

# uArm Swift Pro

开发者手册  
V1.0.6

# 目录

使用须知.....	3
简介 .....	3
1.链接 .....	3
2.硬件结构 .....	4
3.坐标.....	5
4.安装.....	6
5. 按钮与指示灯 .....	8
6.扩展简介 .....	10
规格参数.....	12
应用程序信息 .....	14
1.通过 USB 线发送指令.....	14
2.通过蓝牙发送指令 .....	17
3. 第二 UART .....	20
4. ARDUINO .....	21
5.ROS & PYTHON .....	23
6.OPENMV 示例 .....	24
7.从错误的代码中恢复 .....	24
通讯协议.....	24
发布记录.....	34

# 使用须知

1. 机械臂工作时，请勿将手放在机械臂的连杆之间;
2. 请使用官方标配的电源适配器;
3. 使用前请确认桌面有足够空间，避免机械臂运行过程中碰到障碍物

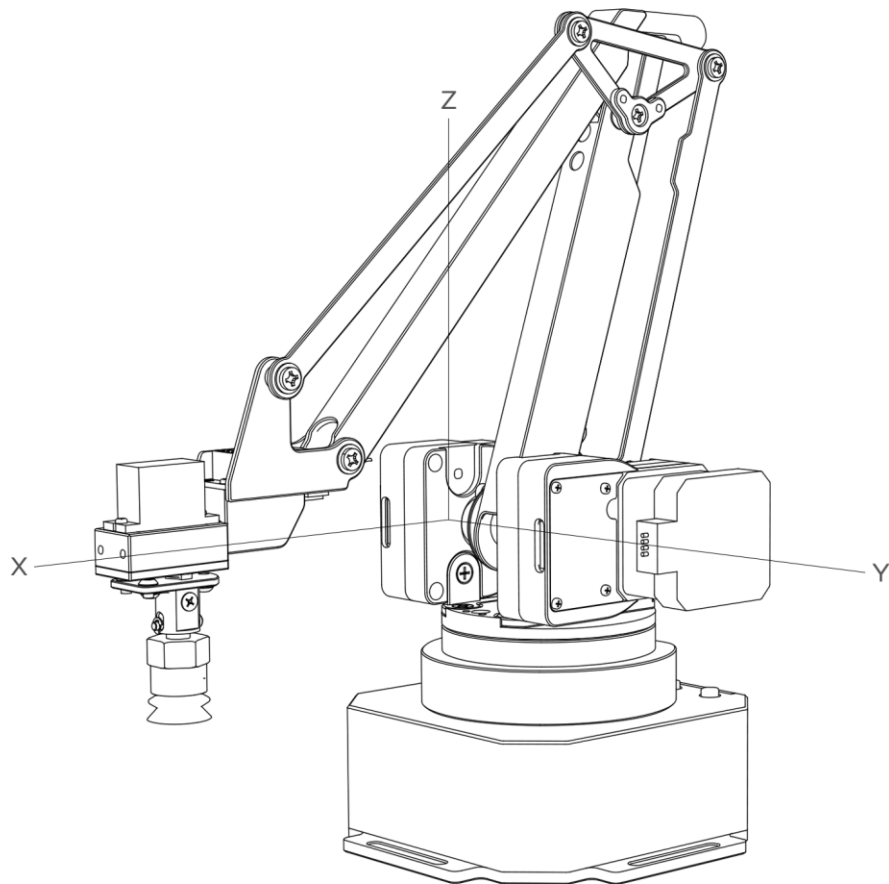
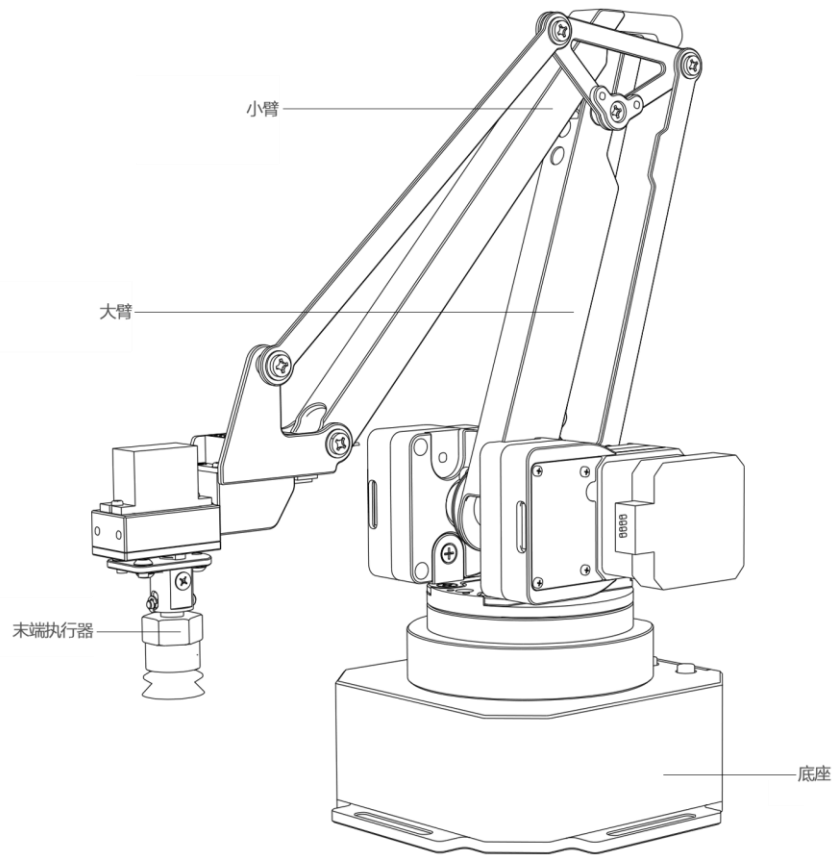
## 简介

开发前建议您阅读以下信息

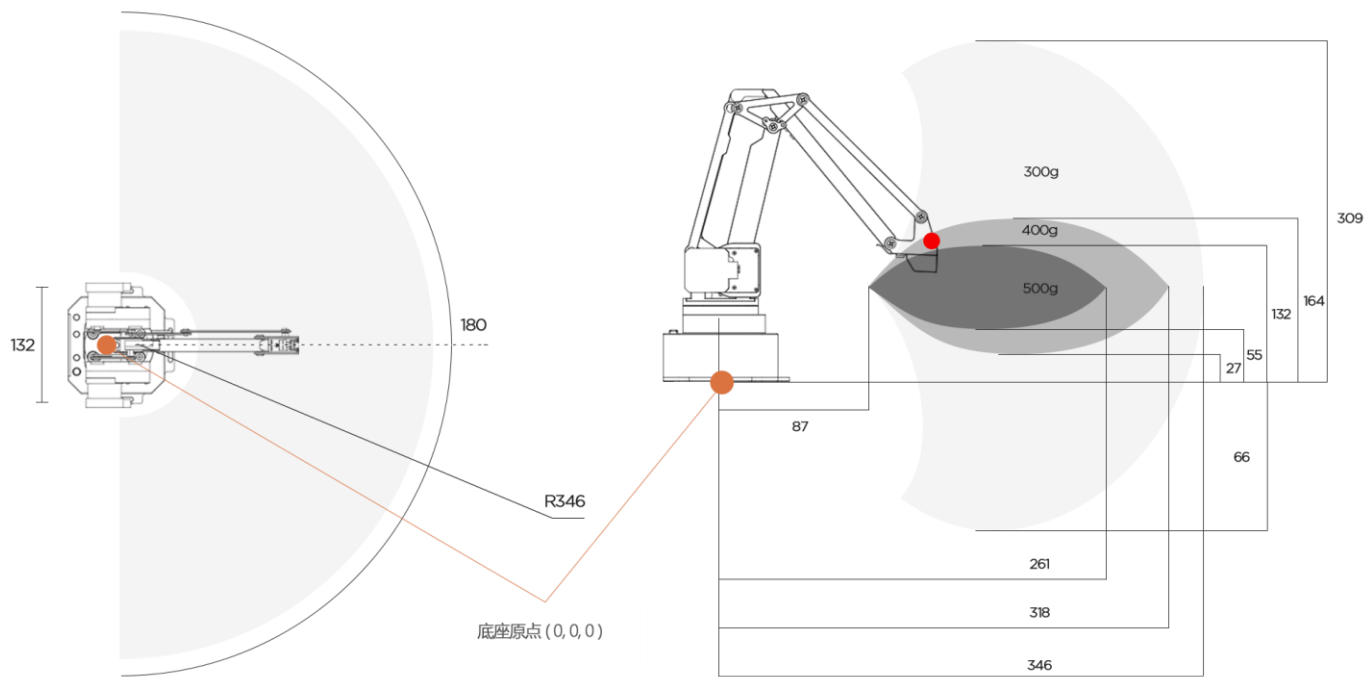
### 1.链接

- uArm Swift Pro 固件代码:  
<https://github.com/uArm-Developer/SwiftProForArduino>
- uArm Swift Pro ROS 代码  
<https://github.com/uArm-Developer/SwiftproForROS>
- Swift Pro Python 库  
<https://github.com/uArm-Developer/pyuf>
- OpenMV 示例  
<https://github.com/uArm-Developer/OpenMV-Examples>
- 即将推出(Arduino, C++, Raspberry Pi)

## 2.硬件结构

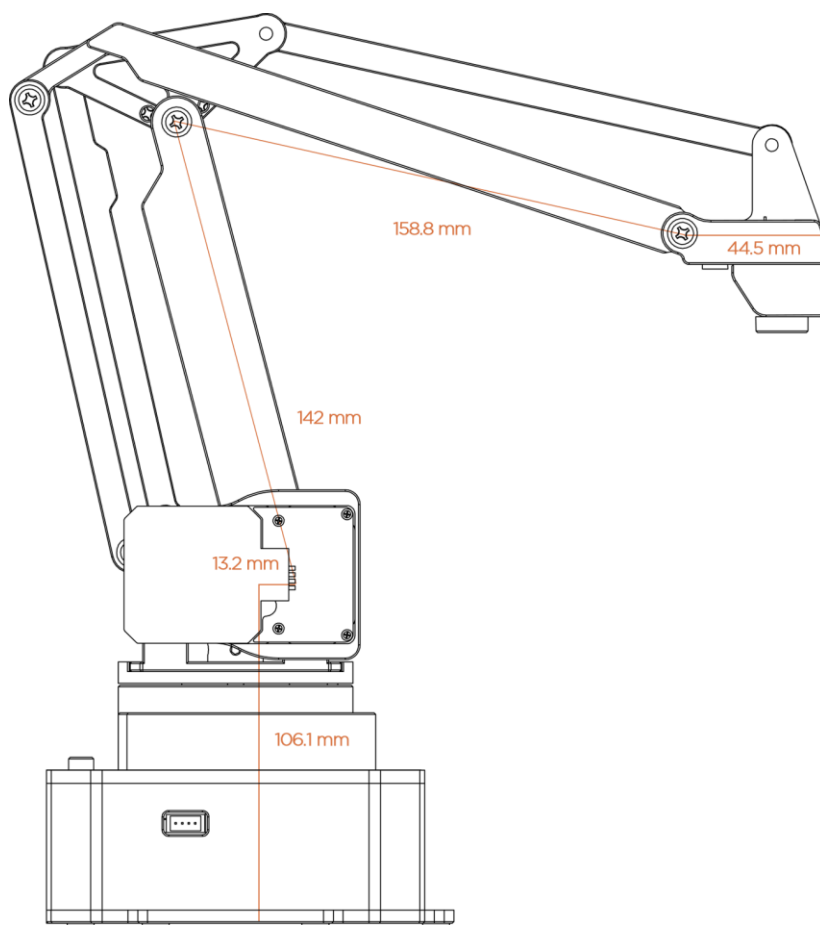


### 3.坐标



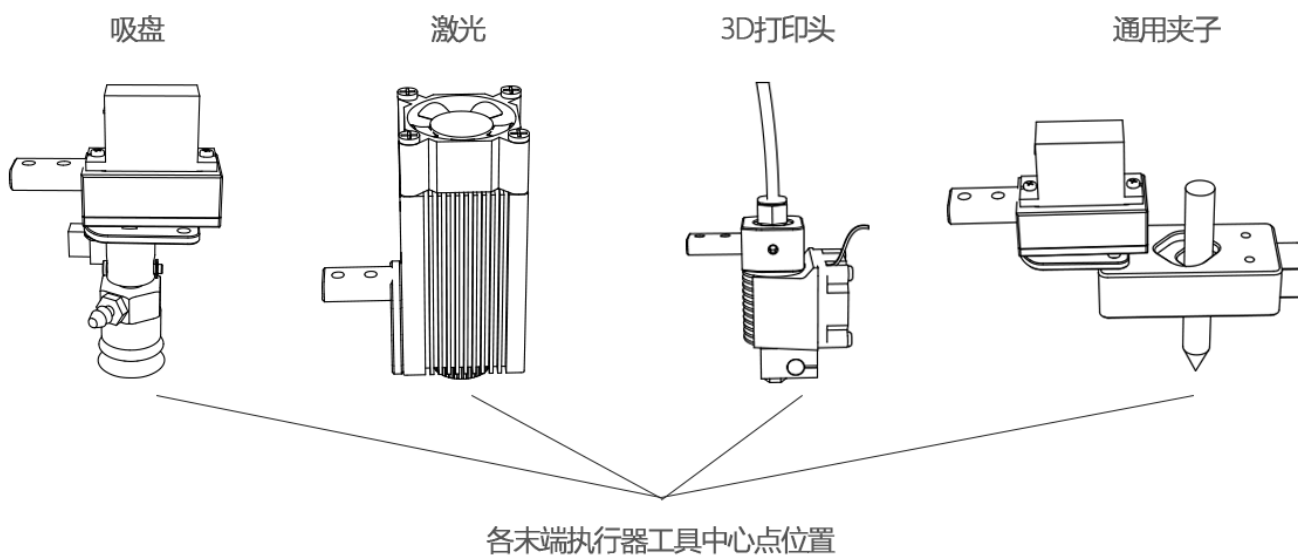
(右侧图片显示了 uArm 的动态负载)

