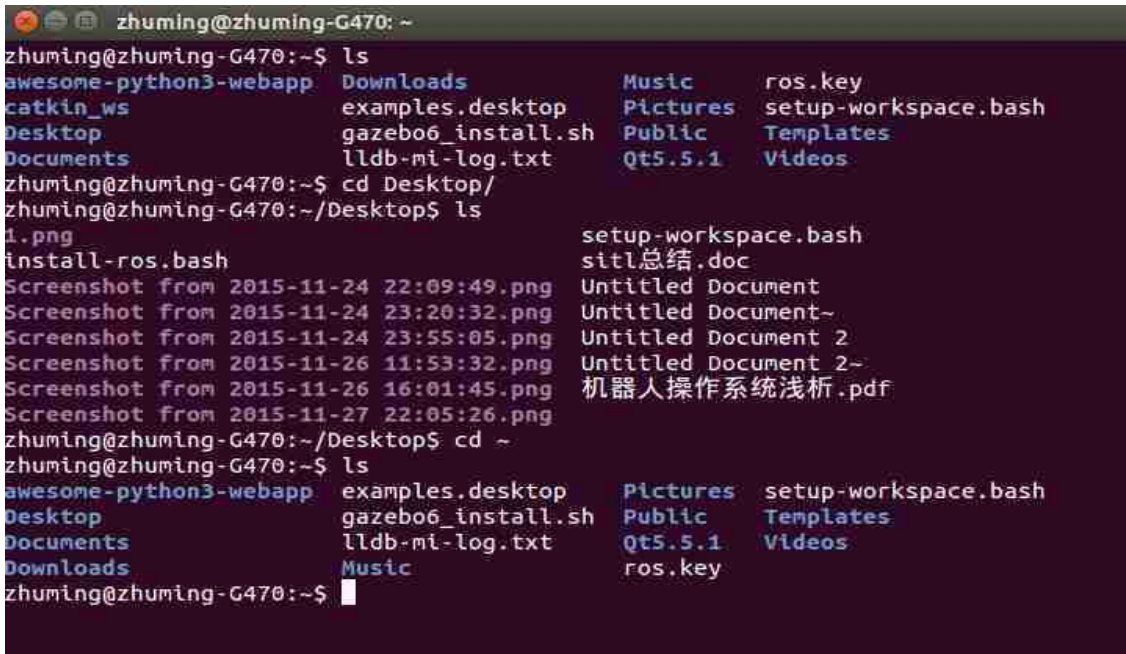


(一) 建立源代码

今天总结一下 sitl 仿真 px4 的过程，主要是在 ubuntu 上搭建一个仿真平台最后出现一个四旋翼模型，借助这个平台进行代码结构的学习，仿真，并逐步提高自身的技术能力。主要是根据 <http://dev.px4.io/index.html> 上的内容进行一步一步操作的，有一些难点。下面我在主文件夹中建立相应的文件夹并完成这个过程。

从 1.3 Building PX4 Software 开始:



