宇树四足机器人开发入门：ROS篇

宇树四足机器人开发入门：ROS篇

本文旨在介绍从一个新的Ubuntu系统开始跑宇树SDK `unitree_legged_sdk` 、宇树ROS包 `unitree_ros` 所需要的步骤以及需要配置的环境。根据内容，本文分为基础篇、SDK篇、ROS篇。

本篇内容以**在客户PC上运行unitree_ros、unitree_ros_to_real包**为例。如果是在机载板卡上运行unitree_ros_to_real包，ROS环境出厂时已经安装好，无需再次安装。unitree_ros包主要是仿真用，无需在机载板卡上编译运行。

1. 安装ROS

ROS安装直接参考ROS Wiki的说明安装即可，与Ubuntu 18.04对应的ROS版本是ROS-melodic，安装一般没什么问题，但由于网络污染，ROS初始化，一般都是不成功的，可以看看对应的解决办法，也可以使用第三方提供的“一键安装”方法。

安装ROS参考链接：https://blog.csdn.net/qq_43310597/article/details/105756819 <https://blog.csdn.net/qq_43310597/article/details/105756819>

1.1 调整Ubuntu镜像源

Software Updater中设置下载源为 http://mirrors.ustc.edu.cn/ubuntu 。

这一步是为了让接下来的下载过程使用中国的服务器，加快速度，ROS官方推荐软件源设置为中国科技大学（USTC）或清华大学（Tsinghua），这里将软件源切换成中科大的源。

1.2 添加source.list

Plain Text | 复制代码

```
1  sudo sh -c '. /etc/lsb-release && echo "deb http://mirrors.ustc.edu.cn/ros/ubu
```

这一步配置将镜像添加到Ubuntu系统源列表中，建议使用国内的镜像源，这样能够保证下载速度。这里使用的是中国科技大学的源。

1719 字

1.3 添加keys

Plain Text 复制代码

```
sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key 42
```

公钥是Ubuntu系统的一种安全机制，也是ROS安装中不可缺的一部分。

1.4 安装ROS

Plain Text 复制代码

```
sudo apt update
```

首先要保证系统的Debian安装包在最新状态，一定要确保所有的执行项都被命中或者获取。

> 如果出现错误：
> - W: GPG 错误：http://packages.ros.org/ros/ubuntu xenial InRelease: 由于没有公钥，无法验证下列签名： NO_PUBKEY F42ED6FBAB17C654
> - W: 仓库 “http://packages.ros.org/ros/ubuntu xenial InRelease” 没有数字签名。
> - N: 无法认证来自该源的数据，所以使用它会带来潜在风险。
> - N: 参见 apt-secure(8) 手册以了解仓库创建和用户配置方面的细节。
>
> 则：
> sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys F42ED6FBAB17C654
> 其中最后一串码与之前的报错中签名的内容保持一致。

Plain Text 复制代码

```
sudo apt install ros-melodic-desktop-full
```

安装桌面完整版，包含ROS、rqt、rviz、机器人通用库、2D/3D 模拟器、导航以及2D/3D感知等。

> 也可以安装某个指定的ROS软件包（使用软件包名称替换掉下面的PACKAGE）：
> sudo apt install ros-melodic-PACKAGE
> 例如：
> sudo apt install ros-melodic-slam-gmapping

1719 字

1.5 初始化rosdep

Plain Text 复制代码

```
sudo rosdep init
```

如果出现错误：

- --找不到命令
  sudo apt install python-rosdep

- --Website may be down
  方法一：
  sudo gedit /etc/hosts
  增加199.232.68.133 raw.githubusercontent.com
  方法二：
  参考链接
  https://blog.csdn.net/weixin_43311920/article/details/114796748
<https://blog.csdn.net/weixin_43311920/article/details/114796748>
  https://blog.csdn.net/nanianwochengshui/article/details/105702188
<https://blog.csdn.net/nanianwochengshui/article/details/105702188>

Plain Text 复制代码

```
rosdep update
```

如果报错：
解决办法参考：
https://blog.csdn.net/weixin_43311920/article/details/114796748
<https://blog.csdn.net/weixin_43311920/article/details/114796748>
https://blog.csdn.net/nanianwochengshui/article/details/105702188
<https://blog.csdn.net/nanianwochengshui/article/details/105702188>

1.6 配置环境

Plain Text 复制代码

```
echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc
source ~/.bashrc
```

1719 字

1.7 测试ROS

```
roscore
```

rosinstall
rosinstall是ROS中一个经常使用的命令行工具，它使你能够轻松地下载和安装ROS中的功能包程序。这个工具暂时不是必需的，但是为了便于后续开发，还是建议通过如下命令安装：

```
sudo apt-get install python-rosinstall python-rosinstall-generator python-wstoo
```

1.8 创建ROS工作空间

```
mkdir -p ~/catkin_ws/src
cd ~/catkin_ws/
catkin_make
source devel/setup.bash
echo "source ~/catkin_ws/devel/setup.bash" >> ~/.bashrc
```

