

# 通过手柄实现控制机械臂运动

## 实验设备：

睿尔曼 RM65-B 机械臂



罗技 F710 手柄



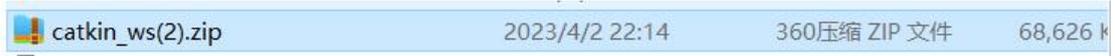
英伟达 Jetson Xavier NX 开发板



# 实验前期准备

## 1. 通讯方式

机械臂网口连接 NX 开发板网口，将手柄的接收器插到开发板的接口上，配置好 NX 的基础功能，之后替换 catkin 功能包



将 U 盘导入的 catkin 功能包解压，删除原有的 catkin 文件，将解压后的 catkin 文件粘贴到 home 目录下，粘贴完成后输入编译指令，**注意解压后的 catkin\_ws (2) 文件内部还有一个 catkin\_ws 文件，要把这个文件复制到主目录中。**

```
cd ~/catkin_ws  
catkin build
```

## 2 安装游戏手柄驱动

1 安装手柄驱动：

```
sudo apt-get install ros-melodic-joy  
sudo apt-get install joystick
```

2 查看手柄串口号

```
ls -l /dev/input/js0
```

返回信息：

```
will@will-OptiPlex-9010:~$ ls -l /dev/input/js0  
crw-rw-r-- 1 root input 13, 0 May 18 21:56 /dev/input/js0
```

3 测试手柄信号

```
sudo jstest /dev/input/js0
```

通过按移动手柄上各个按键会有对应的值变化

```

will@will-OptiPlex-9010:~$ sudo jstest /dev/input/js0
[sudo] password for will:
Driver version is 2.1.0.
Joystick (Logitech Gamepad F710) has 8 axes (X, Y, Z, Rx, Ry, Rz, Hat0X, Hat0Y)
and 11 buttons (BtnA, BtnB, BtnX, BtnY, BtnTL, BtnTR, BtnSelect, BtnStart, BtnMode, BtnThumbL, BtnThumbR).
Testing ... (interrupt to exit)
Axes: 0: 0 1: 0 2: 0 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off
Axes: 0: 0 1: 0 2: 0 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off
Axes: 0: 0 1: 0 2: 0 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off
Axes: 0: 0 1: 0 2: 0 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off
Axes: 0: 0 1: 0 2: 0 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off
Axes: 0: 0 1: 0 2: 0 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off
Axes: 0: 0 1: 0 2: 0 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off
Axes: 0: 0 1: 0 2: 0 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off
Axes: 0: 0 1: 0 2: 0 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off
Axes: 0: 0 1: 0 2: 0 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off
Axes: 0: 0 1: 0 2:-32767 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off
Axes: 0: 0 1: 0 2:-32767 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off
Axes: 0: 0 1: 0 2:-32767 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off
Axes: 0: 0 1: 0 2:-32767 3: 0 4: 0 5:-32767 6: 0 7: 0 Buttons: 0:off
Axes: 0: 0 1: 0 2:-32767 3: 0 4: 0 5:-32767 6: 0 7: 0 Buttons: 0:off
Axes: 0: 0 1: 0 2:-32767 3: 0 4: 0 5:-32767 6: 0 7: 0 Buttons: 0:off
1:off 2:off 3:off 4:off 5:off 6:off 7:off 8:off 9:off 10:off

```

[https://blog.csdn.net/Will\\_Ye](https://blog.csdn.net/Will_Ye)

#### 4 运行 ROS 节点

```
roslaunch joy joy_node
```

```

will@will-OptiPlex-9010:~$ roslaunch joy joy_node
[ WARN ] [1621346428.271123034]: Couldn't set gain on joystick force feedback: Bad file descriptor
[ INFO ] [1621346428.272911599]: Opened joystick: /dev/input/js0. deadzone_: 0.050000.

