通过安装包安装：

1、下载链接：

[Index of /archive/qt/5.9/5.9.9]()

2、离线安装

1. 给予权限

```
chmod a+x qt-opensource-linux-x64-5.9.9.run
```

2. 双击运行或者 `./qt-opensource-linux-x64-5.9.9.run`
3. 进行选择配置

> 路径选择家目录或者/opt目录
>
> 组件选择 `Desktop gcc 64-bit`
>
> 许可协议选择LGPL

4. 点击安装，等待安装完毕

3、添加qtcreator环境变量

1. 编辑/usr/bin/qtcreator，修改为实际的安装路径：

   ```
    sudo nano /usr/bin/qtcreator
   ```

   写入

   ```
   #!/bin/sh
   export QT_HOME=/opt/Qt5.9.9/Tools/QtCreator/bin
   $QT_HOME/qtcreator $*
   ```

   或者

   ```
   #!/bin/sh
   export QT_HOME=/opt/Qt5.9.9/Tools/QtCreator/bin
   $QT_HOME/qtcreator $*
   ```

2. 更改权限

   ```
   sudo chmod a+x /usr/bin/qtcreator
   ```

3. 终端运行

   ```
   qtcreator
   ```

4、安装依赖

```
sudo apt-get install ros-melodic-qt-create
```

```
sudo apt-get install ros-melodic-qt-build
```

5、创建功能包运行

```
mkdir catkin_qt&&cd catkin_qt&&mkdir src&&cd src
```

```
catkin_create_qt_pkg class1_ros_qt_demo roscpp rviz
```

若编译出错可进行修改class1_ros_qt_demo的CMakeLists.txt，没有出错则无需修改

在CMake下方设置包含当前目录

```
set(CMAKE_INCLUDE_CURRENT_DIR ON)
```

在qt build下方添加qt库

```
find_package(Qt5 REQUIRED Core Widgets)
set(QT_LIBRARIES Qt5::Widgets)
```

注释改行如下：

```
#rosbuild_prepare_qt4(QtCore QtGui)
```

将QT4

```
QT4_ADD_RESOURCES(QT_RESOURCES_CPP ${QT_RESOURCES})
QT4_WRAP_UI(QT_FORMS_HPP ${QT_FORMS})
QT4_WRAP_CPP(QT_MOC_HPP ${QT_MOC})
```

改成QT5

```
QT5_ADD_RESOURCES(QT_RESOURCES_CPP ${QT_RESOURCES})
QT5_WRAP_UI(QT_FORMS_HPP ${QT_FORMS})
QT5_WRAP_CPP(QT_MOC_HPP ${QT_MOC})
```

修改~/catkin_qt/src/class1_ros_qt_demo/include/class1_ros_qt_demo/main_window.hpp

将

```
#include <QtGui/QMainWindow>
```

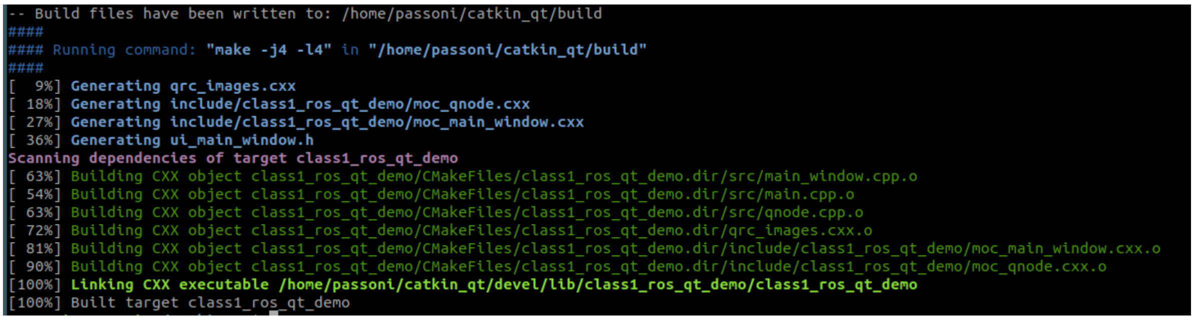
改成

```
#include <QtWidgets/QMainWindow>
```

