和订单验证包裹。 备有需求中列出的所需零件。

安装时： 满足环境条件。 在牢固地固定住机械爪并清除危险区域之前，请勿操作机械爪或打开电源。 机械爪的手指可能会移动并造成伤害或损坏。

2.1 发货清单

BIO机械爪G2套件通常包括以下物品（如下图所示）：

- BIO机械爪G2（1个）
- BIO机械爪G2安装件（1个）
- 十字沉头螺丝M6*8（4个）
- 十字沉头螺丝M6*10（6个）
- 十字沉头螺丝M6*12（4个）



BIO 机械爪 G2安装件



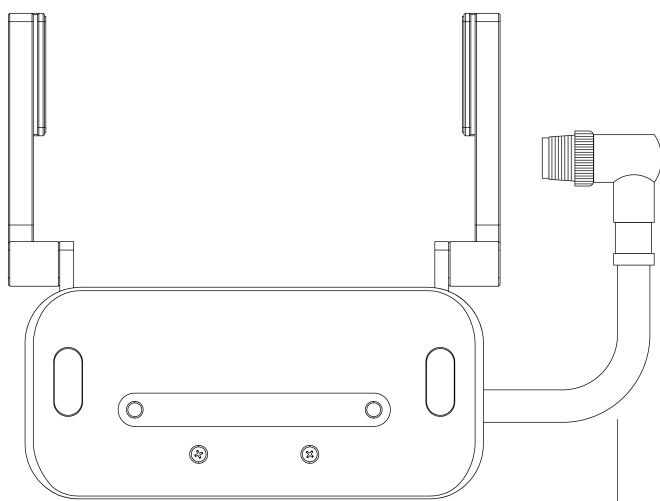
十字沉头螺丝 M6*8(4)



十字沉头螺丝 M6*10(6)



十字沉头螺丝 M6*12(4)



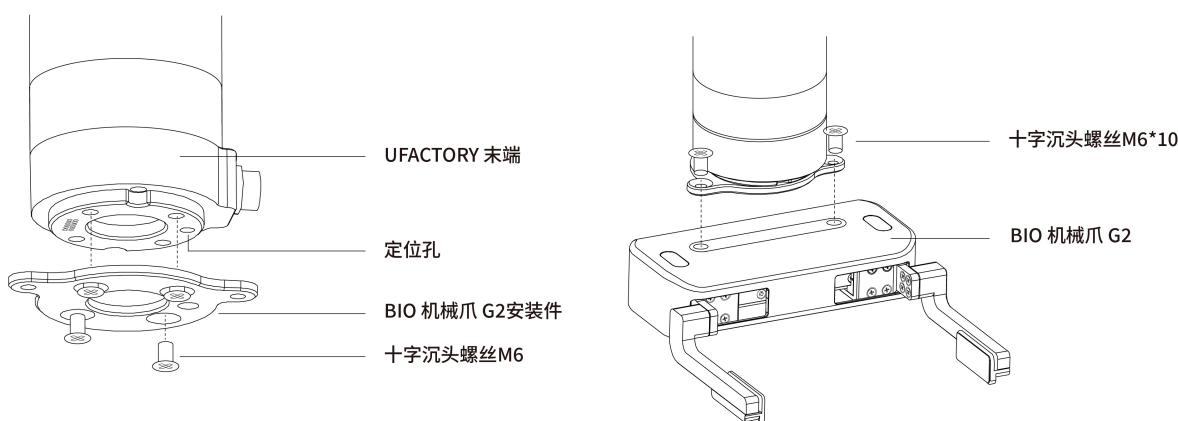
BIO 机械爪 G2

机械爪转接线

2.2 机械安装

BIO机械爪安装步骤（如下图所示）：

1. 将BIO机械爪G2安装板用螺丝固定到机械臂的工具末端，注意将安装板上的定位柱对齐机械臂末端的定位孔。
2. 将BIO机械爪G2用螺丝固定到BIO机械爪G2安装板上。
3. 用机械爪连接线将机械臂与BIO机械爪G2连接。



注意

- 接通机械爪连接线时一定要使机械臂断电，急停开关处于按下状态，机械臂电源指示灯 熄灭，避免热插拔引起机械臂故障；
- 因机械爪连接线长度限制，机械爪接口与末端接口需在相同的方向；
- 机械爪连接线接通机械爪跟机械臂时注意务必对齐两端接口的定位孔，连接线的公针较为纤细，避免在拆装时使公针弯曲。

