

# ROS1-单臂R5-SDK



## 一、硬件配置

### 1、硬件清单

| 类目      | 型号  | 数量 |
|---------|-----|----|
| 机械臂     | R5  | 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

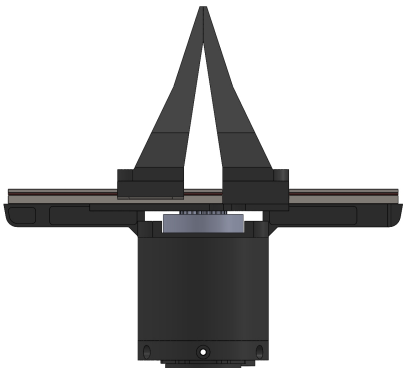


G型夹



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_R5_python    #python SDK
9
10 |— ROS                  #ros1 SDK
11 |   |— R5_ws
12 |       |— src
13 |           |— ARX_R5_Remote_SDK_ROS    #遥操作包
14 |           |— arx_r5_ros              #单臂包
15 |— ROS2                  #ros2 SDK
16 |   |— R5_ws
17 |       |— src
18 |           |— ARX_R5_Remote_SDK_ROS2    #遥操作包
19 |           |— ARX_R5_ros2              #单臂包

```

在ARX\_R5/ROS/R5\_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，因为遥操作包和单臂包都在一个工作空间（R5\_ws）中，所以这两个包都会编译。

## 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\_R5/ROS/R5\_ws/目录下，运行

```

1 #在工作空间，即ros_ws文件夹中
2 #每次打开新终端都要运行
3 source devel/setup.bash
4 #不要忘记运行ros核心
5 roscore

```



```

6
7 #运行相应的节点
8 rosrun arx_r5_controller R5Controller

```

```

nuc@nuc-04: ~/Q_work/ts_new_sdk/ros1_ws
[ INFO] [1732329124.796636050]: Publishing RobotStatus message
[ INFO] [1732329124.806627791]: Publishing RobotStatus message
[ INFO] [1732329124.816657259]: Publishing RobotStatus message
[ INFO] [1732329124.826666749]: Publishing RobotStatus message
[ INFO] [1732329124.836669778]: Publishing RobotStatus message
[ INFO] [1732329124.846681198]: Publishing RobotStatus message
[ INFO] [1732329124.856721309]: Publishing RobotStatus message
[ INFO] [1732329124.866663801]: Publishing RobotStatus message
[ INFO] [1732329124.876688845]: Publishing RobotStatus message
[ INFO] [1732329124.886676889]: Publishing RobotStatus message
[ INFO] [1732329124.896666980]: Publishing RobotStatus message
[ INFO] [1732329124.906688854]: Publishing RobotStatus message

```

新开终端启动按键控制：

```

1 #每次打开新终端都要运行
2 source devel/setup.bash
3 rosrun arx_r5_controller KeyBoard

```

终端需要“Ctrl+c”关闭，不可以直接关闭。

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

| ros话题说明   |                |            |               |
|-----------|----------------|------------|---------------|
|           | 节点             | topic 名称   | 作用            |
| 通过话题控制机械臂 | R5Controller   | /r5_status | 发布机械臂信息       |
|           | R5Controller   | /r5_cmd    | 订阅控制信息        |
| 通过VR控制机械臂 | R5ControllerVr | /r5_status | 发布机械臂信息       |
|           | R5ControllerVr | /ARX_VR_L  | 订阅控制信息（VR左手柄） |

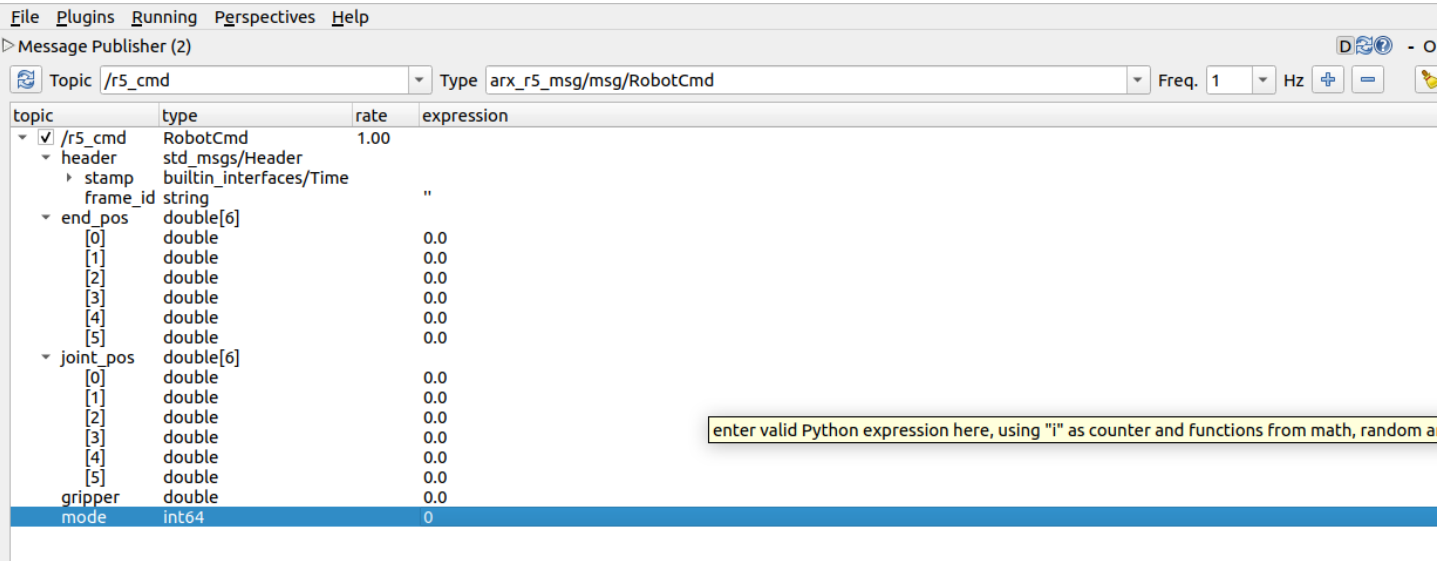
通过话题控制机械臂：

在运行 “ rosrun arx\_r5\_controller r5Controller ” 后，在相同目录下再开一个终端，运行：