测试环境: G2202 F1000;红点 是前端工具中心点)



底座和连杆尺寸

机械臂的坐标原点位于机械臂底座中心，末端执行器的原点因末端执行器不同而不同，不同使用场景下各末端执行器的指令也不同。



当前我们有 4 种模式:

M2400 S0 : 常规模式 (末端执行器: 吸盘)

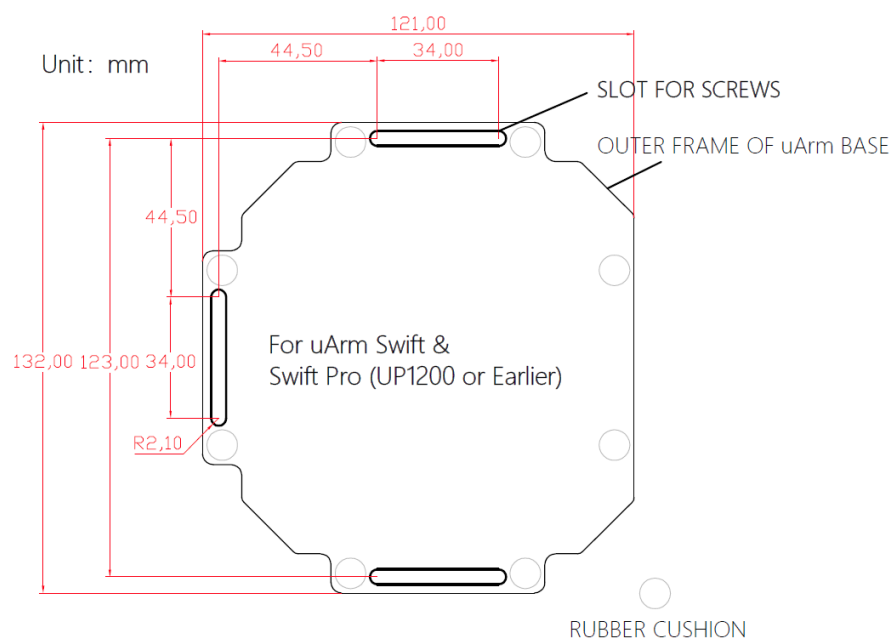
M2400 S1 : 激光雕刻模式 (末端执行器: 激光头)

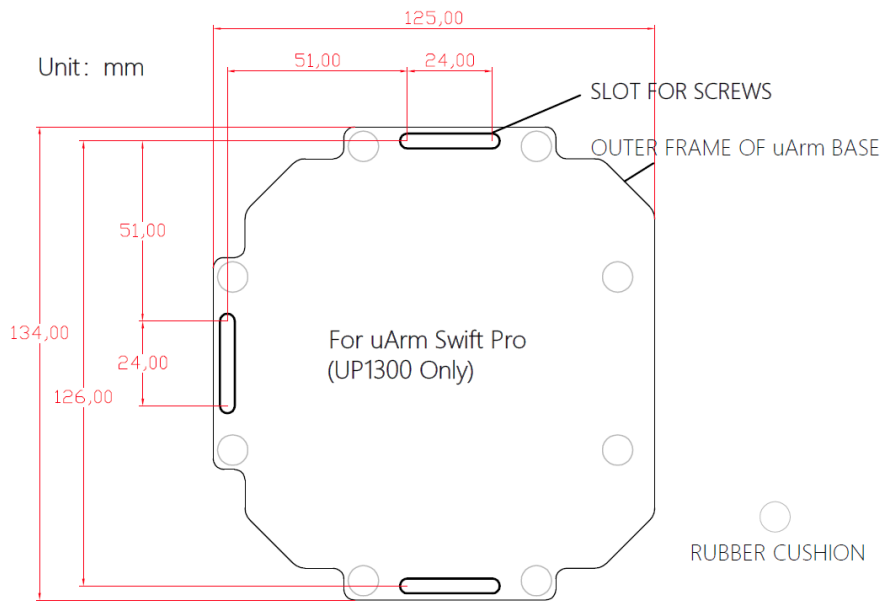
M2400 S2 : 3D 打印模式 (末端执行器: 3D 打印头)

M2400 S3 : 通用笔夹模式 (末端执行器: 通用笔夹)