```

#### 5 查看手柄发出的信号

```
rostopic echo joy
```

按几下手柄，看看输出的信号对不对

```

^Cwill@will-OptiPlex-9010:~/catkin_ws$ rostopic echo joy
WARNING: no messages received and simulated time is active.
Is /clock being published?
header:
  seq: 1
  stamp:
    secs: 0
    nsecs: 0
  frame_id: "/dev/input/js0"
axes: [0.0, -0.0, 0.0, 0.47996458411216736, -0.13873553276062012, 0.0, -1.0, 0.0]
buttons: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
---
header:
  seq: 2
  stamp:
    secs: 0
    nsecs: 0
  frame_id: "/dev/input/js0"
axes: [0.0, -0.0, 0.0, -0.0, -0.0, 0.0, -0.0, 0.0]
buttons: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
---
header:
  seq: 3
  stamp:
    secs: 0
    nsecs: 0
  frame_id: "/dev/input/js0"
axes: [0.0, -0.0, 0.0, -0.0, -0.0, 0.0, -0.0, 0.0]
buttons: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
---

```

[https://blog.csdn.net/Will\\_Ye](https://blog.csdn.net/Will_Ye)

## 3 ROS 节点开机自启动

### 一、安装 robot-upstart

打开终端，执行以下命令，安装 robot-upstart。

```
sudo apt-get install ros-melodic-robot-upstart
```

若出现报错可能是未进行更新需要执行以下命令在进行安装

```
sudo apt-get update
```

### 二、添加自启动任务

#### 3.1 启动 ros 核心节点

由于 robot\_upstart 是 ROS 节点，启动前需要先开启 roscore。

```
cd ~/catkin_ws  
source devel/setup.bash  
roscore
```

#### 3.2 装载自启动 launch 文件

打开一个新终端，运行以下命令，装载自启动文件。

```
cd ~/catkin_ws  
source devel/setup.bash  
roslaunch robot_upstart install rm_65_demo/launch/start_joy.launch --job joy_robot --logdir  
~/joy_robot.log
```

#### 3.3 启动任务

在终端运行以下命令，启动任务。

```
sudo systemctl daemon-reload && sudo systemctl start joy_robot
```

```
realman@ubuntu:~$ cd catkin_ws/
realman@ubuntu:~/catkin_ws$ source devel/setup.bash
realman@ubuntu:~/catkin_ws$ rosrn robot_upstart install rm_65_demo/launch/start_joy.launch --job joy_robot --logdir ~/joy_robot.log
/lib/systemd/systemd
Preparing to install files to the following paths:
  /etc/ros/melodic/joy_robot.d/.installed_files
  /etc/ros/melodic/joy_robot.d/start_joy.launch
  /etc/systemd/system/multi-user.target.wants/joy_robot.service
  /lib/systemd/system/joy_robot.service
  /usr/sbin/joy_robot-start
  /usr/sbin/joy_robot-stop
Now calling: /usr/bin/sudo /opt/ros/melodic/lib/robot_upstart/mutate_files
[sudo] realman 的密码:
Filesystem operation succeeded.
** To complete installation please run the following command:
sudo systemctl daemon-reload && sudo systemctl start joy_robot
realman@ubuntu:~/catkin_ws$
```

### 三、测试

- (1) 关闭机械臂与 NX
- (2) 重启机械臂
- (3) 待机械臂启动成功后，启动 NX
- (4) 测试手柄控制机械臂
- (5) 若无法正常控制机械臂，需重启 NX

**注：NX 启动之前，必须确保机械臂已完成正常启动**

## 四、手柄控制机械臂方法



首先,手柄上方中间的开关拨到靠右边;

然后,如果手柄长时间无操作会进入休眠状态,需要先按几下A键唤醒;

1. 手柄控制机械臂:

- 1) 按键 X, Y, B, A, LB, RB 分别代表关节 1~6
- 2) 按键 X, Y, B 分别代表位置示教的 X, Y, Z 方向
- 3) 方向键左和右代表关节示教时的正向和负向
- 4) 方向键上和下代表位置示教时的正向和负向
- 5) 代表关节的按键和方向键左或右同时按下时,相应关节开始关节示教运动,其中一个键松开或都松开后停止示教

6) 代表位置示教的 X, Y, Z 方向的一个按键和方向键上或下同时按下时, 机械臂开始沿对应方向位置示教运动, 其中一个键松开或都松开后停止示教;

## 2. 手柄控制升降:

- 1) "X"和"B"都松开时, 只控制摇杆 A 上下使用默认速度控制;
- 2) "X"按下且"B"松开时, 同时控制摇杆 A 上下使用快速模式控制;
- 3) "X"松开且"B"按下时, 同时控制摇杆 A 上下使用慢速模式控制;

