



# 华清远见

## QT 开发文档

(V0.1)

华清远见教育集团 • 研发中心



4007061880



微信

farsight2013



<http://dev.hqyj.com>



[support@farsight.com.cn](mailto:support@farsight.com.cn)





# 目录

目录 .....	I
1 Qt 开发环境搭建 .....	1
1.1 主机开发环境说明 .....	1
1.2 主机 Qt 开发环境搭建 .....	1
1.3 Qt-everywhere 移植 .....	5
1.4 Qt-Creater 设置 .....	7
1.5 创建第一个工程 HelloWorld .....	14
版本历史: .....	22



## 1 Qt 开发环境搭建

### 1.1 主机开发环境说明

- 1) Linux 为华清远见开发环境 V12B(Ubuntu 12.04)
- 2) 主机 gcc 版本为 gcc-4.6.3
- 3) 交叉工具链版本为 arm-none-linux-gnueabi-gcc-4.6.4  
工具链使用需要统一, 可根据实际情况切换工具链版本

### 1.2 主机 Qt 开发环境搭建

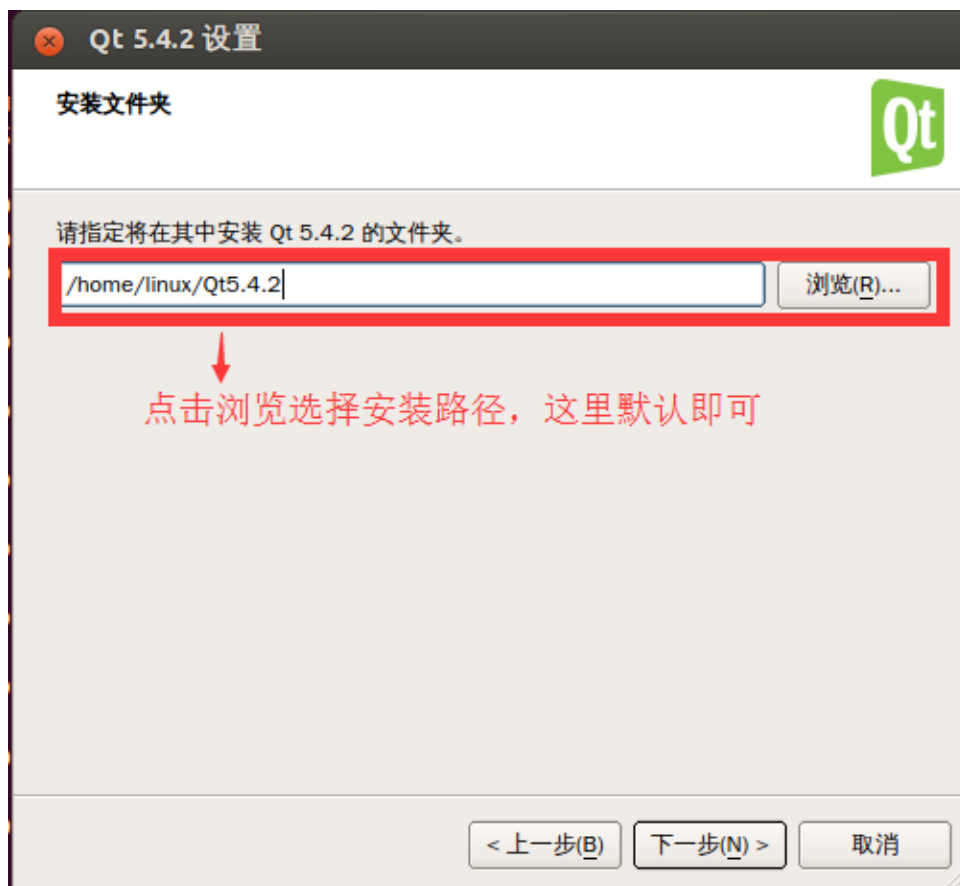
- 4) 将 qt-opensource-linux-x64-5.4.2.run 拷贝到 Linux 任意目录下
- 5) 安装 Qt 开发环境

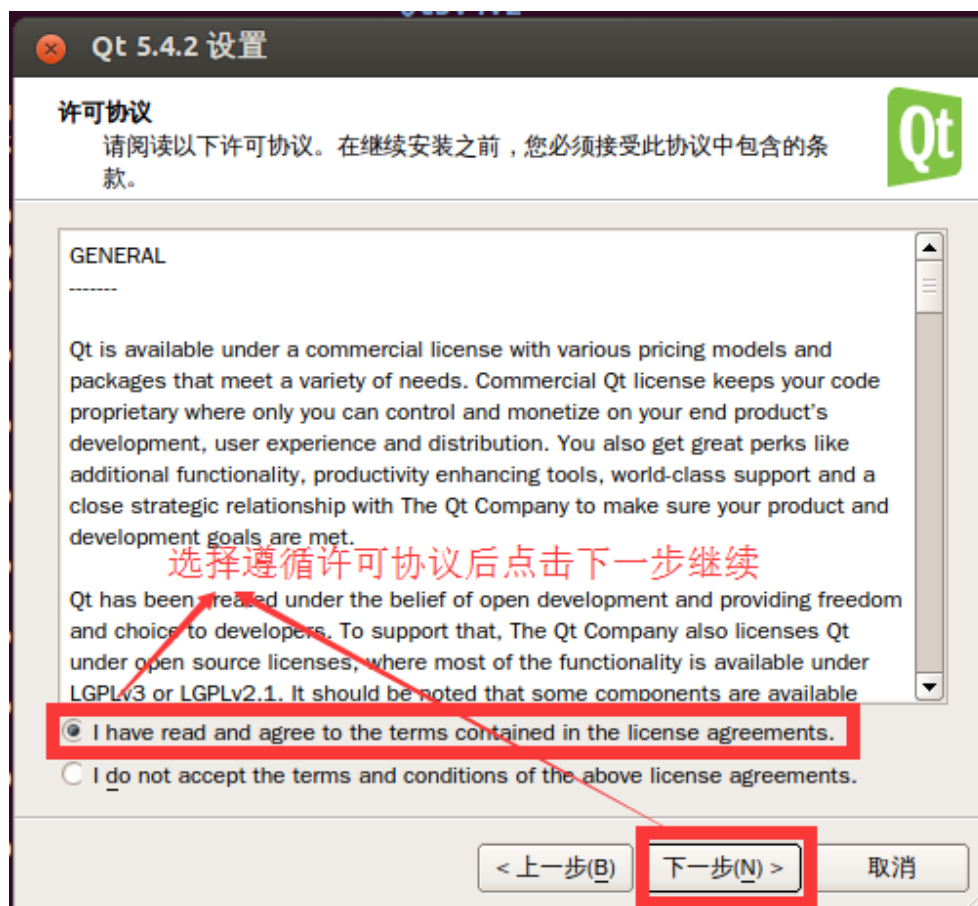
执行:

```
$ chmod 777 ./qt-opensource-linux-x64-5.4.2.run  
$ ./qt-opensource-linux-x64-5.4.2.run
```