至此，一个ROS工作环境搭建完成。

1.9 ROS安装工具

下面放上两个ROS“一键安装”的方法，供大家参考使用。
https://github.com/RocShi/rostaller <https://github.com/RocShi/rostaller>
https://blog.csdn.net/qq_27865227/article/details/120277420
<https://blog.csdn.net/qq_27865227/article/details/120277420>

2. 编译使用unitree_ros及unitree_ros_to_real

如果只是想通过ros控制机器狗，仅需将GitHub上unitree_ros_to_real包放到工作空间的/src目录下，并按readme里配置环境及编译。

1719 字

如果只想在个人PC上仿真，仅需要将unitree_ros包及unitree_ros_to_real包里的unitree_legged_msgs一起放到工作空间的/src目录下，并按readme里配置环境及编译。

如果在个人PC上仿真及通过ros控制机器狗，需要将unitree_ros包及unitree_ros_to_real包一起放到工作空间的/src目录下，并按readme里配置环境及编译。

下面本文的工作空间以~/catkin_ws为例，介绍前两种情况，第三种情况不再赘述。

2.1 unitree_ros

Step1. 首先新建工作空间文件夹~/catkin_ws，并在目录下新建src文件夹。

Step2. 将unitree_ros包里的所有内容以及unitree_ros_to_real（v3.2.1 <https://github.com/unitreerobotics/unitree_ros_to_real/tree/v3.2.1> ）包里的unitree_legged_msgs放到~/catkin_ws/src下。

Step3. 根据readme里build的相关内容，按ROS版本安装相应的ROS依赖。

> For ROS Melodic:
> sudo apt-get install ros-melodic-controller-interface ros-melodic-gazebo-ros-control ros-melodic-joint-state-controller ros-melodic-effort-controllers ros-melodic-joint-trajectory-controller ros-melodic-joint-state-publisher-gui
>
> For ROS Kinetic:
> sudo apt-get install ros-kinetic-controller-manager ros-kinetic-ros-control ros-kinetic-ros-controllers ros-kinetic-joint-state-controller ros-kinetic-effort-controllers ros-kinetic-velocity-controllers ros-kinetic-position-controllers ros-kinetic-robot-controllers ros-kinetic-robot-state-publisher ros-kinetic-gazebo8-ros ros-kinetic-gazebo8-ros-control ros-kinetic-gazebo8-ros-pkgs ros-kinetic-gazebo8-ros-dev

Step4. 修改unitree_gazebo/worlds/stairs.world文件里最后一段building_editor_models/stairs的真实路径。

> 以系统用户名unitree，工作空间~/catkin_ws为例：
> <include>
> <uri>model:///home/unitree/catkin_ws/src/unitree_gazebo/worlds/building_editor_models/stairs</uri>
> </include>