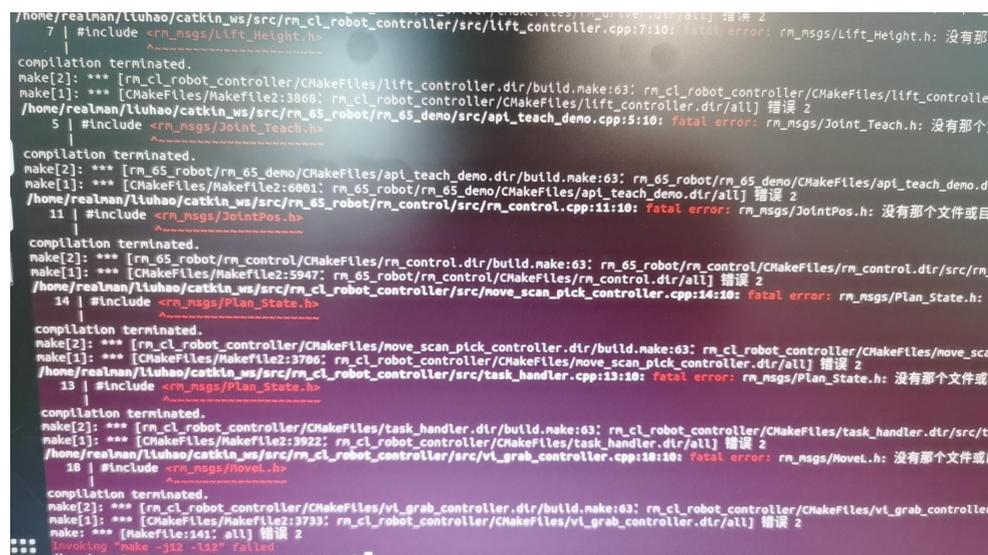
## 五、常见问题

### 1. 通讯问题

(1) 机械臂默认 IP 为 192. 168. 1. 18, 需要做到与 NX 开发板通讯, 需要将 NX 的 IP 修改至和机械臂为同一网段如: 192. 168. 1. 11

(2) 修改完成后在终端 ping192. 168. 1. 18 判断是否修改成功

### 2. Catkin\_ws 编译不通过



```
~/home/realnan/luhao/catkin_ws/src/rm_cl_robot_controller/src/lift_controller.cpp:7:10: fatal error: rm_msgs/Lift_Height.h: 没有那个文件或目录
7 | #include <rm_msgs/Lift_Height.h>
  |          ^
compilation terminated.
make[2]: *** [rm_cl_robot_controller/CMakeFiles/lift_controller.dir/build.make:63: rm_cl_robot_controller/CMakeFiles/lift_controller.dir/src/lift_controller.o] 错误 2
make[1]: *** [CMakeFiles/Makefile2:3868: rm_cl_robot_controller/CMakeFiles/lift_controller.dir/all] 错误 2
~/home/realnan/luhao/catkin_ws/src/rm_65_robot/rm_65_deno/src/api_teach_demo.cpp:5:10: fatal error: rm_msgs/JoInt_Teach.h: 没有那个文件或目录
5 | #include <rm_msgs/JoInt_Teach.h>
  |          ^
compilation terminated.
make[2]: *** [rm_65_robot/rm_65_deno/CMakeFiles/api_teach_demo.dir/build.make:63: rm_65_robot/rm_65_deno/CMakeFiles/api_teach_demo.dir/src/api_teach_demo.o] 错误 2
make[1]: *** [CMakeFiles/Makefile2:6001: rm_65_robot/rm_65_deno/CMakeFiles/api_teach_demo.dir/all] 错误 2
~/home/realnan/luhao/catkin_ws/src/rm_65_robot/rm_control/src/rm_control.cpp:11:10: fatal error: rm_msgs/JoIntPos.h: 没有那个文件或目录
11 | #include <rm_msgs/JoIntPos.h>
   |          ^
compilation terminated.
make[2]: *** [rm_65_robot/rm_control/CMakeFiles/rm_control.dir/build.make:63: rm_65_robot/rm_control/CMakeFiles/rm_control.dir/src/rm_control.o] 错误 2
make[1]: *** [CMakeFiles/Makefile2:5947: rm_65_robot/rm_control/CMakeFiles/rm_control.dir/all] 错误 2
~/home/realnan/luhao/catkin_ws/src/rm_cl_robot_controller/src/move_scan_pick_controller.cpp:14:10: fatal error: rm_msgs/Plan_State.h: 没有那个文件或目录
14 | #include <rm_msgs/Plan_State.h>
   |          ^
compilation terminated.
make[2]: *** [rm_cl_robot_controller/CMakeFiles/move_scan_pick_controller.dir/build.make:63: rm_cl_robot_controller/CMakeFiles/move_scan_pick_controller.dir/src/move_scan_pick_controller.o] 错误 2
make[1]: *** [CMakeFiles/Makefile2:3786: rm_cl_robot_controller/CMakeFiles/move_scan_pick_controller.dir/all] 错误 2
~/home/realnan/luhao/catkin_ws/src/rm_cl_robot_controller/src/task_handler.cpp:13:10: fatal error: rm_msgs/Plan_State.h: 没有那个文件或目录
13 | #include <rm_msgs/Plan_State.h>
   |          ^
compilation terminated.
make[2]: *** [rm_cl_robot_controller/CMakeFiles/task_handler.dir/build.make:63: rm_cl_robot_controller/CMakeFiles/task_handler.dir/src/task_handler.o] 错误 2
make[1]: *** [CMakeFiles/Makefile2:3792: rm_cl_robot_controller/CMakeFiles/task_handler.dir/all] 错误 2
~/home/realnan/luhao/catkin_ws/src/rm_cl_robot_controller/src/vt_grab_controller.cpp:18:10: fatal error: rm_msgs/Move1.h: 没有那个文件或目录
18 | #include <rm_msgs/Move1.h>
   |          ^
compilation terminated.
make[2]: *** [rm_cl_robot_controller/CMakeFiles/vt_grab_controller.dir/build.make:63: rm_cl_robot_controller/CMakeFiles/vt_grab_controller.dir/src/vt_grab_controller.o] 错误 2
make[1]: *** [CMakeFiles/Makefile2:3793: rm_cl_robot_controller/CMakeFiles/vt_grab_controller.dir/all] 错误 2
make: *** [Makefile:141: all] 错误 2
Invoking "make -j12 -l12" failed
(base) catkin_ws$ cd .. && rm -rf catkin_ws
```