执行上述操作后, 会弹出如下界面, 按提示安装即可。

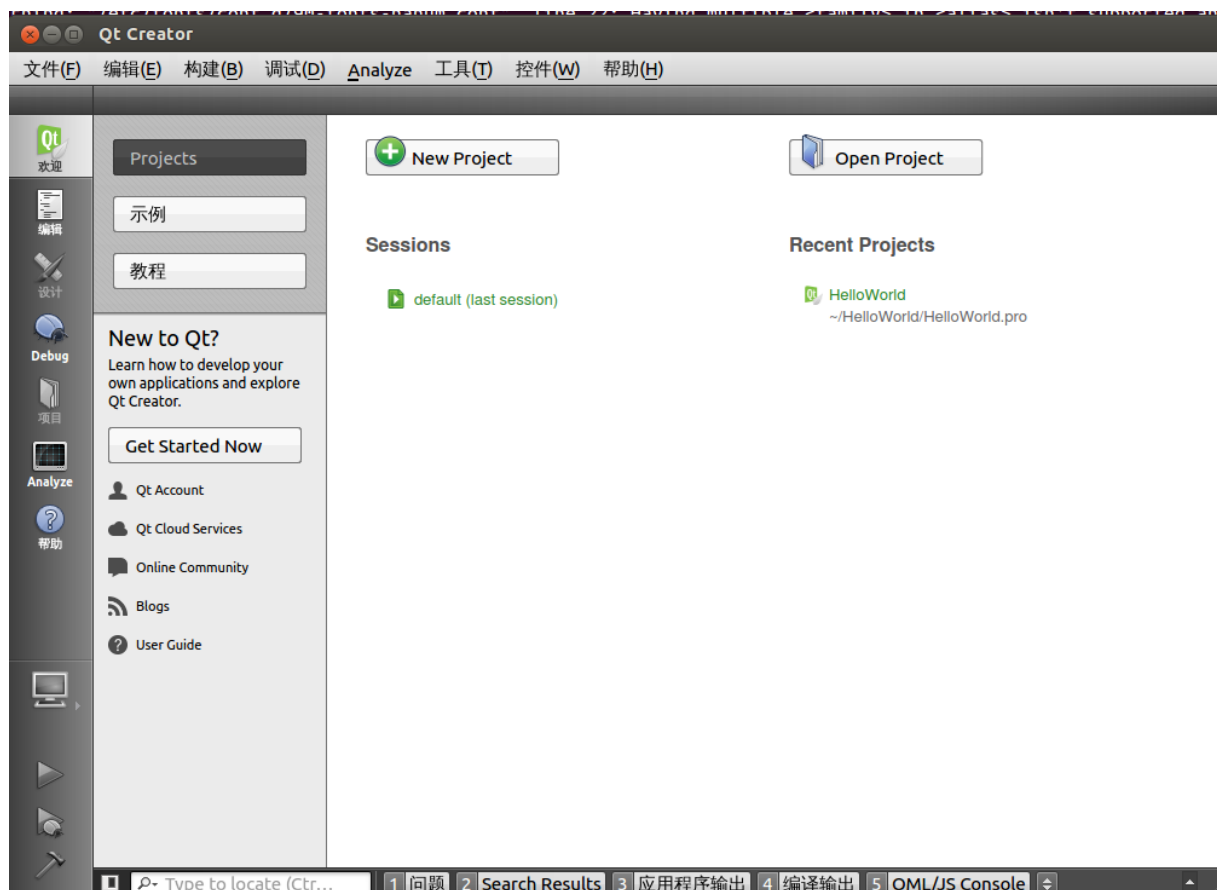








安装后开发界面如下：





## 1.3 Qt-everywhere 移植

1) 拷贝源码 qt-everywhere-opensource-src-5.3.1.tar.xz 到 Linux 任意目录下

2) 解压源码

执行:

```
$ tar xvf qt-everywhere-opensource-src-5.3.1.tar.xz
```

3) 修改源码

修改源码中 qtbase/mkspecs/linux-arm-gnueabi-g++/qmake.conf 内容

```
13 # modifications to g++.conf
14 QMAKE_CC           = arm-linux-gnueabi-gcc
15 QMAKE_CXX          = arm-linux-gnueabi-g++
16 QMAKE_LINK         = arm-linux-gnueabi-g++
17 QMAKE_LINK_SHLIB   = arm-linux-gnueabi-g++
18
19 # modifications to linux.conf
20 QMAKE_AR            = arm-linux-gnueabi-ar cqs
21 QMAKE_OBJCOPY       = arm-linux-gnueabi-objcopy
22 QMAKE_NM            = arm-linux-gnueabi-nm -P
23 QMAKE_STRIP         = arm-linux-gnueabi-strip
```

为

```
13 # modifications to g++.conf
14 QMAKE_CC           = arm-none-linux-gnueabi-gcc
15 QMAKE_CXX          = arm-none-linux-gnueabi-g++
16 QMAKE_LINK         = arm-none-linux-gnueabi-g++
17 QMAKE_LINK_SHLIB   = arm-none-linux-gnueabi-g++
18
19 # modifications to linux.conf
20 QMAKE_AR            = arm-none-linux-gnueabi-ar cqs
21 QMAKE_OBJCOPY       = arm-none-linux-gnueabi-objcopy
22 QMAKE_NM            = arm-none-linux-gnueabi-nm -P
23 QMAKE_STRIP         = arm-none-linux-gnueabi-strip
```

4) 拷贝 configure.sh 到解压后的源码目录下

5) 配置源码

执行:

```
$ chmod 777 configure.sh
```





```
$ ./configure.sh
```

输入"yes"后按回车继续

```
You are licensed to use this software under the terms of
the Lesser GNU General Public License (LGPL) versions 2.1.
You are also licensed to use this software under the terms of
the GNU General Public License (GPL) versions 3.

Type '3' to view the GNU General Public License version 3.
Type 'L' to view the Lesser GNU General Public License version 2.1.
Type 'yes' to accept this license offer.
Type 'no' to decline this license offer.

Do you accept the terms of either license? 
```

输入"yes"后按回车继续

属性如下内容便是配置成功

```
NOTE: Qt is using double for qreal on this system. This is binary incompatible against Qt 5.1.
Configure with '-qreal float' to create a build that is binary compatible with 5.1.
Info: creating super cache file /home/linux/Qt/qt-everywhere-opensource-src-5.3.1/qmake.super

Qt is now configured for building. Just run 'make'.
Once everything is built, you must run 'make install'.
Qt will be installed into /opt/qt

Prior to reconfiguration, make sure you remove any leftovers from
the previous build.
```

#### 6) 编译安装源码

执行:

```
添加工具链路径到环境变量中, 本例使用工具链版本为 4.6.4
$ export PATH=$PATH: /usr/local/toolchain/toolchain-4.6.4/bin/
$ make
```

编译过程中没有提示错误即可实行下边操作

```
$ sudo make install
```

安装后会在/opt/目录下生成 qt 目录。

#### 7) 根文件系统修改

首先需要有一个可以使用的根文件系统, 标准 Linux 根文件系统即可。

解压标准根文件 rootfs.tar.xz 到 Linux 下/source/目录下

```
$ tar xvf rootfs.tar.xz -C /source
```

将上文编译好的 qt 库拷贝到根文件系统中

```
$ cp /opt/qt /source/rootfs -a
```

修改根文件系统中的 profile 文件添加如下内容:



```
$ vi /source/rootfs/etc/profile
```

添加如下内容:

```
export QTDIR=/opt/qt
export PATH=$QTDIR:$PATH
export QT_QPA_PLATFORM_PLUGIN_PATH=$QTDIR/plugin
export QT_QPA_PLATFORM=linuxfb:tty=/dev/fb0
export QT_QPA_FONTDIR=$QTDIR/lib/fonts
export TSDEVICE=/dev/input/event4
export QT_QPA_GENERIC_PLUGINS=evdevtouch:$TSDEVICE
export QWS_MOUSE_PROTO=evdevtouch:$TSDEVICE
export set QWS_DISPLAY=LinuxFb:/dev/fb0
#export set QWS_SIZE=800x600
export set QWS_SIZE=1024x600
export set QWS_DISPLAY="LinuxFb:mmWidth300:mmHeight400:0"
export set QT_QWS_FONTDIR=$QTDIR/lib/fonts/
export set

LD_LIBRARY_PATH=$QTDIR:$QTDIR/lib/:$QT_QPA_PLATFORM_PLUGIN_PATH:$QT_QPA_F
ONTDIR:$LD_LIBRARY_PATH
```