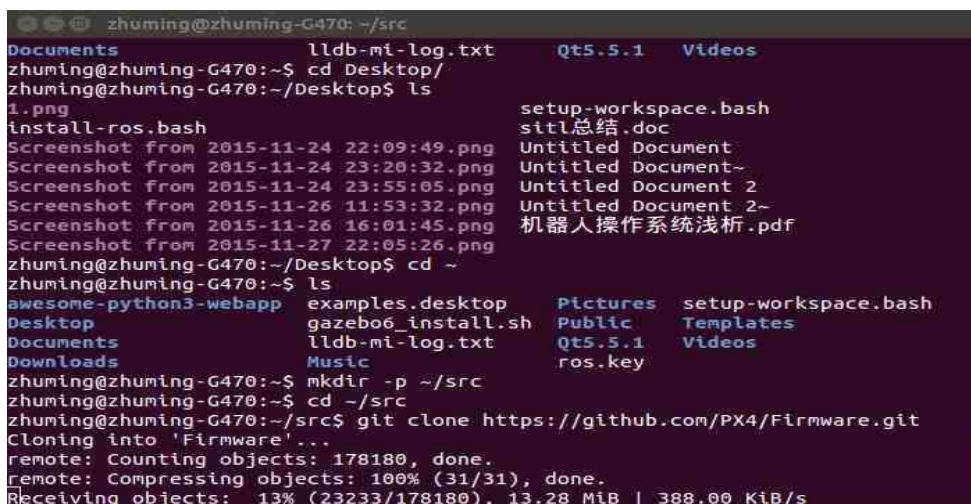
```
zhuming@zhuming-G470: ~  
zhuming@zhuming-G470:~$ ls  
awesome-python3-webapp  Downloads          Music             ros.key  
catkin_ws               examples.desktop  Pictures          setup-workspace.bash  
Desktop                 gazebo6_install.sh  Public           Templates  
Documents              lldb-mi-log.txt   Qt5.5.1         Videos  
zhuming@zhuming-G470:~$ cd Desktop/  
zhuming@zhuming-G470:~/Desktop$ ls  
1.png                  setup-workspace.bash  
install-ros.bash      sitl总结.doc  
Screenshot from 2015-11-24 22:09:49.png  Untitled Document  
Screenshot from 2015-11-24 23:20:32.png  Untitled Document~  
Screenshot from 2015-11-24 23:55:05.png  Untitled Document 2  
Screenshot from 2015-11-26 11:53:32.png  Untitled Document 2~  
Screenshot from 2015-11-26 16:01:45.png  机器人操作系统浅析.pdf  
Screenshot from 2015-11-27 22:05:26.png  
zhuming@zhuming-G470:~/Desktop$ cd ~  
zhuming@zhuming-G470:~$ ls  
awesome-python3-webapp  examples.desktop  Pictures          setup-workspace.bash  
Desktop                 gazebo6_install.sh  Public           Templates  
Documents              lldb-mi-log.txt   Qt5.5.1         Videos  
Downloads              Music             ros.key
```

这是桌面上的主文件显示，接下来建立 src 文件夹。复制以下代码到终端：



```
mkdir -p ~/src  
cd ~/src  
git clone https://github.com/PX4/Firmware.git
```

显示如下：



```
zhuming@zhuming-G470: ~/src  
Documents              lldb-mi-log.txt   Qt5.5.1         Videos  
zhuming@zhuming-G470:~$ cd Desktop/  
zhuming@zhuming-G470:~/Desktop$ ls  
1.png                  setup-workspace.bash  
install-ros.bash      sitl总结.doc  
Screenshot from 2015-11-24 22:09:49.png  Untitled Document  
Screenshot from 2015-11-24 23:20:32.png  Untitled Document~  
Screenshot from 2015-11-24 23:55:05.png  Untitled Document 2  
Screenshot from 2015-11-26 11:53:32.png  Untitled Document 2~  
Screenshot from 2015-11-26 16:01:45.png  机器人操作系统浅析.pdf  
Screenshot from 2015-11-27 22:05:26.png  
zhuming@zhuming-G470:~/Desktop$ cd ~  
zhuming@zhuming-G470:~$ ls  
awesome-python3-webapp  examples.desktop  Pictures          setup-workspace.bash  
Desktop                 gazebo6_install.sh  Public           Templates  
Documents              lldb-mi-log.txt   Qt5.5.1         Videos  
Downloads              Music             ros.key  
zhuming@zhuming-G470:~$ mkdir -p ~/src  
zhuming@zhuming-G470:~$ cd ~/src  
zhuming@zhuming-G470:~/src$ git clone https://github.com/PX4/Firmware.git  
Cloning into 'Firmware'...  
remote: Counting objects: 178180, done.  
remote: Compressing objects: 100% (31/31), done.  
Receiving objects: 13% (23233/178180), 13.28 MiB | 388.00 KiB/s
```

