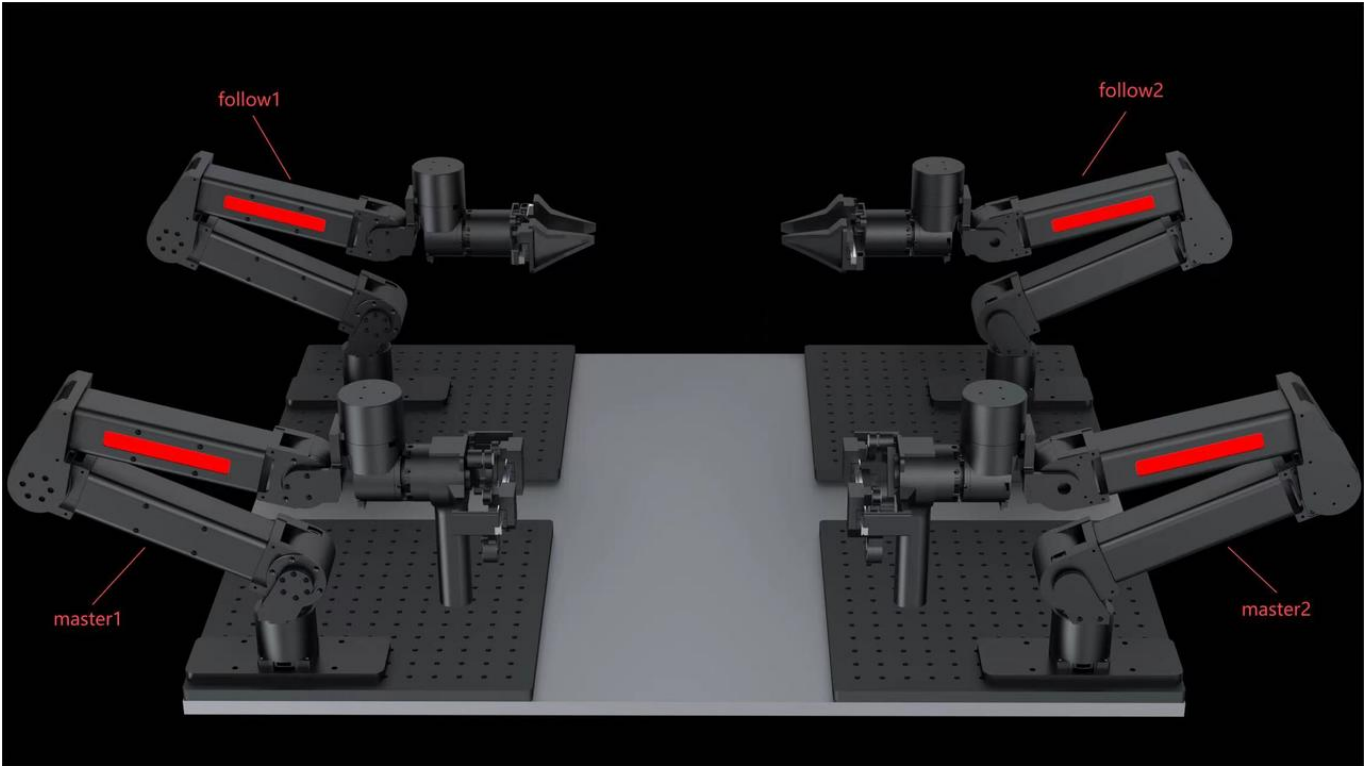


# R5-ROS2-aloha



前言：

方案1：四台机械臂，两两遥控。两台作为操作器（master，主臂），两台作为执行器（follow，从臂）

方案2：两台机械臂，配合VR。VR手柄作为操作器（master），两台机械臂作为执行器（follow）。

## 一、硬件配置

### 1、硬件清单

方案1：

类目	型号	数量	备注
机械臂主臂	ARX R5	2	末端为示教器
机械臂从臂	ARX R5	2	末端为夹爪
夹爪	ARX-Play-G	2	
示教器	ARX-Play-E	2	