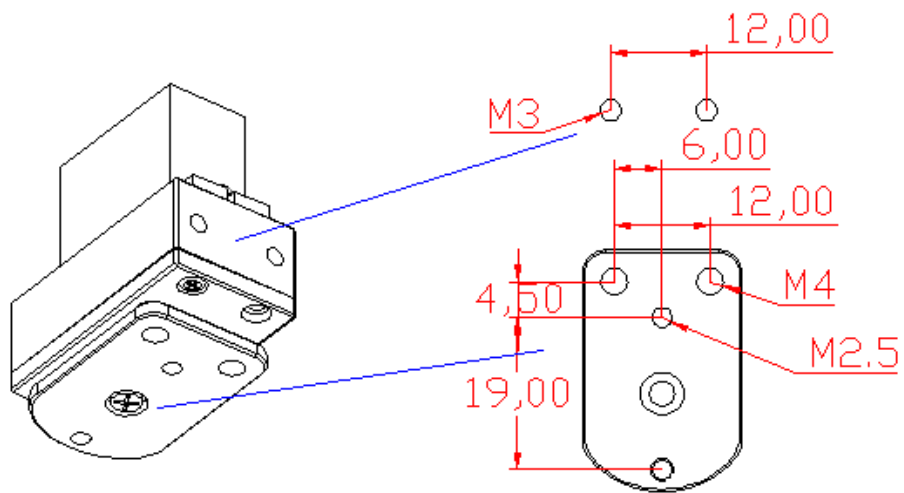
电动夹子由于可以水平旋转所以没有专门设定模式

## 4.安装

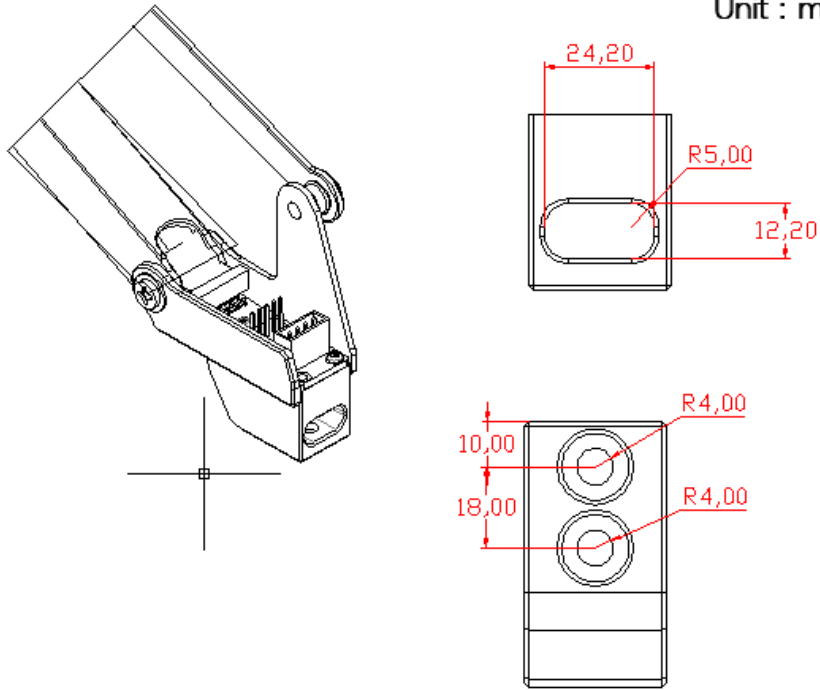




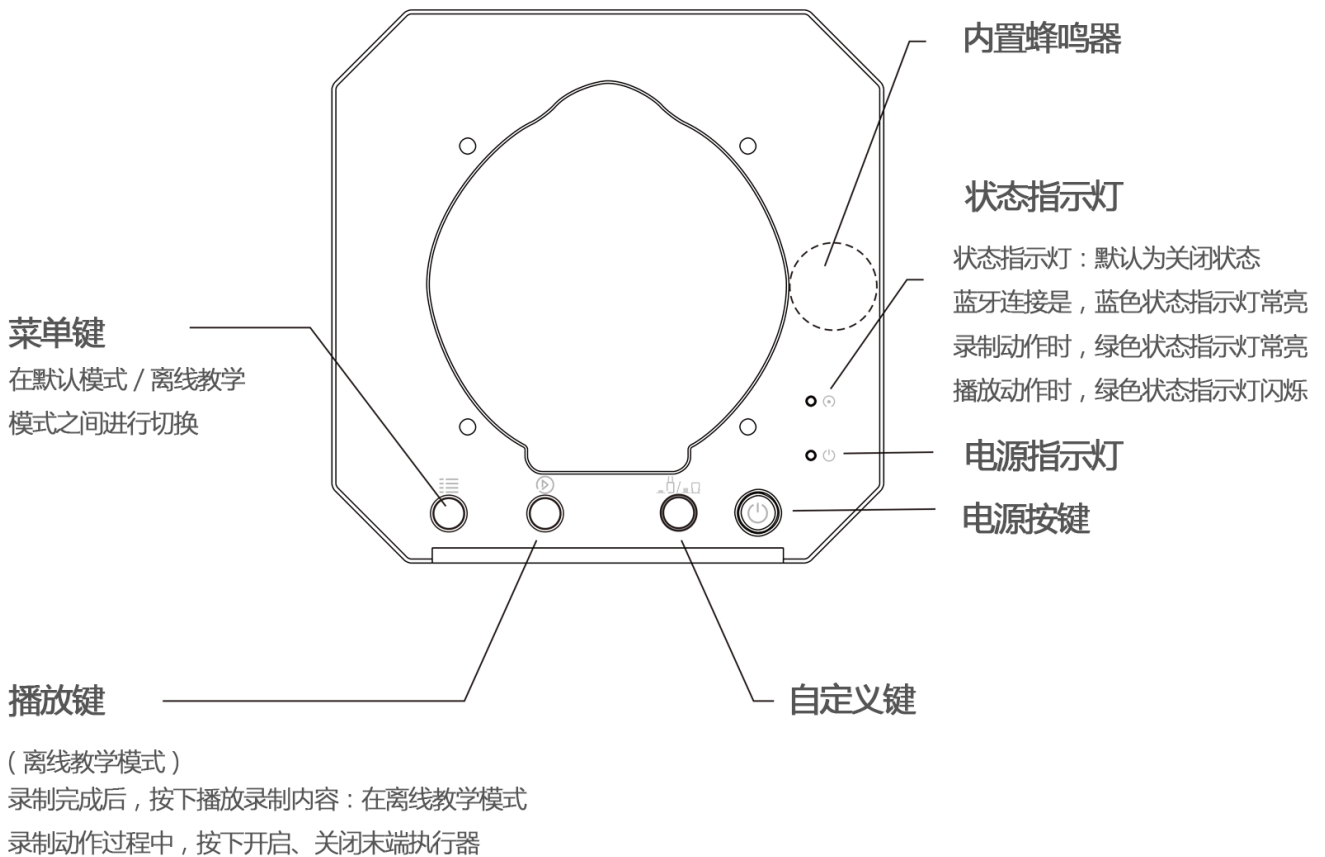
Unit : mm



Unit : mm

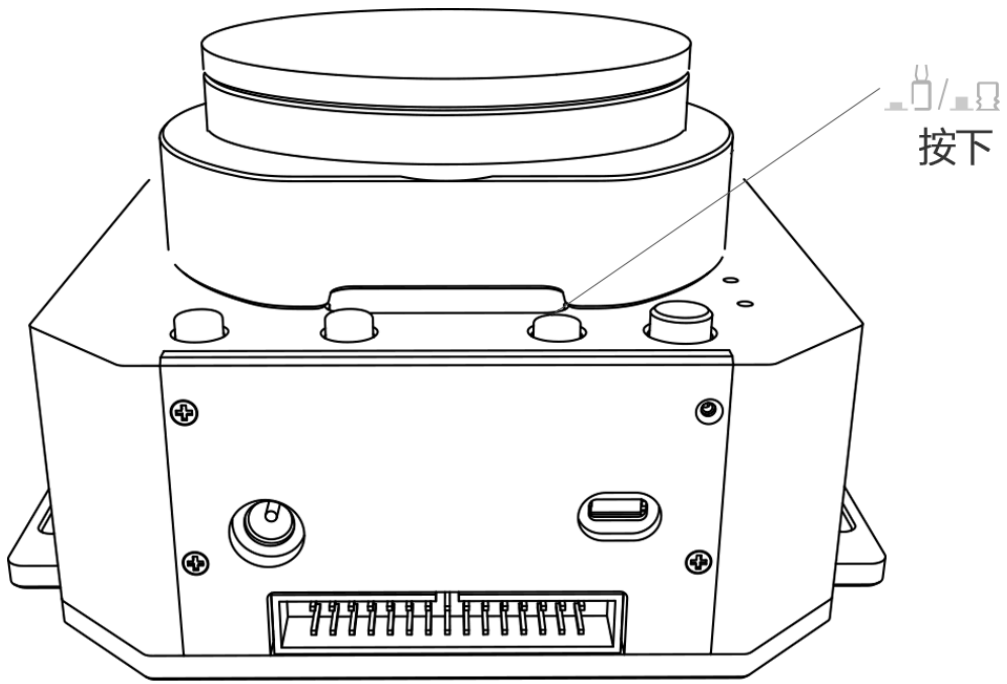


## 5. 按钮与指示灯

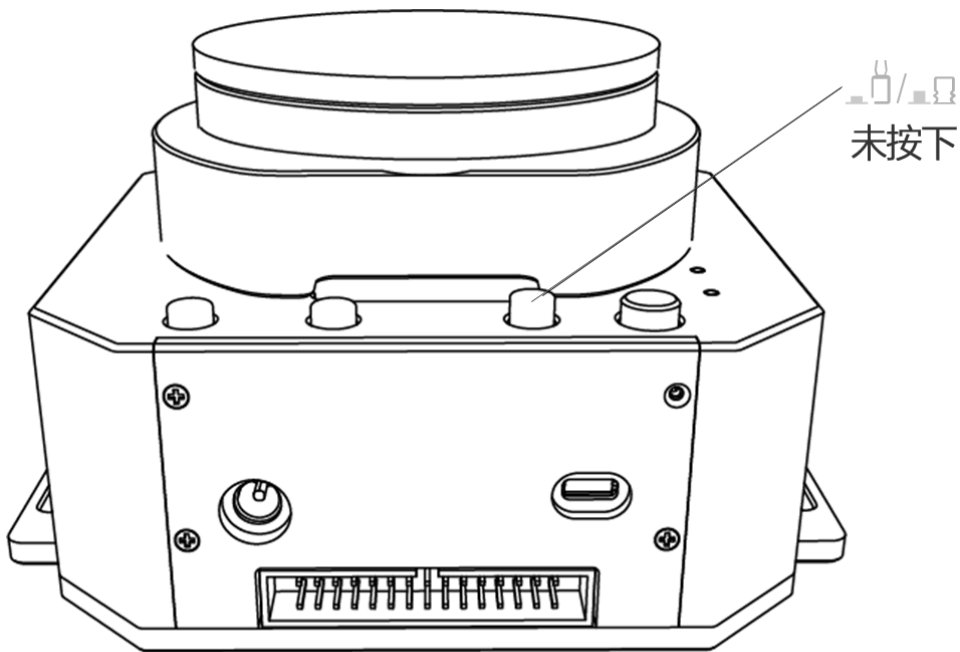


**注意:** 默认情况下, 自定义键用于蓝牙/USB 切换, 使用 USB 模式时, 请确保自定义键处于弹起状态。



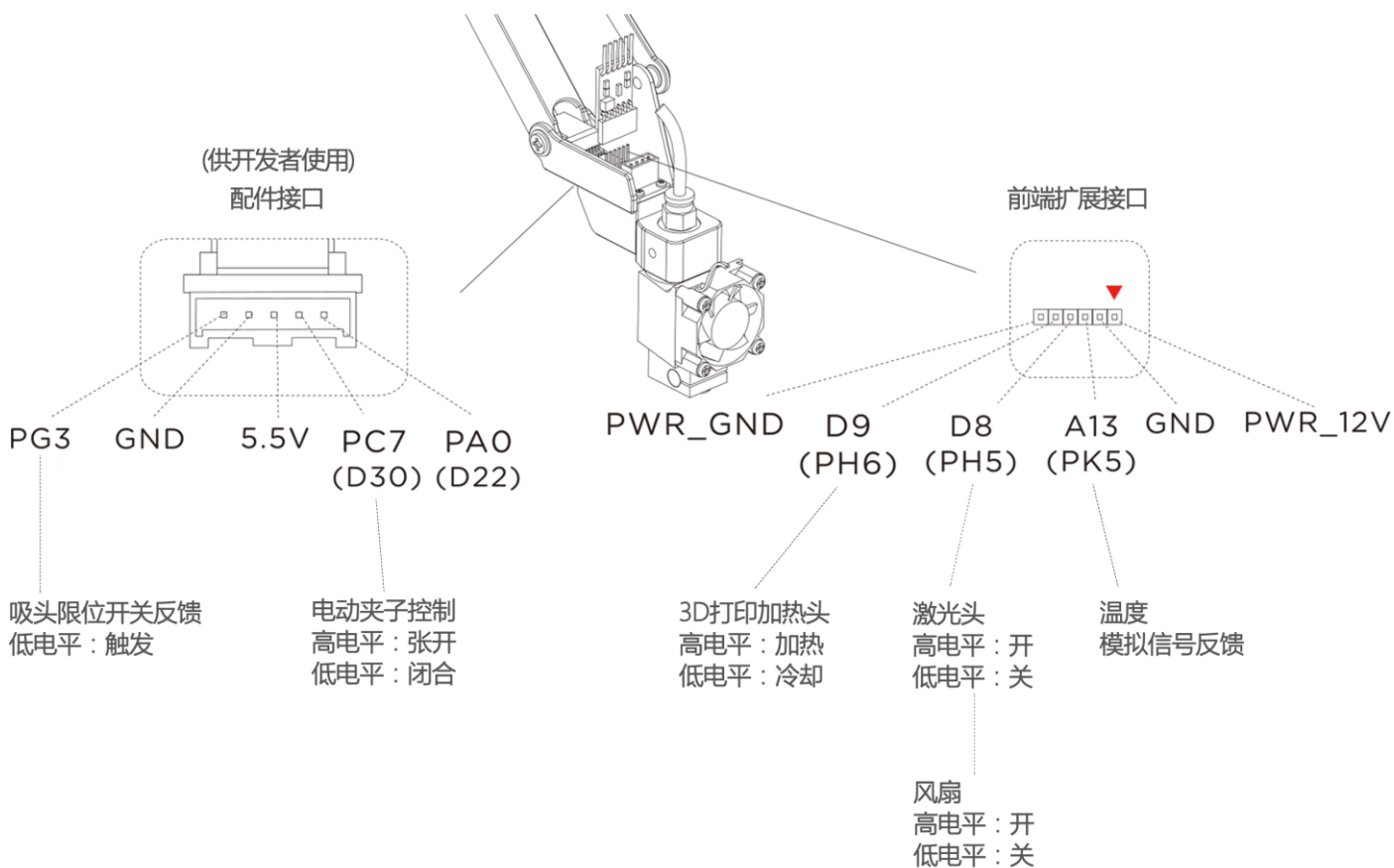
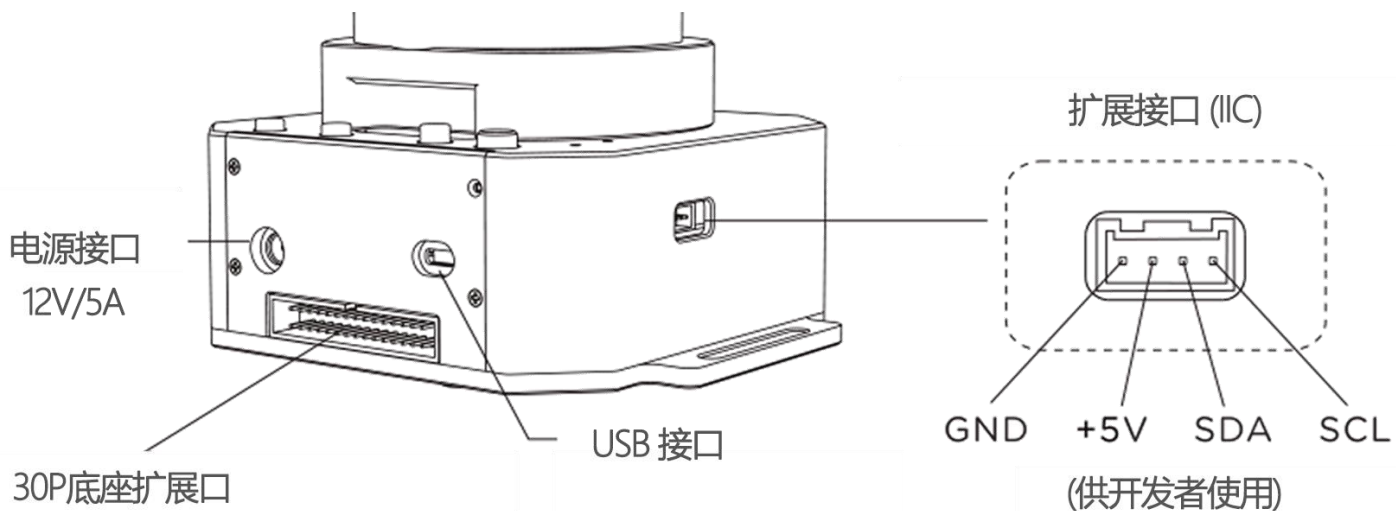


蓝牙

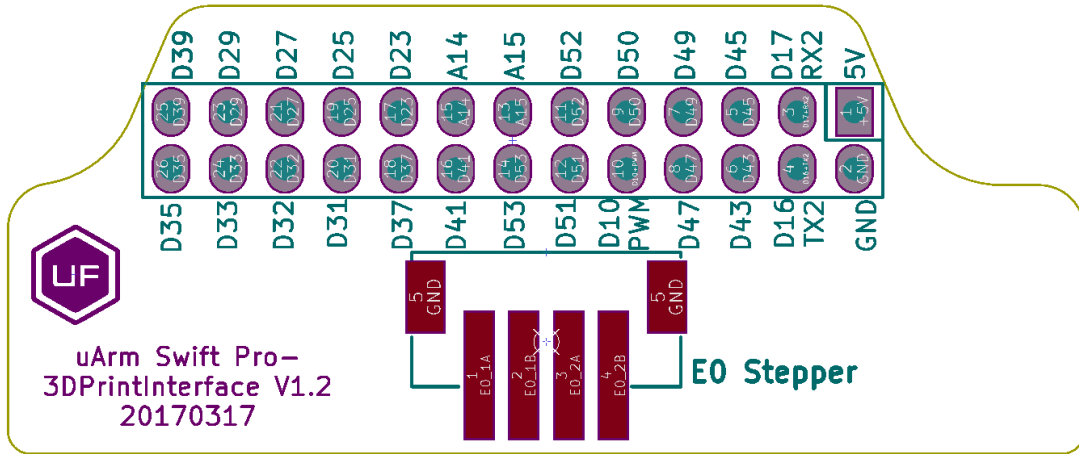


USB

## 6. 扩展简介



# 30P 底座扩展板



# 规格参数

uArm Swift Pro 规格参数				
重量	2.2kg			
自由度	4			
精度	0.2mm			
最大载荷	500g			
工作范围	50mm ~ 320mm			
最大速度	100mm/s			
有线连接	Micro USB			
无线连接	蓝牙 4.0			
输入电压	DC 12V			
电源适配器	输入:100 ~ 240V 50/60Hz; 输出: 12V5A 60W			
操作温度 & 湿度	0°C-35°C 30%RH-80%RH 无凝结			
储存温度 & 湿度	-20°C-60°C 30%RH-80%RH 无凝结			
硬件				
电机型号	定制变速箱 + 步进电机			
位置反馈	12 bit 编码			
减速器	定制减速器			
尺寸(L*W*H)	150mm*140mm*281mm			
主板	Arduino MEGA 2560			
机身材质	铝合金			
波特率	115200bps			
扩展 I/O 接口	I/O *27, IIC *1, 5V*1, 12V*1, Stepper*1			
软件				
PC	uArm Studio			
App	uArm Play			
开发平台	Python/Arduino/ROS			
特征	开源			
电机速度 & 扭矩				
	角度	速度	使用寿命	扭矩
主电机	0° ~ 180°	40°/s	>3000h	12kg·cm
左电机	0° ~ 130°	40°/s	>3000h	12kg·cm
右电机	0° ~ 106°	40°/s	>3000h	12kg·cm
第四轴电机	0° ~ 180°	60°/s	500h	2kg·cm