接下来，复制以下到终端：

```
cd Firmware
make px4fmu-v2_default
```

此过程可能出现一直停在地下那个界面不懂的问题，并有 ssh 相关的错误提示，在后续会有相应的解决办法。

```
Submodule 'src/modules/uavcan/libuavcan' (git://github.com/UAVCAN/libuavcan.git)
  registered for path 'src/modules/uavcan/libuavcan'
Submodule 'unittests/googletest' (https://github.com/google/googletest.git) reg
  istered for path 'unittests/googletest'
Cloning into 'Nuttx'...
```

上述内容完成后如图所示：

```
zhuming@zhuming-G470: ~/src/Firmware
n_params.c.obj]
[ 98%] Building CXX object src/modules/uavcan/CMakeFiles/modules__uavcan.dir/actuators/esc.cpp.obj
Linking CXX static library libuavcan_stm32_driver.a
[ 98%] Built target uavcan_stm32_driver
[ 98%] Building CXX object src/modules/uavcan/CMakeFiles/modules__uavcan.dir/sensors/sensor_bridge.cpp.obj
[ 98%] Building CXX object src/modules/uavcan/CMakeFiles/modules__uavcan.dir/sensors/gnss.cpp.obj
[100%] Building CXX object src/modules/uavcan/CMakeFiles/modules__uavcan.dir/sensors/mag.cpp.obj
[100%] Building CXX object src/modules/uavcan/CMakeFiles/modules__uavcan.dir/sensors/baro.cpp.obj
Linking CXX static library libmodules__uavcan.a
[100%] Built target modules__uavcan
Scanning dependencies of target firmware_nuttx
[100%] Building C object src/firmware/nuttx/CMakeFiles/firmware_nuttx.dir/built_in_commands.c.obj
Linking CXX executable firmware_nuttx
[100%] Built target firmware_nuttx
Scanning dependencies of target build_firmware_px4fmu-v2
[100%] Generating nuttx-px4fmu-v2-default.px4
[100%] Built target build_firmware_px4fmu-v2
zhuming@zhuming-G470:~/src/Firmware$
```

接下来可以在 qt 中查看源代码并进行自己的代码编写：

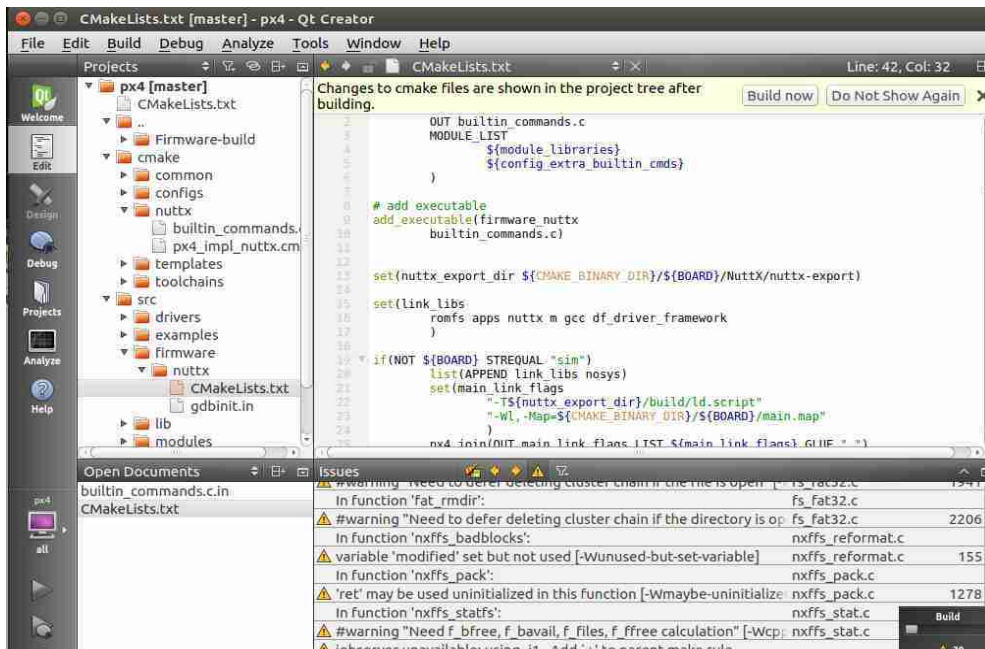
Qt Creator on Linux

NOTE: Linux users can just load the CMakeLists.txt in the root firmware folder via File -> Open File or Project -> Select the CMakeLists.txt file.

