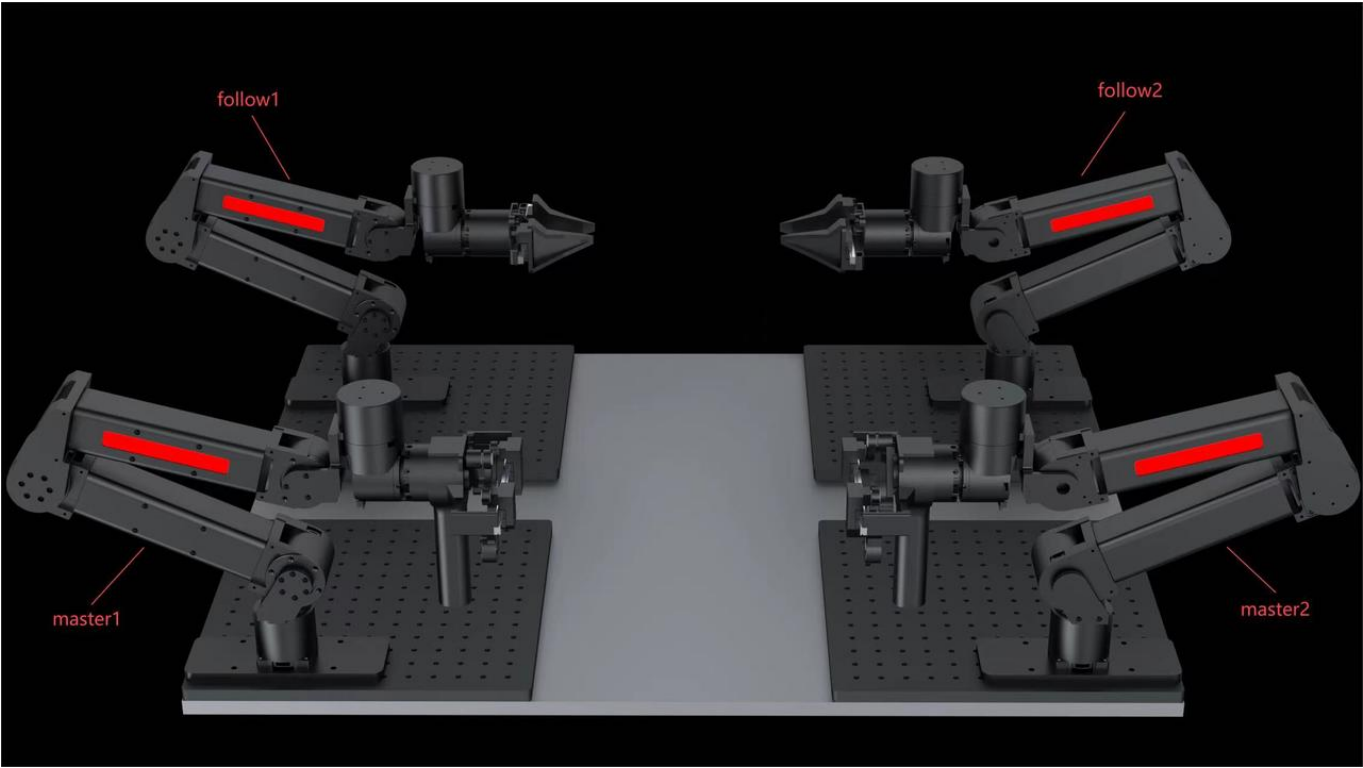


# R5-ROS-aloha



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	R5	2	末端为示教器
机械臂从臂	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	