## 配件

气泵	吸盘尺寸	5mm ~ 10mm
	最大压力	33kPa
	最大提升载荷	1000g
	特征	有反馈
通用笔夹	重量	36g
	尺寸(L*W*H)	62mm*25mm*15mm
	材质	铝合金
	被夹物体最大直径	14mm
电动夹子	重量	58g
	尺寸(L*W*H)	92mm*50mm*18mm
	材质	铝合金
	最大载荷	750~800g
	被夹物体最大直径	40mm
	最大运动速度	20mm/s
	驱动方式	电动
	工作电压/电流	6V/300mA
OpenMV 相机	焦距	2.8mm
	FOV	115°
	光圈	f2.0
	编程方式	Micro Python
	重量	16g
	尺寸(L*W*H)	45mm*36mm*30mm
	频段	<b>100M</b>
内置蓝牙	工作频率	<b>2400Mhz – 2483.5MHz</b>
	带宽	<b>2MHz</b>
	发射功率	<b>0dBm</b>
	工作模式	<b>GFSK</b>
	单工 / 双工	<b>simplex</b>
	占空比	<b>100ms</b>
	访问协议	<b>低功耗蓝牙 4.0</b>
	发射机类型	<b>CLASS 2</b>
	3D 打印套件 (Pro 专属)	型号
功耗		35W
喷头直径		0.4mm
最高温度		270 °C
打印材料		PLA
最大打印速度		20mm/s
文件格式		Gcode
打印尺寸(L*W*H)		10mm*10mm*10mm
激光雕刻套件	激光功率	500mW

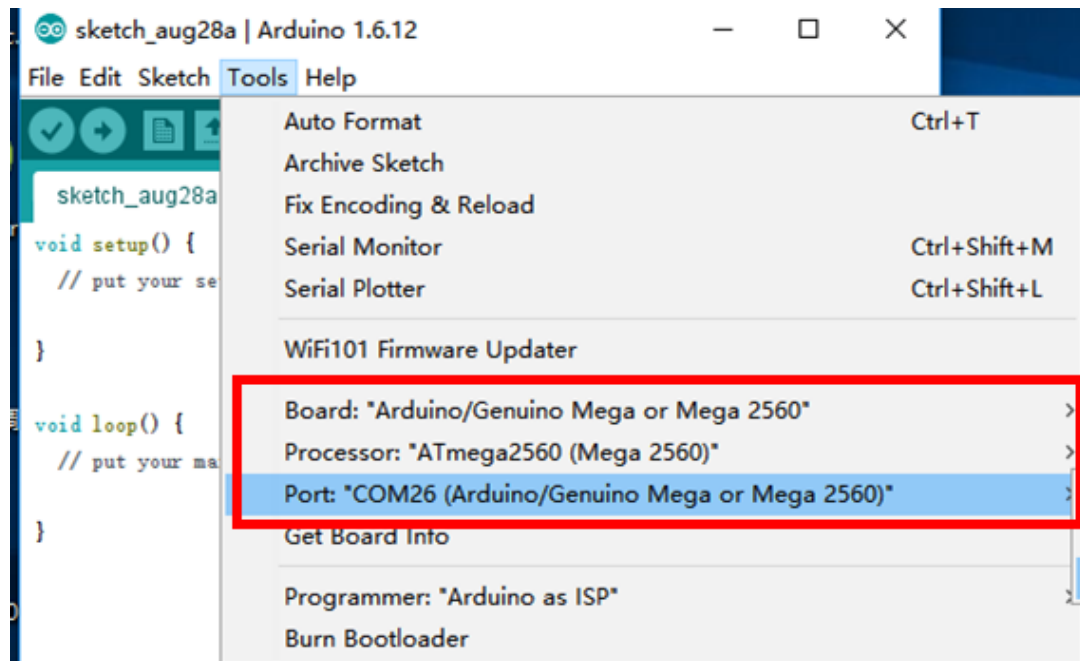
(Pro 专属)	工作电压/电流	12V/5A
	波长	405nm
	重量	140g
	尺寸(L*W*H)	55mm*33mm*88mm
	可雕刻材料	木头、塑料、皮革、绒毛、纸等

## 应用程序信息

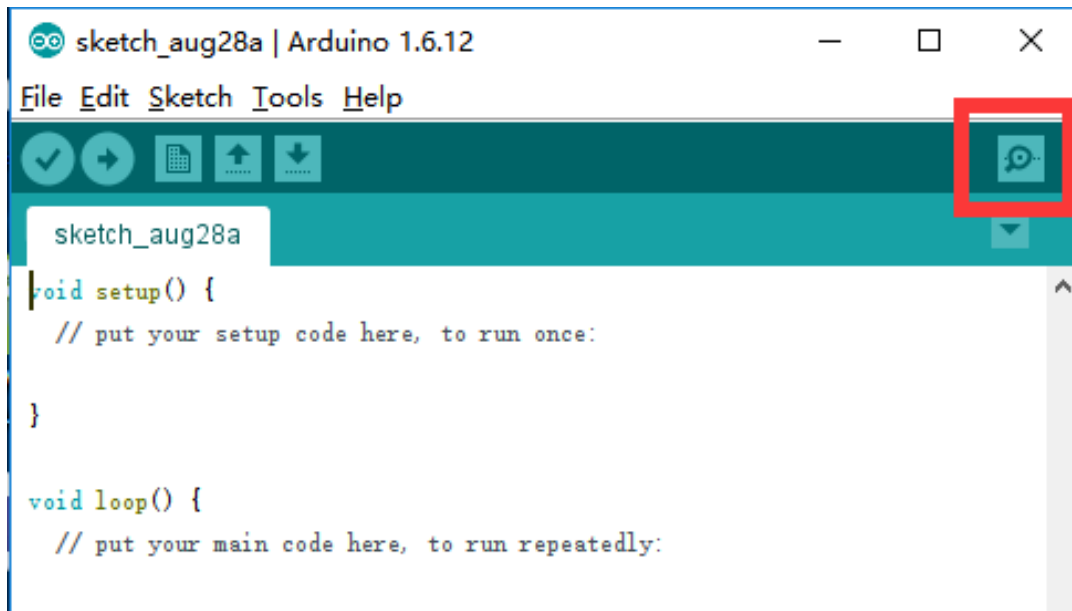
下面内容将介绍不同平台下 uArm 的工作方式

### 1. 通过 USB 口发送指令

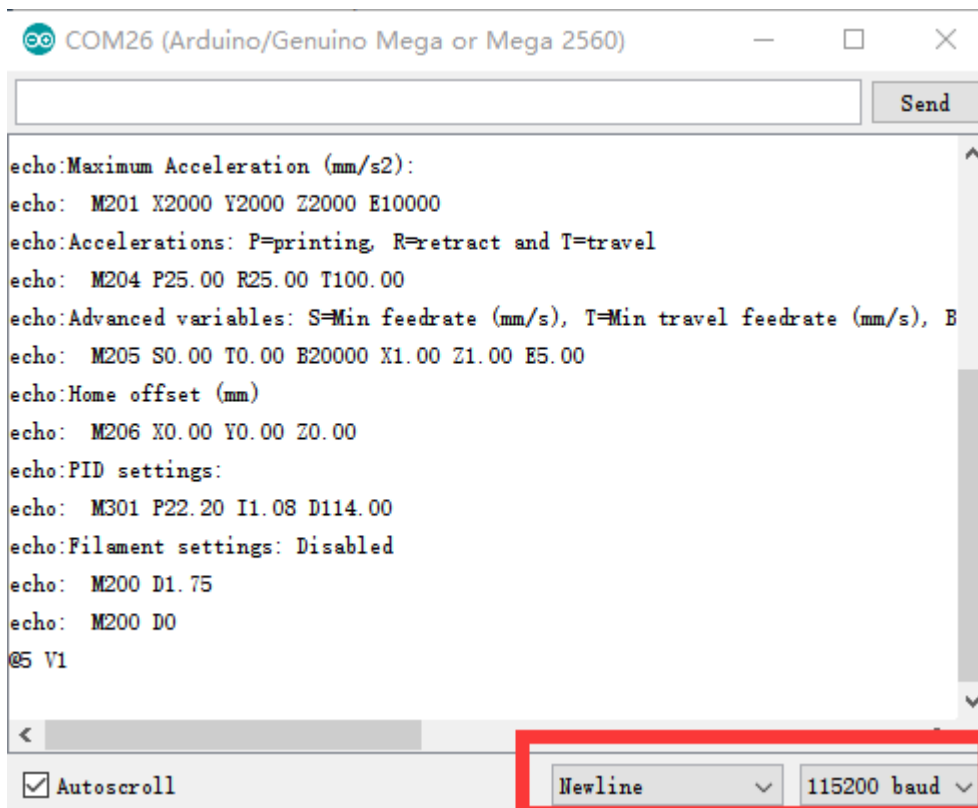
开启 uArm 电源，启动 Arduino IDE，按下图所示设置参数，请确保使用正确的接口



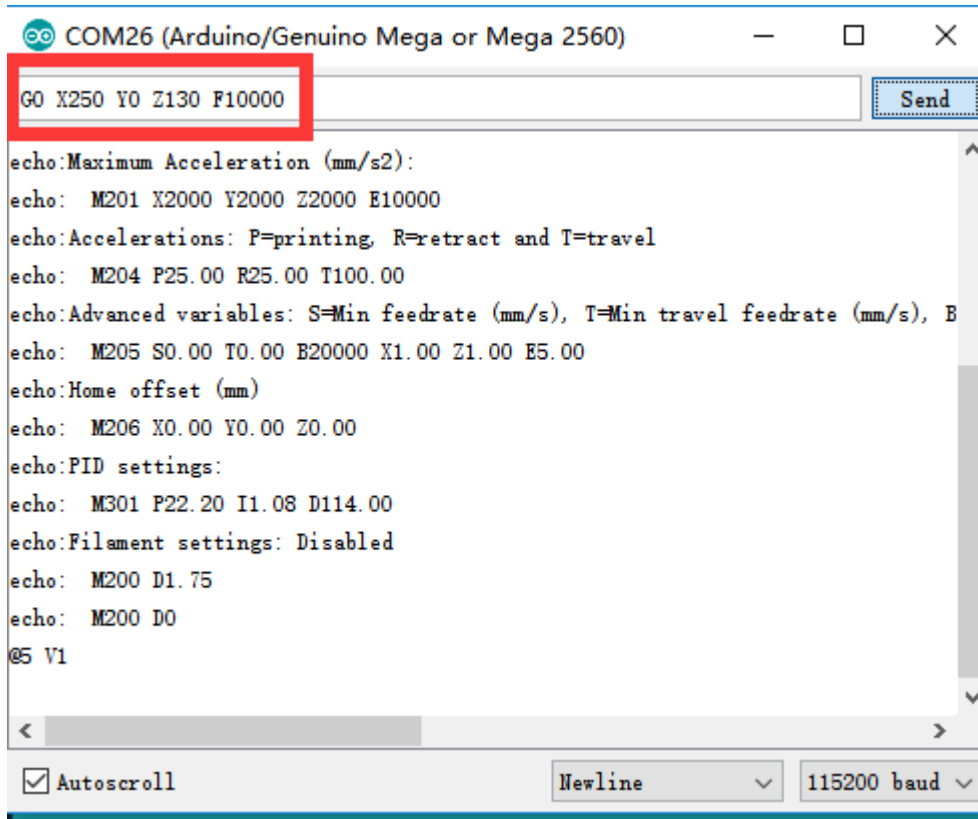
打开界面右上方串口监视器，点击图标后听到“嘀”的一声表示 uArm 已连接



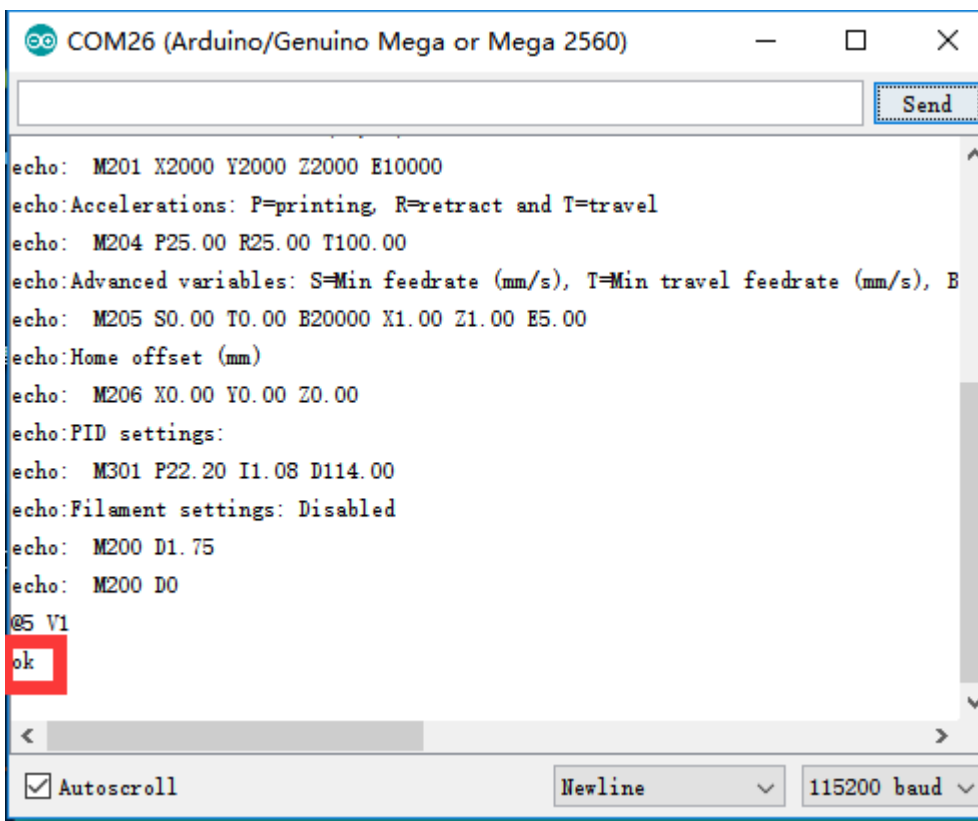
按下图设置串口参数，如果参数设置正确，界面会出现如下提示



参数设置完成后，可以尝试向 uArm 发送指令，例如发送“G0 X250 Y0 Z130 F10000”