1719 字

> 这部分根据实际情况修改！！

Step5. 编译

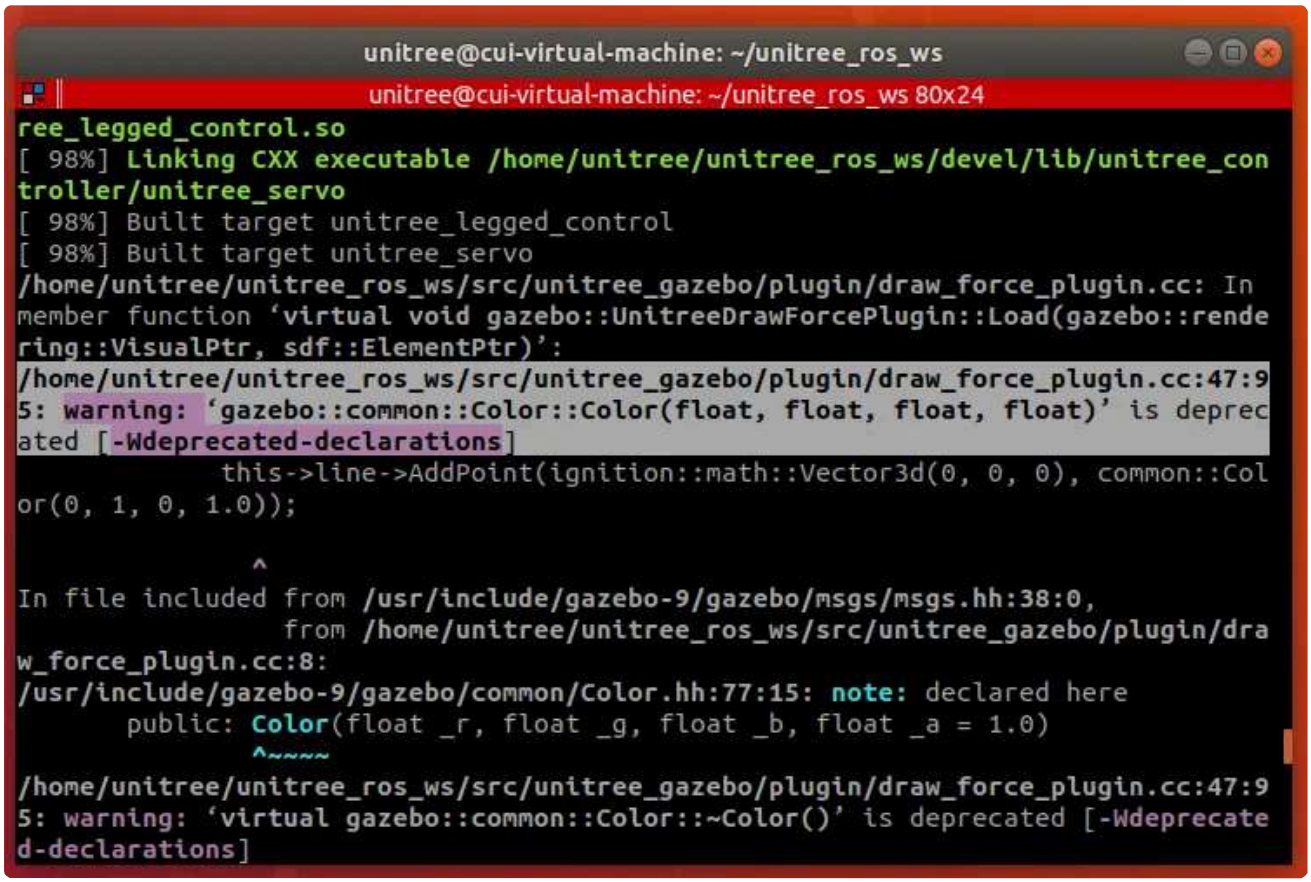
```
cd ~/catkin_ws
catkin_make
```

Step6. 使用。可以根据提供的示例尝试。

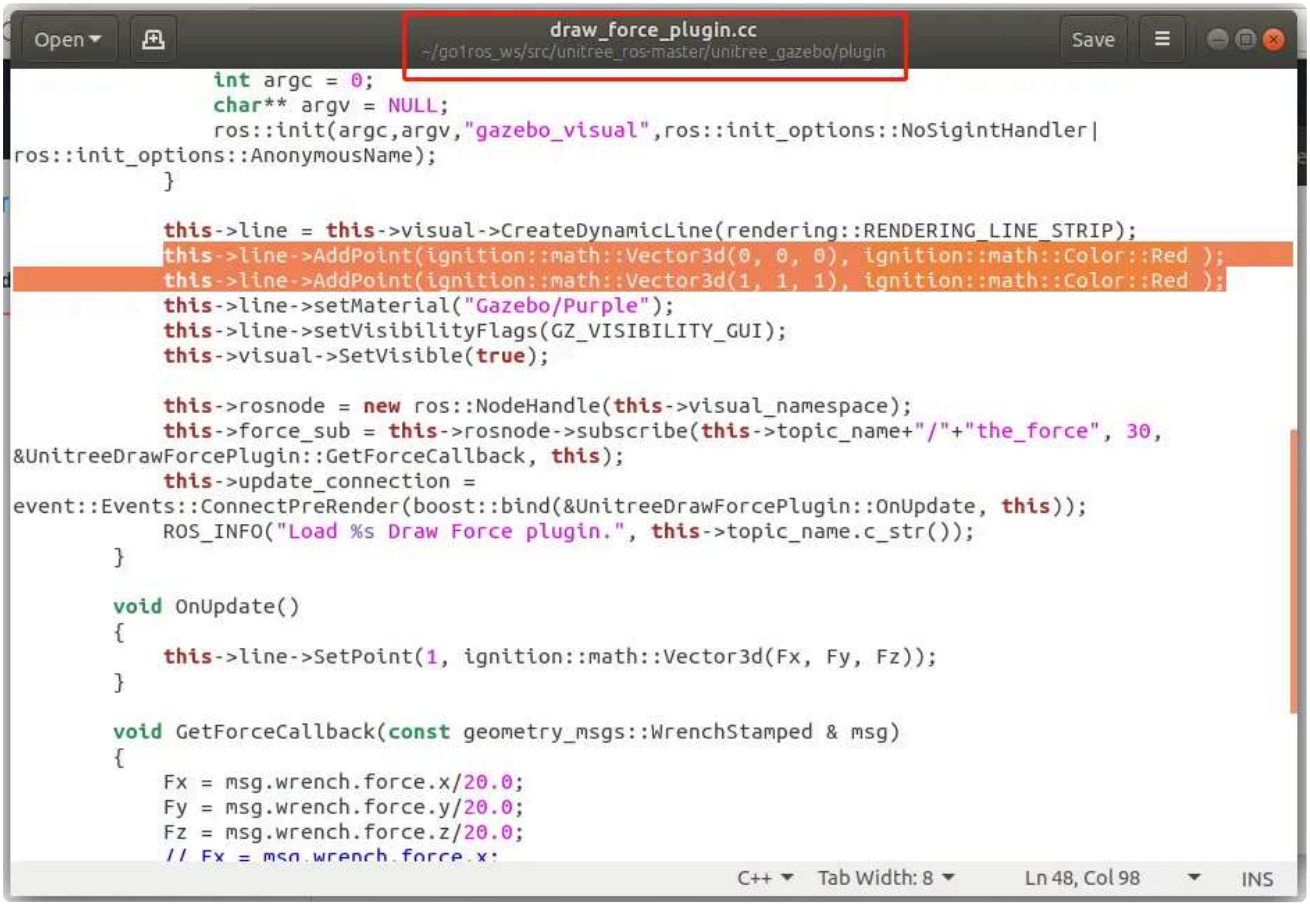
常见错误

1．draw_force_plugin.cc文件报警告

警告：



解决办法：



2．无法运行示例（提示找不到包之类的）

Plain Text | 复制代码

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

每个终端窗口都需要配置一下。

3．Rviz joint_state_publisher_gui报错

报错：

```
process[robot_state_publisher-3]: started with pid [31480]
process[rviz-4]: started with pid [31481]
[ INFO] [1652960707.053002084]: rviz version 1.13.24
[ INFO] [1652960707.053050330]: compiled against Qt version 5.9.5
[ INFO] [1652960707.053076130]: compiled against OGRE version 1.9.0 (Ghadamon)
[ INFO] [1652960707.061705865]: Forcing OpenGl version 0.
[WARN] [1652960707.204924]: The 'use_gui' parameter was specified, which is deprecated.  We'll attemp
t to find and run the GUI, but if this fails you should install the 'joint_state_publisher_gui' packa
ge instead and run that.  This backwards compatibility option will be removed in Noetic.
[ERROR] [1652960707.205831]: Could not find the GUI, install the 'joint_state_publisher_gui' package
[joint_state_publisher-2] process has died [pid 31476, exit code 1, cmd /opt/ros/melodic/lib/joint_st
ate_publisher/joint_state_publisher __name:=joint_state_publisher __log:=/home/unitree/.ros/log/208c9
bd8-d769-11ec-b32a-000c2966374f/joint_state_publisher-2.log].
log file: /home/unitree/.ros/log/208c9bd8-d769-11ec-b32a-000c2966374f/joint_state_publisher-2*.log
[ INFO] [1652960708.758782126]: Stereo is NOT SUPPORTED
[ INFO] [1652960708.758861589]: OpenGL device: SVGA3D; build: RELEASE;  LLVM;
[ INFO] [1652960708.758895422]: OpenGl version: 2.1 (GLSL 1.2).