如遇QtGui相关报错，则将报错相关头文件的QtGui均改为QtWidgets

如遇 'UnicodeUTF8' is not a member of 'QApplication'，因为UnicodeUTF8已经启用，直接删除即可

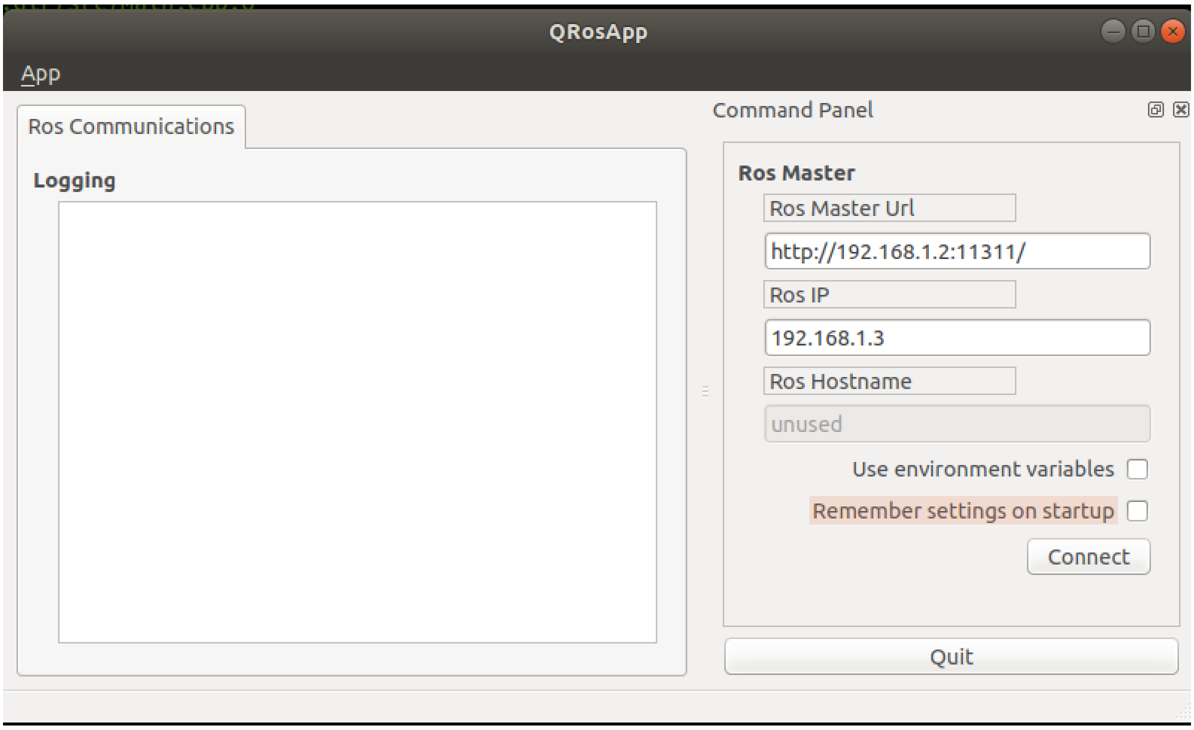
重新编译



```
-- Build files have been written to: /home/passoni/catkin_qt/build
####
#### Running command: "make -j4 -l4" in "/home/passoni/catkin_qt/build"
####
[  9%] Generating qrc_images.cxx
[ 18%] Generating include/class1_ros_qt_demo/moc_qnode.cxx
[ 27%] Generating include/class1_ros_qt_demo/moc_main_window.cxx
[ 36%] Generating ui_main_window.h
Scanning dependencies of target class1_ros_qt_demo
[ 63%] Building CXX object class1_ros_qt_demo/CMakeFiles/class1_ros_qt_demo.dir/src/main_window.cpp.o
[ 54%] Building CXX object class1_ros_qt_demo/CMakeFiles/class1_ros_qt_demo.dir/src/main.cpp.o
[ 63%] Building CXX object class1_ros_qt_demo/CMakeFiles/class1_ros_qt_demo.dir/src/qnode.cpp.o
[ 72%] Building CXX object class1_ros_qt_demo/CMakeFiles/class1_ros_qt_demo.dir/qrc_images.cxx.o
[ 81%] Building CXX object class1_ros_qt_demo/CMakeFiles/class1_ros_qt_demo.dir/include/class1_ros_qt_demo/moc_main_window.cxx.o
[ 90%] Building CXX object class1_ros_qt_demo/CMakeFiles/class1_ros_qt_demo.dir/include/class1_ros_qt_demo/moc_qnode.cxx.o
[100%] Linking CXX executable /home/passoni/catkin_qt/devel/lib/class1_ros_qt_demo/class1_ros_qt_demo
[100%] Built target class1_ros_qt_demo
```

6、打开工程

```
rosrun class1_ros_qt_demo class1_ros_qt_demo
```



Ros Master Url: http://192.168.1.2:11311/

Ros IP: 192.168.1.3

Ros Hostname: unused

Use environment variables □

Remember settings on startup □

在点击连接connect之前，要先确定roscore，因为要配置多机通信。

6、后续其他依赖

编译功能包遇到这个报错

```
CMake Error at /usr/lib/x86_64-linux-gnu/cmake/Qt5/Qt5Config.cmake:28
(find_package):
  Could not find a package configuration file provided by "Qt5Multimedia"
  with any of the following names:

    Qt5MultimediaConfig.cmake
    qt5multimedia-config.cmake

  Add the installation prefix of "Qt5Multimedia" to CMAKE_PREFIX_PATH or set
  "Qt5Multimedia_DIR" to a directory containing one of the above files.  If
  "Qt5Multimedia" provides a separate development package or SDK, be sure it
  has been installed.
```

执行

```
sudo apt-get install qtmultimedia5-dev
```

7、切换分支

Go to the branch 转到分支

```
cd Ros_Qt5_Gui_App
git checkout rviz_tree
```

git总结 —— 切换到分支时提示提交或暂存修改