如果机械臂完成动作，将会返回“ok”

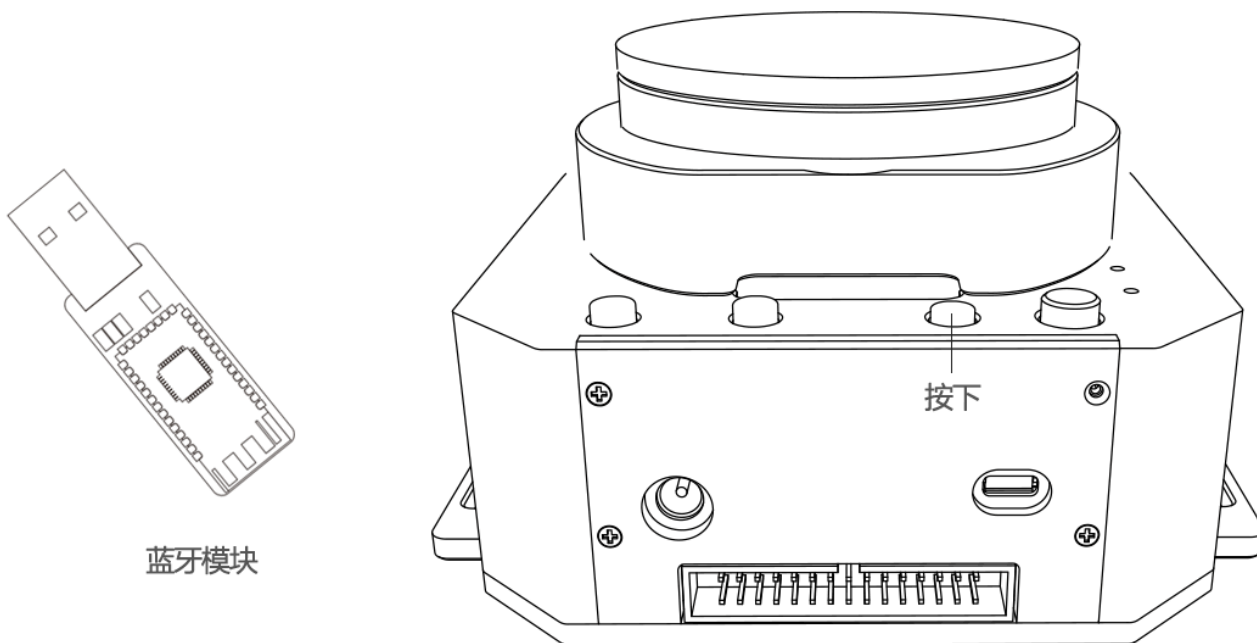


请查看“通讯协议”获取更多测试更多指令.



## 2.通过蓝牙发送指令

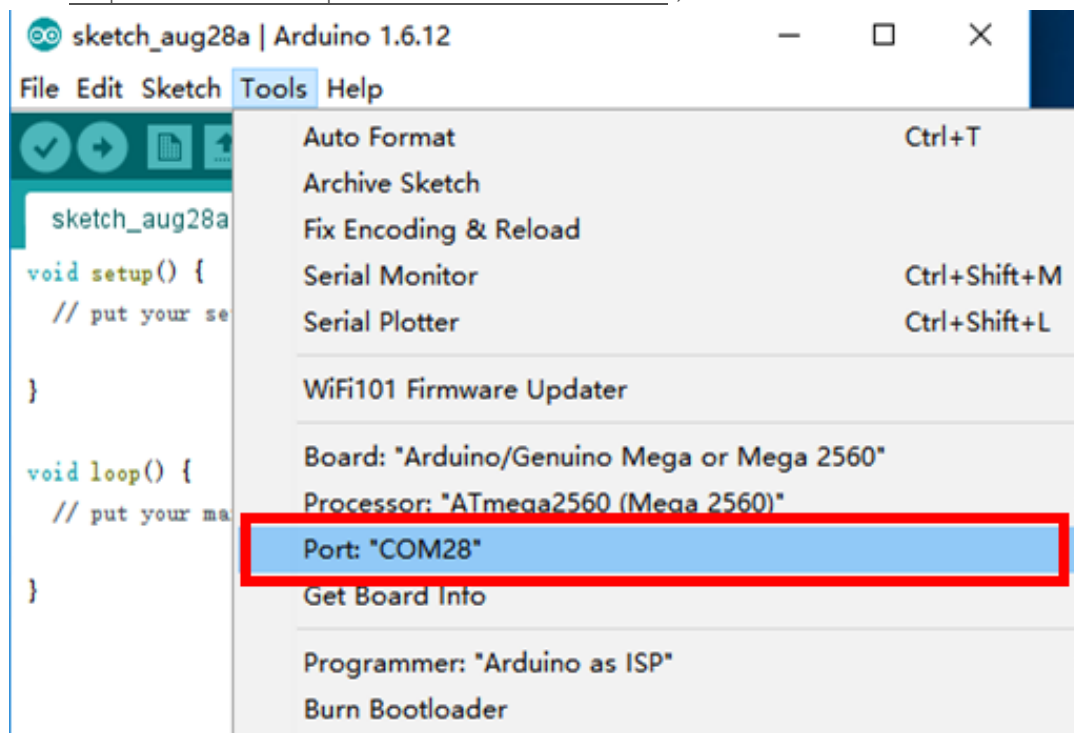
插入蓝牙模块，按下电源键旁边的自定义键



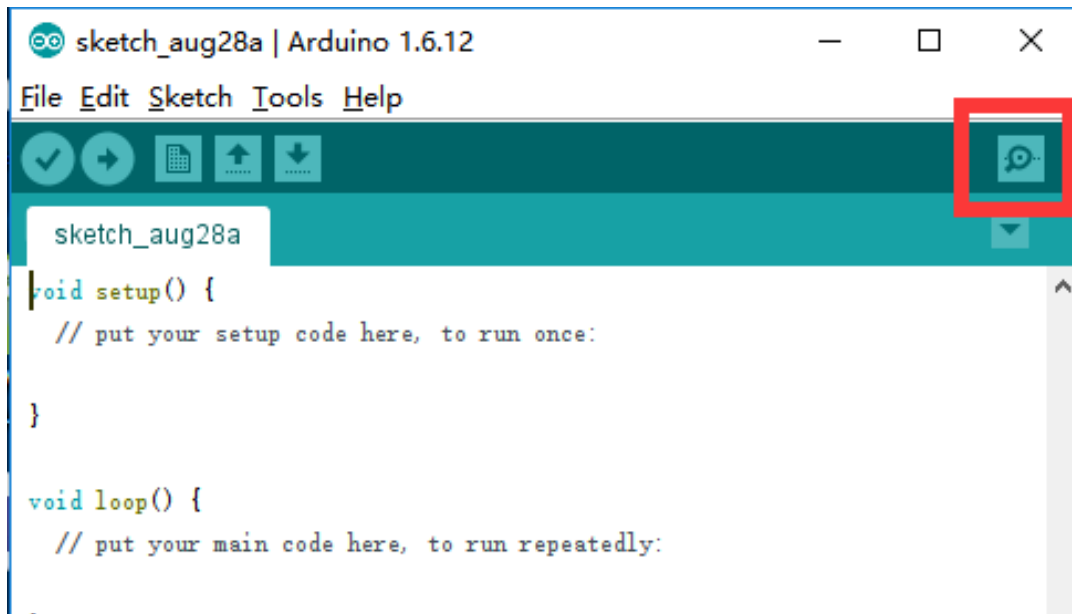
启动 uArm，搜索蓝牙信号时，蓝色指示灯蓝灯闪烁。当蓝牙连接后，蓝牙模块和机械臂上的指示灯将都将蓝灯常亮。

启动 Arduino IDE，按下图所示设置 COM 口，请确保所选择的接口兼容蓝牙模块。

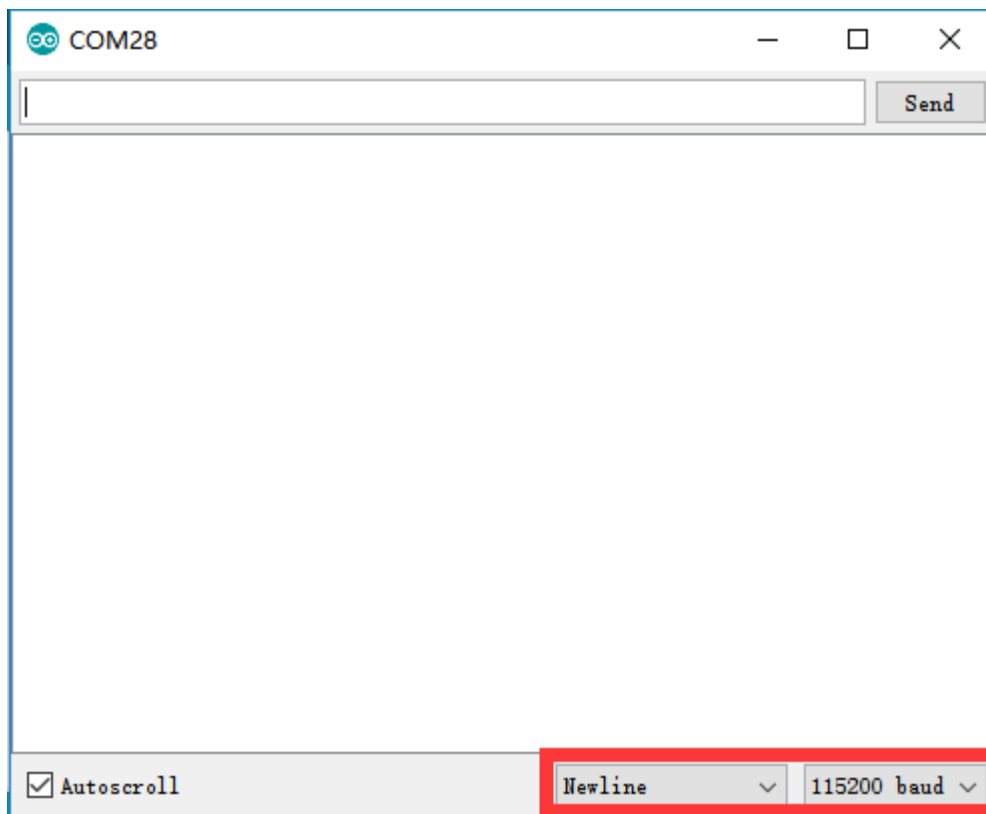
(蓝牙驱动: <http://www.ftdichip.com/Drivers/VCP.htm>)