After loading, the 'play' button can be configured to run the project by selecting 'custom executable' in the run target configuration and entering 'make' as executable and 'upload' as argument.

Qt Creator on Windows

如图所示:



(二) 建立 gazebo 仿真
在左边栏中选中如下所示内容:



Installation

The installation requires to install Gazebo and our simulation plugin.

TIP: Mac OS users should install gazebo7 (not 6). Linux users: If you installed a ROS version earlier than Jade, be sure to uninstall the bundled Gazebo (`sudo apt-get remove ros-indigo-gazebo`) version as it is too old.

Follow the instruction in the [sitsl gazebo documentation](#) to install. Binaries are available for all major operating systems and can be installed via `apt-get` or `brew`.

点击如下图中的网页链接找到网页中下部的 Gazebo for PX4 SITL，可以看到，点击下面的链接，进入 gazebo 的安装配置：

Install Gazebo 6 Simulator

Follow instructions on the [official site](#) to install Gazebo version 6.

Installation

Instructions to install Gazebo on all the platforms supported: major linux distributions and Mac (windows is still under development).

Ubuntu 
Installation instructions for Ubuntu using packages (amd64, i386 or armhf)
Versions: 4.0 5.0 6.0

点击右侧的小眼睛，我们不使用在线安装，进行一步一步安装：

Step-by-step Install

1. Setup your computer to accept software from packages.osrfoundation.org.

Note: there is a list of [available mirrors](#) for this repository which could improve the download speed.

```
sudo sh -c 'echo "deb http://packages.osrfoundation.org/gazebo/ubuntu-stable `lsb_release -cs` main" > /etc/apt/sources.list.d/gazebo-stable.list'
```

2. Setup keys

```
wget http://packages.osrfoundation.org/gazebo.key -O - | sudo apt-key add -
```

3. Install Gazebo.

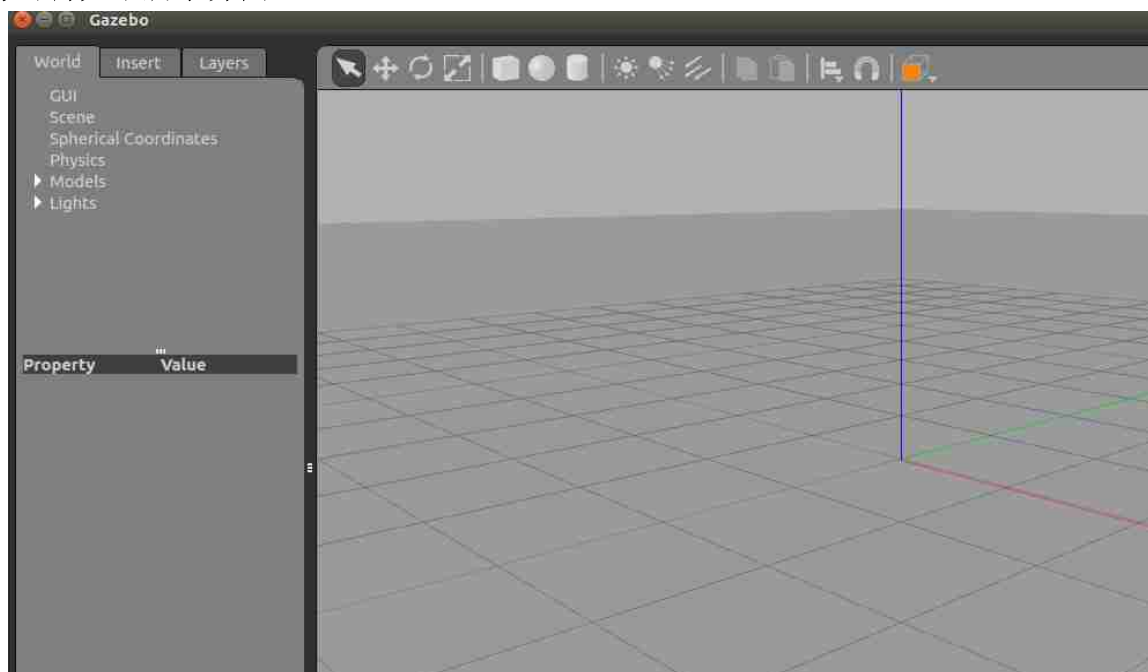
```
sudo apt-get update
sudo apt-get install gazebo6
# For developers that work on top of Gazebo, one extra package
sudo apt-get install libgazebo6-dev
```