笔记本电脑	Lenovo Y7000P(RTX4070)	1	
摄像头	Intel D405	3	

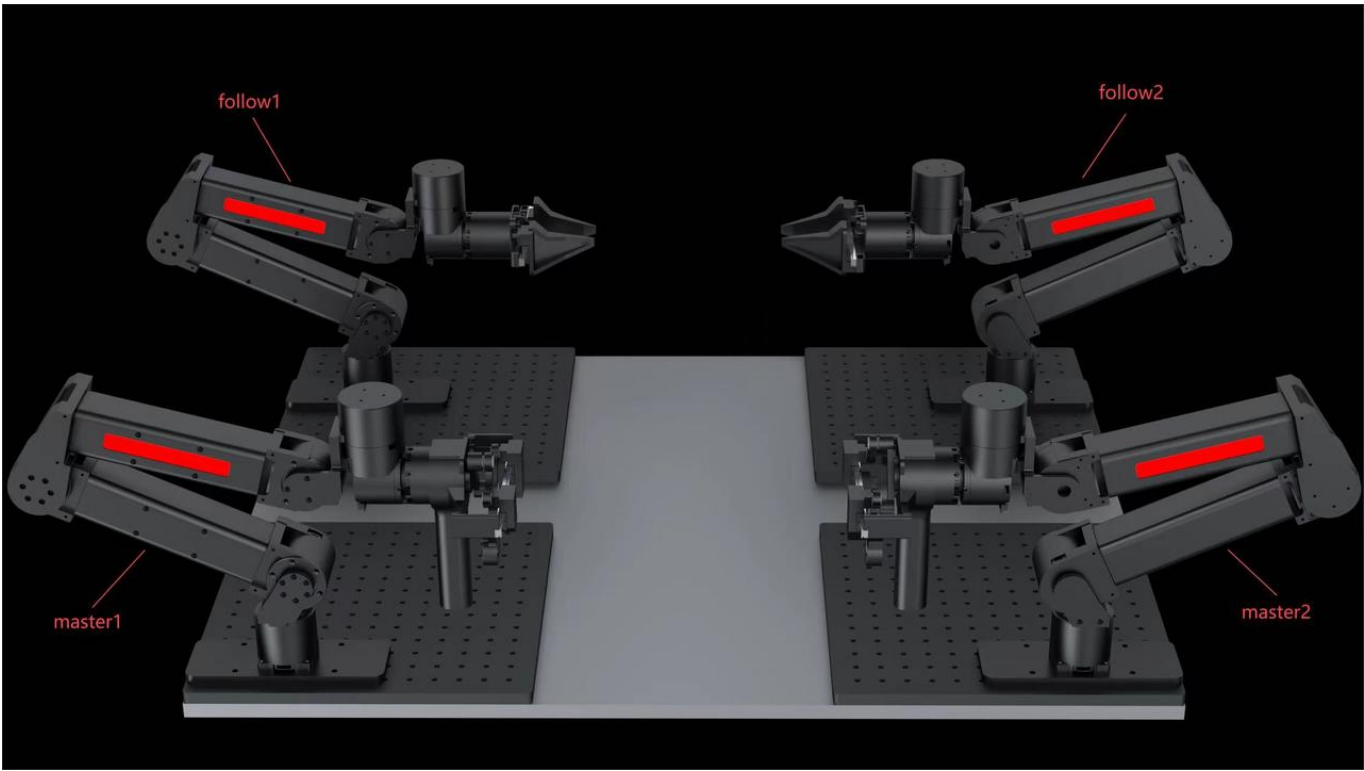
方案2:

类目	型号	数量	备注
机械臂	ARX R5	2	末端为夹爪
夹爪	ARX-Play-G	2	
VR	Quest3	套装	
路由器	AX3000	1	
笔记本电脑	Lenovo Y7000P(RTX4070)	1	
摄像头	Intel D405 ARX定制版	3	

2、整体组装

方案1:

主臂和从臂都可以按照 “ ROS2-单臂R5-SDK ” 进行安装。（方案2，无需主臂）



其中主臂需要安装示教器。

这里四台臂的的usb口先不要连接电脑！

## 方案2:

在安装了两个从臂的基础上，增加连接VR的硬件。



```
1 https://github.com/ARXroboticsX/ARX_VR_SDK.git
```