[rviz-4] process has finished cleanly
log file: /home/unitree/.ros/log/208c9bd8-d769-11ec-b32a-000c2966374f/rviz-4*.log
```

解决办法：

```
1   sudo apt-get install ros-melodic-joint-state-publisher-gui
```

```
unitree@cui-virtual-machine:~/unitree_ros_ws$
unitree@cui-virtual-machine:~/unitree_ros_ws$ sudo apt-get install ros-melodic-joint-state-publisher-
gui
[sudo] password for unitree:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  ros-melodic-joint-state-publisher-gui
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 12.0 kB of archives.
After this operation, 55.3 kB of additional disk space will be used.
Get:1 http://mirrors.tuna.tsinghua.edu.cn/ros/ubuntu bionic/main amd64 ros-melodic-joint-state-publis
her-gui amd64 1.12.15-1bionic.20220127.152450 [12.0 kB]
Fetched 12.0 kB in 0s (122 kB/s)
Selecting previously unselected package ros-melodic-joint-state-publisher-gui.
(Reading database ... 223313 files and directories currently installed.)
Preparing to unpack .../ros-melodic-joint-state-publisher-gui_1.12.15-1bionic.20220127.152450_amd64.d
eb ...
Unpacking ros-melodic-joint-state-publisher-gui (1.12.15-1bionic.20220127.152450) ...
```

4．Gazebo Error in REST request报错

报错：

```
tch_controller
[INFO] [1652961539.309276, 0.164600]: Controller Spawner: Wa
oad_controller
[INFO] [1652961539.314498, 0.165000]: Loading controller: jo
[INFO] [1652961539.322707, 0.169200]: Loading controller: FL
[Err] [REST.cc:205] Error in REST request

libcurl: (51) SSL: no alternative certificate subject name m
l.org'
[INFO] [1652961539.428603, 0.223800]: Loading controller: FL
[INFO] [1652961539.445120, 0.229200]: Loading controller: FL
```