4. Check your installation

Note The first time `gazebo` is executed requires the download of some models and it could take some time, please be patient.

```
gazebo
```

执行可得如下所示界面：



接下来，复制如下所示命令到终端：

Protobuf

Install the protobuf library, which is used as interface to Gazebo.

Ubuntu Linux

```
sudo apt-get install libprotobuf-dev libprotoc-dev protobuf-compiler
```

然后依次如下输入命令，注意看着页面输入命令的时候要做一定修改，注意到前面 `firmware` 文件夹也是下载到 `src`，有类似的命令，此处不宜直接新建文件夹，容易出错。执行 `cd ~/src` 代替以下图片中的前两行命令，然后执行 `git` 对应那条命令，如下。

Build Gazebo Plugins (all operating systems)

Clone the gazebo plugins repository to your computer. IMPORTANT: If you do not clone to `~/src/sitl_gazebo`, all remaining paths in these instructions will need to be adjusted.

```
mkdir -p ~/src
cd src
git clone https://github.com/PX4/sitl_gazebo.git
```

Create a build folder in the top level of your repository:

```
mkdir Build
```

注意，上面的 `Build` 文件夹在 `sitl_gazebo` 文件夹内部，从下面的图片中可以看出这一点，文件夹搞错会出现其他不必要的错误。

Next add the location of this build directory to your gazebo plugin path, e.g. add the following line to your `.bashrc` (Linux) or `.bash_profile` (Mac) file:

```
# Set the plugin path so Gazebo finds our model and sim
export GAZEBO_PLUGIN_PATH=${GAZEBO_PLUGIN_PATH}:$HOME/src/sitl_gazebo/Build
# Set the model path so Gazebo finds the airframes
export GAZEBO_MODEL_PATH=${GAZEBO_MODEL_PATH}:$HOME/src/sitl_gazebo/models
# Disable online model lookup since this is quite experimental and unstable
export GAZEBO_MODEL_DATABASE_URI=""
```

You also need to add the the root location of this repository, e.g. add the following line to your `.bashrc` (Linux) or `.bash_profile` (Mac) file:

```
# Set path to sitl_gazebo repository
export SITL_GAZEBO_PATH=$HOME/src/sitl_gazebo
```

添加以上内容后，文件 `.bashrc` 的内容如图所示（底部最后八行代码）：

```
zhuming@zhuming-G470: ~
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
source /opt/ros/indigo/setup.bash
source /home/zhuming/catkin_ws/devel/setup.bash

# Set the plugin path so Gazebo finds our model and sim
export GAZEBO_PLUGIN_PATH=${GAZEBO_PLUGIN_PATH}:$HOME/src/sitl_gazebo/Build
# Set the model path so Gazebo finds the airframes
export GAZEBO_MODEL_PATH=${GAZEBO_MODEL_PATH}:$HOME/src/sitl_gazebo/models
# Disable online model lookup since this is quite experimental and unstable
export GAZEBO_MODEL_DATABASE_URI=""

# Set path to sitl_gazebo repository
export SITL_GAZEBO_PATH=$HOME/src/sitl_gazebo
```

