

ROS1-单臂R5Pro-SDK



一、硬件配置

1、硬件清单

类目	型号	数量
机械臂	R5Pro	1
机械臂底座	/	1
USB2CAN	CAN	1

电源	DC24V	1
G型夹	G型	2
螺丝扳手	m3	若干



本体



底座



USB2CAN



电源

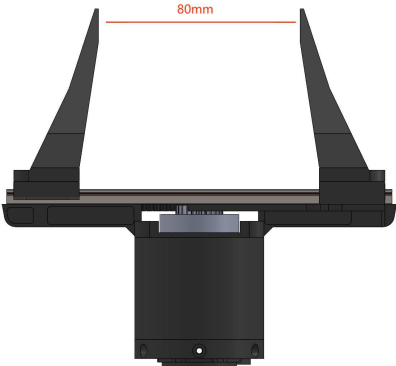
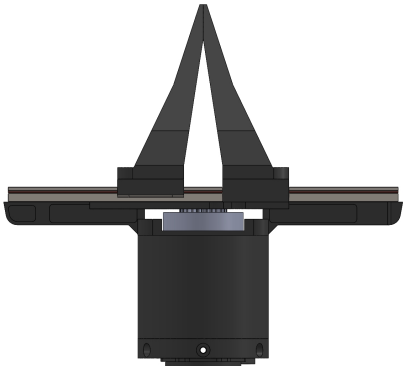


G型夹



螺丝扳手

夹持端参数



夹持范围	0-80mm
反馈及控制方式	位置 速度 扭矩
末端接口	集成机械臂（xt30 2+2）
最大夹持力	10NM
重量	约585g

2、整体组装



连接底座



G型夹固定

将底座与机械臂本体，通过螺丝连接，并通过G型夹将机械臂固定在桌子边缘。



右侧接口



连接

如右图所示连接电源、can模块和机械臂。

将电源插入插排，将usb线插入电脑。

确保机械臂为折叠的状态，确保接口在机械臂右侧



开始配置软件。

二、软件配置

1、环境配置

注意一定按照安装顺序

ROS1-noetic安装：ubuntu系统20.04 推荐鱼香ROS安装

```
1  wget http://fishros.com/install -O fishros && . fishros
```

配置can环境

```
1  配置can
2  sudo apt install can-utils
3  sudo apt install net-tools
```

编译：

```
1  |—— ARX_CAN  #设置CAN（全局适用）
```

```

2 | | | arx_can
3 | | | arx_can.rules
4 | | | can.sh
5 | | | search.sh
6 | | | set.sh
7 | | py
8 | | | ARX_R5Pro_python    #python SDK
9 | | ROS
10 | | | R5_Pro_ws          #ros1 SDK
11 | | | | src
12 | | | | | ARX_R5_ros
13 | | | | | | ARX_R5Pro_ros
14 | | | | | | | arx_msg_ros
15 | | ROS2
16 | | | R5_Pro_ws          #ros2 SDK
17 | | | | src
18 | | | | | ARX_R5_ros2
19 | | | | | | ARX_R5Pro_ros2
20 | | | | | | | arx_msg_ros2
21

```

在ARX_R5Pro/ROS/R5_Pro_ws/目录下打开终端，执行。

```

1  #编译
2  catkin_make

```

此时一个完整的ros项目就搭建完成了。

2、启动机械臂

第一步：开启CAN

参考文档：配置CAN手册。

第二步、启动机械臂

运行

```

1  #在工作空间，即R5_Pro_ws文件夹中
2  #每次打开新终端都要运行
3  source devel/setup.bash
4

```

```

5  #按Tab 可自动补全
6  roslaunch arx_r5pro_controller open_single_arm.launch
7  #终端需要“ Ctrl+c”关闭，不可以直接关闭

```

或者

新开终端（关闭其他程序）启动按键控制：

```

1  #每次打开新终端都要运行
2  source devel/setup.bash
3  #按Tab 可自动补全
4  #键盘启动
5  roslaunch arx_r5pro_controller open_keyboard_control.launch
6  #再打开新终端运行
7  source devel/setup.bash
8  #按Tab 可自动补全
9  rosrn arx_r5pro_controller R5ProKeyBoard
10 #ctrl c或 ctrl x关闭程序
11

```

键位映射：

		W	前			R	复位					I	重力补偿	O	张开			
A	左	S	后	D	右									L	pit+			
				C	闭合			N	roll-	M	roll+	,	yaw-	.	pit-	/	yaw+	UP
																	LEFT	DOW

三、操作方式及注意事项

1、控制说明

```

1  //运行相应的节点
2  roslaunch arx_r5pro_controller open_single_arm.launch
3  //查看如下话题