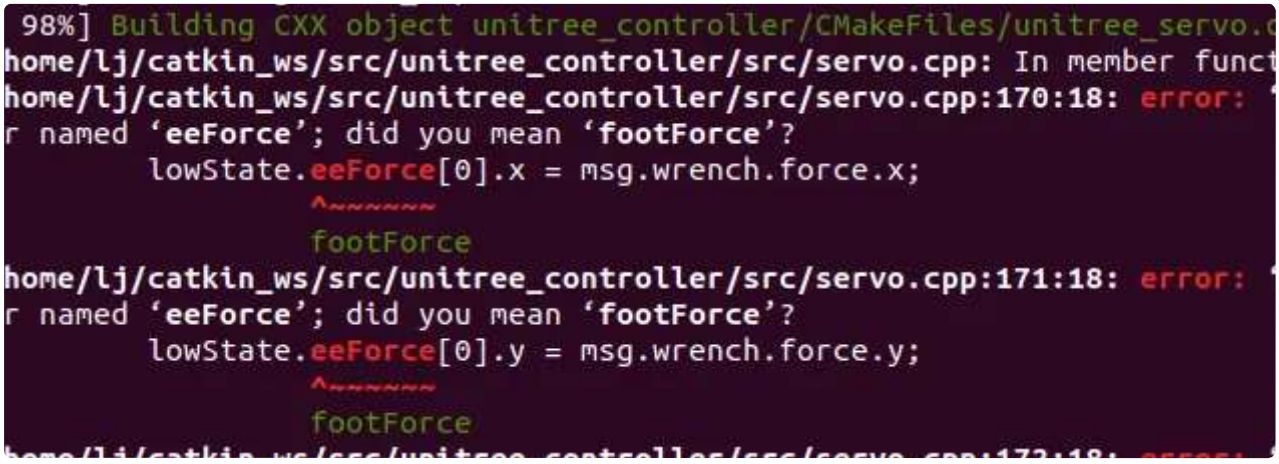
1719 字

解决办法：

https://blog.csdn.net/zc15210073939/article/details/122811637
<https://blog.csdn.net/zc15210073939/article/details/122811637>

5．eeForce报错

报错：



解决办法：

这个是由于 `unitree_legged_msgs` 用错版本导致，选择unitree_ros_to_real（v3.2.1 <https://github.com/unitreerobotics/unitree_ros_to_real/tree/v3.2.1> ）包里的unitree_legged_msgs放到~/catkin_ws/src下，重新编译即可。

或者

增加下列语句到unitree_legged_msgs/msg/LowState.msg文件中后重新编译。

Plain Text 复制代码

```
Cartesian[4] eeForceRaw
Cartesian[4] eeForce          #it's a 1-DOF force infact, but we use 3-DOF her
```

https://github.com/unitreerobotics/unitree_ros_to_real/pull/46/files
<https://github.com/unitreerobotics/unitree_ros_to_real/pull/46/files>

或者

将文件unitree_controller/src/servo.cpp中包含eeForce的语句注释掉后重新编译。

1719 字


```
168     void FRfootCallback(const geometry_msgs::WrenchStamped& msg)
169     {
170         lowState.eeForce[0].x = msg.wrench.force.x;
171         lowState.eeForce[0].y = msg.wrench.force.y;
172         lowState.eeForce[0].z = msg.wrench.force.z;
173         lowState.footForce[0] = msg.wrench.force.z;
174     }
175
176     void FLfootCallback(const geometry_msgs::WrenchStamped& msg)
177     {
178         lowState.eeForce[1].x = msg.wrench.force.x;
179         lowState.eeForce[1].y = msg.wrench.force.y;
180         lowState.eeForce[1].z = msg.wrench.force.z;
181         lowState.footForce[1] = msg.wrench.force.z;
182     }
183
184     void RRfootCallback(const geometry_msgs::WrenchStamped& msg)
185     {
186         lowState.eeForce[2].x = msg.wrench.force.x;
187         lowState.eeForce[2].y = msg.wrench.force.y;
188         lowState.eeForce[2].z = msg.wrench.force.z;
189         lowState.footForce[2] = msg.wrench.force.z;
190     }
191
192     void RLfootCallback(const geometry_msgs::WrenchStamped& msg)
193     {
194         lowState.eeForce[3].x = msg.wrench.force.x;
195         lowState.eeForce[3].y = msg.wrench.force.y;
196         lowState.eeForce[3].z = msg.wrench.force.z;
197         lowState.footForce[3] = msg.wrench.force.z;
198     }
199
200 private:
201     ros::NodeHandle nm;
202     ros::Subscriber servo_sub[12], footForce_sub[4], imu_sub;
```