## 1.4 Qt-Creater 设置

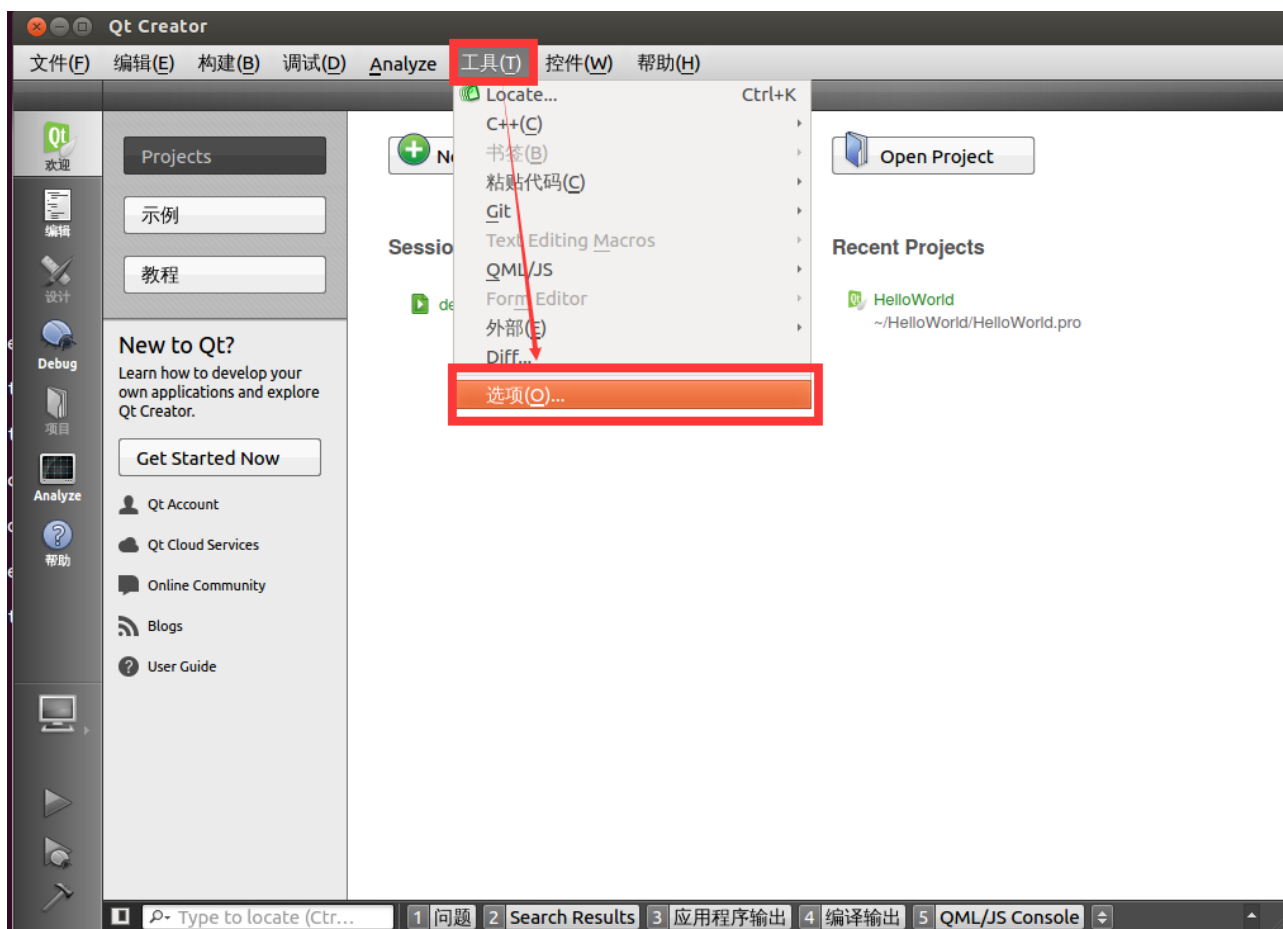
### 1) 打开 qtcreator

按照前文安装, qtcreator 所在路径为: /home/linux/Qt5.4.2/Tools/QtCreator/bin

执行

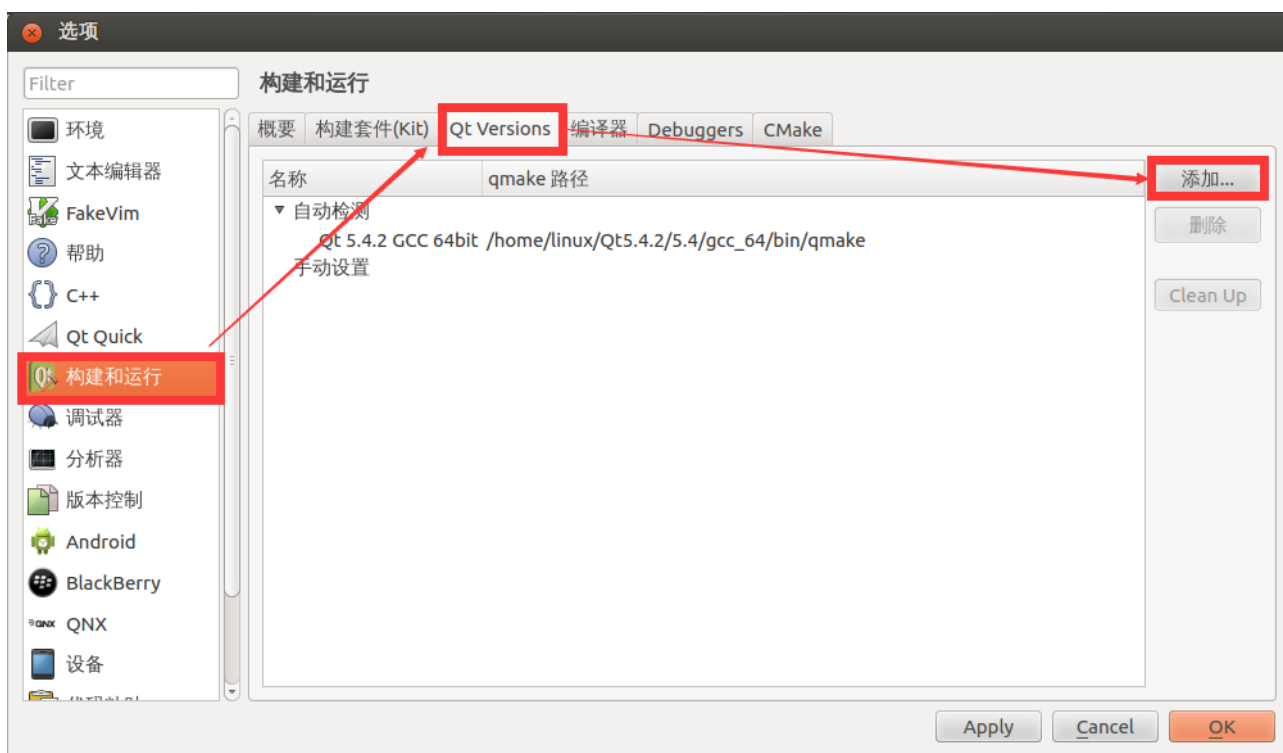
```
$ cd /home/linux/Qt5.4.2/Tools/QtCreator/bin
$ ./qtcreator
```

在主界面中从菜单栏点击“工具”→“选项”