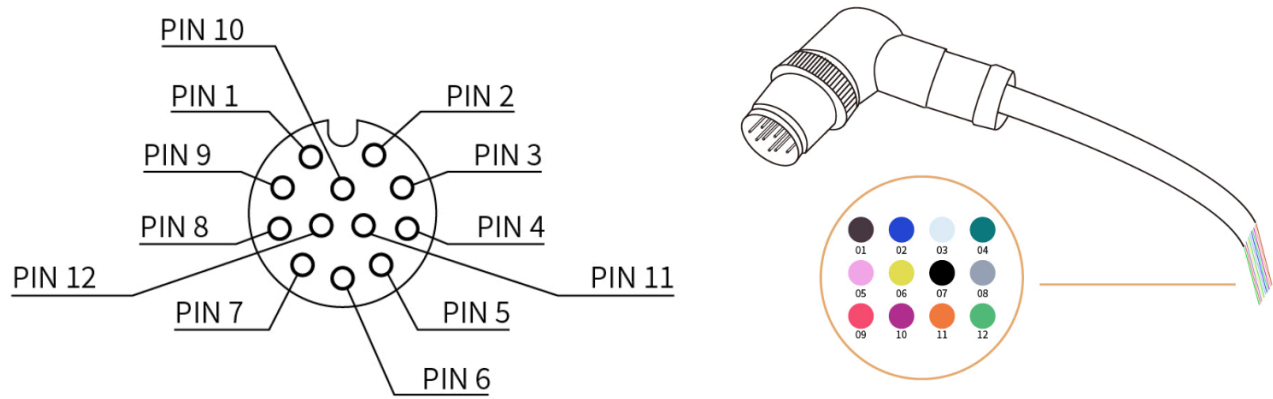
2.3 电气设置

BIO机械爪通过单根设备电缆与xArm机械臂建立电源和通信。设备电缆为机械爪提供24V电源，并实现了与机械臂控制器的串行RS485通信。

请断开机械臂电源后，再用机械爪连接线将机械爪和机械臂连接在一起。

2.3.1 引脚接口

机械爪通过位于其外表面上的12pin接头与机械臂工具端连接。



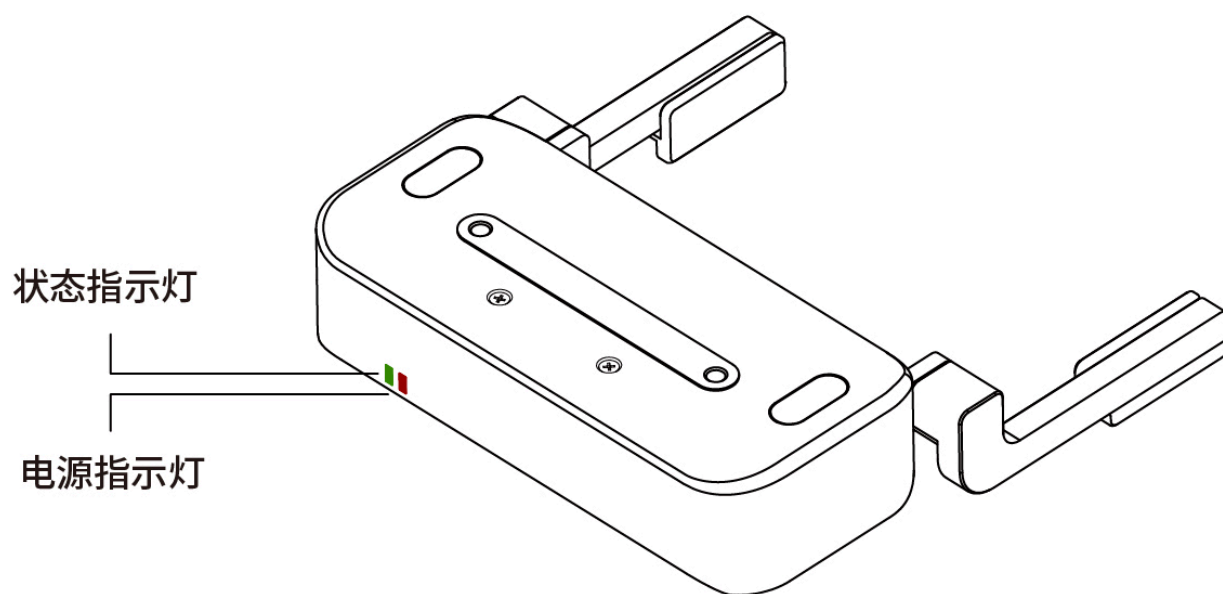
线序	颜色	信号
1	棕	+24V (电源)
2	蓝	+24V (电源)
3	白	0V (GND)
4	绿	0V (GND)
5	粉	用户485-A
6	黄	用户485-B
7	黑	工具输出 0 (TO0)
8	灰	工具输出 1 (TO1)
9	红	工具输入 0 (TI0)
10	紫	工具输入 1 (TI1)
11	橙	模拟输入 0 (AI0)
12	浅绿	模拟输入 1 (AI1)

BIO机械爪G2使用24V(PIN1和PIN2), GND(PIN3和PIN4), 485A(PIN5), 485B(PIN6).