2.2 unitree_ros_to_real v3.2.1

后续有空了写

2.3 unitree_ros_to_real v3.5.0

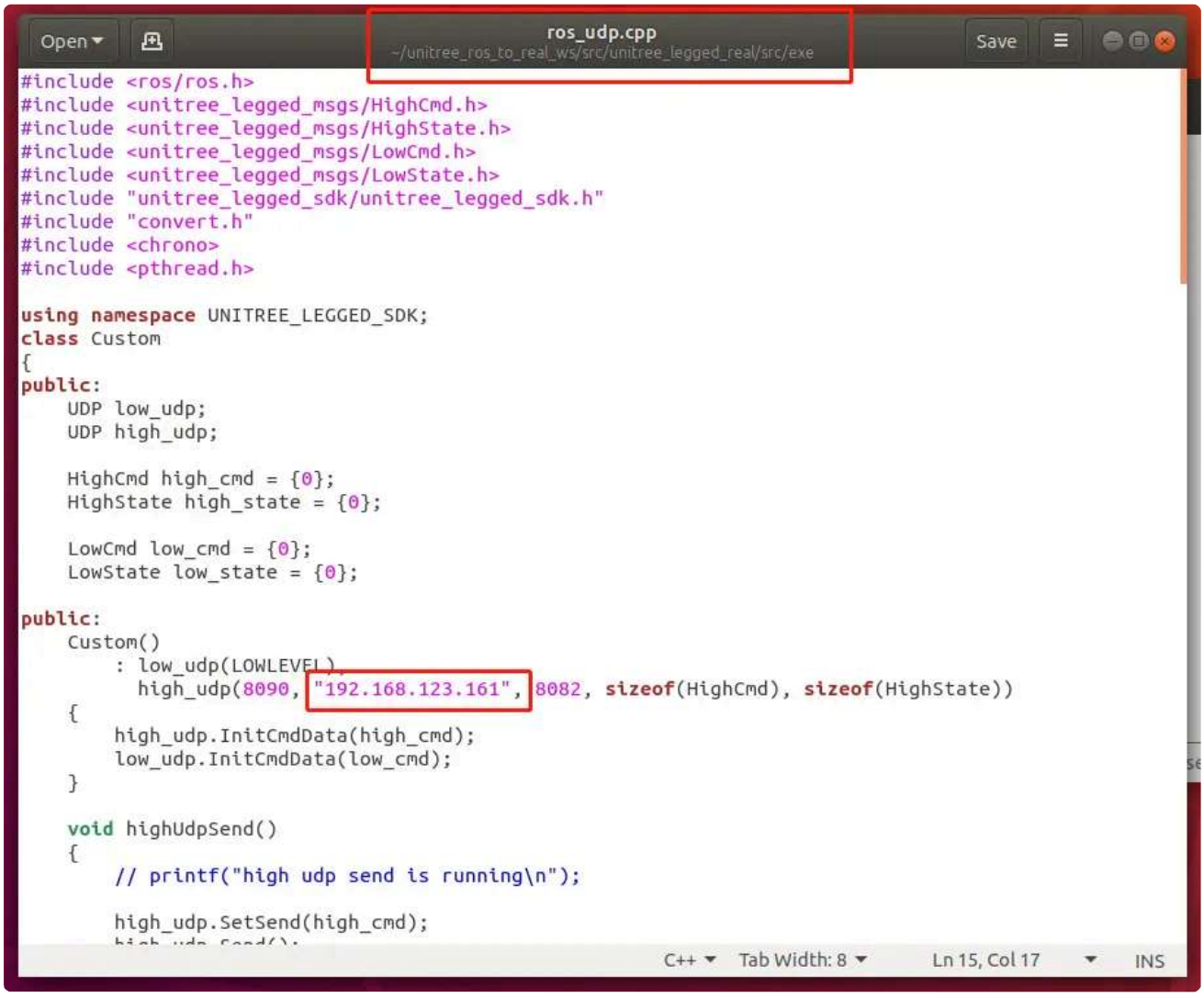
1719 字

仅适用于Go1。 unitree_ros_to_real_ws-3.5.1.zip (502 KB)

Step1. 首先新建工作空间文件夹~/catkin_ws，并在目录下新建src文件夹。

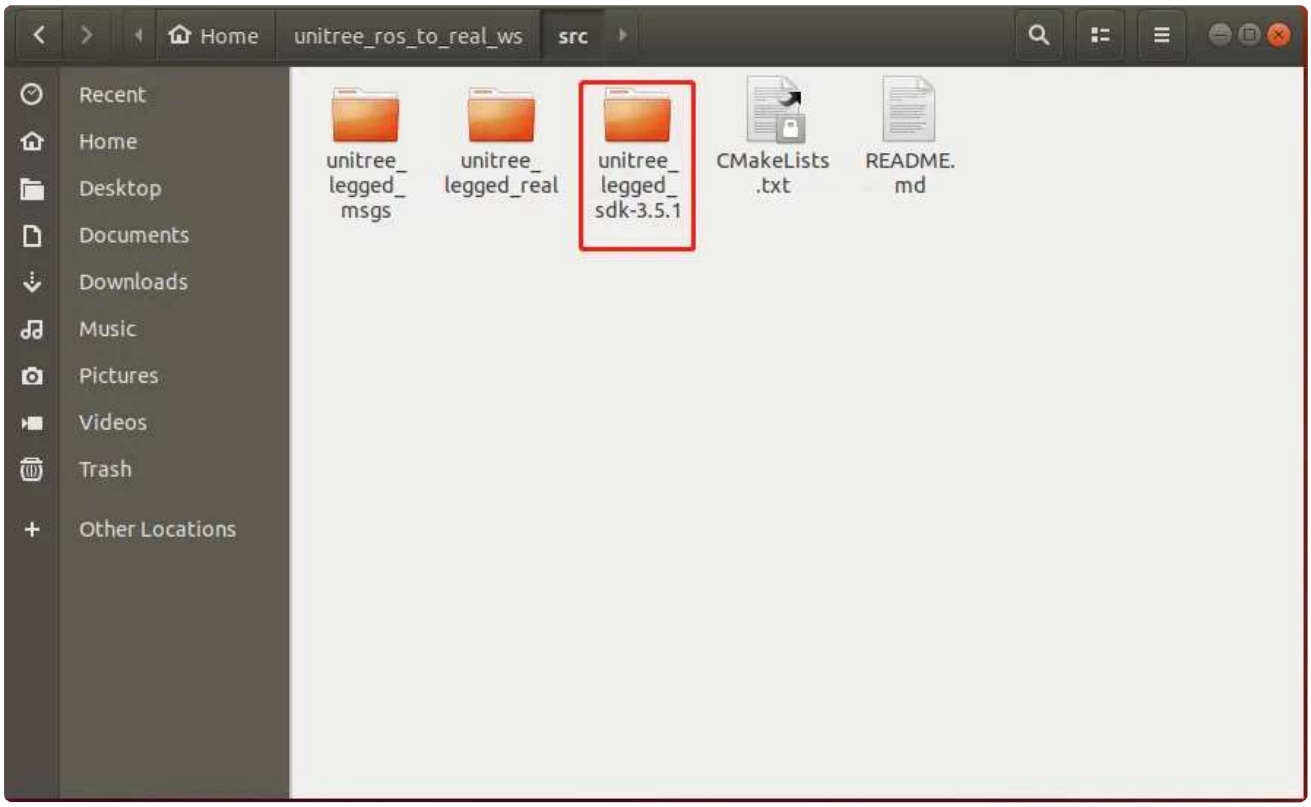
Step2. 将unitree_ros_to_real v3.5.0包里的所有内容和unitree_legged_sdk v3.5.1放到~/catkin_ws/src下。

