TRON1 移动操作拓展套件 SDK

生成时间：2026-10-26

1. 概述

移动操作拓展套件SDK文档主要用于用户购买移动操作拓展套件（TRON1机器人和机械臂）后，可通过该文档快速部署移动操作拓展套件的依赖和全身控制算法，从而实现TRON1机器人和机械臂快速操作。

注意：本文档适用于2.9.0版本机械臂固件

2. 仿真与真机调试

2.1搭建开发环境

- 安装ROS Noetic：我们推荐在Ubuntu 20.04操作系统上建立基于ROS Noetic的算法开发环境。ROS提供了一系列工具和库，如核心库、通信库和仿真工具（如Gazebo），极大地便利了机器人算法的开发、测试和部署。这些资源为用户提供了一个丰富而完整的算法开发环境。
  - ROS Noetic 安装请参考文档：Ubuntu install of ROS Noetic，选择“ros-noetic-desktop-full”进行安装。
  - ROS Noetic 安装完成后，Bash终端输入以下Shell命令，安装开发环境所依赖的库：

    ▼

    bash 复制代码

```
sudo apt-get update
sudo apt install ros-noetic-urdf \
                 ros-noetic-kdl-parser \
                 ros-noetic-urdf-parser-plugin \
                 ros-noetic-hardware-interface \
                 ros-noetic-controller-manager \
                 ros-noetic-controller-interface \
                 ros-noetic-robot-state-* \
                 ros-noetic-joint-state-* \
                 ros-noetic-controller-manager-msgs \
                 ros-noetic-control-msgs \
                 ros-noetic-ros-control \
                 ros-noetic-gazebo-* \
                 ros-noetic-rqt-gui \
                 ros-noetic-rqt-controller-manager \
                 ros-noetic-plotjuggler* \
                 ros-noetic-joy-teleop ros-noetic-joy \
                 cmake build-essential libpcl-dev libeigen3-dev libopencv-dev libma
tio-dev \
```

```
                    python3-pip libboost-all-dev libtbb-dev liburdfdom-dev liborocos-k
dl-dev libspdlog-dev -y
```

2.2 仿真部署

- 打开一个Bash终端
- 下载仿真部署代码及虚拟遥控器节点

▼

bash 复制代码

```
git clone https://github.com/limxdynamics/tron1-rl-deploy-arm.git
git clone https://github.com/limxdynamics/robot-joystick.git
```

- 编译工程

▼

bash 复制代码

```
# 如您安装了Conda，请临时禁用 Conda 环境
# 因为 Conda 会干扰 ROS 的运行环境设置
conda deactivate
cd ~/tron1-rl-deploy-arm
catkin_make install
```