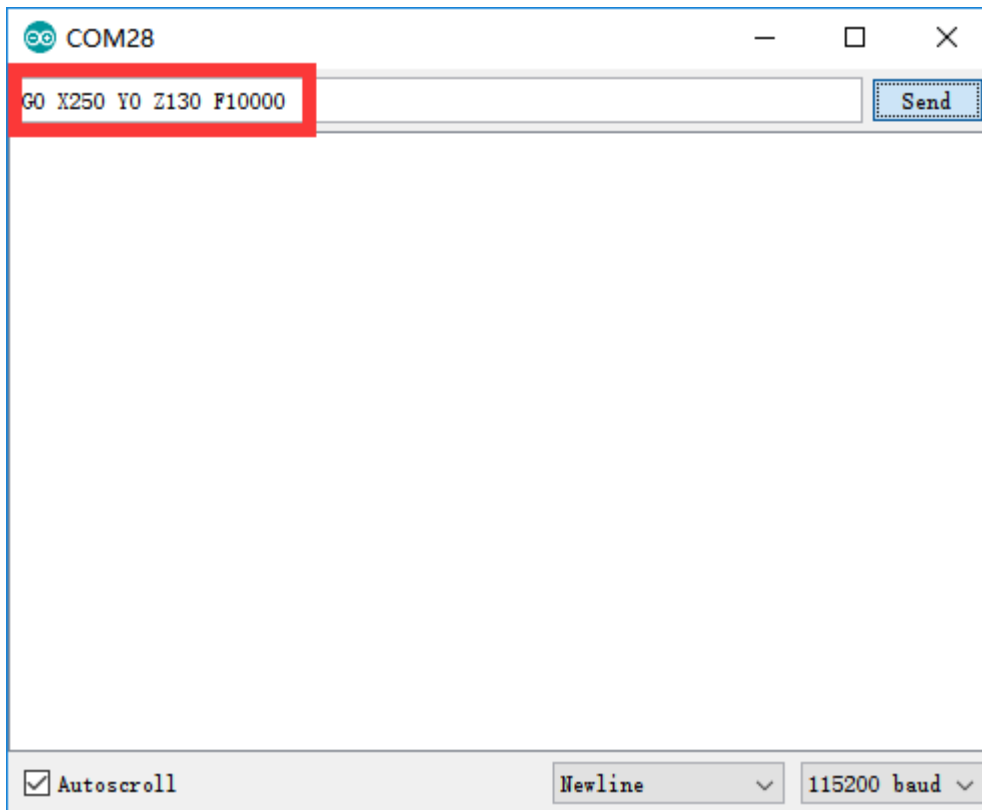
Arduino IDE 中打开串口监视器，如果听到“滴”的一声表示 uArm 已连接



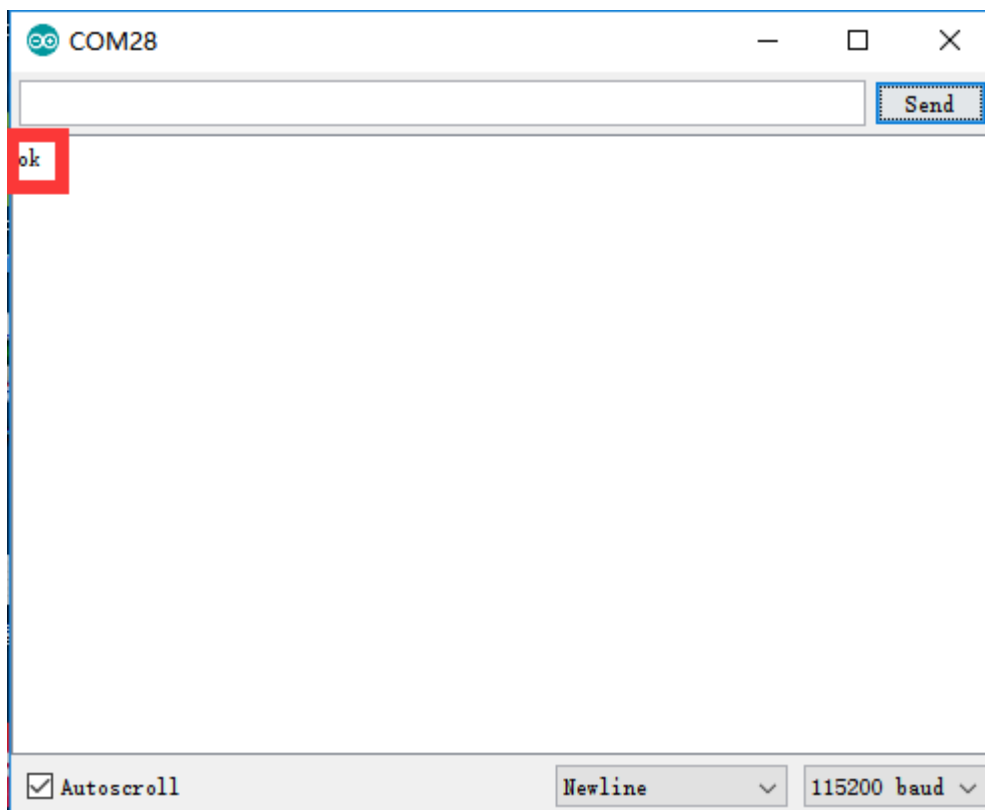
按下图所示，在右下角设置串口监视器参数



参数配置完成后，可以尝试向机械臂发送指令，例如发送“G0 X250 Y0 Z130 F10000”。



如果机械臂完成动作，将会返回“ok”

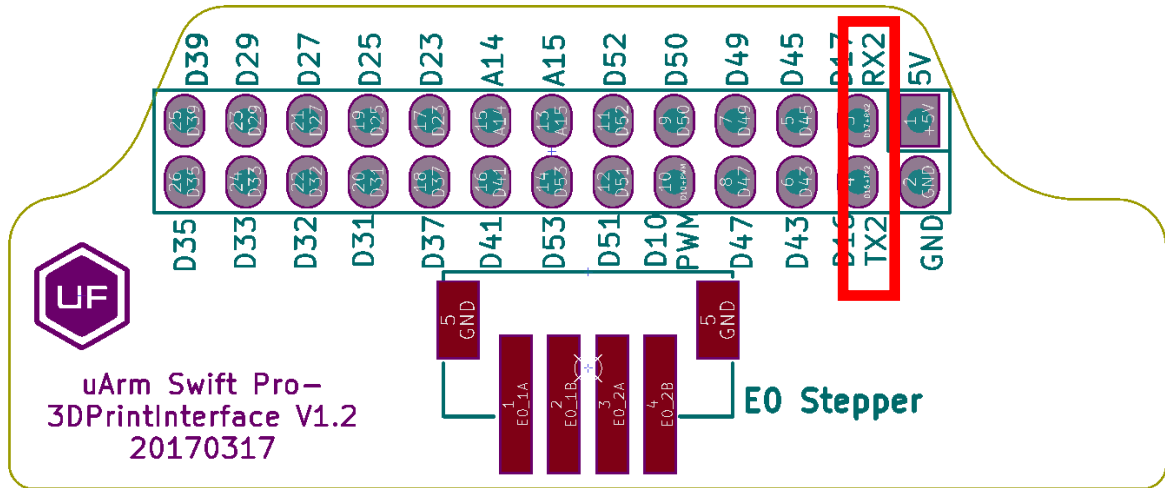


请查看“通讯协议”获取更多测试指令

### 3. 第二 UART

某些时候第二 UART 非常重要, 比如当你想用另一个 Arduino 板来和 uArm 通信的时候。出于这个考虑, 我们在 30P 底座扩展口中预留了第二 UART

底座扩展口的所有针都和 Arduino MEGA 2560 直接相连, 是 TTL 级别。注意: 5V 以上的电压可能会造成 IO 损坏.

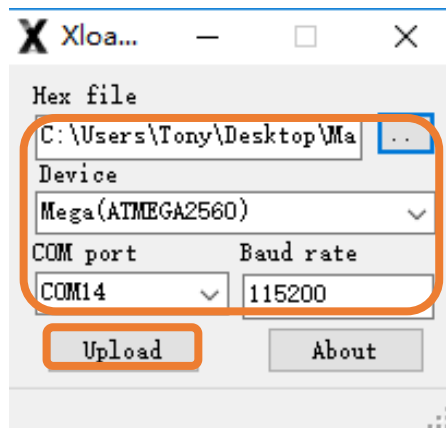


用导线连接 UART 和 GND, 完成硬件配置。由于代码只支持单口通信, 所以主通讯口将由 USB 转移到第二 UART 口。

将 USB 通讯切换为第二 UART 通信

将 uArm Swift Pro 连接到电脑。打开 XLoader ([xloader.russeotto.com/](http://xloader.russeotto.com/)), 加载

uArmSwiftPro\_2ndUART.hex (下载连接: <https://github.com/TonyLeheng/Vision-Pick-and-Place>) 点击“上传”按钮将代码上传至 uArm Swift Pro。



由第二 UART 通讯切回 USB 通信

将 uArm Swift Pro 连接到电脑。打开 XLoader ([xloader.russeotto.com/](http://xloader.russeotto.com/)), 加载

uArmSwiftPro\_Standard.hex (下载连接: <https://github.com/TonyLeheng/Vision-Pick-and-Place>) 点击“上传”按钮将代码上传至 uArm Swift Pro。

## 4. Arduino

主要代码通过 Arduino IDE 编写，请参考以下链接：

<https://github.com/uArm-Developer/SwiftProForArduino>

如何执行和上传文件

1. 下载代码后解压到 Arduino library 文件夹（默认情况下文件地址为 C:/Users/ufactory/documents/Arduino/libraries/）

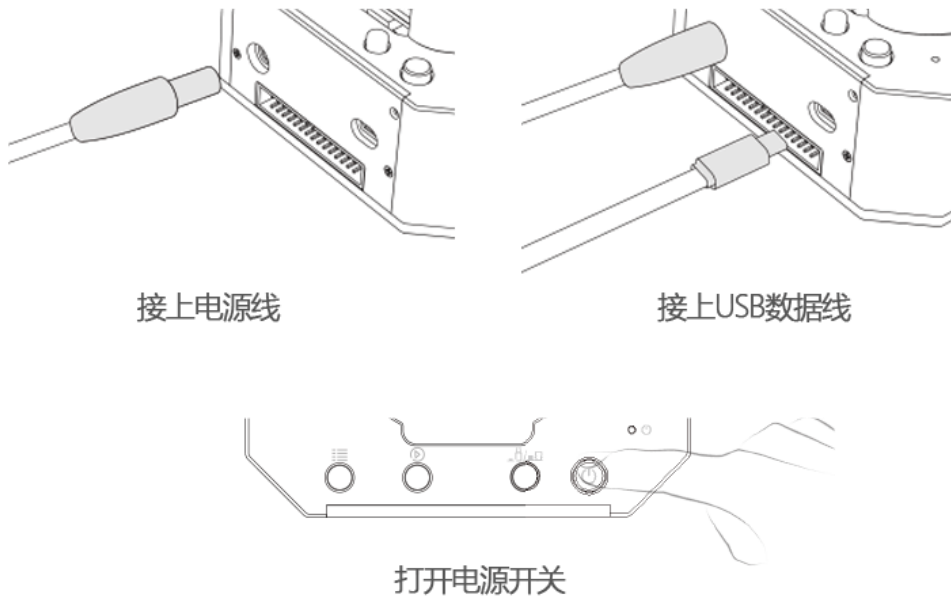


2. Marlin 文件夹中找到名为 Marlin.ino 的文件并打开

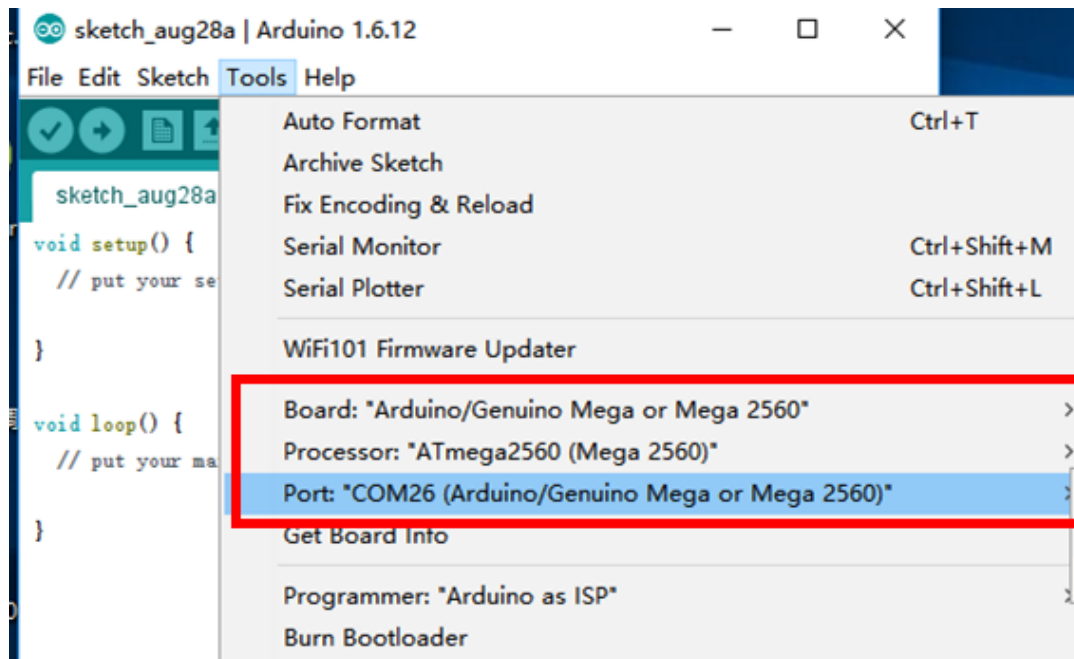
« Arduino » libraries » SwiftProForArduino-master » Marlin » 搜索"Marlin"