2.4 指示灯

BIO机械爪G2提供2个指示灯，电源指示灯与状态指示灯。

- 当机械爪上电时，电源指示灯常亮红色，状态指示灯闪烁绿色。
- 当机械爪夹住东西时，状态指示灯常亮绿色。
- 当机械爪有错误时，状态指示灯快速闪烁绿色。



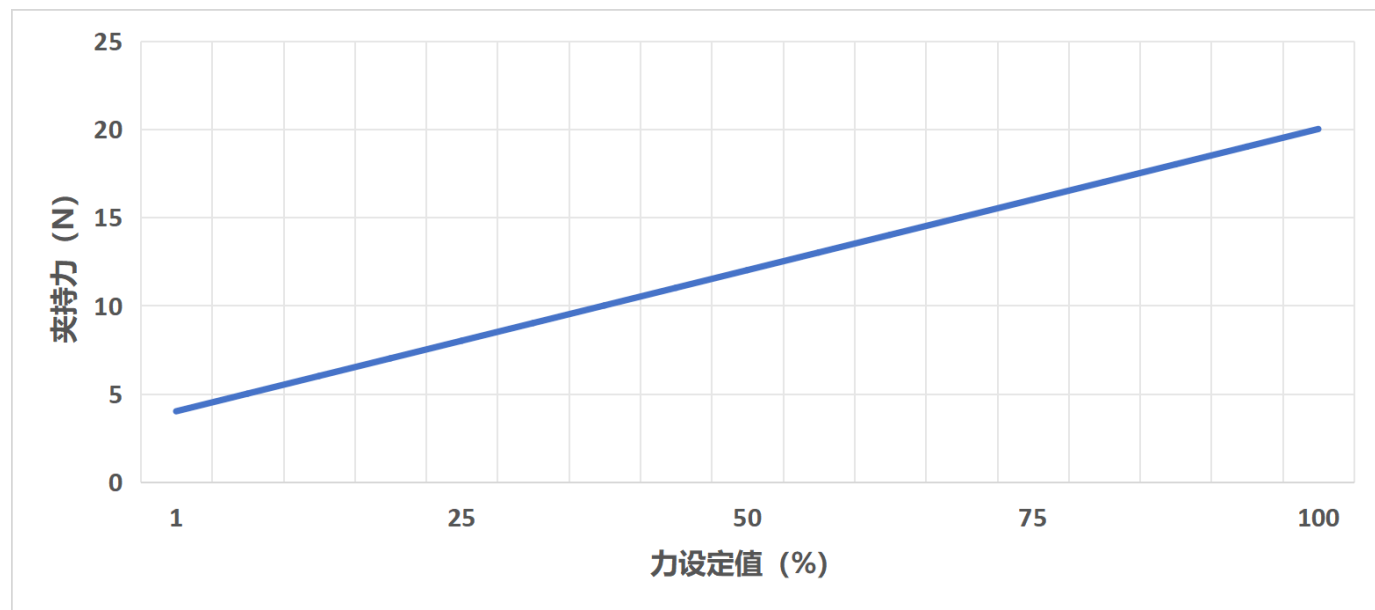
3. 控制方式

BIO机械爪G2提供两种控制模式。切换模式后，需重新使能机械爪。

模式0： 开合模式。（BIO机械爪G2默认的模式）

模式1： 位置模式。支持传入位置，力，速度。

- 位置：71-150（单位：mm）
- 速度：500-4000（单位：rpm，即电机转速）。如需换算为夹爪的开合速度：夹爪速度（mm/s）= 电机速度(rpm) × 5 / 60（如设定3000 rpm，对应夹爪速度250 mm/s）
- 力：1-100（百分比）。力设定值以百分比表示，对应实际力的大小参考下图。