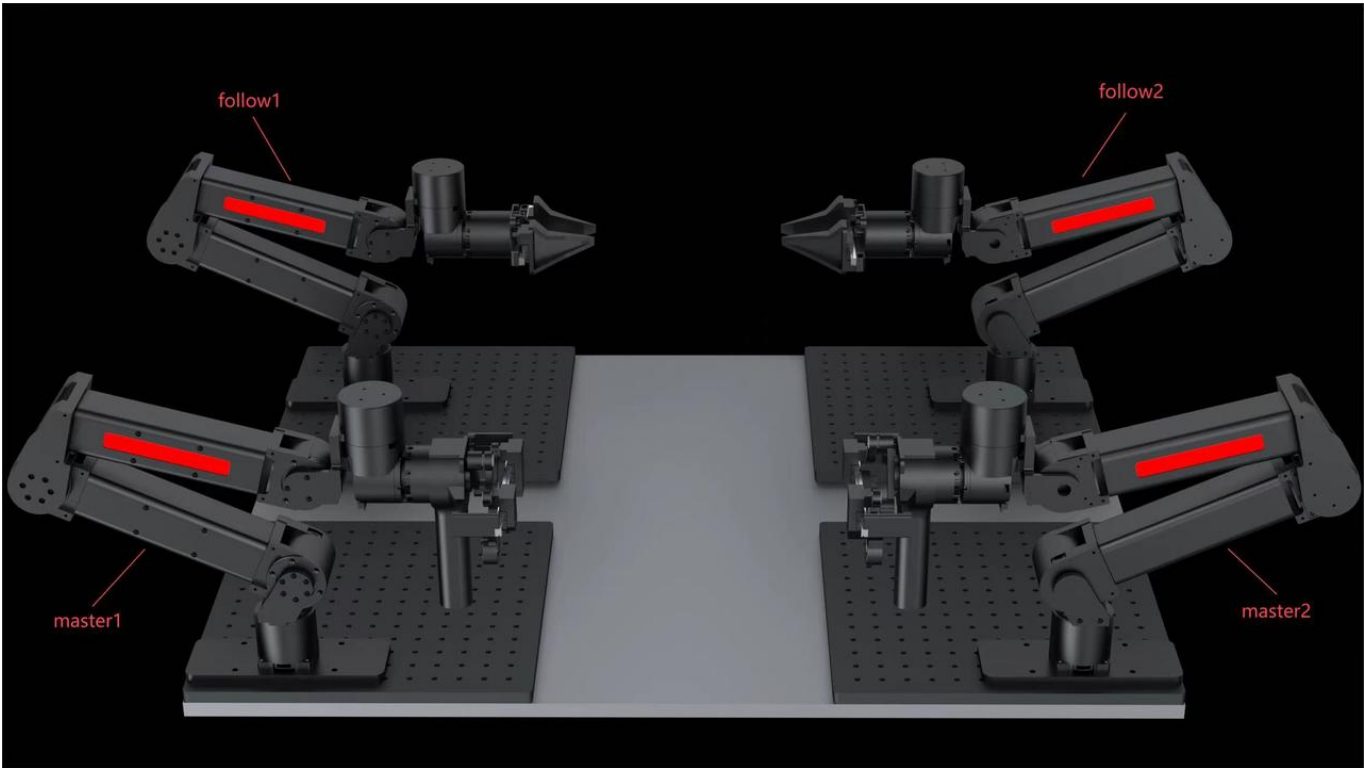
方案2:

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	
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“ROS1-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, 具体可参vr说明手册。

## 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_ros
14 |               |--- arx_msg_ros
15 |— ROS2        #ros2 SDK
16 |   |— R5_ws
17 |       |— src
18 |           |— ARX_R5_ros2
```

在ARX\_R5/ROS/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  catkin_make
```