名称	修改日期	类型	大小
uArmLed.h	2017/6/28 20:07	H 文件	1 KB
uArmParams.h	2017/6/28 20:07	H 文件	3 KB
uArmPump.h	2017/6/28 20:07	H 文件	1 KB
uArmRecorder.h	2017/6/28 20:07	H 文件	1 KB
uArmReportService.h	2017/6/28 20:07	H 文件	1 KB
uArmSerial.h	2017/6/28 20:07	H 文件	2 KB
uArmService.h	2017/6/28 20:07	H 文件	2 KB
uArmServo.h	2017/6/28 20:07	H 文件	1 KB
uArmSwift.h	2017/6/28 20:07	H 文件	3 KB
uArmUtils.h	2017/6/28 20:07	H 文件	1 KB
ultralcd.h	2017/6/28 20:07	H 文件	8 KB
ultralcd_impl_DOGM.h	2017/6/28 20:07	H 文件	23 KB
ultralcd_impl_HD44780.h	2017/6/28 20:07	H 文件	30 KB
ultralcd_st7920_u8glib_rrd.h	2017/6/28 20:07	H 文件	7 KB
Ultrasonic.h	2017/6/28 20:07	H 文件	2 KB
utf_mapper.h	2017/6/28 20:07	H 文件	15 KB
utility.h	2017/6/28 20:07	H 文件	1 KB
vector_3.h	2017/6/28 20:07	H 文件	3 KB
Version.h	2017/6/28 20:07	H 文件	3 KB
watchdog.h	2017/6/28 20:07	H 文件	2 KB
X_IIC.h	2017/6/28 20:07	H 文件	1 KB
Y_IIC.h	2017/6/28 20:07	H 文件	1 KB
Z_IIC.h	2017/6/28 20:07	H 文件	1 KB
Marlin.ino	2017/6/28 20:07	INO 文件	3 KB
Makefile	2017/6/28 20:07	文件	16 KB

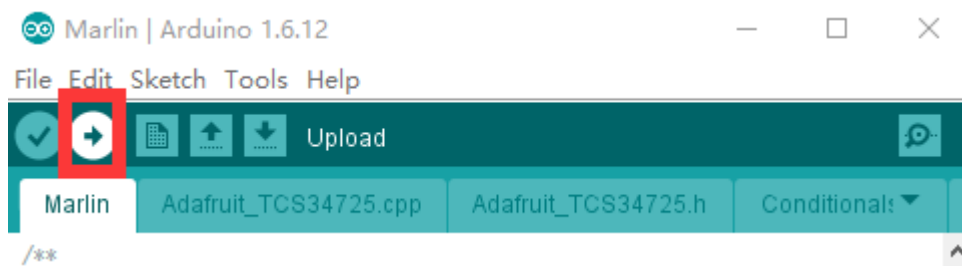
3. 插入 USB 线和电源线，打开电源开关



4. 按下图所示选择正确的接口和 Arduino 板型号



5. 点击“上传”按钮，完成上传



## 5. ROS & Python

目前推出了 Python 和 ROS 库，详情请访问

Swift Pro ROS 代码:

<https://github.com/uArm-Developer/SwiftproForROS>

Swift Pro Python 库:

<https://github.com/uArm-Developer/pyuf>

## 6.OpenMV 示例

OpenMV:

<https://github.com/uArm-Developer/OpenMV-Examples>

您可以在《用户手册》里查看更多的操作细节。

## 7.从错误的代码中恢复

当下载 Arduino 源代码过于复杂或者你写入了错误的代码导致 uArm 无法运行时, 请使用官方 flash 工具恢复官方固件:

<https://drive.google.com/drive/u/0/folders/0B-L-tCvknXU9dDhfSGJwT1JDY1U>

# 通讯协议

1) 简介:

uArm GCode 是 uArm 软件的重要组成部分, 我们基于标准 GCode 协议, 在 GCode 开头增加了新的协议头, 使其更加易用, 更容易 debug。此外, uArm GCode 还与标准 GCode 兼容 (我们提供可解码标准 GCode 的代码)

2) 示例:

- PC 发送指令

```
"#25 G0 X180 Y0 Z150 F5000"
```

```
//速度 5000mm/min 移动到 [180,0,150]
```

- uArm 返回 "\$25 ok"

3) 指令集(TBD).

指令可以分为两部分:

带下划线的指令: 新增的协议头

- PC 发出的指令以 '#' 开头, uArm 发出的指令以 '\$' 开头。



- 符号后面的数字由 PC 决定, uArm 发回的指令, 必须和表示动作完成的指令有相同的数字。(如上述例子中, PC 发送指令'#25', uArm 返回指令'\$25')

不带下划线的指令: 标准的 GCode.

注意:

- 1.每个参数之间需要有空格;
- 2.指令符号必须大写

GCode 指令 (v1.2)	描述	反馈
1. <b>#n</b> 用于 debug, 如果不需要使用可以直接删掉。 (例如: G2202 N <b>0</b> V <b>90</b> \n) 2. '\n' 换行符.		
运动指令 (参数带下划线)		
<b>#n</b> G0 X <b>100</b> Y <b>100</b> Z <b>100</b> F <b>1000</b> \n	移动到 XYZ(mm), F 是速度 (mm/min)	<b>\$n</b> ok \n or <b>\$n</b> Ex \n (参考报错表)
<b>#n</b> G1 X <b>100</b> Y <b>100</b> Z <b>100</b> F <b>1000</b> \n	进入激光模式后, (M2400 S1), 指令 <b>G1</b> 开启激光, 指令 <b>G0</b> 关闭激光	<b>\$n</b> ok \n or <b>\$n</b> Ex \n (参考报错表)
<b>#n</b> G2004 P <b>1000</b> \n	微秒延时	<b>\$n</b> ok \n
<b>#n</b> G2201 S <b>100</b> R <b>90</b> H <b>80</b> F <b>1000</b> \n	极坐标系, S 是长度(mm), R 是旋转角度(°), H 是高度(mm), F 是速度(mm/min)	<b>\$n</b> ok \n or <b>\$n</b> Ex \n (参考报错表)
<b>#n</b> G2202 N <b>0</b> V <b>90</b> \n	移动电机到此位置, N 是电机 ID (0~3), V 是角度(0~180)	<b>\$n</b> ok \n or <b>\$n</b> Ex \n (参考报错表)
<b>#n</b> G2204 X <b>10</b> Y <b>10</b> Z <b>10</b> F <b>1000</b> \n	相对位移	<b>\$n</b> ok \n or <b>\$n</b> Ex \n (参考报错表)

# <u>n</u> G2205 S <u>10</u> R <u>10</u> H <u>10</u> F <u>1000</u> \n	极坐标相对位移	\$ <u>n</u> ok \n or \$ <u>n</u> Ex \n (参考报错表)
设置指令 (参数带下划线)		
# <u>n</u> M17\n	锁住所有电机	\$ <u>n</u> ok \n
#n M204 P200 T200 R200\n	设置加速度并保存 P = 打印运动 R =仅出料 (没有 X, Y, Z 移动) T =仅运动 (不出料)	\$ <u>n</u> ok \n
# <u>n</u> M2019\n	解锁所有电机	\$ <u>n</u> ok \n
# <u>n</u> M2120 V <u>0.2</u> \n	设置反馈周期, 返回笛卡尔坐标, V 时间(秒)	@3 X <u>154.71</u> Y <u>194.91</u> Z <u>10.21</u> \n
# <u>n</u> M2122 V <u>1</u> \n	停止时报告(@9 V0) V1: 开启 V0: 关闭	\$ <u>n</u> ok \n
# <u>n</u> M2201 N <u>0</u> \n	锁住电机, N 是电机 ID(0~3)	\$ <u>n</u> ok \n or \$ <u>n</u> Ex \n (参考报错表)
# <u>n</u> M2202 N <u>0</u> \n	解锁电机, N 是电机 ID (0~3)	\$ <u>n</u> ok \n or \$ <u>n</u> Ex \n (参考报错表)
# <u>n</u> M2203 N <u>0</u> \n	检查电机是否解锁, N 是电机 ID(0~3)	\$ <u>n</u> ok V <u>1</u> \n (1 锁住,0 解锁)
# <u>n</u> M2210 F <u>1000</u> T <u>200</u> \n	蜂鸣器,F 是频率, T 是时间(ms)	\$ <u>n</u> ok \n or \$ <u>n</u> Ex \n (参考报错表)
# <u>n</u> M2211 N <u>0</u> A <u>200</u> T <u>1</u> \n	读取 EEPROM N(0~2,0 是内部 EEPROM,1 是 USR_E2PROM, 2 是 SYS_E2PROM), A 是地址, T 是类(1 char,2 int,4 float)	\$ <u>n</u> ok V <u>10</u> \n
# <u>n</u> M2212 N <u>0</u> A <u>200</u> T <u>1</u> V <u>10</u> \n	写入 EEPROM N(0~2,0 是内部 EEPROM,1 是 USR_E2PROM, 2 是 SYS_E2PROM), A 是地址, T 是类 (1 char,2 int,4 float) V 是 输入数据	\$ <u>n</u> ok\n

# <u>n</u> M2213 V <u>0</u> \n	底座功能按键默认值(0 false, 1 true)	\$ <u>n</u> ok\n
# <u>n</u> M2220 X <u>100</u> Y <u>100</u> Z <u>100</u> \n	将坐标转换为电机角度	\$ <u>n</u> ok B <u>50</u> L <u>50</u> R <u>50</u> \n (B 0 号电机,L 1 号电机,R 2 号电机, 0~180)
# <u>n</u> M2221 B <u>0</u> L <u>50</u> R <u>50</u> \n	将电机角度转换为坐标	\$ <u>n</u> ok X <u>100</u> Y <u>100</u> Z <u>100</u> \n
# <u>n</u> M2222 X <u>100</u> Y <u>100</u> Z <u>100</u> P <u>0</u> \n	确认是否可到达,P1 极坐标 , P0 笛卡尔坐标	\$ <u>n</u> ok V <u>1</u> \n (1 可到达, 0 不可到达)
# <u>n</u> M2231 V <u>1</u> \n	气泵, V1 工作, V0 停止工作	\$ <u>n</u> ok \n or \$ <u>n</u> Ex \n (参考报错表)
# <u>n</u> M2232 V <u>1</u> \n	电动夹子, V1 合上, V0 打开	\$ <u>n</u> ok \n or \$ <u>n</u> Ex \n (参考报错表)
# <u>n</u> M2234 V <u>1</u> \n	开启/关闭蓝牙(1:开启, 0:关闭)	\$ <u>n</u> ok\n
# <u>n</u> M2240 N <u>1</u> V <u>1</u> \n	设置数字 IO 输出	\$ <u>n</u> ok \n or \$ <u>n</u> Ex \n (参考报错表)
# <u>n</u> M2241 N <u>1</u> V <u>1</u> \n	设置数字 IO 方向(V1 输出; V0 输入;)	\$n ok \n
# <u>n</u> M2245 V <u>btname</u> \n	设置蓝牙名称, 限制最长 11 个字母	\$n ok \n
# <u>n</u> M2304 P <u>0</u> \n	请检查 Grove 模块	
# <u>n</u> M2305 P <u>0</u> N <u>1</u> \n	请检查 Grove 模块	
# <u>n</u> M2306 P <u>0</u> V <u>1000</u> \n	请检查 Grove 模块	
# <u>n</u> M2307 P <u>0</u> V <u>1</u> \n	请检查 Grove 模块	
# <u>n</u> M2400 S <u>0</u> \n	设置机械臂工作模式 (0:常规 1:激光 2:3D 打印 3:通用笔夹)	\$ <u>n</u> ok \n
# <u>n</u> M2401\n	设置以当前位置作为参照点	\$ <u>n</u> ok \n