3.1 用UFactory Studio控制

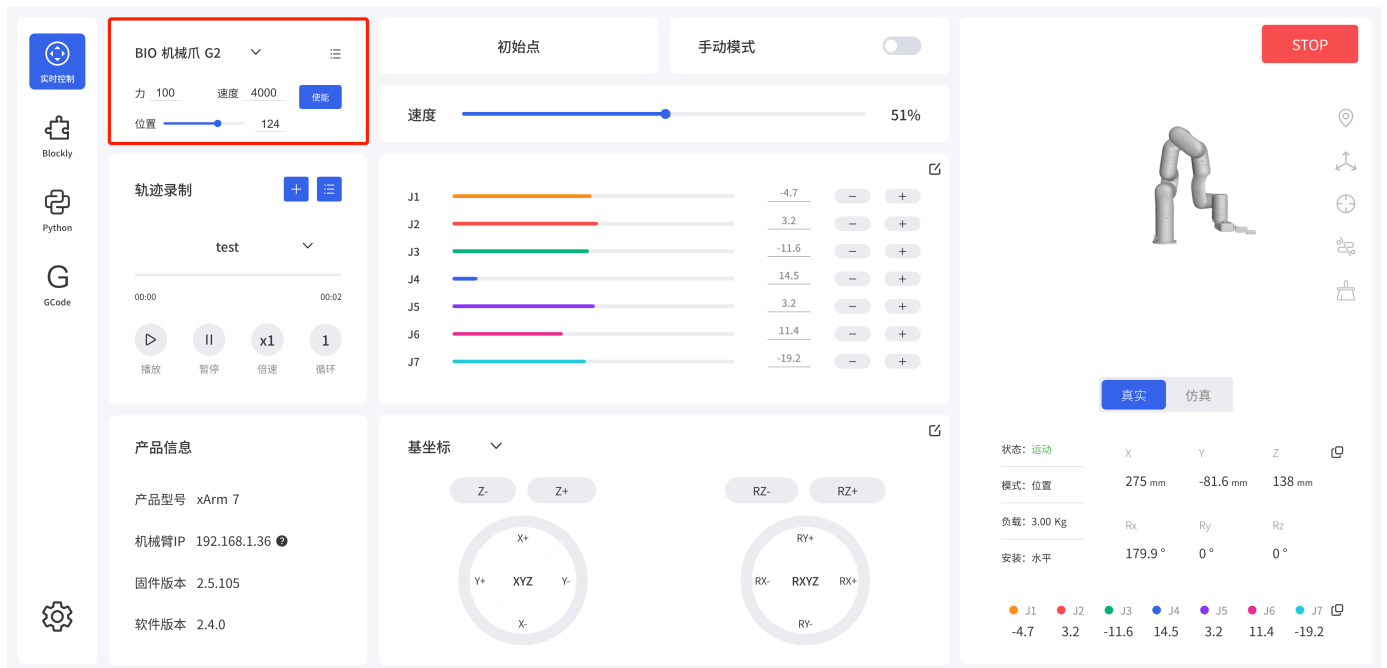
设置BIO机械爪G2

进入设置 - 运动参数 - TCP设置，选择BIO-G2机械爪。



3.1.1 实时控制界面控制

进入实时控制界面，选择BIO机械爪G2，可进行使能，速度，力，位置控制。点击右上角按钮，可关闭位置与力控制（切换模式）。





3.1.2 Blockly控制

Blockly提供3个块控制BIO机械爪G2：

- 初始化BIO机械爪G2
- 设置BIO机械爪G2， 参数：位置，速度，力度，是否等待
- 检测到BIO机械爪G2已经夹住物体， 参数：超时时间



- 通过Blockly编程来控制BIO机械爪G2
[Blockly示例](#)

3.1.3 Modbus RTU控制

进入设置-外接设备-Modbus RTU页面，发送相应的Modbus RTU指令进行控制。

Modbus通讯协议请参考[Modbus RTU通讯协议控制](#)

← 设置

搜索设置

运动参数

外接设备

机械臂IO

控制器IO

力矩传感器

Modbus RTU

Modbus TCP

安全设置

通用设置

我的设备

RS-485 端口 机械臂

Modbus波特率 2000000

超时时间 20 ms

序号	勾选发送	注释	命令	延时ms
1	☐			
2	☐		08 06 11 0A 00 01	
3	☐		08 06 01 00 00 01	
4	☐		08 06 03 03 0b b8	
5	☐		08 06 05 06 00 32	
6	☐		08 10 07 00 00 02 04 00 00 01 2c	
7	☐		08 10 07 00 00 02 04 00 00 00 00	
8	☐		08 03 07 02 00 02	