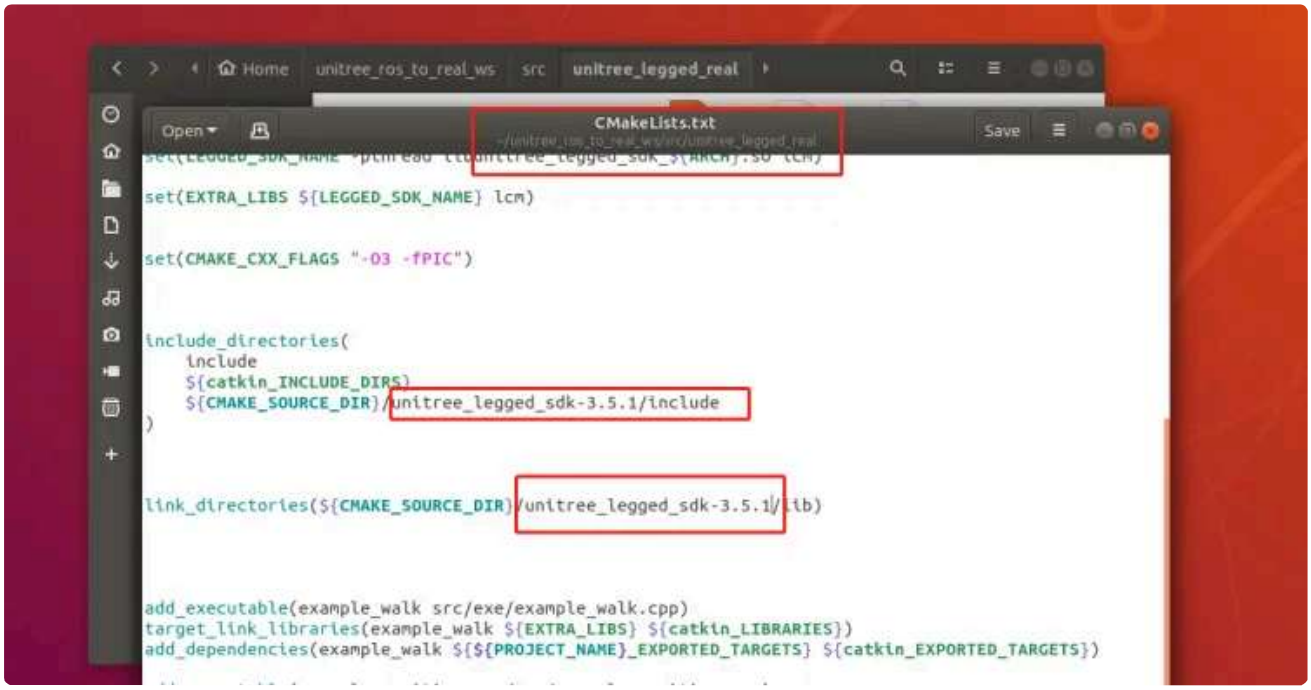
Step3. 根据自己联网的方式，配置自己的IP，修改~/catkin_ws/src/unitree_legged_real/src/exe/ros_udp.cpp文件中的high_udp的IP。有线方式配置192.168.123.161，无线方式配置192.168.12.1。



Step4. 修改自己下载unitree_legged_sdk v3.5.1的文件夹名和unitree_legged_real包的CMakeLists.txt中的路径一致。

1719 字





Step5. 编译。

```
cd ~/catkin_ws
catkin_make
```

Step6. 运行。可以根据README.md提供的运行示例尝试。

1719 字

README.md

Run the package

You can control your real Go1 robot from ROS by this package.

Before you run expamle program, please run command

```
roslaunch unitree_legged_real real.launch ctrl_level:=highlevel
```

or

```
roslaunch unitree_legged_real real.launch ctrl_level:=lowlevel
```

It depends which control mode you want to use.