- 运行仿真器

▼

bash 复制代码

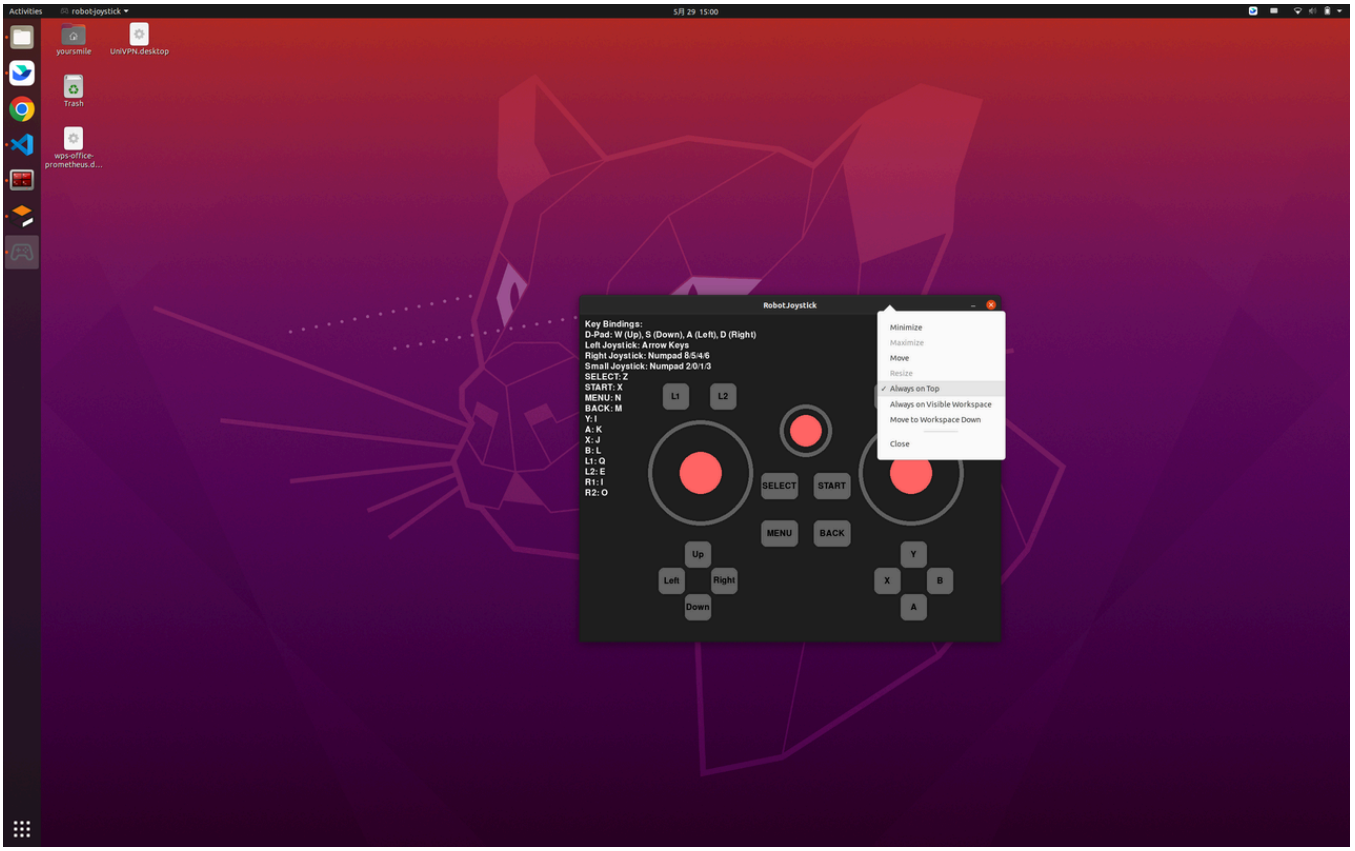
```
source install/setup.bash
export ROBOT_TYPE=SF_TRON1A # 视机器人类型而定，如果使用的是轮足则改为export ROBOT_TYPE=WF_TRON1
A
export IS_SIM=1 # 开启仿真模式
roslaunch robot_hw pointfoot_hw_sim.launch
```

- 开启新的命令行窗口，运行虚拟遥控器，鼠标移动到其上方工具栏，右键选择置顶

▼

bash 复制代码

```
./robot-joystick/robot-joystick
```



- 控制仿真环境中的机器人，正常效果如下，如果出现站不起来或摔倒等情况，点击gazebo窗口，按`Ctrl+Shift+R`即可重置姿态（如果使用`Ctrl+R`会导致ros时间重置，机械臂无法控制）。
- 遥控器按键：
  - 键盘`↑ ↓`控制前后移动，键盘`← →`控制左右移动（仅限脚掌）
  - 小键盘`4 6`控制左转/右转，小键盘`8 5`控制机身高度上升/下降。
  - 键盘`W A S D`控制机械臂四向移动，键盘`I J K L`控制机械臂四向旋转。
- **注意事项**：重置机器人和控制遥控器二者切换时，需要用鼠标点击要控制的窗口（如果要重置机器人，则点击gazebo窗口任意位置；如果要控制遥控器，则点击遥控器面板任意位置），否则会出现指令无响应的情况。

2.3真机部署

> **前提条件**：
>
> - 机器人与机械臂、机械臂与夹爪均已接好电源线、通信线。
>
> **注意事项**：
>
> - 机器人与机械臂的通信：机械臂通信口需要插USB2接口。
> - 目前全身控制算法仅支持机械臂固件版本2.9.0，其余机械臂固件版本不支持。