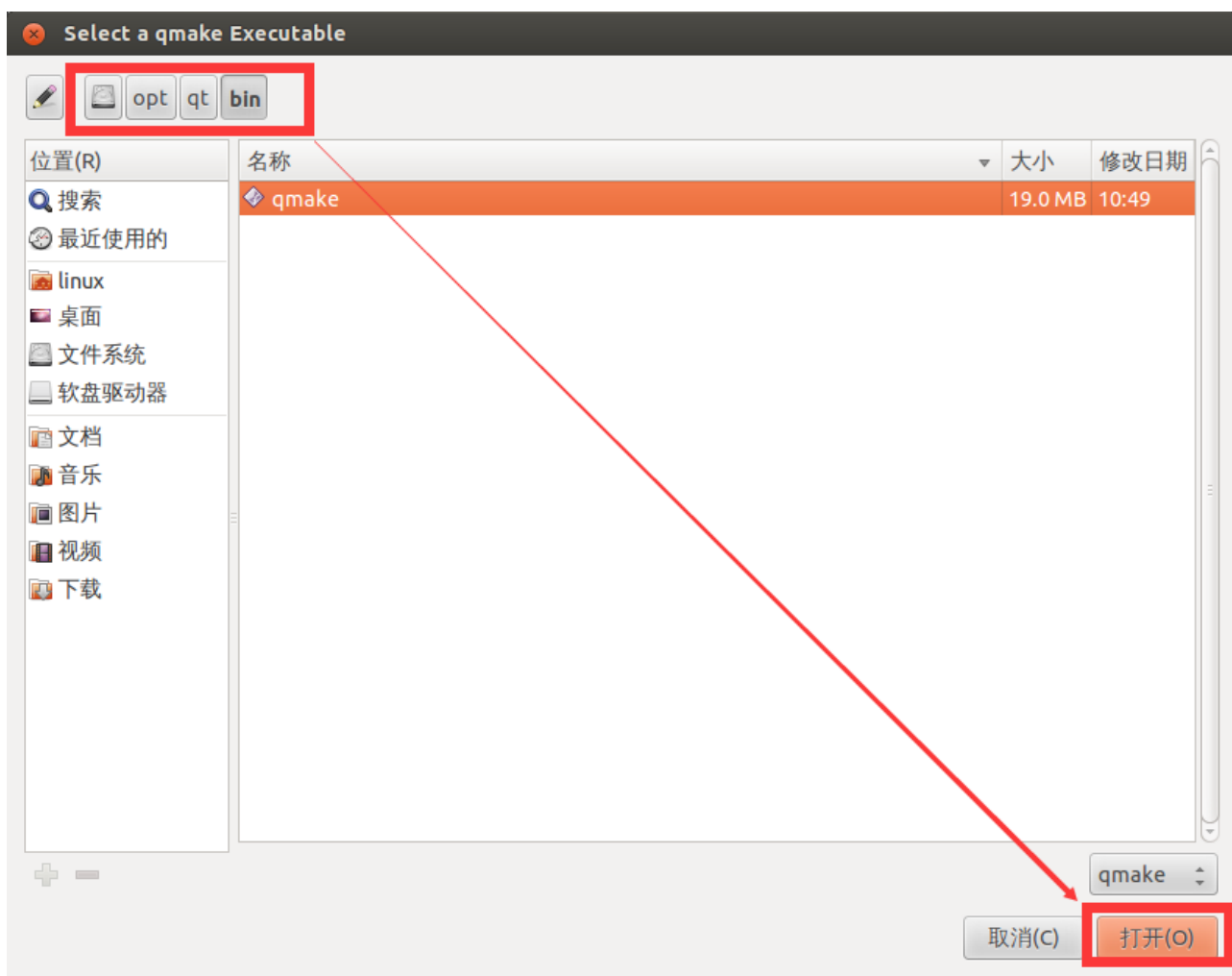
## 2) 配置 qt 版本

点击选项界面左侧“构建和运行”后选择“Qt Versions”选项卡最后点击右侧“添加”按钮

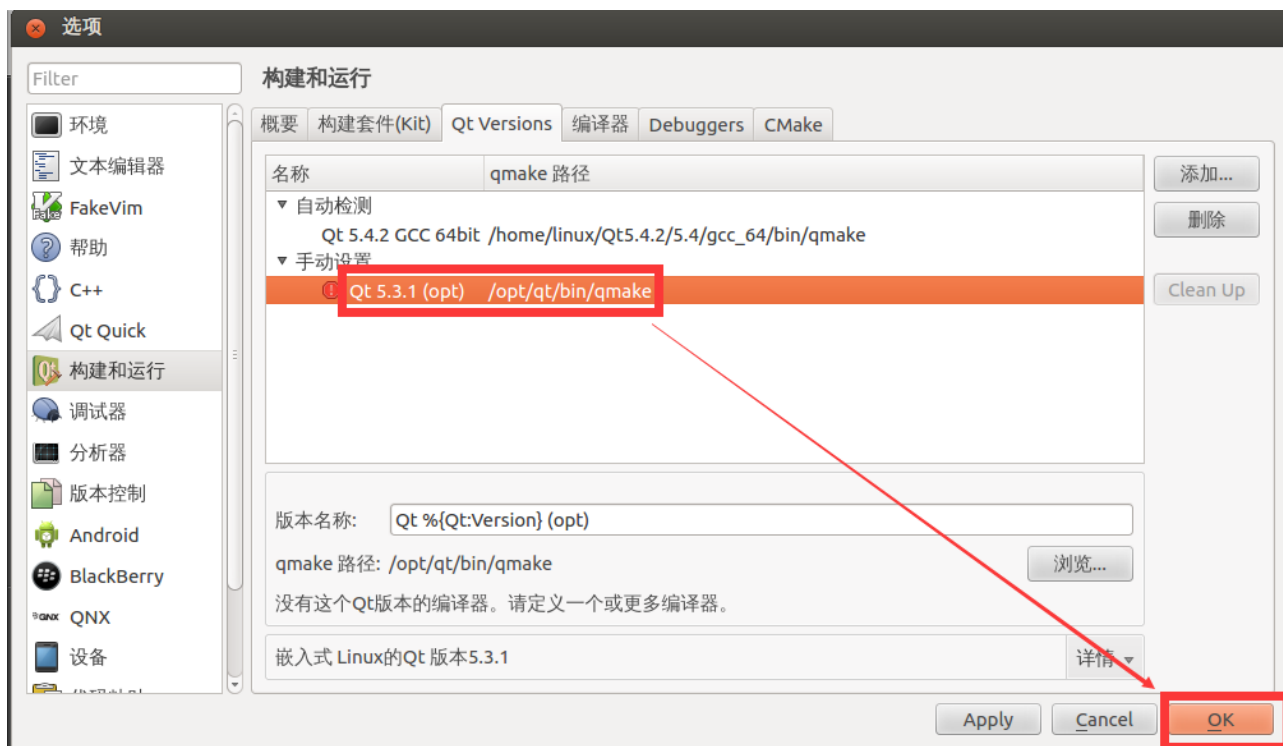




选择上节编译好的 Qt-everywherezh 中 qmake 的路径，本文路径为/opt/qt/bin/，选择后点击“打开”按钮

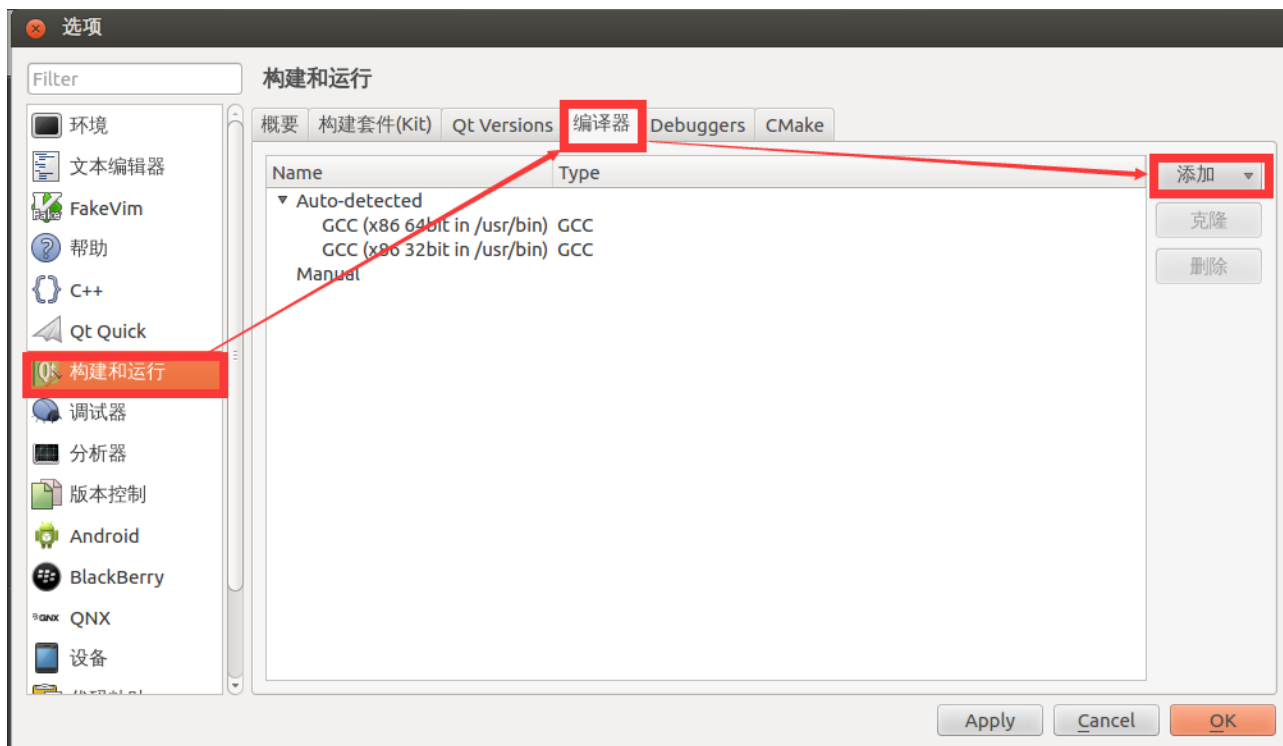


手动设置下会出现一条新的配置，然后点击“OK”按钮

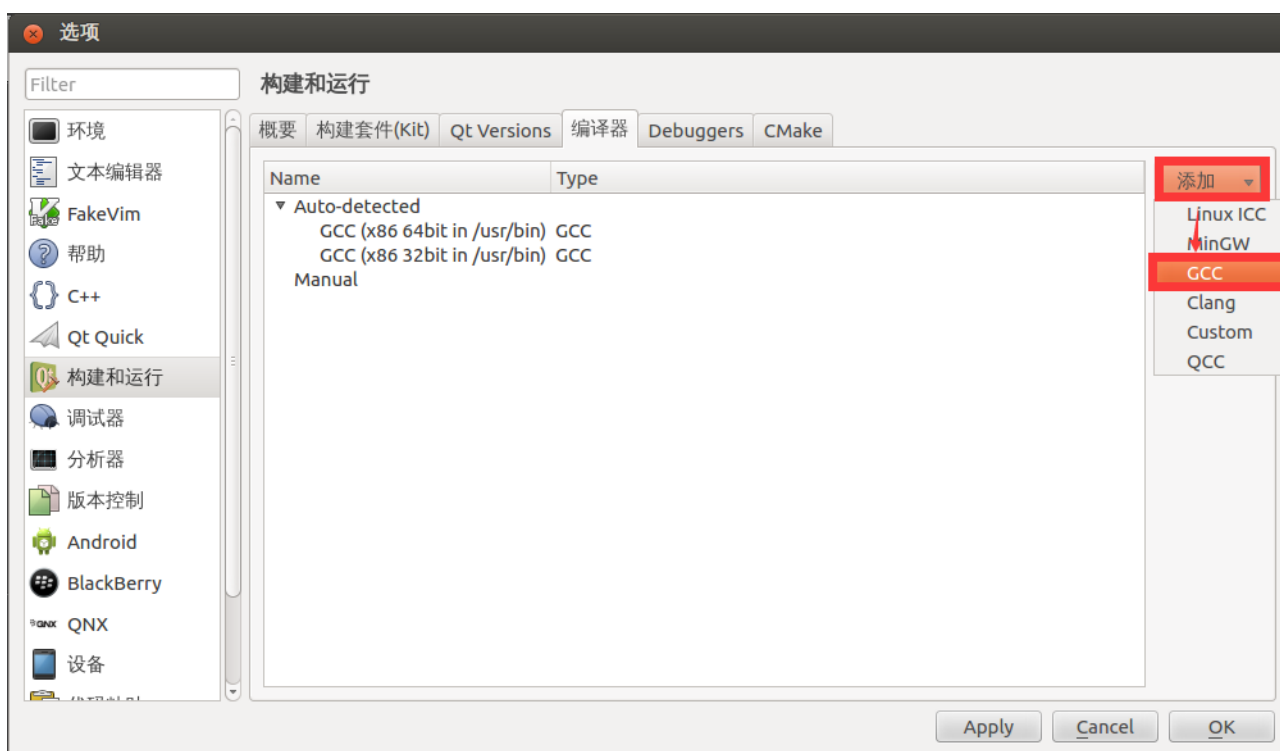


### 3) 配置编译器

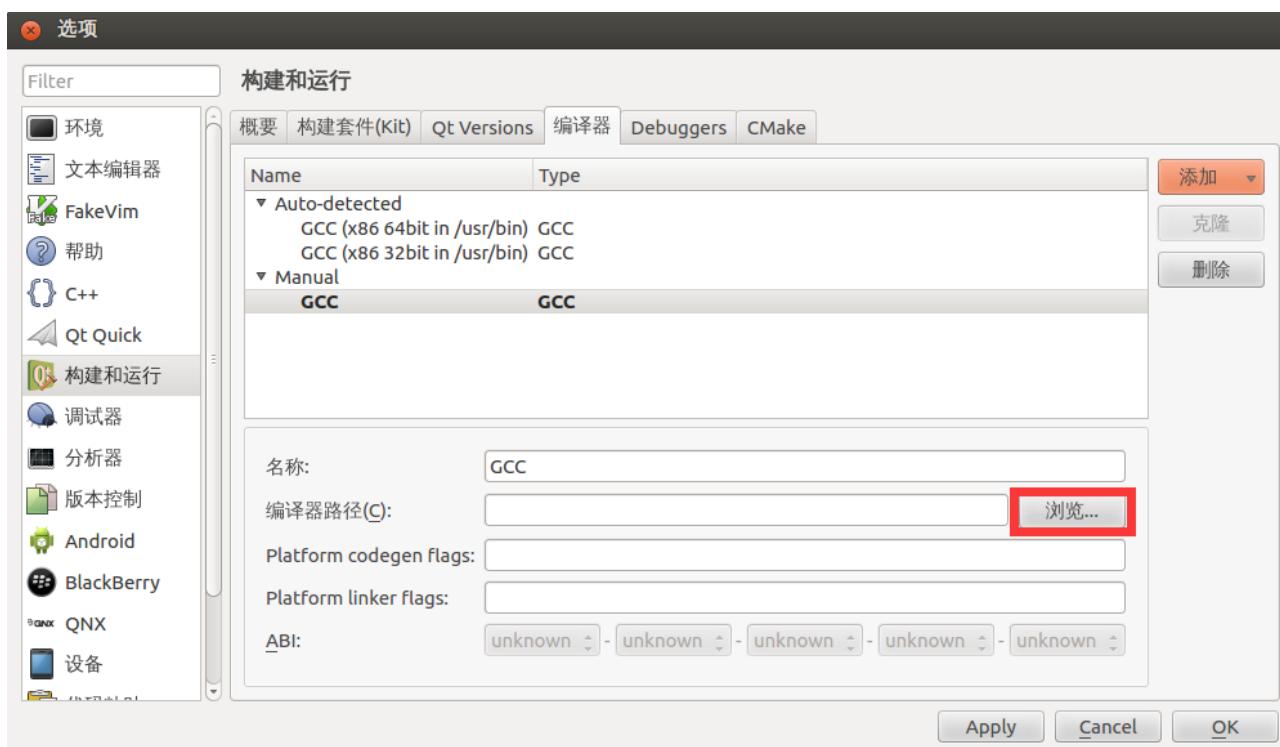
点击选项界面左侧“构建和运行”后选择“编译器”选项卡最后点击右侧“添加”按钮



点击“添加”后选择“GCC”