126,1 Bot

Autogenerate the sdf file with the command

```
make sdf
```

Now build the gazebo plugins by typing:

```
make
```

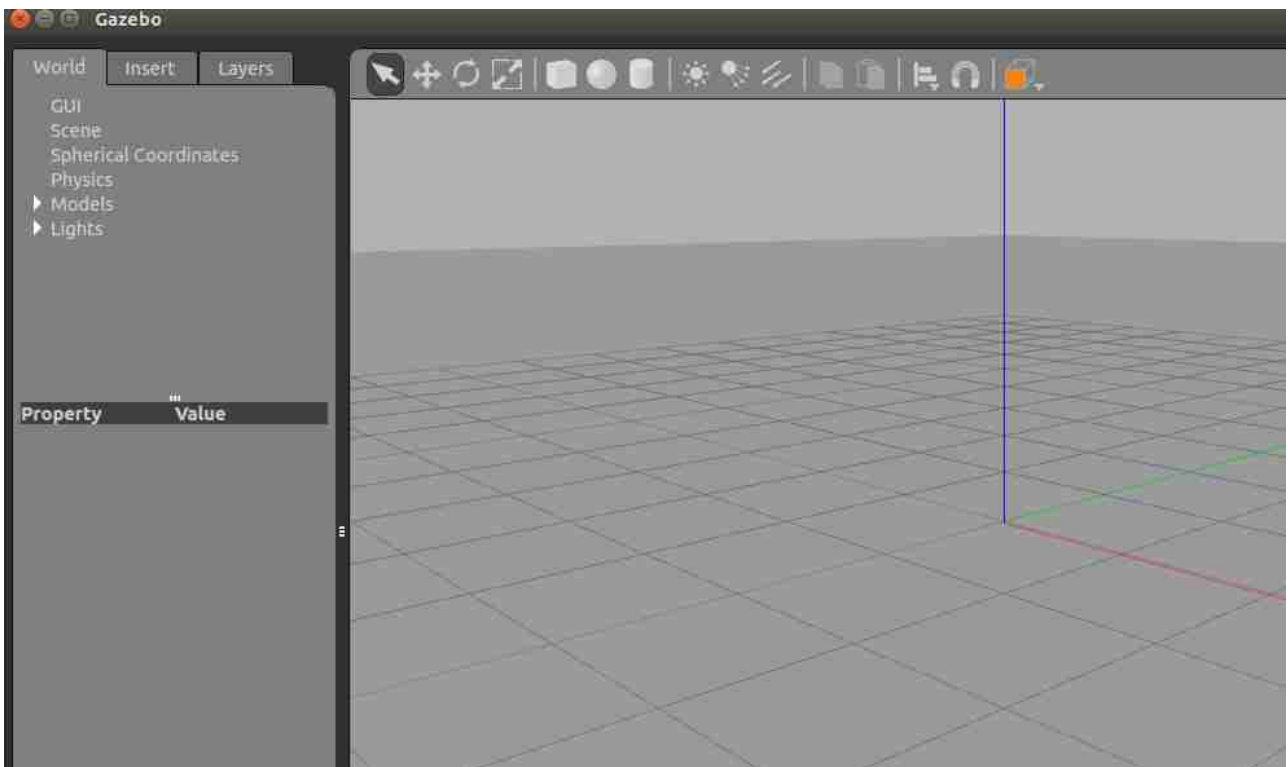
Gazebo will now launch when typing 'gazebo' on the shell:

```
gazebo
```