此时一个完整的ros1项目就搭建完成了。

## 2、启动系统

### 第一步：设置CAN ID号

参考文档：配置CAN手册。

### 第二步、启动机械臂

#### 四臂遥操作

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 devel/setup.bash
5
6  //table 可自动补全
7  //启动从臂
8  roslaunch arx_r5_controller open_remote_slave.launch
9  //启动主臂
10 roslaunch arx_r5_controller open_remote_master.launch
11 //终端需要“ Ctrl+c”关闭，不可以直接关闭
```

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。

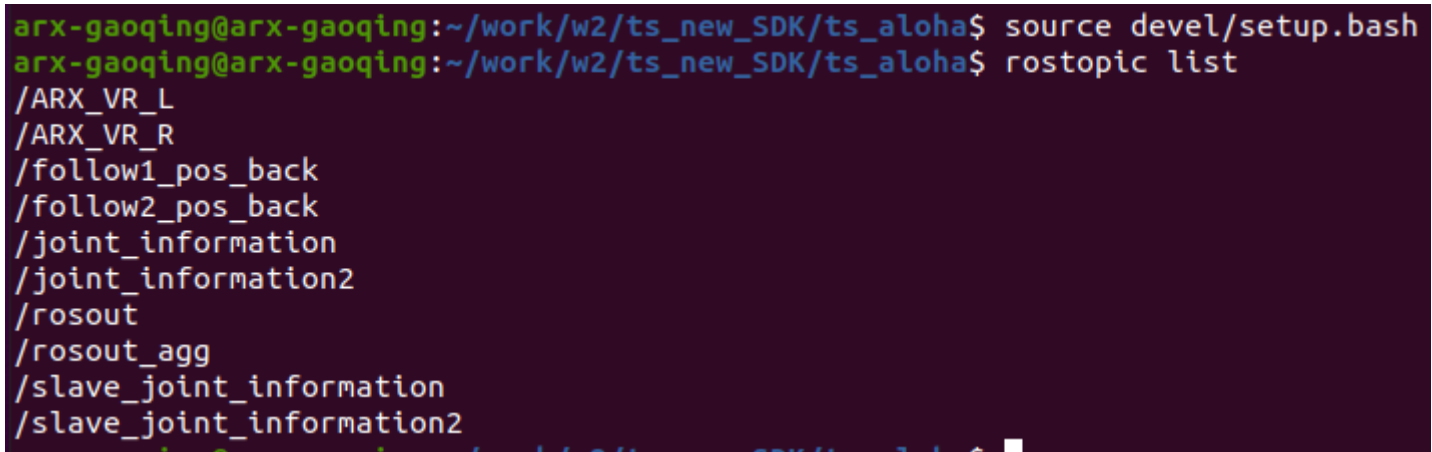
关闭终端前务必先按下：

```
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：同一个目录下，新开终端运行：

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



其中

话题	作用
/follow1_pos_back	sdk 发布左从臂末端位姿信息
/follow2_pos_back	sdk 发布右从臂末端位姿信息
/joint_information	sdk 发布左从臂关节信息
/joint_information2	sdk 发布右从臂关节控制

例如：

```
1  #每次开终端都要运行
2  source devel/setup.bash
3
4  #按Tab 可自动补全
5  #查看左臂从臂关节信息
6  rostopic echo /joint_information
```

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

先启动VR的SDK

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 devel/setup.bash
5 //按Tab 可自动补全
6 //以vr控制模式，启动机械臂
7 roslaunch arx_r5_controller open_vr_double_arm.launch
8 //终端需要“ Ctrl+c”关闭，不可以直接关闭
```

此时就以vr控制模式启动了机械臂。

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 devel/setup.bash
3
4 rostopic list
```

```
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
/ARX_VR_L
/ARX_VR_R
/follow1_pos_back
/follow2_pos_back
/joint_information
/joint_information2
/rosout
/rosout_agg
```

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 devel/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”*

File Plugins Running Perspectives Help			
Message Publisher			
Topic <input type="text" value="/ARX_VR_L"/>		Type <input type="text" value="arx_msgs/PosCmd"/>	
topic	type	rate	expression
▼ <input checked="" type="checkbox"/> /ARX_VR_R	arx_msgs/PosCmd	1.00	
x	float64		0.0
y	float64		0.0
z	float64		0.1
roll	float64		0.0
pitch	float64		0.0
yaw	float64		0.0
gripper	float64		0.0
quater_x	float64		0.0
quater_y	float64		0.0
quater_z	float64		0.0
quater_w	float64		0.0
chx	float64		0.0
chy	float64		0.0
chz	float64		0.0
vel_l	float64		0.0
vel_r	float64		0.0
height	float64		0.0
head_pit	float64		0.0
head_yaw	float64		0.0
▶ tempFloatData	float64[6]		
▶ tempIntData	int32[6]		
mode1	int32		0
mode2	int32		0
timeCount	int32		0