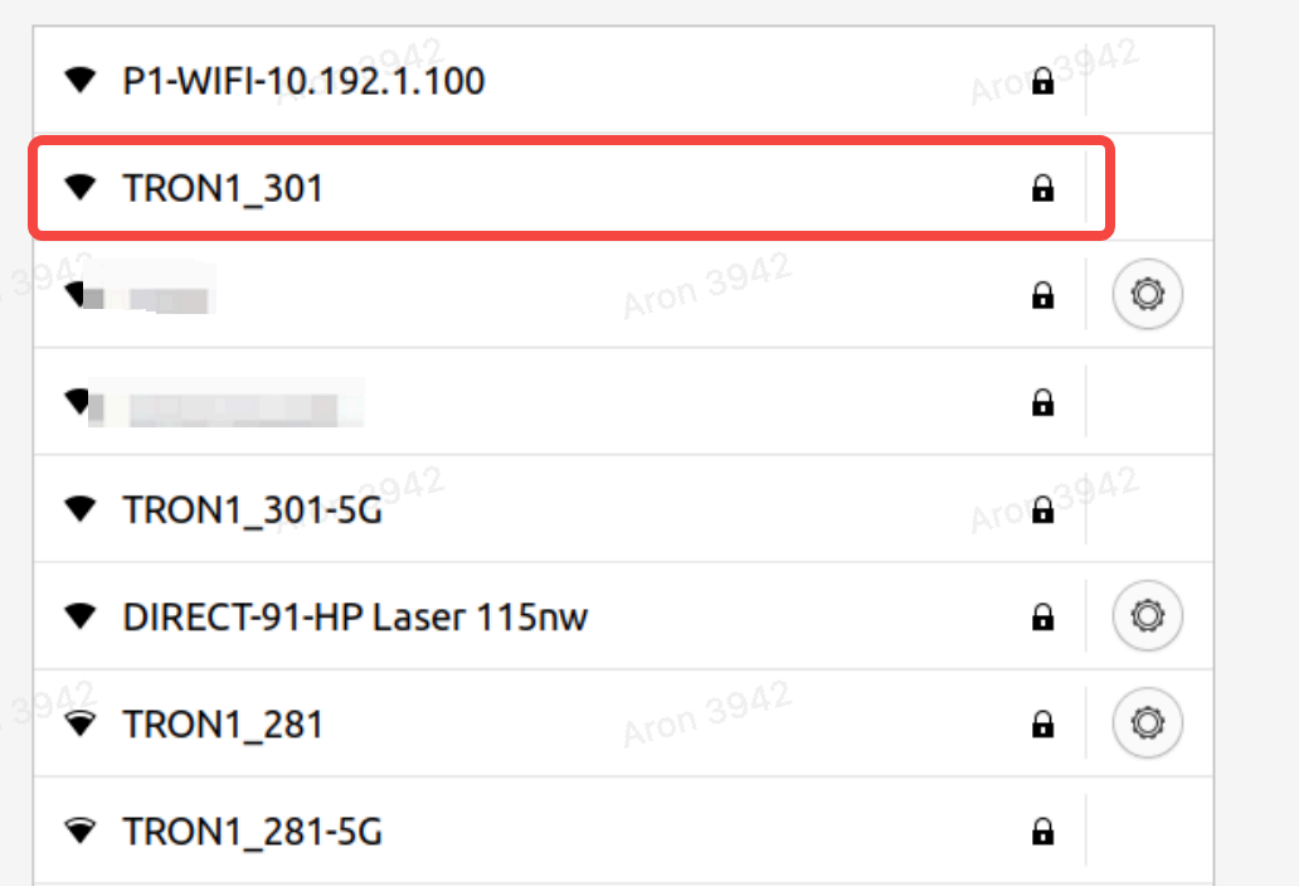
2.3.1 切换机器人至开发者模式

对机器人和机械臂分别开机，待机器人进入空闲状态后（机器人灯语蓝灯常亮），同时按下遥控器按键`R1 + Left`，这时本体主机将会自动重启并切换机器人到开发者模式（机器人灯语由蓝灯常亮变成绿灯常亮）。在模式下，用户可部署移动操作拓展套件全身控制算法，模式设置掉电后不会失效，重新开机后仍然是开发者模式。遥控器切换工作模式的按键列表如下：

按键	模式	说明
R1+Left	开发者模式	部署全身控制算法，可同时操作机器人和机械臂。
R1+Right	遥控模式	运行预安装的控制算法，实现复杂地形的平稳行走，如：上下台阶、过坎等。