```



```

arx-gaoqing@arx-gaoqing:~/work/w2/ts_new_SDK/ts_aloha$ source devel/setup.bash
arx-gaoqing@arx-gaoqing:~/work/w2/ts_new_SDK/ts_aloha$ rostopic list
/arm_cmd
/arm_status_ee
/arm_status_joint
/rosout
/rosout_agg

```

topic 名称	作用
/arm_status_ee	sdk 发布机械臂末端位姿信息
/arm_status_joint	sdk 发布机械臂关节信息
/arm_cmd	sdk 订阅控制信息

通过话题控制机械臂：

在相同目录下再开一个终端，运行：

```

1  source ./devel/setup.bash
2
3  #使用rqt
4  rqt
5
6  #Plugins->Topics->message publisher->选择/arm_cmd->点击右侧加号->勾选话题前的“方
   块”

```

```

1  //单位：米、弧度
2  //[ x y z ]:末端位置
3  //[roll pitch yaw]:末端姿态
4  float64 x //末端位置 前后 范围:[0, 0.5]
5  float64 y //末端位置 左右 范围:[-0.5, 0.5]
6  float64 z //末端位置 上下 范围:[0.5, 0.5]
7  float64 roll //末端roll 正负2.1弧度
8  float64 pitch //末端pitch 正负1.3弧度
9  float64 yaw //末端yaw 正负1.3弧度
10 float64 gripper //夹爪开合 0-5 对应 0-80mm
11 float64 quater_x //四元数 预留位
12 float64 quater_y
13 float64 quater_z
14 float64 quater_w
15 float64 chx //底盘前后
16 float64 chy //底盘左右

```

```

17 float64 chz //底盘旋转
18 float64 vel_l //预留位
19 float64 vel_r //预留位
20 float64 height //高度
21 float64 head_pit //头部俯仰
22 float64 head_yaw //头部左右
23 float64[6] tempFloatData //VR链路
24 int32[6] tempIntData //VR链路
25 int32 mode1 //VR链路
26 int32 mode2 //VR链路
27 int32 timeCount //VR链路
28

```

关节限位：

关节	1	2	3	4	5	6
范围(弧度)	[-3.14, 2.6]	[-3.6, 0.1]	[-1.57, 1.57]	[-1.3, 1.3]	[-1.57, 1.57]	[-2.1, 2.1]

查看机械臂的状态：

在相同目录下再开一个终端，运行：

```

1 source devel/setup.bash
2 #table 可自动补全
3 rostopic echo /arm_status_ee
4
5 #按下 Ctrl+C 退出终端