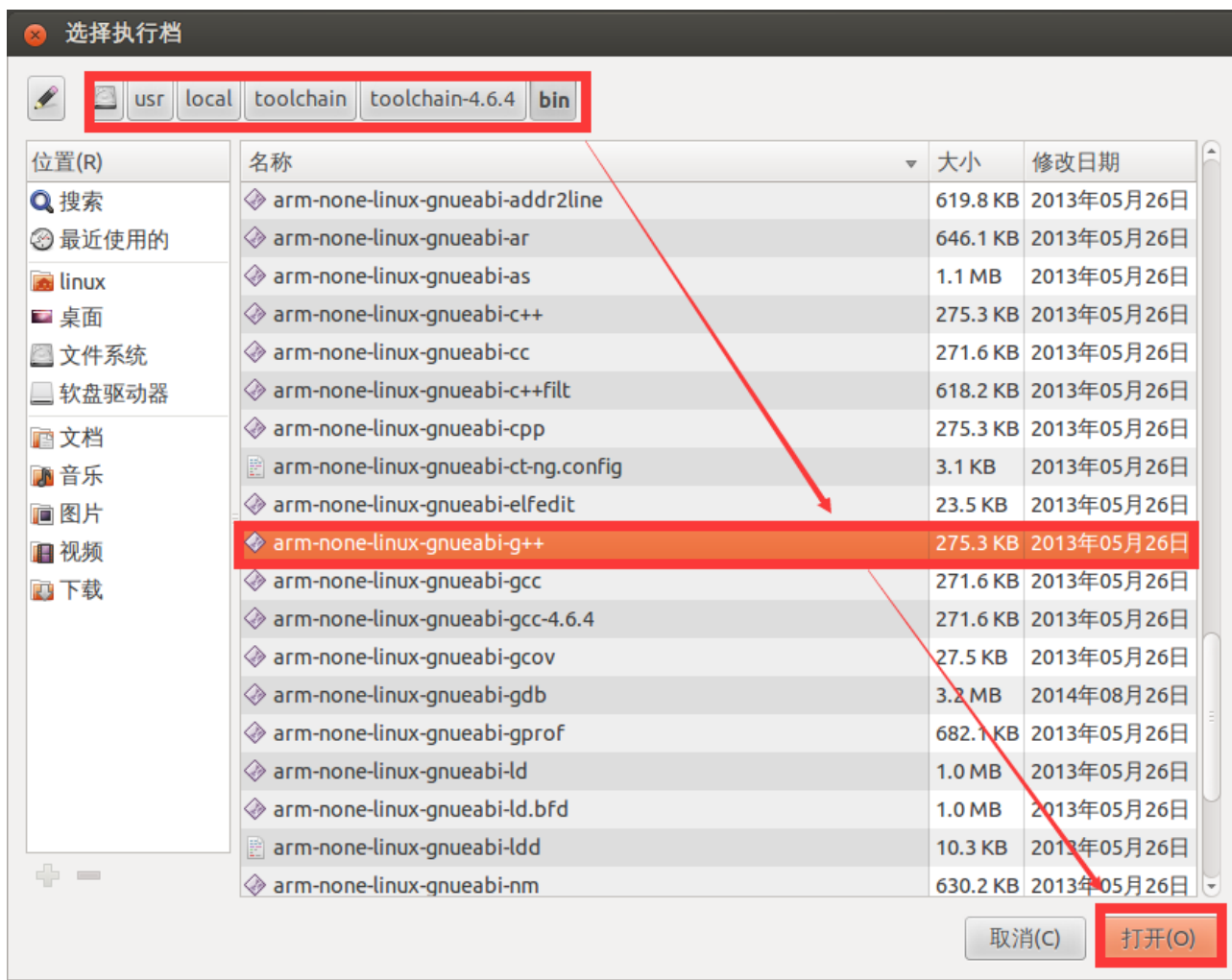


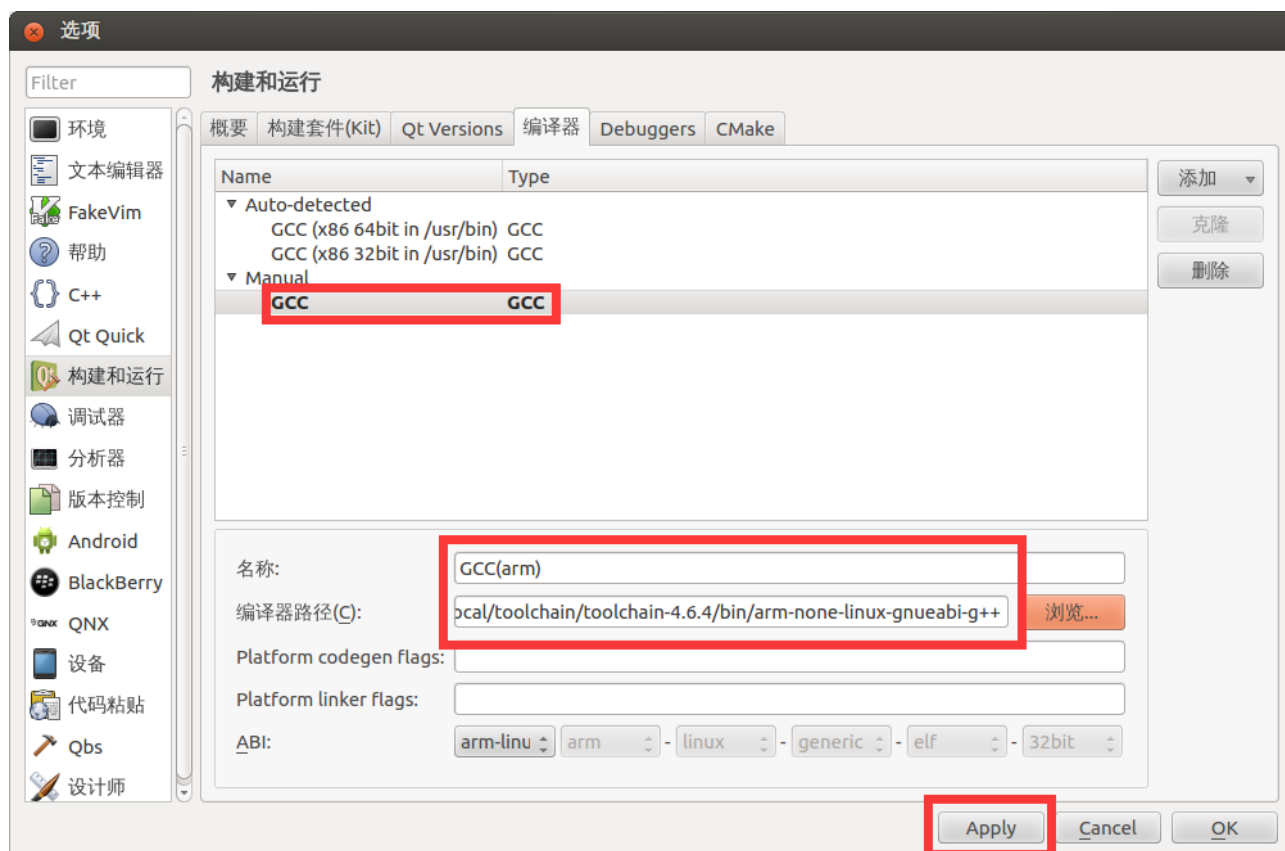
点击“浏览”





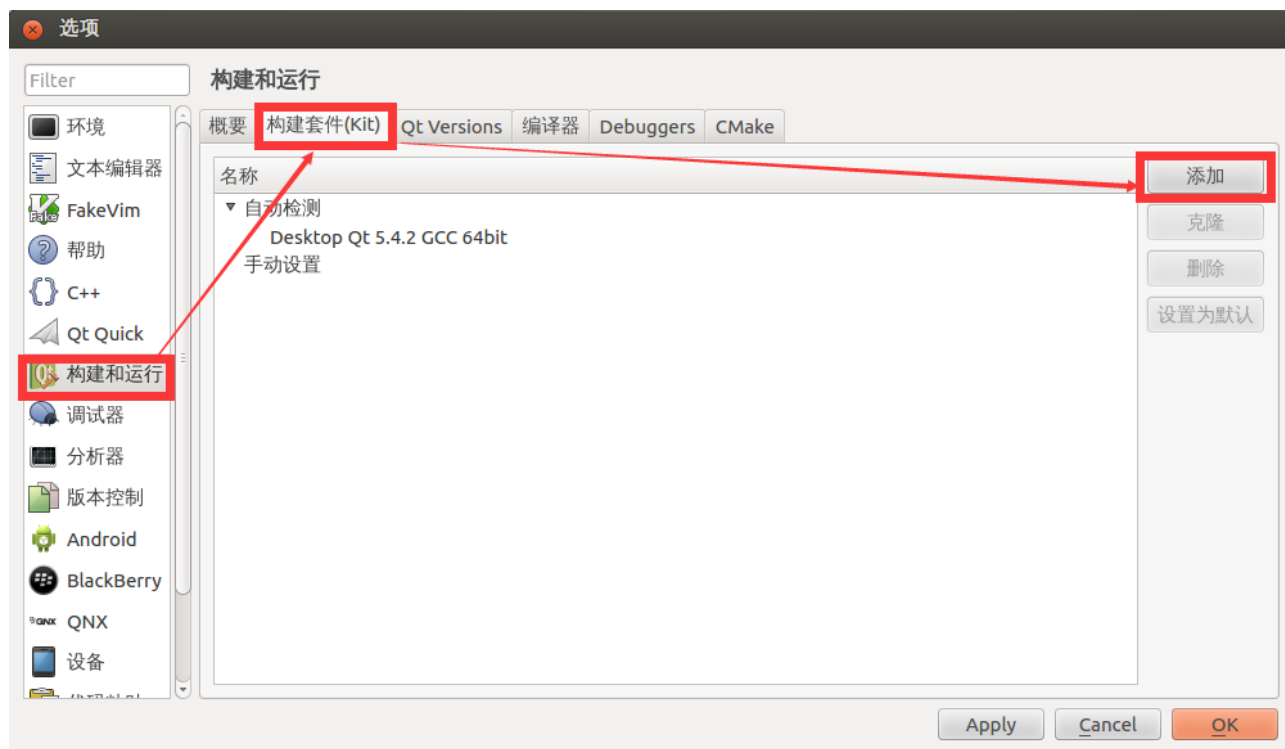
选择 GCC 路径，这里选择交叉工具链路径中 g++ 路径，华清远见提供开发环境 g++ 路径为 /usr/local/toolchain/toolchain-4.6.4/bin/arm-none-linux-gnueabi-g++, 选择后点击“OK”





