

ROS2-单臂R5Pro-SDK

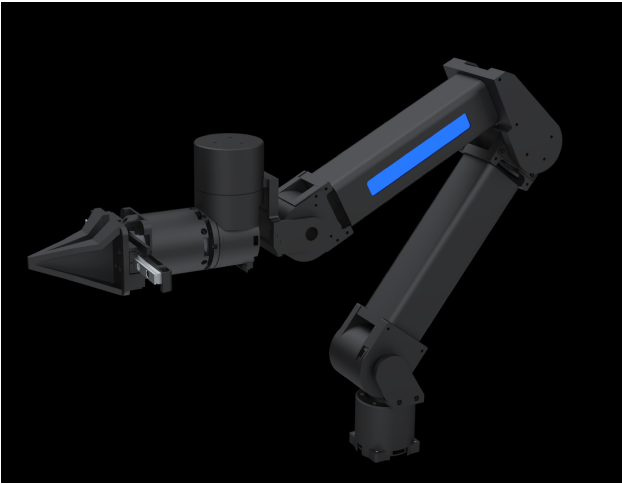


一、硬件配置

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

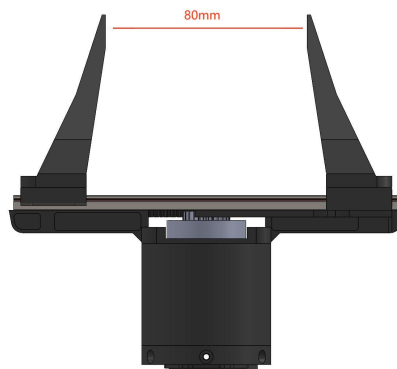
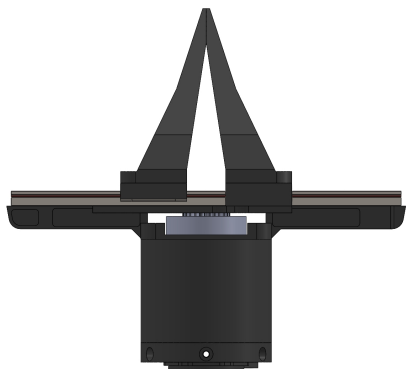


G型夹



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

ROS2-humble安装：ubuntu系统22.04 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/ROS2/R5_Pro_ws/目录下打开终端，执行。

```
1 colcon build
```

此时一个完整的ros2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

在ARX_R5Pro/ROS2/R5_Pro_ws/目录下，运行

```
1  #每次打开新终端都要运行
2  source ./install/setup.bash
3
4  #运行相应的节点
5  ros2 launch arx_r5pro_controller open_single_arm.launch.py
```

如需终结程序，终端需要“Ctrl+c”关闭，不可以直接关闭。

新开终端启动按键控制：

```
1  #每次打开新终端都要运行
2  source ./install/setup.bash
3  ros2 run arx_r5pro_controller KeyBoard
```

终端需要“Ctrl+c”关闭，不可以直接关闭。

按键控制的键位映射:

[illegible]

三、操作方式及注意事项

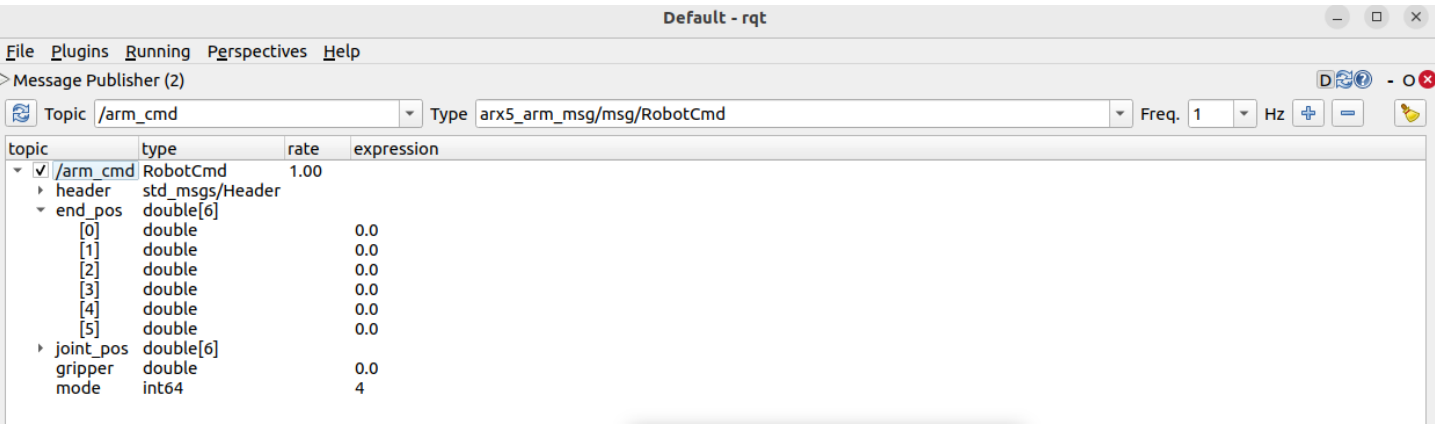
1、控制说明

topic 名称	作用
/arm_status	sdk 发布机械臂信息
/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 ./install/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”
```



变量名称	作用	备注
end_pos	末端位姿	xyz + rpy
joint_pos	关节位置	六个关节
gripper	夹爪	
mode	控制模式	

mode	模式功能	备注
0	力矩清零	所有关节力矩为0
1	机械臂复位	回到初始位形
2	阻尼模式	在“0”的基础上增加阻尼
3	重力补偿	可任意拖动
4	末端位姿控制	通过“end_pos”控制
5	关节控制	通过“joint_pos”控制

下面为PosCmd.mg的内容