错误信息

解决方法：

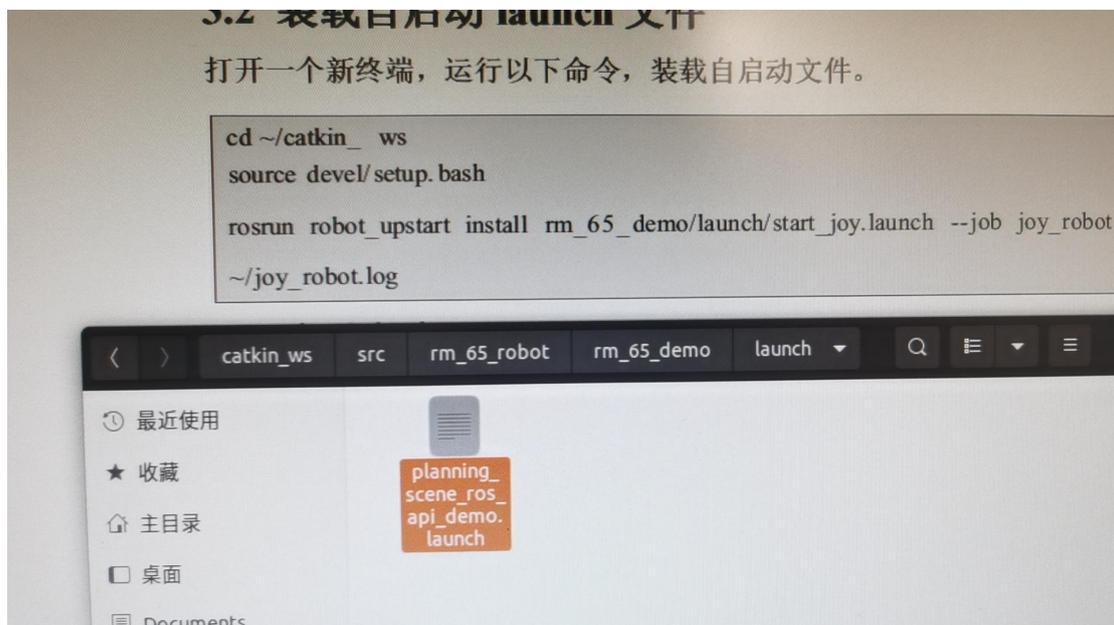
(1) 单独先编译 `rm_msgs`

(2) 使用 `catkin_build` 会自动优化过程 不建议使用 `catkin_make`

参考链接：

<https://blog.csdn.net/benchuspx/article/details/113847854>

### 3. 装载自启动 `launch` 时，没有找到这个 `start_joy.launch`



文件下没有找到对应的 `launch` 文件



在和 `catkin_ws` 同级目录下的手柄控制程序开机自启动压缩包下面，

找到 `launch` 文件后，直接复制到对应的文件夹下。