CRC

循环

发送

清空日志

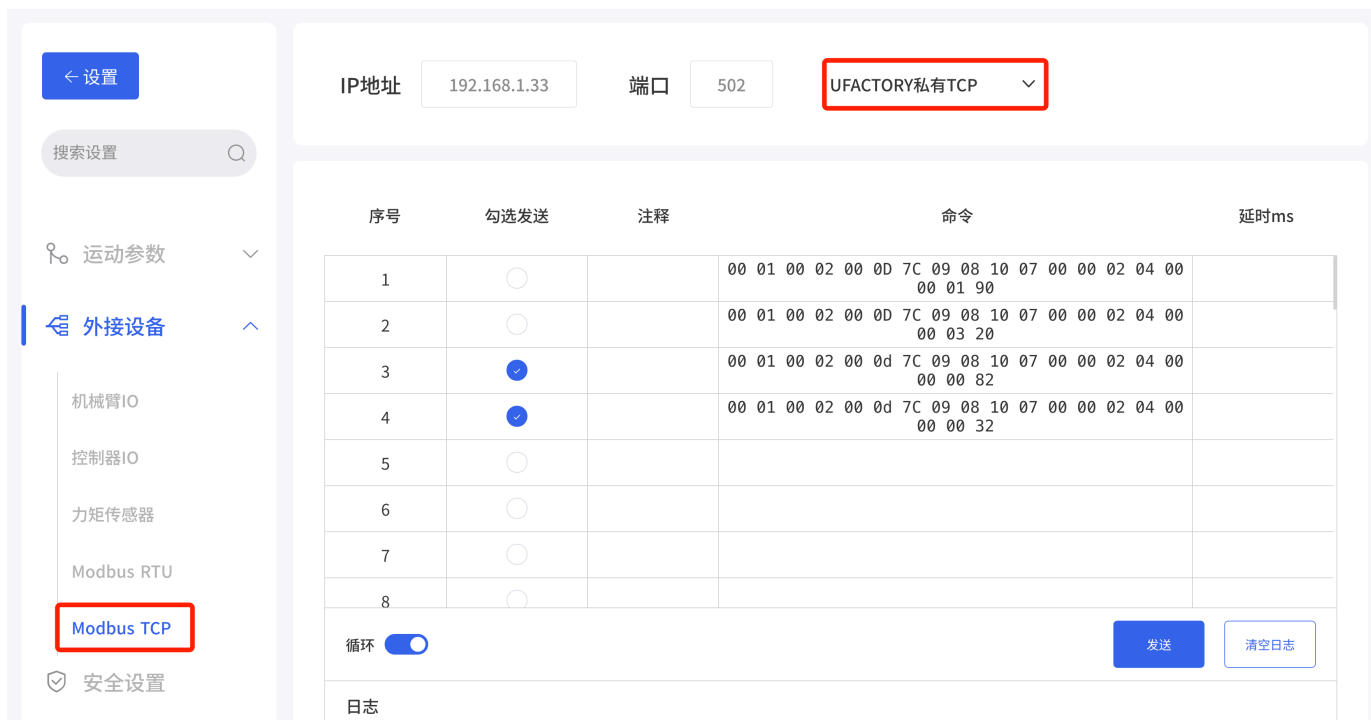
日志

暂无日志

3.1.4 UFACTORY私有协议控制

进入设置-外接设备-Modbus TCP页面，选择'UFACTORY私有TCP'，发送相应的私有TCP指令进行控制。

Modbus通讯协议请参考[UFACTORY私有协议控制](#)



3.2 用Python-SDK控制

3.2.1 模式0：开合模式（默认）

常用接口如下：

`set_bio_gripper_enable`：使能BIO机械爪G2

`set_bio_gripper_speed`：设置BIO机械爪G2速度

`open_bio_gripper`：打开BIO机械爪

`close_bio_gripper`：关闭BIO机械爪

Python示例：[5009-set_bio_gripper.py](#)

3.2.2 模式1：位置模式

常用接口如下：（Python SDK > 1.14.7）

`set_bio_gripper_enable`：使能BIO机械爪G2

`set_bio_gripper_control_mode(mode=1)`：切换为位置模式

`set_bio_gripper_position`：控制BIO机械爪G2的位置，力，速度

Python示例：

```
import os
import sys
import time
sys.path.append(os.path.join(os.path.dirname(__file__), '../..'))

from xarm.wrapper import XArmAPI

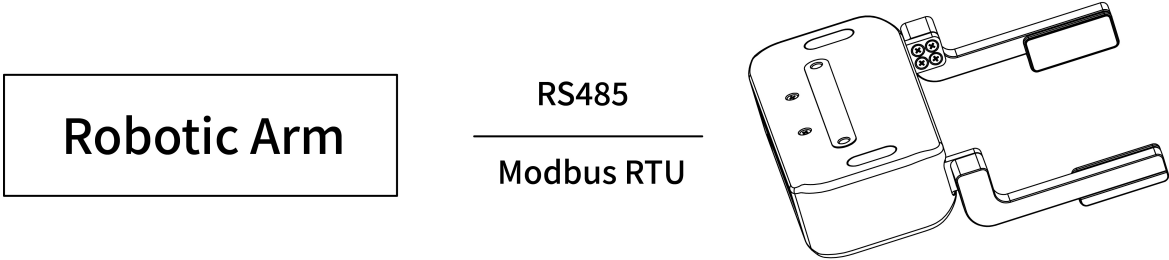
arm = XArmAPI('192.168.1.204')
arm.motion_enable(True)
arm.clean_error()
arm.set_mode(0)
arm.set_state(0)
time.sleep(1)

code = arm.set_bio_gripper_control_mode(mode=1)
print('set_bio_gripper_mode,code={}'.format(code))

code = arm.set_bio_gripper_enable(True)
print('set_bio_gripper_enable,code={}'.format(code))

while True:
    code = arm.set_bio_gripper_position(150, speed=3000, force=50)
    print('set_bio_gripper_position,code={}'.format(code))
    time.sleep(0.2)
    code = arm.set_bio_gripper_position(71, speed=3000, force=100)
    print('set_bio_gripper_position,code={}'.format(code))
    time.sleep(0.2)
```

4.Modbus RTU通讯协议控制



4.1 寄存器地址说明

机械爪支持标准Modbus RTU协议，默认波特率2Mbps，机械爪ID为0x08。目前支持的功能码有：0x03/0x06/0x10。

- 0x03: 读取寄存器
- 0x06: 写单个寄存器
- 0x10: 写多个寄存器

读取：

通讯地址	说明
0x0000	运行状态
0x0001	速度 (r/min)
0x0002	电流百分比
0x0003	电流
0x0004	指令位置
0x0006	电机位置
0x0008	位置误差
0x000F	当前报警代码