# <u>n</u> M2410\n	设置高度 0 点	\$ <u>n</u> ok \n
# <u>n</u> M2411 S <u>100</u> \n	设置末端执行器偏移 (mm)	\$ <u>n</u> ok \n
# <u>n</u> M2500\n	将 uart 0 切换到 uart 2 做外部 TTL uart 通信(例如 OpenMV)	\$ <u>n</u> ok \n
查询指令 (参数带下划线)		
# <u>n</u> P2200\n	获取当前电机角度	\$ <u>n</u> ok B <u>50</u> L <u>50</u> R <u>50</u> \n
# <u>n</u> P2201\n	获取设备名称	\$ <u>n</u> ok V <u>3.2</u> \n
# <u>n</u> P2202\n	获取硬件版本信息	\$ <u>n</u> ok V <u>1.2</u> \n
# <u>n</u> P2203\n	获取软件产品信息	\$ <u>n</u> ok V <u>3.2</u> \n
# <u>n</u> P2204\n	获取 API 版本	\$ <u>n</u> ok V <u>3.2</u> \n
# <u>n</u> P2205\n	获取 UID	\$ <u>n</u> ok V <u>0123456789AB</u> \n
# <u>n</u> P2206 N <u>0</u> \n	获取 0 号电机角度 (0~2)	\$n ok V <u>80</u> \n
# <u>n</u> P2220\n	获取当前坐标	\$ <u>n</u> ok X <u>100</u> Y <u>100</u> Z <u>100</u> \n
# <u>n</u> P2221\n	获取当前极坐标	\$ <u>n</u> ok S <u>100</u> R <u>90</u> H <u>80</u> \n
# <u>n</u> P2231\n	获取气泵状态	\$ <u>n</u> ok V <u>1</u> \n (0 停止, 1 工作, 2 抓取物体)
# <u>n</u> P2232\n	获取电动夹子状态	\$ <u>n</u> ok V <u>1</u> \n (0 停止, 1 工作, 2 抓取物体)
# <u>n</u> P2233\n	获取限位开关状态	\$ <u>n</u> ok V <u>1</u> (1 触发, 0 未触发)
# <u>n</u> P2234\n	获取电源连接状态	\$ <u>n</u> ok V <u>1</u> (1 连接, 0 未连接)
# <u>n</u> P2240 N <u>1</u> \n	获取数字 IO 状态	\$ <u>n</u> ok V <u>1</u> \n (1 高, 0 低)
# <u>n</u> P2241 N <u>1</u> \n	获取模拟 IO 状态	\$ <u>n</u> ok V <u>295</u> \n (返回 ADC 数据)
# <u>n</u> P2242\n	获取每个电机的 AS5600 默认值	\$ <u>n</u> ok B <u>2401</u> L <u>344</u> R <u>1048</u> \n

# <u>n</u> P2400\n	检查当前状态	\$ <u>n</u> ok V <u>1</u> \n (0: 常规; 1: 激光; 2: 3D 打印; 3: 通用笔夹;)
Ticking feedback		
@1	准备就绪	
@3	定时反馈, "M2120"	
@4 N <u>0</u> V <u>1</u> \n	报告按键事件 N: 0 = 菜单键, 1 = 播放键 V: 1 = 短按, 2 = 长按	
@5 V <u>1</u> \n	报告电源连接事件	
@6 N <u>0</u> V <u>1</u> \n	报告末端执行器限位开关事件	
@7 temp error	打印温度错误	
@9 V0\n	停止运动	
报错表		
E20	指令不存在	
E21	参数错误	
E22	地址超出范围	
E23	指令缓存已满	
E24	电源未连接	
E25	操作失败	

Grove 模块					
Grove No.	Module	指令	说明	支持 Ports	Return
1	可连接 RGB LED	# <u>n</u> M2304 P <u>3</u> \n	反初始化	3, 4, 5	\$ <u>n</u> ok\n
		# <u>n</u> M2305 P <u>3</u> N1 V <u>2</u> \n	在 Port <u>3</u> 初始化模块 1 V 是 LEDs 序号	3, 4, 5	\$ <u>n</u> ok\n or E25 初始化失败

		#n M2307 P3 V0 R228 G128 B100\n	设置 <u>0</u> <sup>th</sup> LED 颜色	3, 4, 5	\$n ok\n
2	按钮	#n M2304 P3\n	反初始化	3, 4, 5	
		#n M2305 P3 N2\n	在 Port <u>3</u> 初始化模 2	3, 4, 5	
		Press down			@11 P <u>3</u> N2 V0\n
		Click			@11 P <u>3</u> N2 V1\n
		Long pressed			@11 P <u>3</u> N2 V2\n
3	滑动电位计	#n M2304 P <u>1</u> \n	反初始化	1, 2	\$n ok \n
		#n M2305 P <u>1</u> N3\n	在 Port1 初始化模块 3	1, 2	\$n ok \n
		#n M2306 P <u>1</u> V <u>1000</u> \n	设置报告时间间隔 (ms)	1, 2	@11 P <u>1</u> N3 V583\n
4	振动电机	#n M2304 P <u>3</u> \n	反初始化	3, 4, 5, 8, 9	\$n ok \n
		#n M2305 P <u>3</u> N4\n	在 Port <u>3</u> 初始化模块 4	3, 4, 5, 8, 9	\$n ok \n
		#n M2307 P <u>3</u> V1\n	V1: 启动; V0: 关闭	3, 4, 5, 8, 9	\$n ok \n
5	光线传感器	#n M2304 P <u>1</u> \n	反初始化	1, 2, 13	\$n ok \n
		#n M2305 P <u>1</u> N5\n	在 Port <u>1</u> 初始化模块 5	1, 2, 13	\$n ok \n
		#n M2306 P <u>1</u> V <u>1000</u> \n	设置报告时间间隔 (ms)	1, 2, 13	@11 P <u>1</u> N5 V583\n
6	角度传感器	#n M2304 P <u>1</u> \n	反初始化	1, 2, 13	\$n ok \n
		#n M2305 P <u>1</u> N6\n	在 Port <u>1</u> 初始化模块 6	1, 2, 13	\$n ok \n
		#n M2306 P <u>1</u> V <u>1000</u> \n	设置报告时间间隔 (ms)	1, 2, 13	@11 P <u>1</u> N6 V583\n
7	空气质量传感器	#n M2304 P <u>1</u> \n	反初始化	1, 2, 13	\$n ok \n
		#n M2305 P <u>1</u> N7\n	在 Port <u>1</u> 初始化模块 7	1, 2, 13	\$n ok \n
		#n M2306 P <u>1</u> V <u>1000</u> \n	设置报告时间间隔 (ms)	1, 2, 13	@11 P <u>1</u> N7 V583\n
8	声音传感器	#n M2304 P <u>1</u> \n	反初始化	1, 2, 13	\$n ok \n
		#n M2305 P <u>1</u> N8\n	在 Port <u>1</u> 初始化模块 8	1, 2, 13	\$n ok \n
		#n M2306 P <u>1</u> V <u>1000</u> \n	设置报告时间间隔 (ms)	1, 2, 13	@11 P <u>1</u> N8 V583\n
9		#n M2304 P <u>0</u> \n	反初始化	0	\$n ok \n

	6-轴加速度计 & 指南针	# <u>n</u> M2305 P <u>0</u> N9\n	在 Port <u>0</u> 初始化模块 9	0	\$ <u>n</u> ok \n
		# <u>n</u> M2306 P <u>0</u> V <u>1000</u> \n	设置报告时间间隔 (ms) XYZ 是每个轴的旋转角度。 H 是地磁北极与 x-轴顺时针夹角。 T 是地磁北极与正 x-轴在水平面的投影的顺时针夹角	0	@11 P <u>0</u> N9 X2.0 Y2.0 Z2.0 H2.0 T2.0\n
10	颜色传感器	# <u>n</u> M2304 P <u>0</u> \n	反初始化	0	\$ <u>n</u> ok \n
		# <u>n</u> M2305 P <u>0</u> N10\n	在 Port <u>0</u> 初始化模块 10	0	\$ <u>n</u> ok \n
		# <u>n</u> M2306 P <u>0</u> V <u>1000</u> \n	设置报告时间间隔 (ms)	0	@11 P <u>0</u> N10 R218 G31 B128\n
11	手势传感器	# <u>n</u> M2304 P <u>0</u> \n	反初始化	0	\$ <u>n</u> ok \n
		# <u>n</u> M2305 P <u>0</u> N11\n	在 Port <u>0</u> 初始化模块 11	0	\$ <u>n</u> ok \n
		# <u>n</u> M2306 P <u>0</u> V <u>1000</u> \n	设置报告时间间隔 (ms)	0	@11 P <u>0</u> N11 V1\n 1: 右 2: 左 4: 上 8: 下 16: 前进 32: 后退 64: 顺时针 128: 逆时针
12	超声波传感器	# <u>n</u> M2304 P <u>3</u> \n	反初始化	4, 8, 9	\$ <u>n</u> ok \n
		# <u>n</u> M2305 P <u>3</u> N12\n	在 Port <u>3</u> 初始化模块 12	4, 8, 9	\$ <u>n</u> ok \n
		# <u>n</u> M2306 P <u>3</u> V1000\n	设置报告时间间隔 (ms)	4, 8, 9	@11 P <u>3</u> N12 V4\n Value in cm
13	风扇	# <u>n</u> M2304 P <u>4</u> \n	反初始化	4, 8, 9	\$ <u>n</u> ok \n
		# <u>n</u> M2305 P <u>4</u> N13\n	在 Port <u>4</u> 初始化模块 13	4, 8, 9	\$ <u>n</u> ok \n
		# <u>n</u> M2307 P <u>4</u> V120\n	设置风扇速度(0~255)	4, 8, 9	\$ <u>n</u> ok \n

14	电磁开关	#n M2304 P <u>3</u> \n	反初始化	3, 4, 5, 8, 9	\$n ok \n
		#n M2305 P <u>3</u> N14\n	在 Port <u>3</u> 初始化模块 14	3, 4, 5, 8, 9	\$n ok \n
		#n M2307 P <u>3</u> V1\n	1:开启 0: 关闭	3, 4, 5, 8, 9	\$n ok \n
15	温度 & 湿度传感器	#n M2304 P <u>0</u> \n	反初始化	0	\$n ok \n
		#n M2305 P <u>0</u> N15\n	在 Port <u>0</u> 初始化模块 15	0	\$n ok \n
		#n M2306 P <u>0</u> V <u>1000</u> \n	设置报告时间间隔 (ms)	0	@11 P <u>0</u> N15 T23.3 H82.2\n
16	红外传感器	#n M2304 P <u>3</u> \n	反初始化	3, 4, 5, 8, 9	\$n ok \n
		#n M2305 P <u>3</u> N16\n	在 Port <u>3</u> 初始化模块 16	3, 4, 5, 8, 9	\$n ok \n
		#n M2306 P <u>3</u> V1000\n	设置报告时间间隔 (ms)	3, 4, 5, 8, 9	@11 P <u>3</u> N16 V1\n 1: 检测到运动 0: 未检测到运动
17	1602 LCD	#n M2304 P <u>0</u> \n	反初始化	0	\$n ok \n
		#n M2305 P <u>0</u> N17\n	在 Port <u>0</u> 初始化模块 17	0	\$n ok \n
		#n M2307 P <u>0</u> R128 G120 B10\n	设置背光颜色	0	\$n ok \n
		#n M2307 P <u>0</u> T <u>0</u> \n	0: 关掉屏幕 1: 开启屏幕 2: 清除	0	\$n ok \n
		#n M2307 P <u>0</u> V <u>0</u> S <u>Test</u> \n	V(0~1):行选择 S: 显示字符串	0	\$n ok \n
18	行定位计	#n M2304 P <u>3</u> \n	反初始化	3, 4, 5, 8, 9	\$n ok \n
		#n M2305 P <u>3</u> N18\n	在 Port <u>3</u> 初始化模块 18	3, 4, 5, 8, 9	\$n ok \n
		#n M2306 P <u>3</u> V1000\n	设置报告时间间隔 (ms)	3, 4, 5, 8, 9	@11 P <u>3</u> N18 V1\n 0: 检测到物体 1: 未检测到物体
19		#n M2304 P <u>3</u> \n	反初始化	3, 4, 5, 8, 9	\$n ok \n



	反射式红外传感器	# <u>n</u> M2305 P <u>3</u> N19\n	在 Port <u>3</u> 初始化模块 19	3, 4, 5, 8, 9	\$ <u>n</u> ok \n
		# <u>n</u> M2306 P <u>3</u> V1000\n	设置报告时间间隔 (ms)	3, 4, 5, 8, 9	@11 P <u>3</u> N19 V1\n 1: 检测到物体 0:未检测到物体
20	EMG 探测器	# <u>n</u> M2304 P <u>1</u> \n	反初始化	1, 2, 13	\$ <u>n</u> ok \n
		# <u>n</u> M2305 P <u>1</u> N20\n	在 Port <u>1</u> 初始化模块 20	1, 2, 13	\$ <u>n</u> ok \n
		# <u>n</u> M2306 P <u>1</u> V <u>1000</u> \n	设置报告时间间隔 (ms)	1, 2, 13	@11 P <u>1</u> N20 V583\n

uArm Swift Pro 的各种模式：

M2400 S0 : 常规模式 (末端执行器: 吸盘)

M2400 S1 : 激光雕刻模式 (末端执行器: 激光头)

M2400 S2 : 3D 打印模式 (末端执行器: 3D 打印头)

M2400 S3 : 通用笔夹模式 (末端执行器: 通用笔夹)

电动夹子由于可以水平旋转所以没有专门设定模式

# uArm 社区

[UFACTORY 官方论坛](#)

[uArm Facebook](#)

[uArm 技术支持](#)

[产品视频](#)



UF 微信公众号

## 发布记录

版本	更新	
1.0.0	发布文档	Tony
1.0.1	发布工作范围	Tony
1.0.2	增加各个部分的安装尺寸，增加 Arduino 上载的详细步骤。	Tony
1.0.3	增加左右电机跟机械臂上下臂的关联	Tony
1.0.4	改变了若干 Gcode 指令	David
1.0.5	修正若干 Gcode 指令	Daniel
1.0.6	变更了切换第二 UART 的内容 增加了 uArm Swift Pro V1.1 底座的尺寸图	Daniel