2.3.2 环境配置

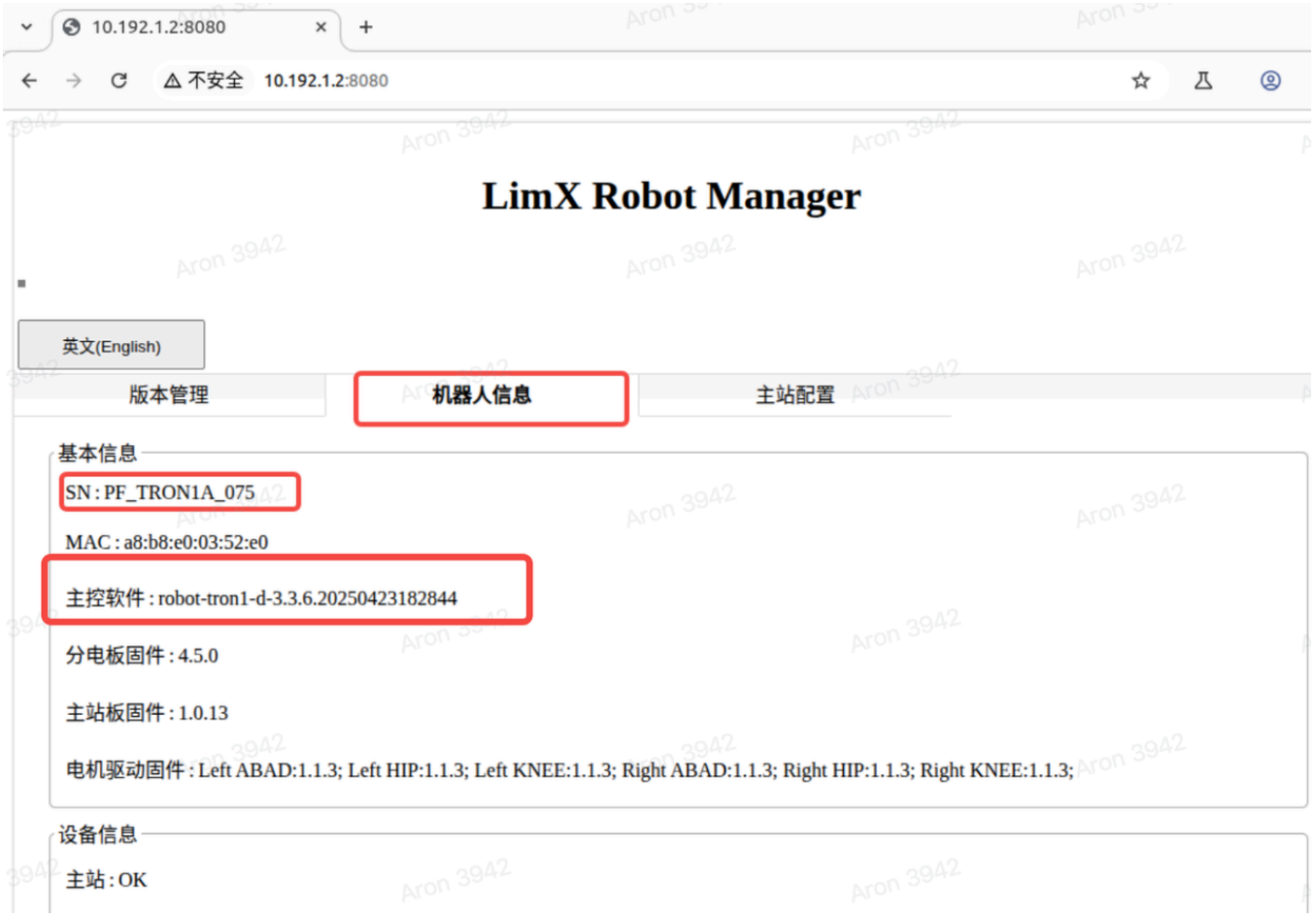
2.3.2.1 连接机器人

- 请使用个人电脑连接机器人Wi-Fi，名称格式通常为「TRON1_01」，Wi-Fi密码：`12345678`。
- 

- 在浏览器中输入`http://10.192.1.2:8080`可以进入“机器人信息页”，并查看机器人信息。页面中显示的SN (序列号) 为PF_TRON1A_075，其中PF_TRON1A是机器人型号类型。

> 机器人主控固件版本必须为3.1.7以上版本，方可支持加载机械臂进行全身控制，若是旧主控固件版本，

> 请升级至最新固件版本后进行真机部署。



2.3.2.2 环境配置

将仿真部署仓库编译好的install文件夹通过scp拷贝至机器人（密码123456）

▼

bash 复制代码

```
scp -r install/ guest@10.192.1.2:/home/guest/
```

▼

bash 复制代码

```
ssh guest@10.192.1.2 # 登录到机器人的电脑
# 密码 123456
source ~/install/setup.bash
airbot_read_params
# 能够正常读取每个电机参数说明环境配置成功
```