写入：

通讯地址	说明	设定范围	单位	出厂设定
0x0100	机械爪使能	0-1;		0
0x010A	控制模式	0-2; 0: 开合模式, 1: 位置模式。		0
0x0303	速度指令	0-4000	r/min	1000
0x0505	夹取检测阈值	30-100	0.01A	50
0x0506	保持电流限幅(力)	10-100	0.01A	50
0x0508	跌落检测阈值	500-2000	r/min	1000
0x0601	通讯波特率	0: 4800 1: 9600 2: 192008: 9216009: 1M10: 1.5M11: 2M	bps	11
0x0700	位置命令高位	0-0xFFFF	-	0
0x0701	位置命令低位	0-0xFFFF	-	0
0x0702	位置反馈高位	0-0xFFFF	-	只读
0x0703	位置反馈低位	0-0xFFFF	-	只读

运动状态（0x0000）

位1:0	00: 停止状态	位3:2	00: 未使能
	01: 运动状态		10: 使能状态
	10: 夹取状态		
	11: 机械爪有错误		

4.2读取保持寄存器

读取保持寄存器			
请求指令格式			

读取保持寄存器			
Modbus RTU 数据	机械爪 ID	1 Byte	0x08
	功能码	1 Byte	0x03
	寄存器起始地址	2 Bytes	Address
	寄存器数量	2 Bytes	N*
	Modbus CRC 16	2 Bytes	CRC*
响应指令格式			
Modbus RTU 数据	机械爪 ID	1 Byte	0x08
	功能码	1 Byte	0x03
	字节数	1 Byte	N*x2
	寄存器值	N*x2 Bytes	Value
	Modbus CRC16	2 Bytes	CRC*

注： N* = 寄存器数量
Address= 寄存器起始地址（见下面列表）
CRC* = 循环冗余校验

寄存器：

	寄存器起始地址	寄存器值	
获取机械爪状态	0x0000	2 Bytes	未使能状态： 0x0000 使能中状态： 0x0004 使能完成状态： 0x0008 停止状态： 0x0008 运动状态： 0x0009 夹取状态： 0x000A 报错状态： 0x000B
获取机械爪错误	0x000F	2 Bytes	有错误： 其他返回值都代表有错误（除0以外） 无错误： 0x0000

4.3 写入寄存器

写入寄存器			
请求指令格式			
Modbus RTU 数据	机械爪 ID	1 Byte	0x08
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	Address
	寄存器数量	2 Bytes	N*
	字节数	1 Byte	N*x2
	寄存器	N*x2 Bytes	Value
	Modbus CRC 16	2 Bytes	CRC*
响应指令格式			
Modbus RTU 数据	机械爪 ID	1 Byte	0x08
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	Address
	寄存器数量	2 Bytes	N*
	Modbus CRC 16	2 Bytes	CRC*

注： N* = 寄存器数量 Address= 寄存器起始地址（见下面列表） CRC* = 循环冗余校验

寄存器：

	寄存器起始地址	寄存器值	
使能/关闭机械爪	0x0100	2 Bytes	使能: 0x0001 停用: 0x0000
设置机械爪位置	0x0700	4 Bytes	打开机械爪： 0x0000 0x0082 闭合机械爪： 0x0000 0x0032
设置机械爪速度	0x0303	2 Bytes	0x0000-0x0BB8 单位： r/min

	寄存器起始地址	寄存器值	
清除机械爪错误	0x000F	2 Bytes	0x0000

4.4 Modbus RTU示例

使用Modbus RTU控制BIO机械爪G2开合，模式1。

1. 设置机械爪模式1。地址：0x010A，最后两位为CRC: 6D AD，断电后仍生效。

```
发：08 06 11 0A 00 01 6D AD
收：08 06 11 0A 00 01 6D AD
```

2. 使能机械爪。地址：0x0100。

```
发：08 06 01 00 00 01 49 6F
收：08 06 01 00 00 01 49 6F
```

3. 打开机械爪，位置150，速度3000，力50。

```
设置速度3000
发：08 06 03 03 0b b8 7E 55
收：08 06 03 03 0b b8 7E 55

设置力50（百分比）
发：08 06 05 06 00 32 E8 4B
收：08 06 05 06 00 32 E8 4B

打开机械爪G2到150，发送的位置需要做转换：（150-70）*3.75=(OCT)100=(HEX)01 2c
发：08 10 07 00 00 02 04 00 00 01 2c FB 4E
收：08 10 07 00 00 02 40 25
```

4. 关闭机械爪，位置71。

发：08 10 07 00 00 02 04 00 00 00 00 FB 03

收：08 10 07 00 00 02 40 25

5. 读取机械爪位置。

机械爪位置71

发：08 03 07 02 00 02 64 26

收：08 03 04 00 00 00 00 63 33

6. 读取报警代码。地址：0x000F

发：08 03 00 0F 00 01 B4 90

收：08 03 02 00 00 64 45

5. 报警与处理

5.1 错误代码

报警代码	报警说明	报警处理
BG01	FOC运行超时	BIO机械爪FOC运行超时，请清除错误后重试。
BG02	过压保护	BIO机械爪过压，请联系技术支持。
BG03	欠压保护	BIO机械爪机械爪欠压，请联系技术支持。
BG04	过温保护	BIO机械爪温度过高，请联系技术支持。
BG05	启动失败	BIO机械爪启动失败，请联系技术支持。
BG06	转速反馈故障	BIO机械爪转速反馈故障，请联系技术支持。
BG07	过流保护	BIO机械爪过流，请联系技术支持。
BG08	MCSDK软件错误	BIO机械爪MCSDK软件错误，请联系技术支持。
BG09	驱动器保护	BIO机械爪驱动器保护，请联系技术支持。
BG11	机械爪过流	BIO机械爪电流过大。请点击‘确认’重新使能机械爪，如反复报错，请联系技术支持。
BG12	抓取的物体滑落	BIO机械爪错误。BIO机械爪抓取的物体滑落，请清除错误后重试。