```
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1.3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
```

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]	[-1.4, 1.4]	[-1.57, 1.57]	[-1.57, 1.57]

注意，只有在对应的mode下，对应的变量才会其作用。其中gripper在任何模式下都可以控制夹爪

查看机械臂的状态：

在相同目录下再开一个终端，运行：

```
1 source ./install/setup.bash
2
```



```
3 ros2 topic echo /arm_status
```

```
header:
  stamp:
    sec: 1732473587
    nanosec: 335993885
  frame_id: ''
end_pos:
- -0.00047054141759872437
- 0.00015679412172175944
- 0.004743590950965881
- -0.0018394031794741748
- -0.024989567697048184
- 0.004960644990205765
joint_pos:
- 0.00019073486328125
- 0.00209808349609375
- 0.00934600830078125
- 0.01773834228515625
- -0.00476837158203125
- -0.00171661376953125
- -0.038338661193847656
joint_vel:
- -0.010990142822265625
- -0.010990142822265625
- -0.03296661376953125
- 0.03296661376953125
- -0.03296661376953125
- -0.010990142822265625
- -0.010990142822265625
joint_cur:
- -0.07472610473632812
- 0.9010982513427734
- 3.9692306518554688
- 2.0263729095458984
- 0.17142868041992188
- 0.0043964385986328125
- -0.9186820983886719
---
```

变量名称	作用	备注
end_pos	末端位姿	xyz + rpy
joint_pos	关节位置	六个关节+夹爪
joint_vel	关节速度	六个关节+夹爪
joint_cur	关节力矩	六个关节+夹爪

2、注意事项

关闭终端前务必先输入：

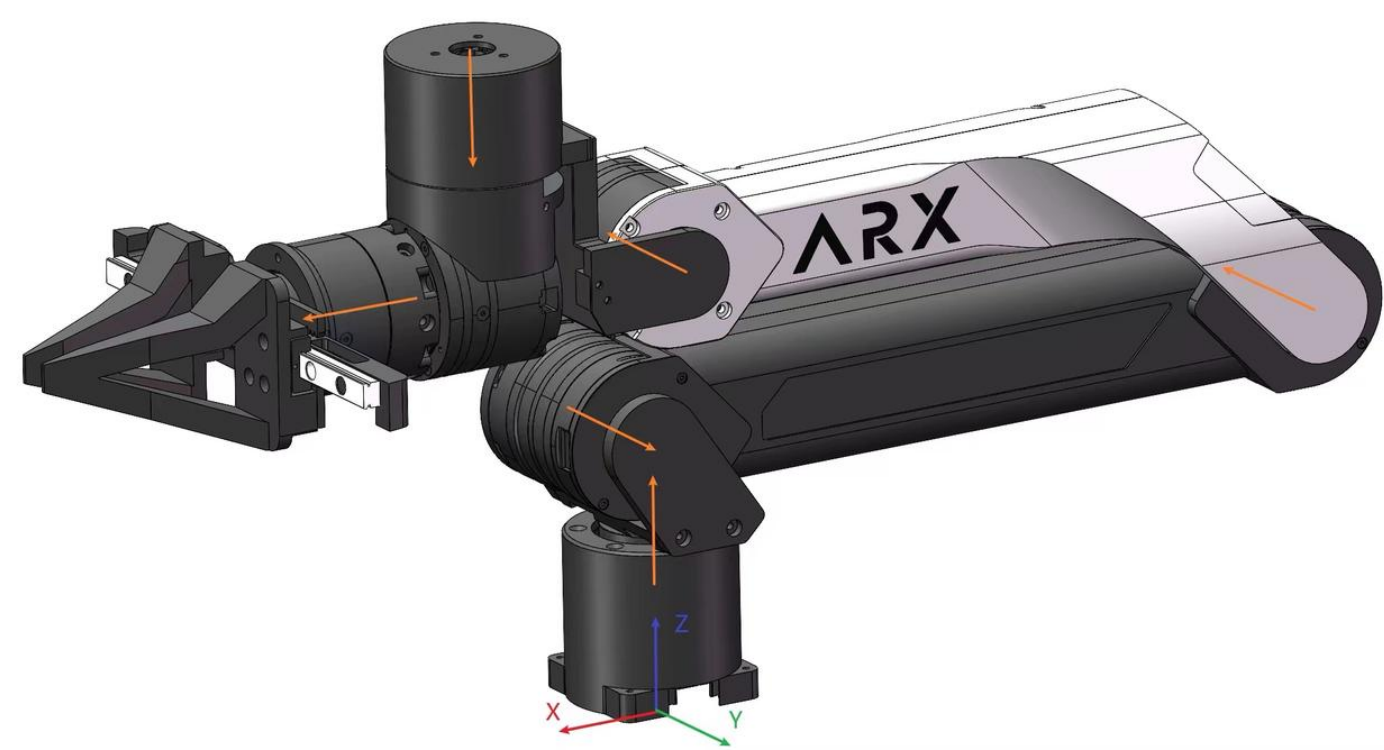
```
1 Ctrl + c
```

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。

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