#### 4) 配置开发套件

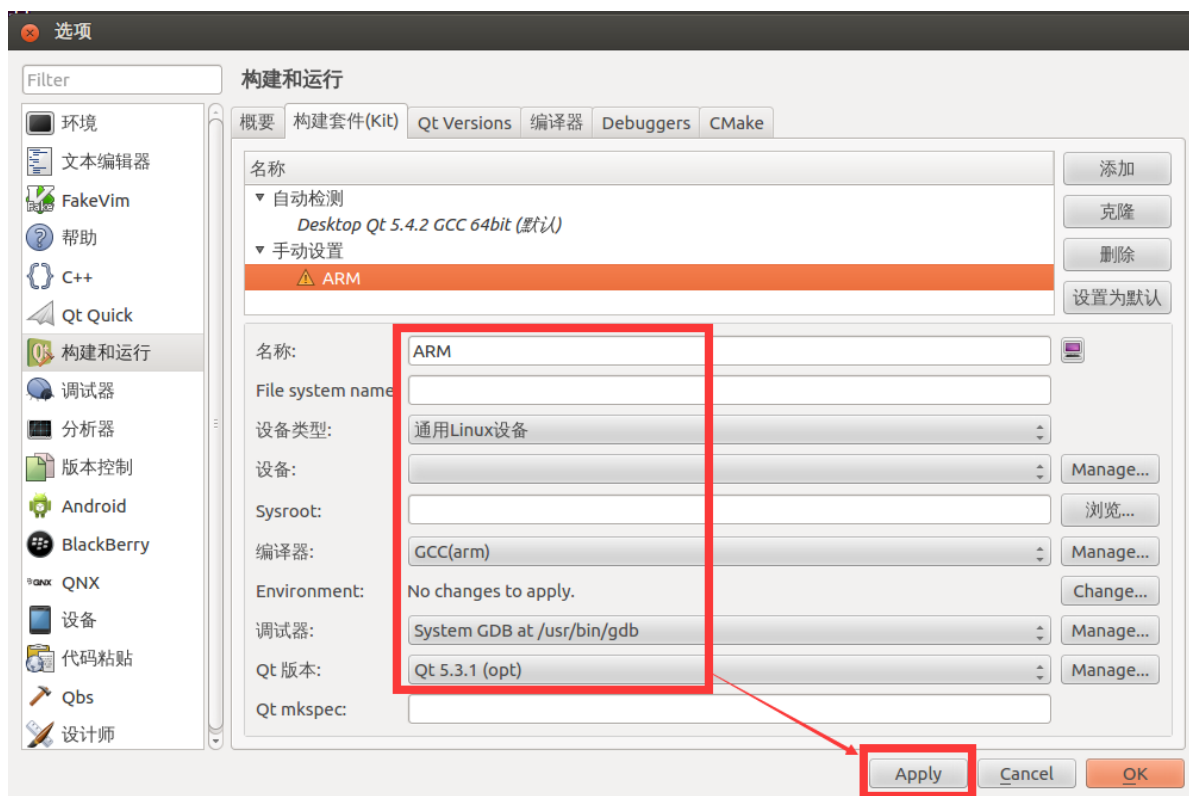
点击选项界面左侧“构建和运行”后选择“构建套件(Kit)”选项卡最后点击右侧“添加”按钮