5.2 报警处理

5.2.1 通过UFACTORY Studio清除错误

- 通过控制器上的紧急停止按钮重新对机械臂上电
- 使能机械臂：点击报错弹窗的引导按钮或者首页的使能机械臂按钮。
- 重新使能机械爪:选择BIO机械爪G2，点击'使能'



5.2.2 通过xArm-Python-SDK清理错误

在用 Python 库设计机械臂运动规划时，如果机械臂出现故障，需要手动清除错误。清除错误后，仍需重新使能机械臂，并将机械臂设置为运动模式，方可使机械臂正常运动。

清除错误步骤：(下面接口具体细节请参考https://github.com/xArm-Developer/xArm-Python-SDK/blob/master/doc/api/xarm_api.md)

1. 通过控制器上的紧急停止按钮重新对机械臂上电
2. 清除错误: `clean_error()`
3. 重新使能机械臂: `motion_enable(enable=True)`
4. 设置运动模式: `set_mode(0)`
5. 设置运动状态: `set_state(0)`
6. 使能BIO机械爪G2: `set_bio_gripper_enable(enable=True)`

python示例:

```
python
import os
import sys
import time

sys.path.append(os.path.join(os.path.dirname(__file__), '../..'))

from xarm.wrapper import XArmAPI

arm = XArmAPI('192.168.1.75')
arm.clean_error() #清除错误
```

```
arm.motion_enable(enable=True) #重新使能机械臂  
arm.set_mode(0) #设置运动模式  
arm.set_state(0) #设置运动状态  
arm.set_bio_gripper_enable(enable=True) #使能bio机械爪
```

6. 技术规格

BIO机械爪G2 (AB1100)	
额定电源电压	24V DC
最大电源电压	28V DC
静态功耗 (最低功耗)	0.72W
峰值电流	1.5A
质量	790g(含转接件)
最大夹持力度	20N
行程	80mm
行程范围	71-150mm (可更换手指)
通信方式	RS-485
通信协议	Modbus RTU
可编程参数	速度(500-4000rpm)、位置(71-150mm)、夹持力(1-100%)
状态	工作状态、电源状态
反馈	抓取检测、滑落检测

7. 售后服务

售后政策： 对于产品的质量保证以及维修和退换货的详情，见官网的售后政策：

<https://www.cn.ufactory.cc/warranty>

售后服务流程：

- 联系技术支持（support@ufactory.cc），确认产品是否需要寄回维修，确定需要寄回的组件。
- 我司根据售后政策，判定产品保修状况，付费或免费维修。
- 维修、测试完成后，我们会将产品寄回给客户，一般情况下，整个维修流程需要1-2周。

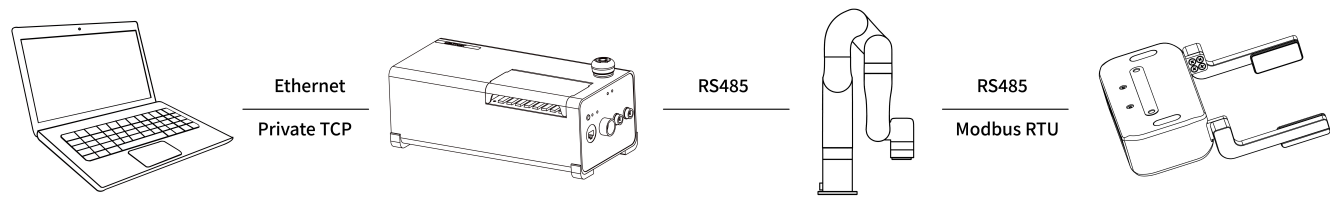
注意

当需要将产品寄回我司进行维修时，需要将产品用包装箱打包好，避免在运输过程中发生不必要的碰撞，导致BIO机械爪G2受损。

8.附录-UFACTORY私有TCP协议控制

本附录关于如何使用UFACTORY私有TCP协议控制BIO机械爪G2

8.1 私有TCP通讯协议



8.1.1 寄存器地址说明

参考[寄存器地址说明](#)

Modbus协议是一项应用层报文传输协议，有ASCII、RTU、TCP三种报文类型。标准Modbus协议物理层接口有RS232、RS422、RS485和以太网接口，采用master/slave方式通信。BIO机械爪G2支持**私有TCP**协议，与标准Modbus TCP类似但不完全相同。