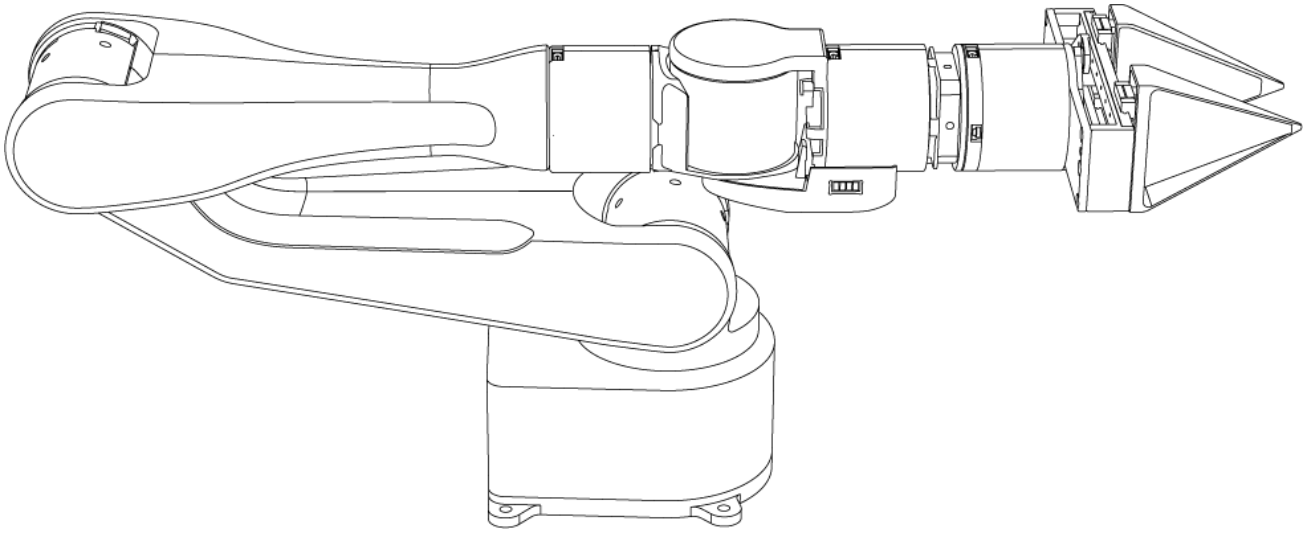
2.3.3机械臂标零

2.3.3.1 常规标零

▼

bash 复制代码

```
ssh guest@10.192.1.2 # 登录到机器人的电脑
# 密码 123456
source ~/install/setup.bash
airbot_set_zero
```

- 将机械臂所有关节移动到零点位置，按下回车，等待终端显示标零完成，则标零成功。
- 
- 若机械臂执行常规校零后，运行控制器代码时出现如下报错：
- ```
  auto-starting new master
  process[master]: started with pid [22030]
  ROS_MASTER_URI=http://localhost:11311

  setting /run_id to 0f61c3f2-d0f2-11f0-ab9c-6932fb54c448
  process[rosout-1]: started with pid [22062]
  started core service [/rosout]
  process[pointfoot_hw-2]: started with pid [22070]
  INFO: RobotIPAddress - "10.192.1.2"
  [ WARN] [1764839852.256687223]: Connect to the robot: 10.192.1.2
  robot_type_ = WF_TRON1A
  [ INFO] [1764839852.269350180]: Real World Deploy , Airbot Arm SDK Init!
  [ INFO] [1764839852.270823180]: Initializing arm interface
  ================================================================================REQUIRED process [pointfoot_hw-2] has died!
  process has died [pid 22070, exit code -11, cmd /home/guest/install/lib/robot_hw/pointfoot_node 10.192.1.2 __name:=pointfoot_hw __log:=
  /home/guest/.ros/log/0f61c3f2-d0f2-11f0-ab9c-6932fb54c448/pointfoot_hw-2.log].
  log file: /home/guest/.ros/log/0f61c3f2-d0f2-11f0-ab9c-6932fb54c448/pointfoot_hw-2*.log
  Initiating shutdown!
  ================================================================================
  [pointfoot_hw-2] killing on exit
  [rosout-1] killing on exit
  [master] killing on exit
  shutting down processing monitor...
  ... shutting down processing monitor complete
  done
  ```
- 请执行下方的绝对标零：