```
1 source devel/setup.bash
2
3 #按tab可自动补全
4 rostopic pub /r5_cmd arx_r5_msg/RobotCmd
5 #或者使用
6 rqt
7 #Plugins->Topics->message publisher->选择/r5pro_cmd->点击右侧加号->勾选话题前的“方
  块”
```

```
nuc@nuc-04:~/Q_work/ts_new_sdk/ros1_ws$ rostopic pub /r5_cmd arx_r5_msg/RobotCmd
"header:
  seq: 0
  stamp: {secs: 0, nsecs: 0}
  frame_id: ''
end_pos: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
joint_pos: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
gripper: 0.0
mode: 0"
```

左右移动方向键，更改对应的值，回车即可控制。



topic中各个变量的作用：

| 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 | 六种模式      |

| 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 |

```
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
11
```

机械臂末端位姿坐标系的原点，在第6号电机的转子前端，也就是和夹爪电机的连接处。  
当然这个参考原点，可前移：

```
37
38     arx_arm ARX_ARM((int) CONTROL_MODE, 0.0);
39
```



更改图中原本为0的参数，就可以让坐标系原点往前移动，单位为米。

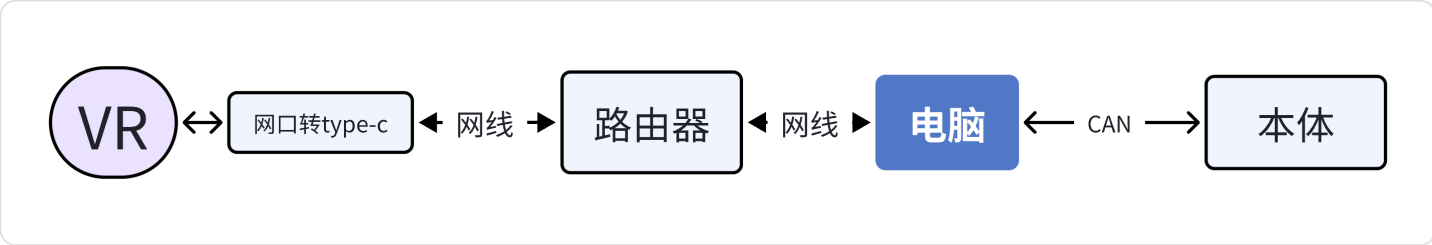
关节限位：

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

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

通过VR控制机械臂：

使用VR时需要在原本的控制机械臂的基础上，增加连接VR的硬件。



在启动机械臂的SDK之前需要先启动VR的SDK，具体可参考VR SDK中的readme文件。

通过VR控制机械臂启动的节点和上述不同，需要运行：

```
1 rosrun arx_r5_controller R5ControllerVR
```

控制VR手柄（左手）运动，机械臂就会跟随。

## 查看机械臂的状态：

在运行 “roslaunch arx\_r5\_controller R5Controller” 后, 在相同目录下再开一个终端，运行：