私有TCP通信过程：

- (1) 建立TCP连接
- (2) 准备私有TCP报文
- (3) 使用send命令发送报文
- (4) 在同一连接下等待应答
- (5) 使用recv命令读取报文，完成一次数据交换
- (6) 通信任务结束时，关闭TCP连接

参数：

默认TCP端口：502

协议标识：0x00 0x02 控制(当前只有这一个)

在本章节中，数据解析均为**大端解析**。

8.1.2 读取保持寄存器

读取保持寄存器			
请求指令格式			

读取保持寄存器			
私有TCP 包头	事务标识	2 Bytes	0x00, 0x01
	协议标识	2 Bytes	0x00, 0x02
	长度	2 Bytes	0x00, 0x08
	寄存器	1 Byte	0x7C
	主机 ID（内部使用）	1 Byte	0x09
Modbus RTU 数据	机械爪 ID	1 Byte	0x08
	功能码	1 Byte	0x03
	寄存器起始地址	2 Bytes	Address
	寄存器数量	2 Bytes	N*
响应指令格式			
私有TCP 包头	事务标识	2 Bytes	0x00, 0x01
	协议	2 Bytes	0x00, 0x02
	长度	2 Bytes	*6+N*x2
	寄存器	1 Byte	0x7C
	状态	1 Byte	0x00
	主机 ID（内部使用）	1 Byte	0x09
Modbus RTU 数据	机械爪 ID	1 Byte	0x08
	功能码	1 Byte	0x03
	字节数	1 Byte	N*x2
	寄存器值	N*x2 Bytes	Value

注：

N* = 寄存器数量

Address = 寄存器起始地址（见下面列表）

寄存器：

	寄存器起始地址	寄存器值	
获取机械爪状态	0x0000	2 Bytes	未使能状态: 0x0000 使能中状态: 0x0004 使能完成状态: 0x0008 停止状态: 0x0008 运动状态: 0x0009 夹取状态: 0x000A 报错状态: 0x000B
获取机械爪错误	0x000F	2 Bytes	有错误: 其他返回值都代表有错误（除0以外） 无错误: 0x0000

8.1.3 写入寄存器

写入寄存器			
请求指令格式			
私有TCP 包头	事务标识	2 Bytes	0x00, 0x01
	协议	2 Bytes	0x00, 0x02
	长度	2 Bytes	9+N*x2
	寄存器	1 Byte	0x7C
	主机ID（内部使用）	1 Byte	0x09
Modbus RTU 数据	机械爪ID	1 Byte	0x08
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	Address
	寄存器数量	2 Bytes	N*
	字节数	1 Byte	N*x2
	寄存器	N*x2 Bytes	Value
响应指令格式			
私有TCP 包头	事务标识	2 Bytes	0x00, 0x01
	协议	2 Bytes	0x00, 0x02
	长度	2 Bytes	0x00, 0x09

写入寄存器			
	寄存器	1 Byte	0x7C
	状态	1 Byte	0x00
	主机 ID	1 Byte	0x09
Modbus RTU 数据	机械爪ID（内部使用）	1 Byte	0x08
	功能码	1 Byte	0x10
	寄存器起始地址	2 Bytes	Address
	寄存器数量	2 Bytes	N*

注： N* = 寄存器数量
Address= 寄存器起始地址（见下面列表）

寄存器：

	寄存器起始地址	寄存器值	
使能/关闭机械爪	0x0100	2 Bytes	使能: 0x0001 停用: 0x0000
设置机械爪位置	0x0700	4 Bytes	打开机械爪: 0x0000 0x0082 闭合机械爪: 0x0000 0x0032
设置机械爪速度	0x0303	2 Bytes	0x0000-0x0BB8
清除机械爪错误	0x000F	2 Bytes	0x0000

8.1.4 私有TCP示例

使用私有TCP控制BIO机械爪G2开合，模式0。

- 1. 设置机械爪模式0。地址：0x010A，断电后仍生效。

发： 00 01 00 02 00 08 7C 09 08 06 11 0A 00 00

收: 00 01 00 02 00 09 7C 50 09 08 06 11 0A 00 00

2. 使能机械爪。地址: 0x0100

发: 00 01 00 02 00 0B 7C 09 08 10 01 00 00 01 02 00 01

收: 00 01 00 02 00 09 7C 50 09 08 10 01 00 00 01

3. 打开机械爪。

模式0没有位置控制, 发送位置 >90 , 则打开机械爪。 (OCT)130=(HEX)0082

发: 00 01 00 02 00 0d 7C 09 08 10 07 00 00 02 04 00 00 00 82

收: 00 01 00 02 00 09 7C 50 09 08 10 07 00 00 02

4. 关闭机械爪。

模式0没有位置控制, 发送位置 ≤ 90 , 则关闭机械爪。 (OCT)50=(HEX)0032

发: 00 01 00 02 00 0d 7C 09 08 10 07 00 00 02 04 00 00 00 32

收: 00 01 00 02 00 09 7C 50 09 08 10 07 00 00 02