按如上所示输入 make 即可出现一下界面:


```
zhuming@zhuming-G470: ~/src/sitl_gazebo/Build
'void gazebo::physics::Joint::SetAngle(unsigned int, gazebo::math::Angle)' is de
precated (declared at /usr/include/gazebo-6.5/gazebo/physics/Joint.hh:396) [-Wde
precated-declarations]
    right_elevon_joint_ ->SetAngle(0, -inputs.control[0]);
                                ^
/home/zhuming/src/sitl_gazebo/src/gazebo_mavlink_interface.cpp:429:53: warning:
'void gazebo::physics::Joint::SetAngle(unsigned int, gazebo::math::Angle)' is de
precated (declared at /usr/include/gazebo-6.5/gazebo/physics/Joint.hh:396) [-Wde
precated-declarations]
    elevator_joint_ ->SetAngle(0, inputs.control[1]);
                                ^
Linking CXX shared library librotors_gazebo_mavlink_interface.so
[ 93%] Built target rotors_gazebo_mavlink_interface
Scanning dependencies of target rotors_gazebo_motor_model
[ 96%] Building CXX object CMakeFiles/rotors_gazebo_motor_model.dir/src/gazebo_m
otor_model.cpp.o
Linking CXX shared library librotors_gazebo_motor_model.so
[ 96%] Built target rotors_gazebo_motor_model
Scanning dependencies of target rotors_gazebo_multirotor_base_plugin
[100%] Building CXX object CMakeFiles/rotors_gazebo_multirotor_base_plugin.dir/s
rc/gazebo_multirotor_base_plugin.cpp.o
Linking CXX shared library librotors_gazebo_multirotor_base_plugin.so
[100%] Built target rotors_gazebo_multirotor_base_plugin
zhuming@zhuming-G470:~/src/sitl_gazebo/Build$
```

输入 gazebo 后，即可得如下所示：



（三）在 gazebo 中显示四旋翼

在页面 <http://dev.px4.io/simulation-gazebo.html> 中跳到 Running the Simulation，并输入以下代码：

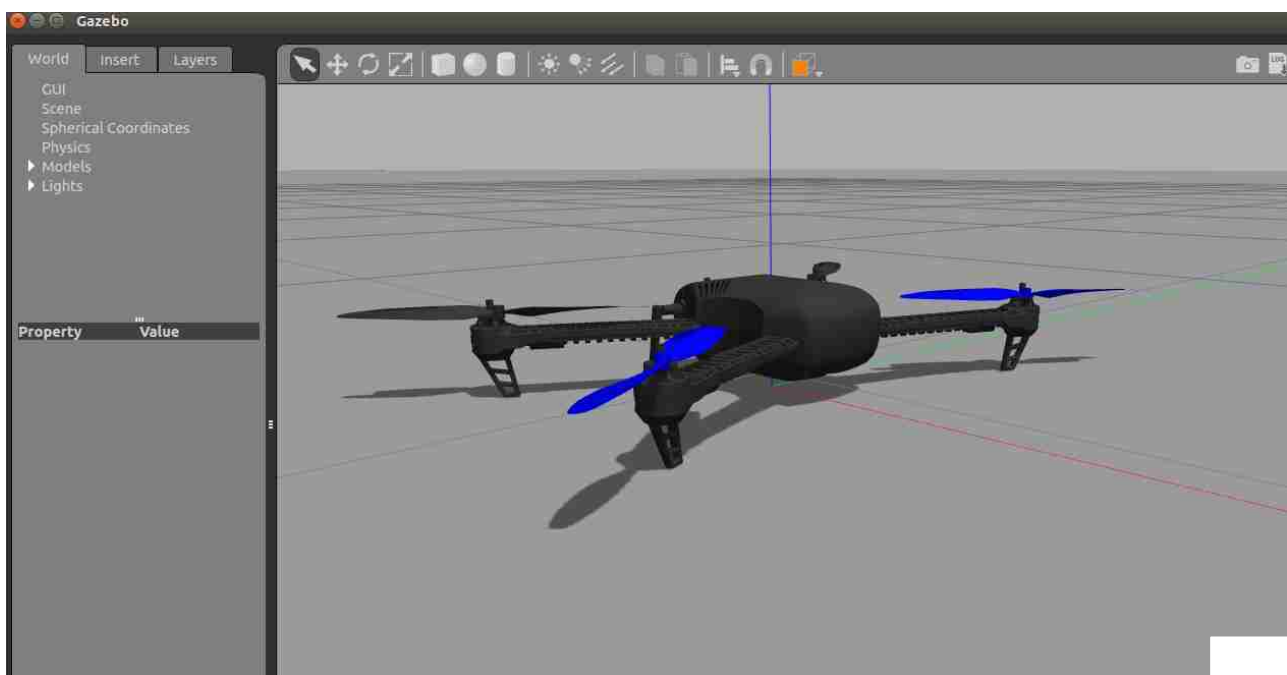
Run the PX4 SITL with the Iris configuration in the Firmware directory:

```
cd ~/src/Firmware
make posix_sitl_default gazebo
```