Then, if you want to run high-level control mode, you can run example_walk node like this

```
rosrun unitree_legged_real example_walk
```

If you want to run low-level control mode, you can run example_position program node like this

```
rosrun unitree_legged_real example_postion
```

You can also run the node state_sub to subscribe the feedback information from Go1 robot

```
rosrun unitree_legged_real state_sub
```

You can also run the launch file that enables you control robot via keyboard like you can do in turtlesim package

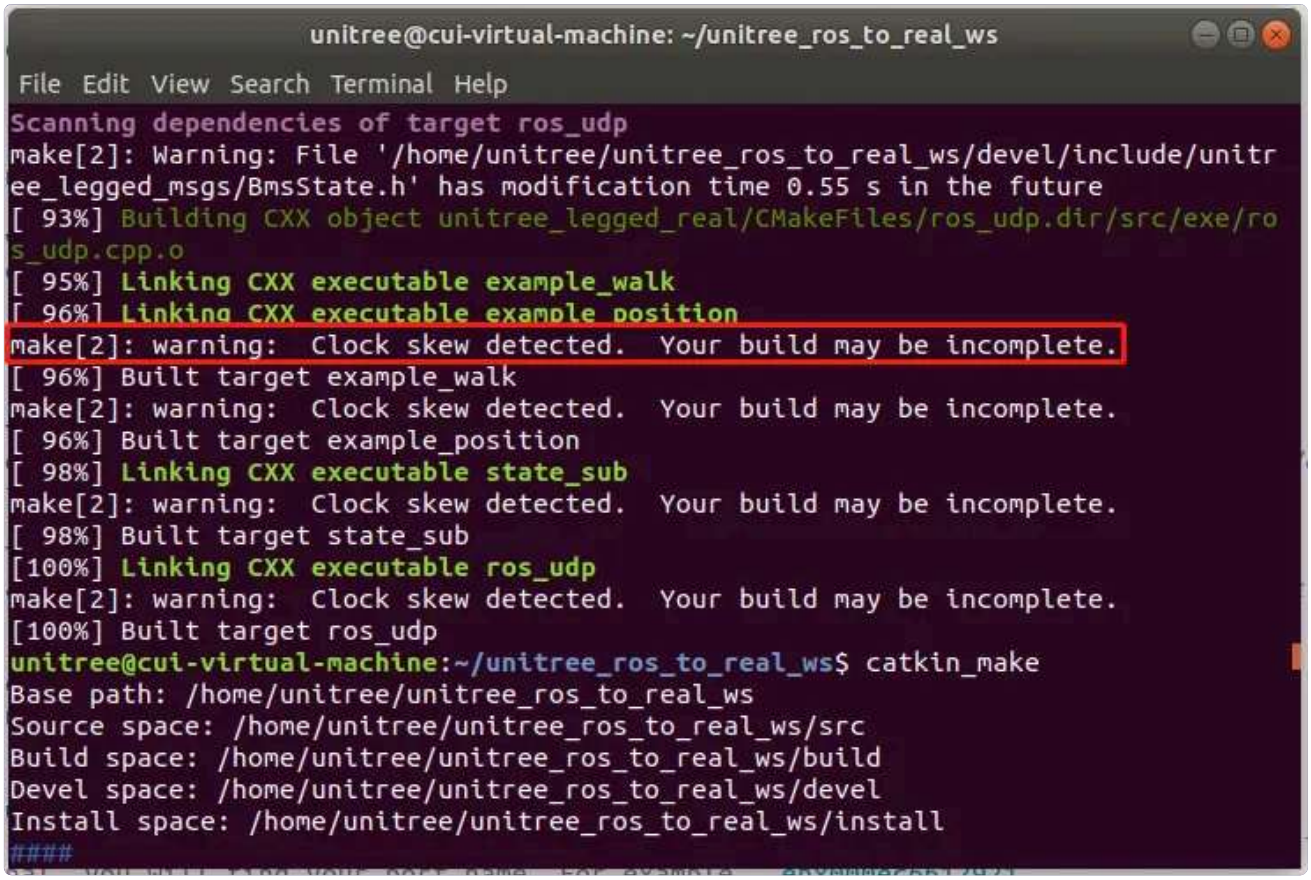
```
roslaunch unitree_legged_real keyboard_control.launch
```

And before you do the low-level control, please press L2+A to sit the robot down and then press L1+L2+start to make the robot into mode in which you can do joint-level control, finally make sure you hang the robot up before you run low-level control.

常见错误

1．Clock skew detected警告

警告：



解决办法：

https://blog.csdn.net/weixin_34191734/article/details/86430127

<https://blog.csdn.net/weixin_34191734/article/details/86430127>

2.4 unitree_ros_to_real v3.8.0

仅适用于Go1。 unitree_ros_to_real_ws-3.8.0.zip (497 KB)

操作方法同2.3节unitree_ros_to_real v3.5.0。

f70ddaf2965e.png&title=%E5%AE%87%E6%A0%91%E5%9B%9B%E8%B6%B3%E6%9C%BA%E5%9

1719 字