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2.1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

```

若需要进行模型的推理，或者通过关节控制机械臂，则启动：

```

1 //每次开终端都要运行
2 source devel/setup.bash

```



```
3
4 //按Tab 可自动补全
5 roslaunch arx_r5_controller open_double_arm.launch
6 //终端需要“ Ctrl+c”关闭，不可以直接关闭
```

```

arx-gaoqing@arx-gaoqing:~/work/w2/ts_new_SDK/ts_aloha$ rostopic list
/follow1_pos_back
/follow2_pos_back
/joint_control
/joint_control2
/joint_information
/joint_information2
/rosout
/rosout_agg

```

话题	作用
/joint_control	sdk 订阅左从臂关节信息
/joint_control2	sdk 订阅右从臂关节信息

总结：

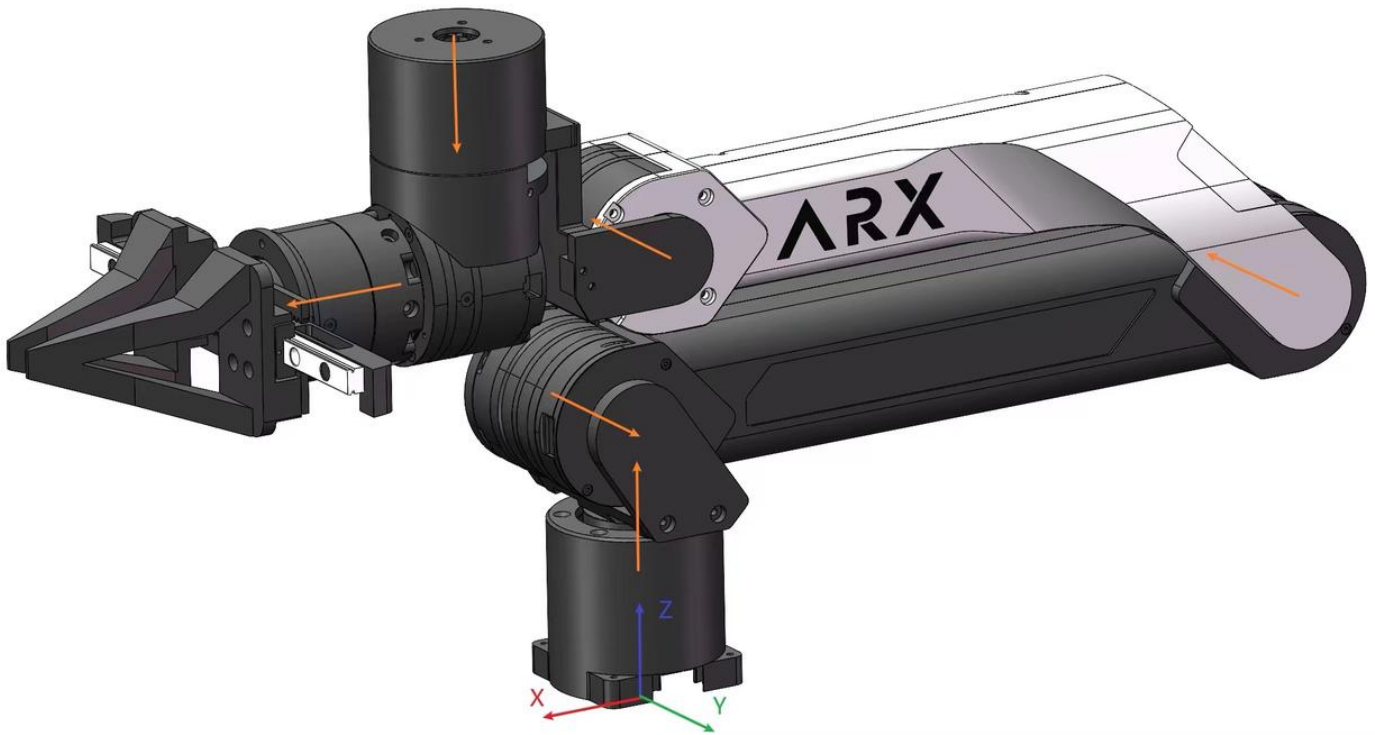
四臂采集	roslaunch arx_r5_controller open_remote_slave.launch roslaunch arx_r5_controller open_remote_master.launch	
vr采集	roslaunch arx_r5_controller open_vr_double_arm.launch	
关节控制双臂	roslaunch arx_r5_controller open_double_arm.launch	通过话题
末端位姿控制双臂	roslaunch arx_r5_controller open_vr_double_arm.launch	通过话题

### 三、注意事项

关闭终端前务必先输入：

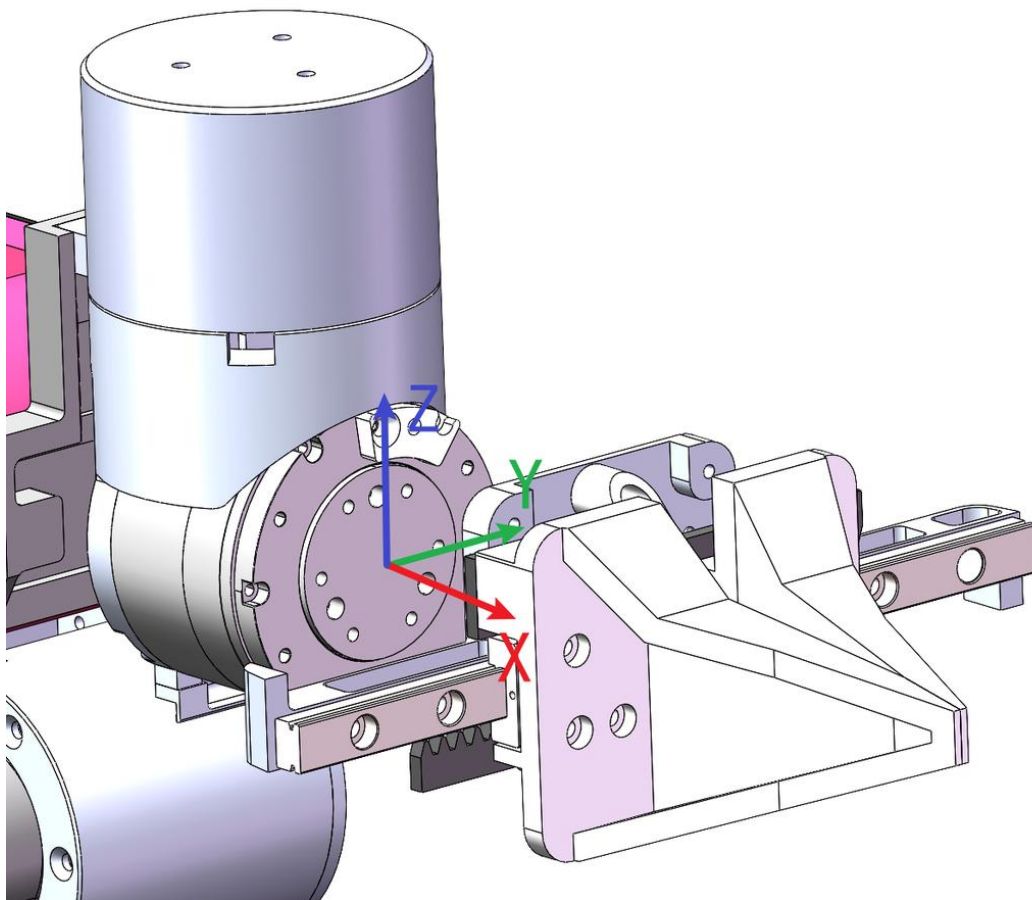
不可直接关闭终端

## 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