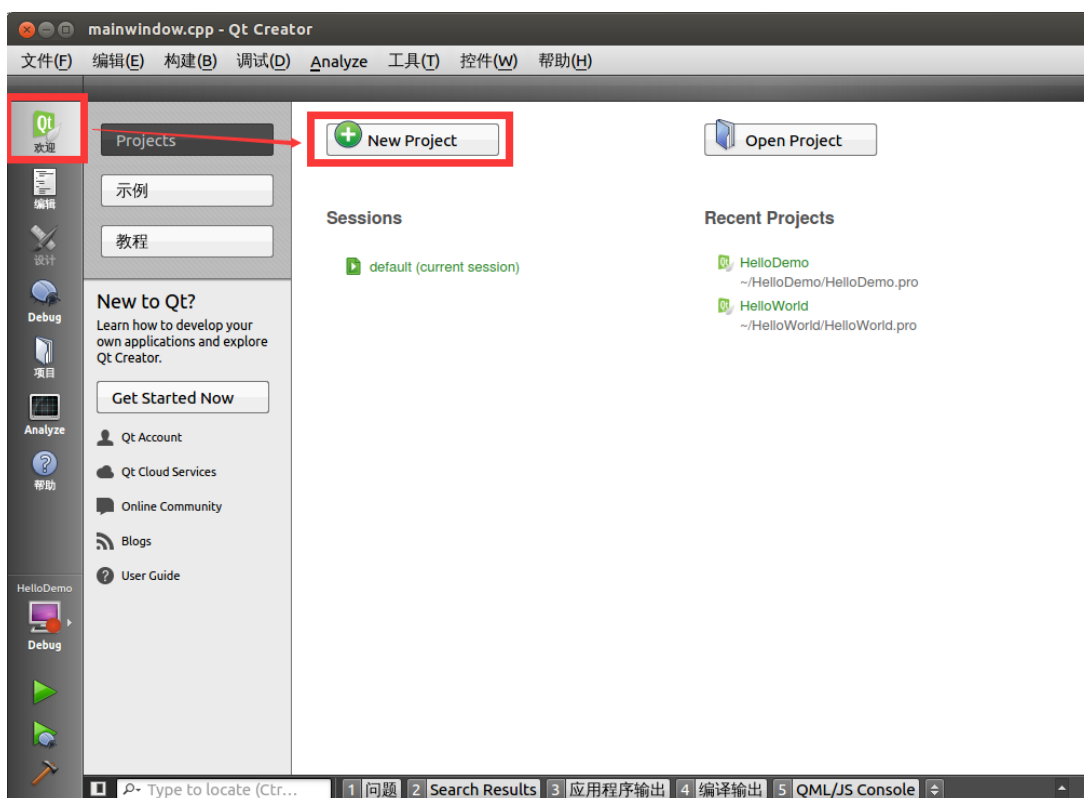


按图选择相应选择后，点击“Apply”然后点击“OK”关闭选项窗口

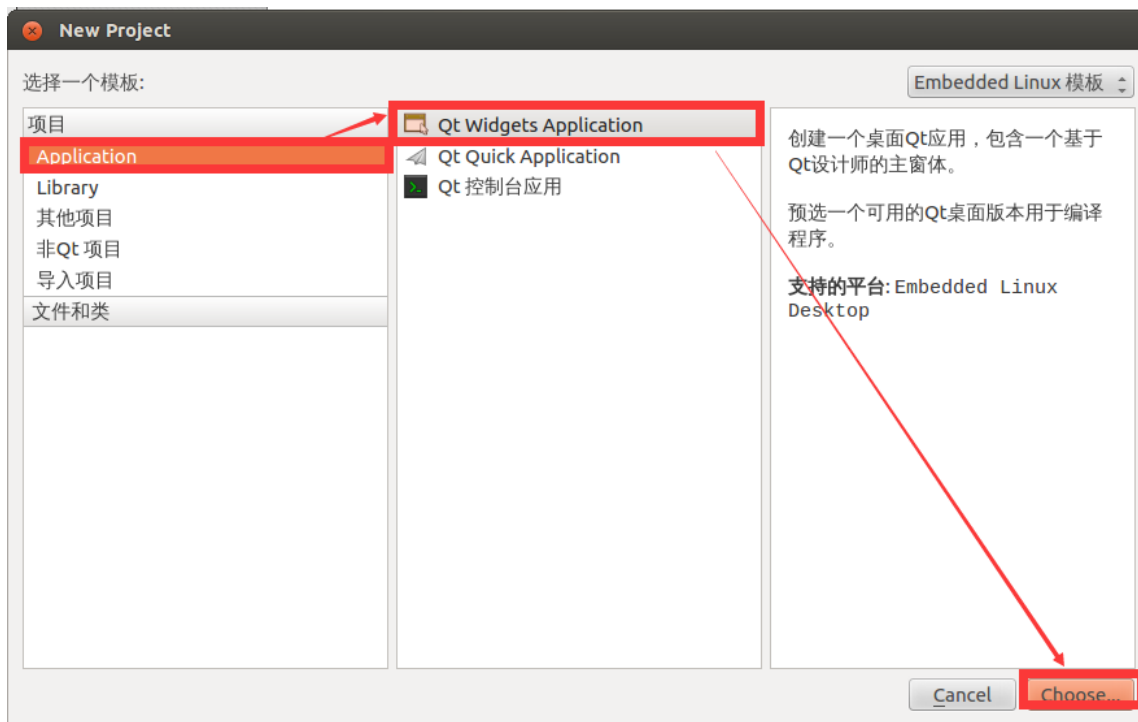


## 1.5 创建第一个工程 HelloWorld

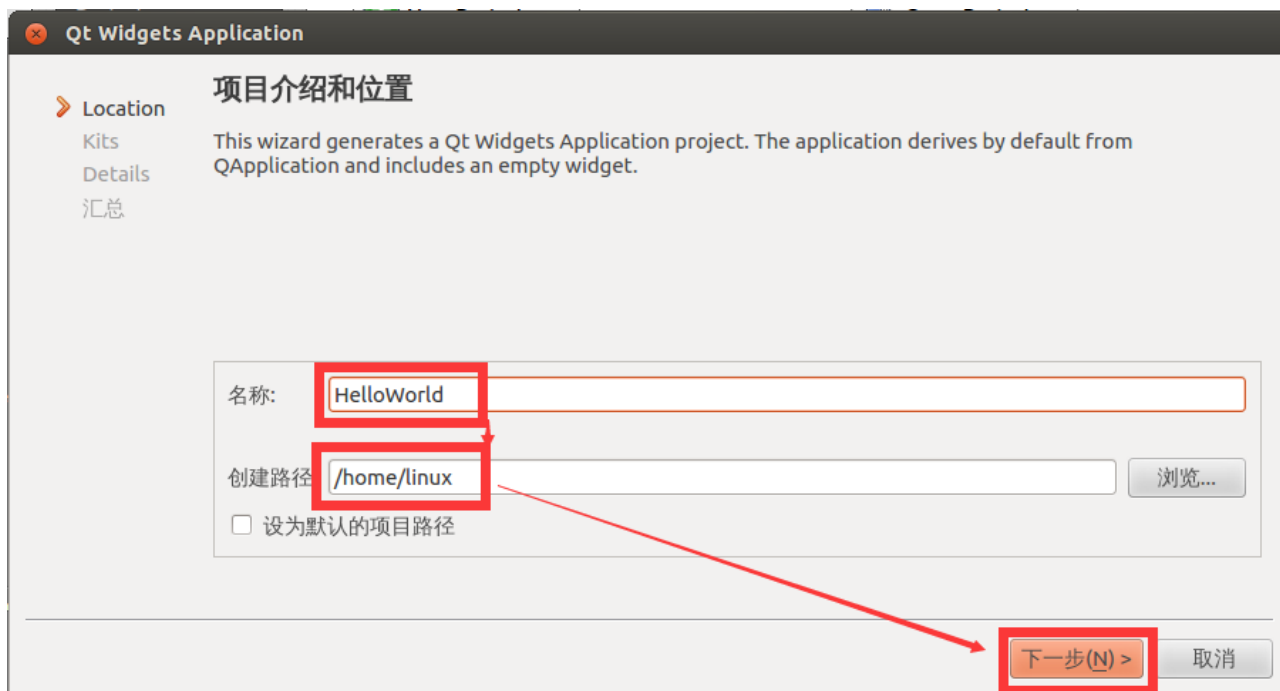
打开 qtcreator, 单机左侧菜单“欢迎”，后点击中间“New Project”



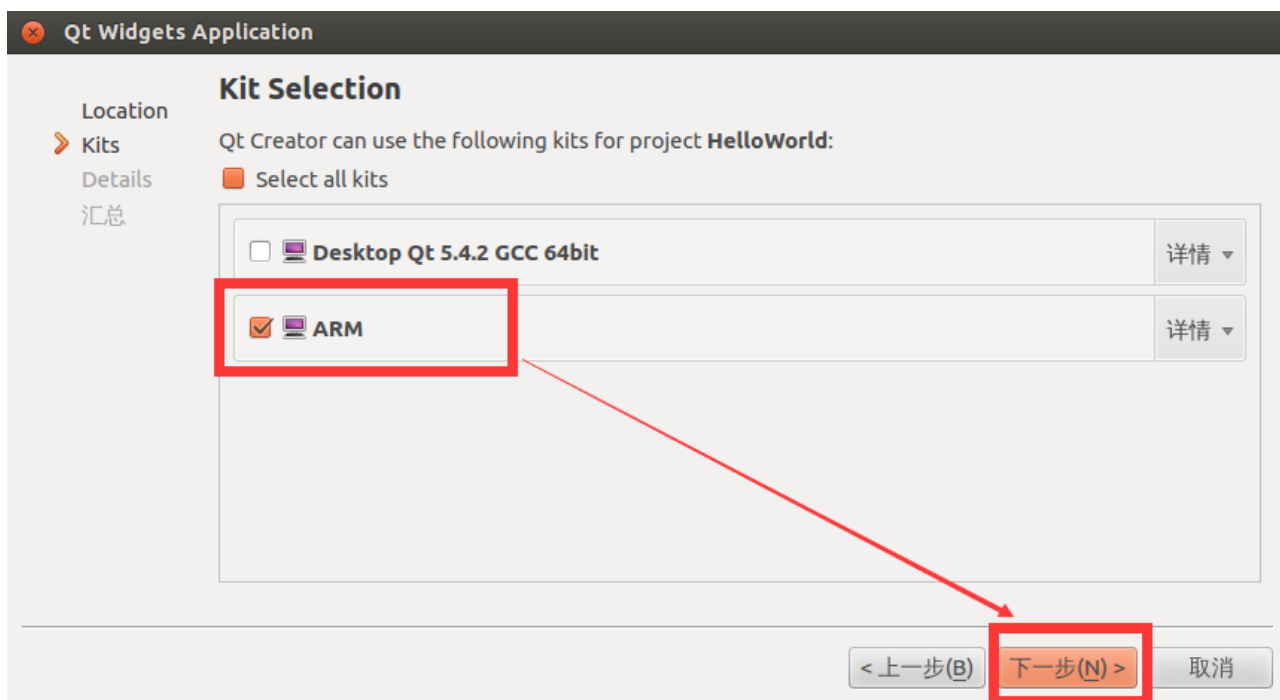
选择“Application”—“Qt Widgets Application”后点击“Choose”按钮



填写工程名称和工程所在路径后点击下一步



选择开发套件，本例的程序需要运行在 ARM 平台，所以选择 ARM 开发套件



指定类名，根据自己的需求完成，本例使用默认类名



Qt Widgets Application

**类信息**

Location  
Kits  
Details  
汇总

指定您要创建的源码文件的基本类信息。

类名(C):

基类(B):

头文件(H):

源文件(S):

创建界面(C): ☒

界面文件(F):

< 上一步(B) **下一步(N) >** 取消

添加到版本控制系统中，本例不添加

Qt Widgets Application

**项目管理**

Location  
Kits  
Details  
汇总

作为子项目添加到项目中:

添加到版本控制系统(V):