从这里下载VR的sdk，按照readme的方式配置网络和VR，先不启动VR 的sdk。

## 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

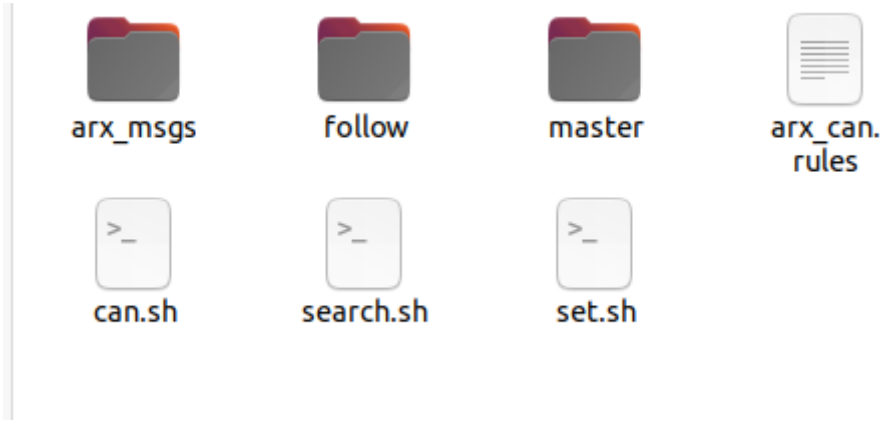
```
1 wget http://fishros.com/install -O fishros && . fishros
```

配置can环境

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

## 2、SDK文件结构

其中ARX\_R5\_Remote\_SDK\_ROS2包的文件结构如下图所示：



包名	用途
master	启动主机械臂
follow	启动从机械臂
arx_msgs	存放消息文件

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 |   | ROS
10 |     | R5_Pro_ws      #ros1 SDK
11 |       | src
12 |         | ARX_R5_ros
13 |         | arxmsgros2
14 | ROS2              #ros2 SDK
```

```

15      └─ R5_ws
16          └─ src
17              └─ ARX_R5_ros2          #单臂包
18                  └─ arxmsgros2

```

在ARX\_R5/ROS2/R5\_ws/目录下打开终端，执行。

```

1 # 回到工作空间，即R5_ws文件夹中
2 colcon build

```

此时一个完整的ros2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、启动系统

### 第一步：开启CAN

参考文档：配置CAN手册(方案2，只需要配置从臂)。

### 第二步、启动机械臂

#### 四臂遥操作

运行

```

1 // 回到工作空间，即R5_ws文件夹中
2
3 //每次开终端都要运行
4 source install/setup.bash
5
6 //启动主臂
7 ros2 launch arx_r5_controller open_remote_master.launch.py
8 //启动从臂
9 ros2 launch arx_r5_controller open_remote_slave.launch.py
10
11 //上述命令可以按下tab建自动补全，若无法补全，请检查是否执行source，以及终端所在路径是否正确

```

启动后所有机械臂的灯由红转为绿，主臂为重力补偿模式，可以随意拖动。

从臂会实时的跟踪主臂的运动。

关闭终端前务必先按下：

```
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。

话题查看,在同一个目录下，新开终端运行：

```
1 #每次开终端都要运行
2 source install/setup.bash
3
4 ros2 topic list
```

```
arx@arx:~/Q_work/ts_R5_SDK$ ros2 topic list
/arm_master_l_status
/arm_master_r_status
/arm_slave_l_status
/arm_slave_r_status
/parameter_events
/rosout
```

其中

话题	作用
/arm_master_l_status	sdk 发布左主臂信息
/arm_master_r_status	sdk 发布右主臂信息
/arm_slave_l_status	sdk 发布左从臂信息
/arm_slave_r_status	sdk 发布右从臂控制

例如：

```
1 #每次开终端都要运行
2 source install/setup.bash
3
4 #查看左臂从臂信息
5 ros2 topic echo /arm_slave_l_status
```

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]

## VR遥操作

则运行

```
1 //在工作空间，即R5_ws文件夹中
2
3 //每次开终端都要运行
4 source install/setup.bash
5
6 //以vr控制模式，启动机从臂
7 ros2 launch arx_r5_controller open_vr_double_arm.launch.py
```