```
1 source devel/setup.bash
2
3 rostopic echo /r5_status
```

```
header:
  seq: 13363
  stamp:
    secs: 1732332053
    nsecs: 601799391
  frame_id: ''
end_pos: [-0.0006374642252922058, -6.500052404589951e-05, 0.0050298795104026794, -0.0014597536064684395, -0.025371035560965538, 0.002671871101483703]
joint_pos: [-0.00209808349609375, 0.00057220458984375, 0.01049041748046875, 0.01544952392578125, -0.00476837158203125, -0.00133514404296875, -0.045586585998535156]
joint_vel: [0.010990142822265625, -0.010990142822265625, -0.010990142822265625, -0.03296661376953125, 0.010990142822265625, -0.010990142822265625, 0.010990142822265625]
joint_cur: [0.15384674072265625, 1.0153846740722656, 3.828571319580078, 2.0615386962890625, 0.1802196502685547, -0.0043964385986328125, -0.9714279174804688]
---
```

| 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：

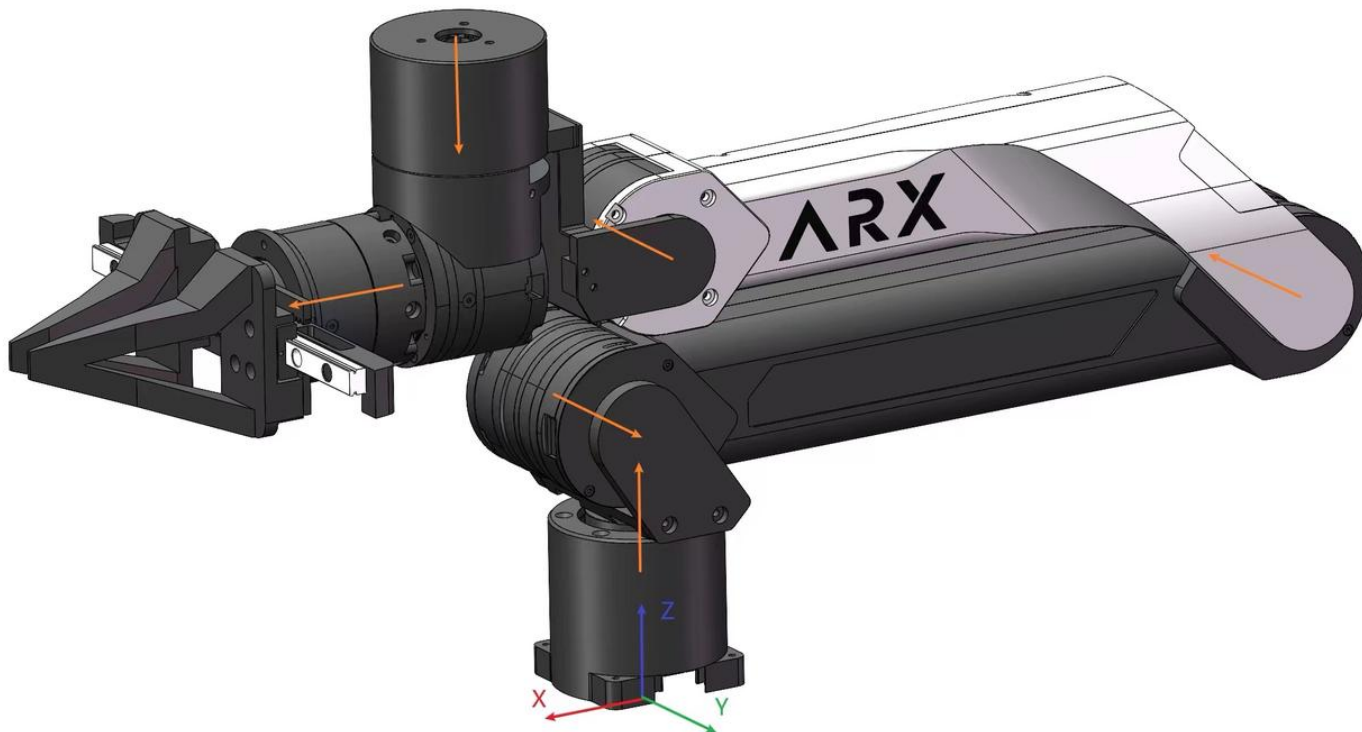
```
1 Ctrl + c
```

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                           |
| 程序一直在初始化   | 保证usb接口带宽足够，不要和usb wifi等数据量较大设备公用一个usb |