2.3.3.2 绝对标零（正常情况不需要运行）

▼

bash 复制代码
  ```

```
ssh guest@10.192.1.2 # 登录到机器人的电脑
# 密码 123456
source ~/install/setup.bash
airbot_absolute_set_zero
```

将机械臂J1关节顺时针转动至限位，按下回车，全程扶着机械臂后三关节在零点位置，机械臂会自动将前三关节运动到零点，等待终端显示标零完成并退出程序后完成。

> 绝对标零仅在以下情况才执行，正常情况下执行常规标零即可：
>
> - 机械臂重新刷新固件版本。
> - 机械臂丢失零点，此时机械臂灯语会黄灯闪烁。

2.3.4 运行控制器

- 导入环境变量并启动程序

▼

bash 复制代码

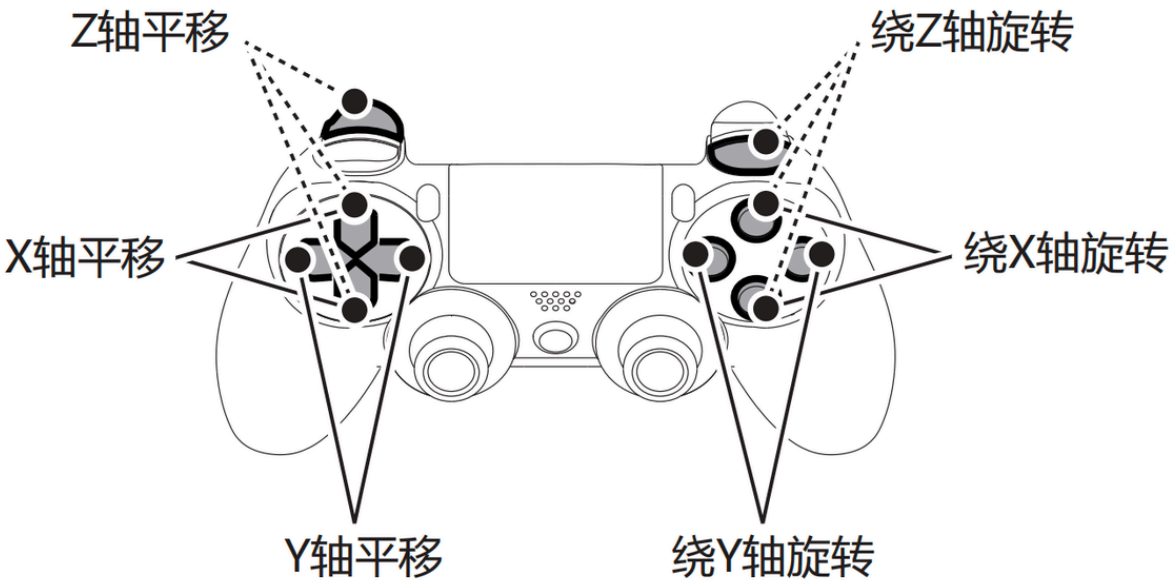
```
ssh guest@10.192.1.2 # 登录到机器人的电脑
# 密码 123456
source ~/install/setup.bash
export ROBOT_TYPE=WF_TRON1A # 一定要和机器人形态保持一致，脚掌则为export ROBOT_TYPE=SF_TRON1A
export IS_SIM=0 # 设置真机模式
roslaunch robot_hw pointfoot_hw.launch
```

- 遥控器按键：
  - `L1+△` **请先回到准备状态再站起，站起时请用手扶助机身直至站起。**
  - `L1+□` 关闭控制程序，**会直接下电，需要拉住机身侧把手，关闭程序后将其手摆至下蹲姿态。**

3. 遥控器操作说明

3.1遥控器操作

3.1.1 遥控器操作按键定义



- 左摇杆，机器人前后左右移动。
- 右摇杆，左右拨动为控制机器人旋转方向，上下拨动为控制机器人底盘高度。
- 方向键，控制末端位置。
  - 上：末端水平向前移动（x轴正向）
  - 下：末端水平向后移动（x轴负向）
  - 左：末端水平向右移动（y轴正向）
  - 右：末端水平向左移动（y轴负向）
  - L2+上：末端竖直向上移动（z轴正向）
  - L2+下：末端竖直向下移动（z轴负向）
- Δ O X □ ，控制末端旋转。
  - X：增加末端pitch轴旋转角度。
  - Δ：减小末端pitch轴旋转角度。
  - O：增加yaw轴旋转角度。
  - □：减小yaw轴旋转角度。
  - L2+O：增加roll轴旋转角度。

    - L2+□：减小roll轴旋转角度。
- R1 控制夹爪开合，按下夹爪打开，松开夹爪闭合。