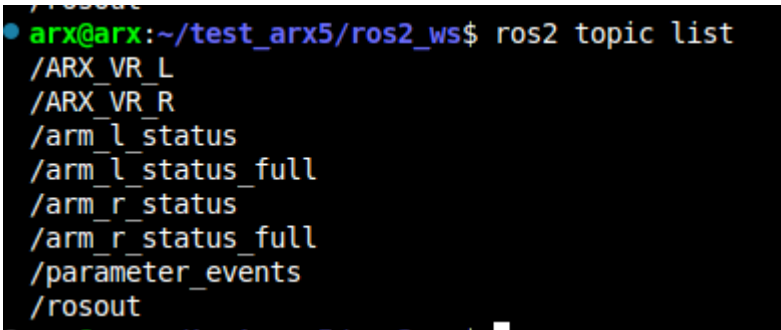
此时就以vr控制模式启动了从臂。

然后启动VR的sdk，当两个sdk都正确启动后，根据readme的说明即可控制机械臂运动了。

注意如果想要结束程序，需按下“catrl+c”退出终端。

话题查看：同一个目录下，新开终端运行：

```
1 #每次开终端都要运行
2 source install/setup.bash
3
4 ros2 topic list
```



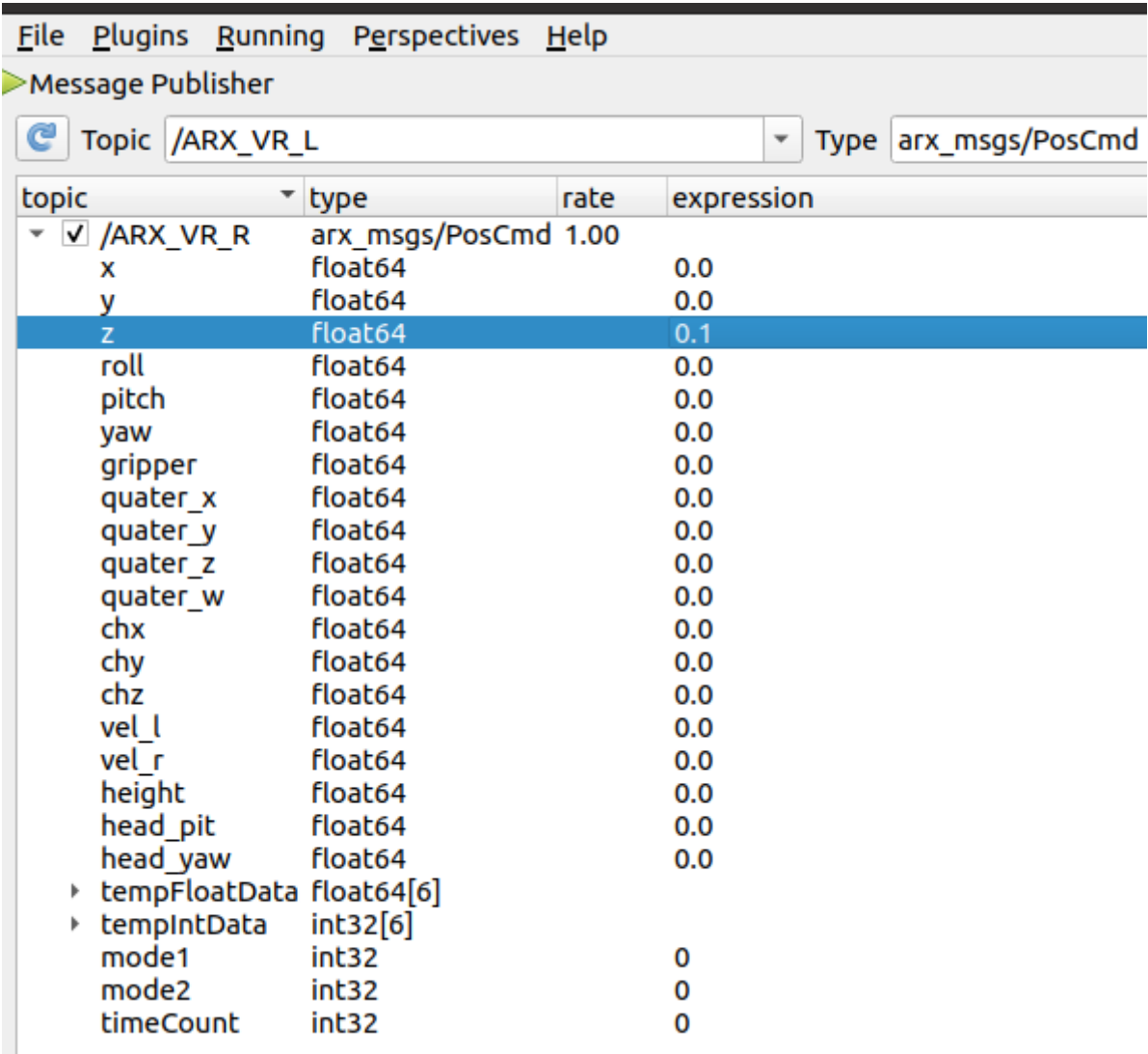
其中 “/ARX\_VR\_L” 和 “/ARX\_VR\_R” 两个话题，这是机械臂订阅的VR控制信号的话题。