```

2、注意事项

关闭终端前务必先输入：

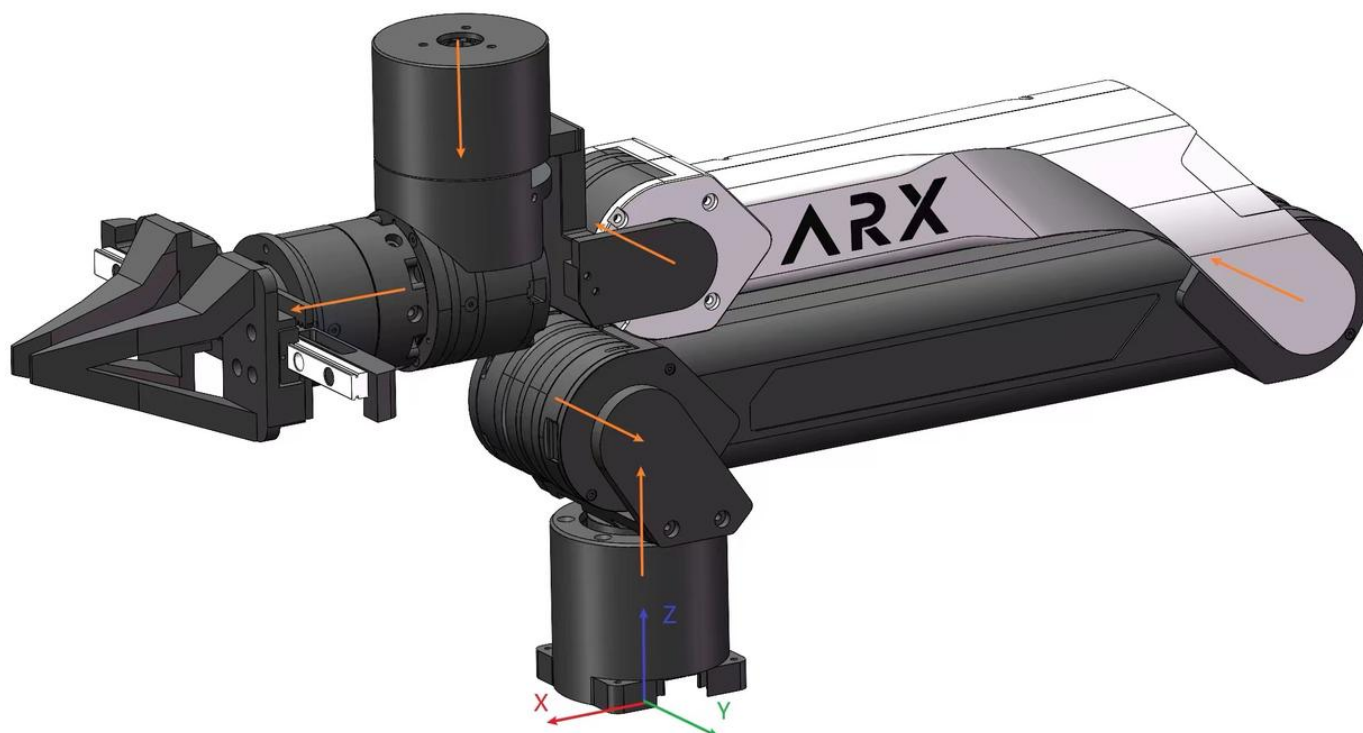
```
1 Ctrl + c
```

不可直接关闭终端，若不正常退出且出现异常，应该重启电脑，关闭后台的线程。

串联机械臂，奇异位置是不可避免的，尽量不要在工作空间边缘进行控制。

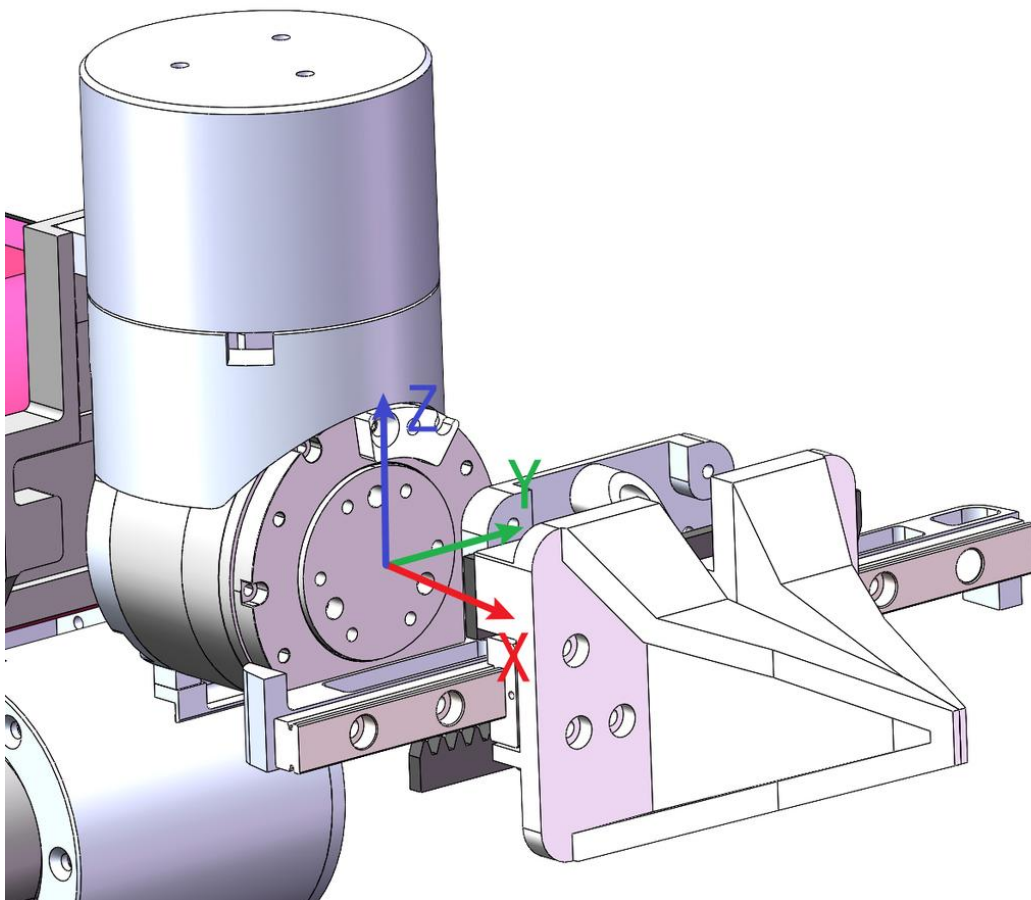
当关节超限时，机械臂会停止运动。

机械臂各个关节轴向



不同型号的机械臂，其关节的轴向都是相同的。关节转向符合右手定理，大拇指的指向关节轴向，四指方向就是电机转动的正方向。

末端坐标系



在初始位置，末端坐标系和参考坐标系重合，位置和姿态都是0，如上图所示。

异常处理

机械臂垂落，无法控制	终端是否提示safe mode（碰撞检测进入保护模式，断电复位，重启即可）
某个can口打不开	检查can连接，重新插拔对应的usb，重新开启can。
电机无法连接	重新插拔机械臂底座的插头