出现以下过程:

```
9c62b7b70eb94'
Submodule path 'Tools/sitl_gazebo': checked out '03a749a3b3238c014329e9e8da57d16
2ede6f4f0'
Submodule path 'mavlink/include/mavlink/v1.0': checked out '86a2b147e4cec5066efd
c6d030d69d902929e66f'
Submodule path 'src/lib/DriverFramework': checked out '2ae8d8118db0e95867cd1946d
5b8b85d7e4ef9d3'
Submodule path 'src/lib/dspal': checked out '94188fb83ab6b2da59eb9a8c8596ab49b67
c42f6'
Submodule path 'src/lib/matrix': checked out '9cd6ac3dd9a255a1a2ef299da413c38a9e
40b8c7'
Submodule path 'src/modules/uavcan/libuavcan': checked out 'ed1d71e639543ea5743a
839c313c53237ab3a27d'
Submodule path 'src/modules/uavcan/libuavcan/dsdl': checked out '9804a3e69728255
86be252ce08dd899f44994b14'
Submodule path 'src/modules/uavcan/libuavcan/libuavcan/dsdl_compiler/pyuavcan':
checked out 'c58477a644d20ccf95a20c151f3a0402f271c3b8'
Submodule path 'src/modules/uavcan/libuavcan/libuavcan/dsdl_compiler/pyuavcan/ds
dl': checked out 'fd12483ddd4e58242d61d74a163e7aeea1e0f466'
```

执行完毕后出现飞机模型:



不过这次出现以下错误提示，带后续研究，也请高手指点一二。

```
zhuming@zhuming-G470: ~/src/Firmware
INFO ToneAlarm::start_note 6384
INFO ToneAlarm::start_note 7166
WARN init ref: lat=47.3666999, lon=8.5500000, alt= 0.0540
telem> home: 47.3666999, 8.5500000, 0.06
INFO TONE_SET_ALARM 15
INFO ToneAlarm::start_tune
INFO TONE_SET_ALARM 0
[Err] [InsertModelWidget.cc:298] Missing model.config for model "/home/zhuming/s
rc/sitl_gazebo/models/rotors_description"
[Err] [InsertModelWidget.cc:298] Missing model.config for model "/home/zhuming/s
rc/Firmware/Tools/sitl_gazebo/models/rotors_description"
[Wrn] [ModelDatabase.cc:211] Unable to connect to model database using [//databa
se.config]. Only locally installed models will be available.
INFO TONE_SET_ALARM 4
INFO ToneAlarm::start_tune
INFO ToneAlarm::start_note 11375
INFO ToneAlarm::start_note 11375
INFO TONE_SET_ALARM 0
WARN mavlink sim timeout for 10 ms
WARN mavlink sim timeout for 10 ms
WARN mavlink sim timeout for 10 ms
WARN mavlink sim timeout for 10 ms
WARN mavlink sim timeout for 10 ms
```

SSH 问题解决

转载: [ubuntu 下 github connect 错误解决](#)

<http://www.douban.com/note/303388058/>

Generating SSH keys

<https://help.github.com/articles/generating-ssh-keys/>

欢饮加入飞控漫谈交流群239920039，一起打造开源飞控