话题	作用
/ARX_VR_L	sdk 订阅左手从臂末端位姿态控制
/ARX_VR_R	sdk 订阅右手从臂末端位姿态控制

手动发布上述两个话题，就可以控制机械臂的末端位姿态。

可以在终端运行rqt（与上述终端在同一路径），进行话题的控制：

```
1 #每次开终端都要运行
2 source install/setup.bash
3
4 rqt
5 #Plugins->Topics->message publisher->选择对应话题->点击右侧加号->勾选话题前的“方块”
```





其中几个控制末端位姿的变量，说明如下

```
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2.0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
```

话题	作用
/arm_l_status_full	sdk 发布左手从臂末端位姿和关节弧度
/arm_r_status_full	sdk 发布右手从臂末端位姿和关节弧度

```
1 #每次开终端都要运行
2 source install/setup.bash
3 ros2 topic echo /arm_l_status_full
```

```
---
header:
  stamp:
    sec: 1737030936
    nanosec: 870340272
  frame_id: ''
end_pos:
- -0.0006801411509513855
- 0.0001310795923927799
- 0.0030713528394699097
- -0.010122501291334627
- -0.007442252710461615
- 0.0022889352403581134
joint_pos:
- 0.00095367431640625
- -0.00133514404296875
- 0.00972747802734375
- -0.00362396240234375
- -0.00133514404296875
- -0.01010894775390625
- -0.034333229064941406
joint_vel:
- -0.010990142822265625
- -0.03296661376953125
- 0.010990142822265625
- -0.0549468994140625
- 0.010990142822265625
- -0.03296661376953125
- -0.03296661376953125
joint_cur:
- -0.11868095397949219
- 0.0659332275390625
- 3.925273895263672
- 3.2659339904785156
- 0.03076934814453125
- 0.14505577087402344
- -0.9626369476318359
---
```

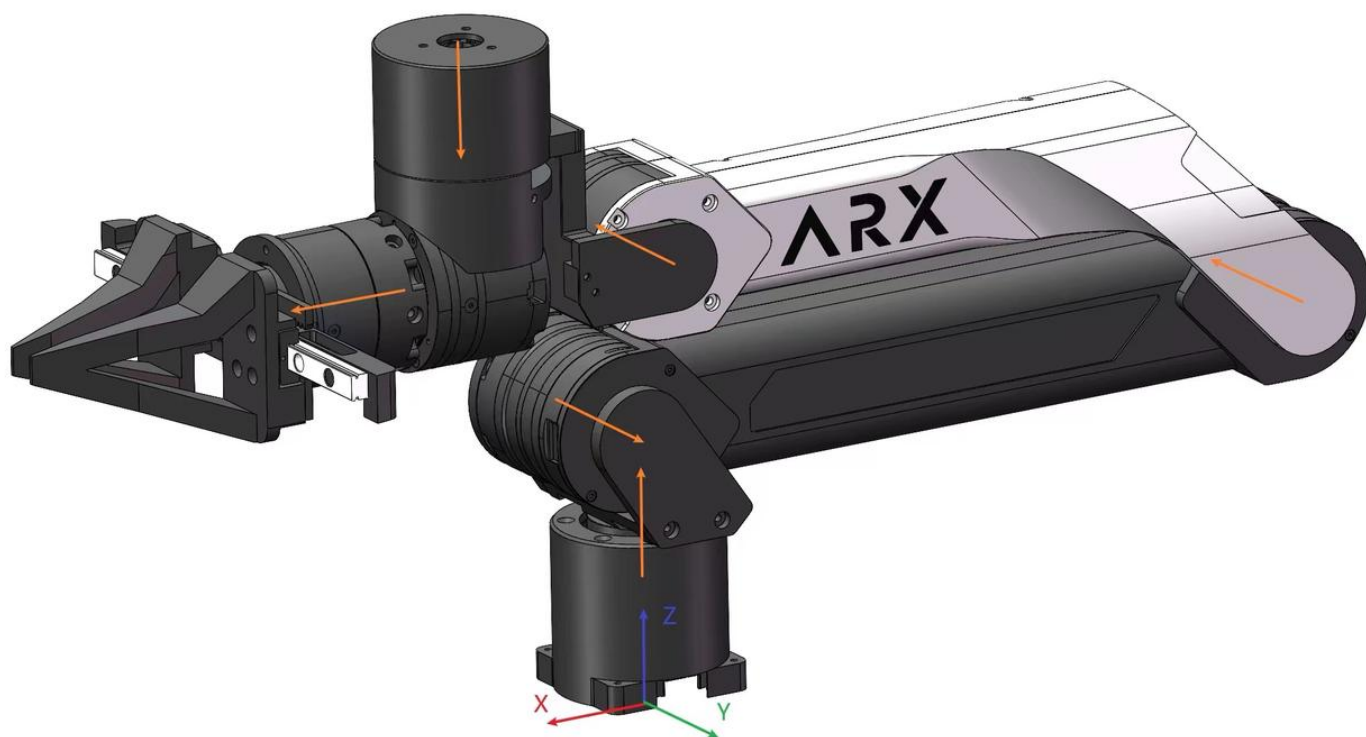
### 三、注意事项

关闭终端前务必先输入：

```
1 Ctrl + c
```

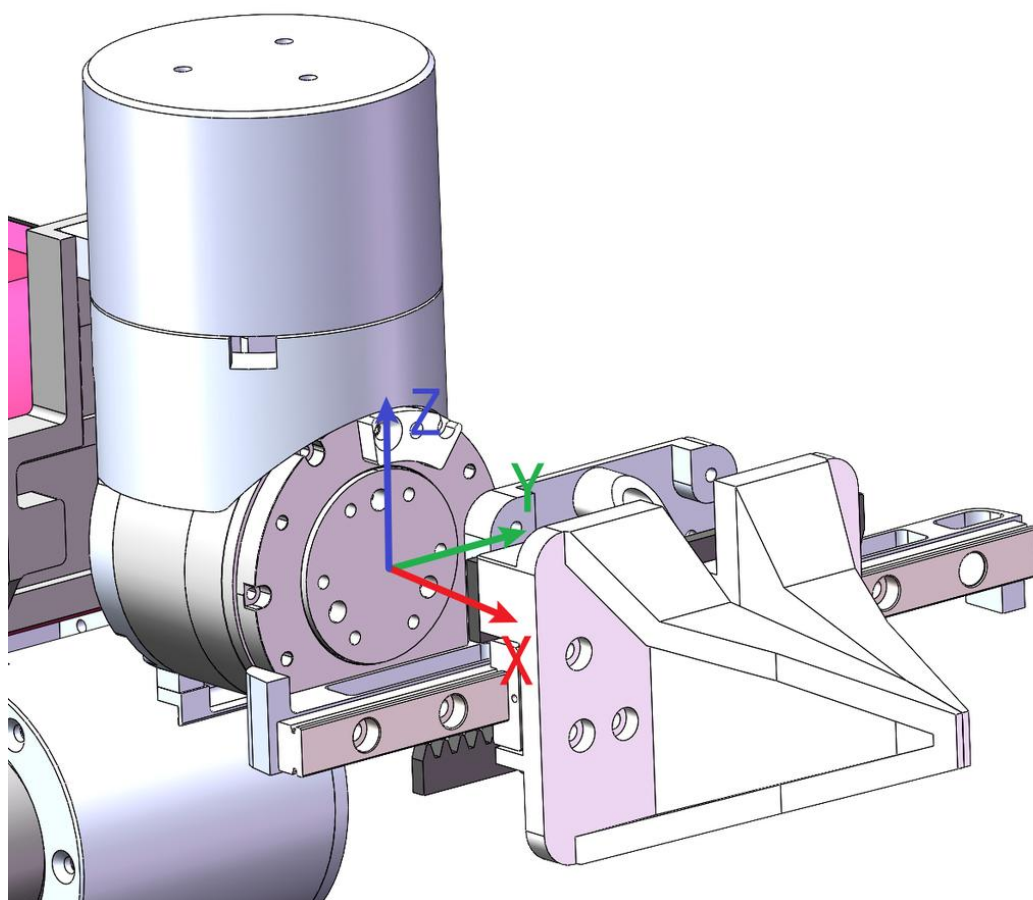
不可直接关闭终端

#### 1、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。

## 2、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