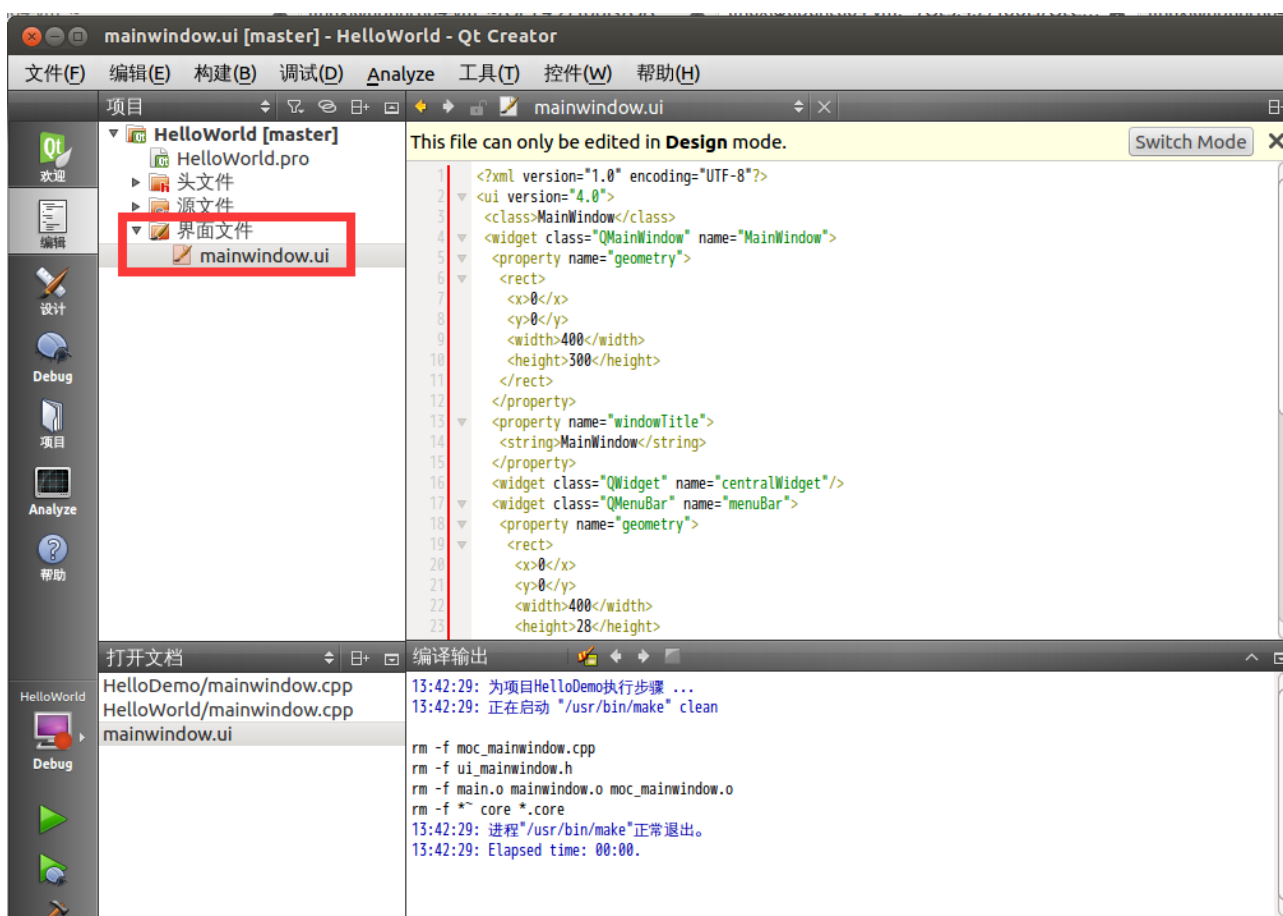
要添加的文件

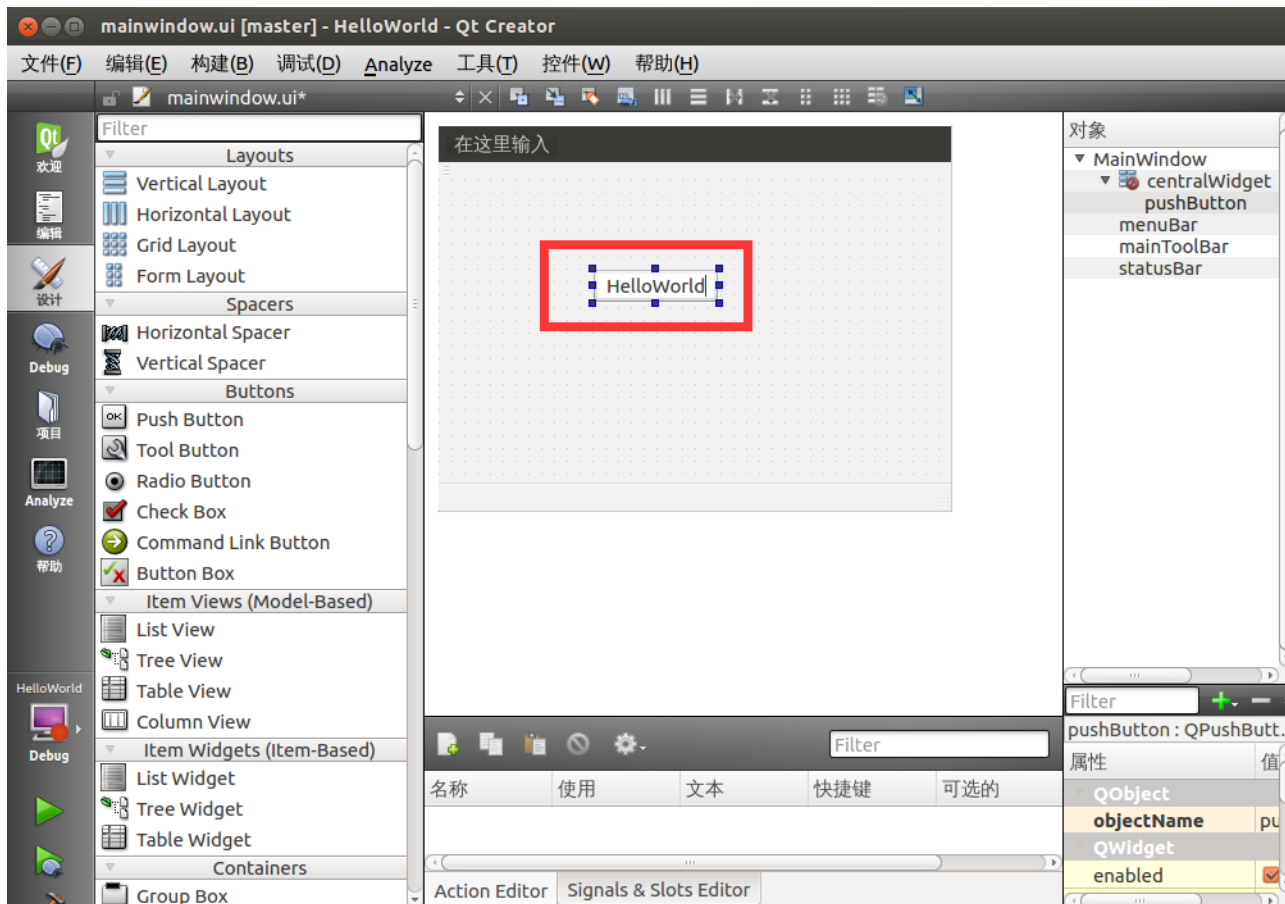
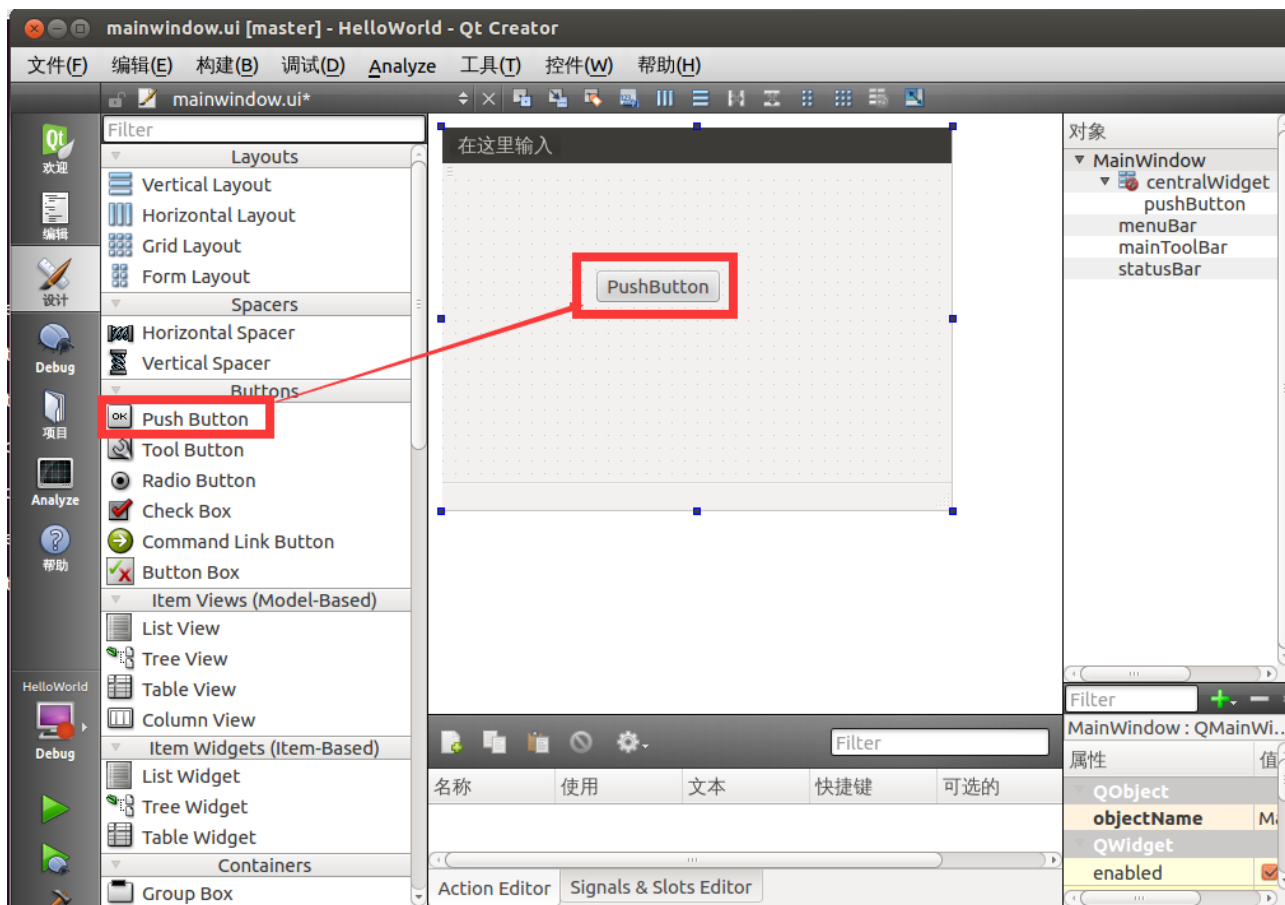
/home/linux/HelloWorld:

HelloWorld.pro  
main.cpp  
mainwindow.cpp  
mainwindow.h  
mainwindow.ui

< 上一步(B) **完成(F)** 取消

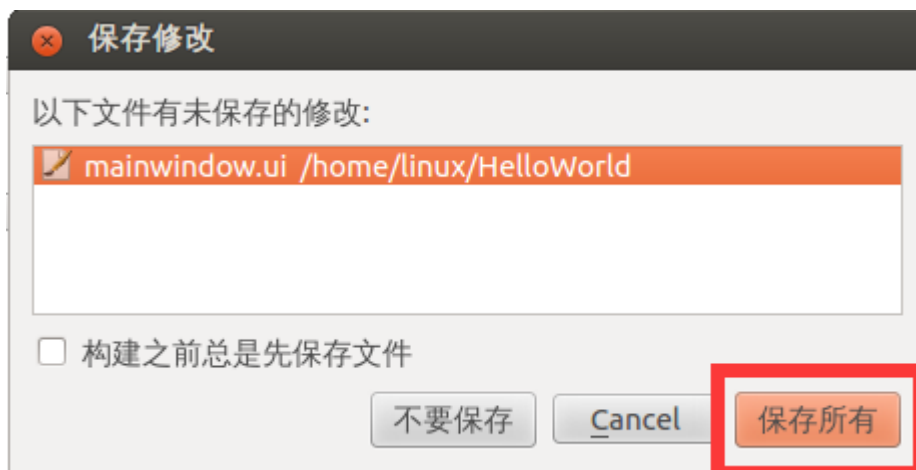
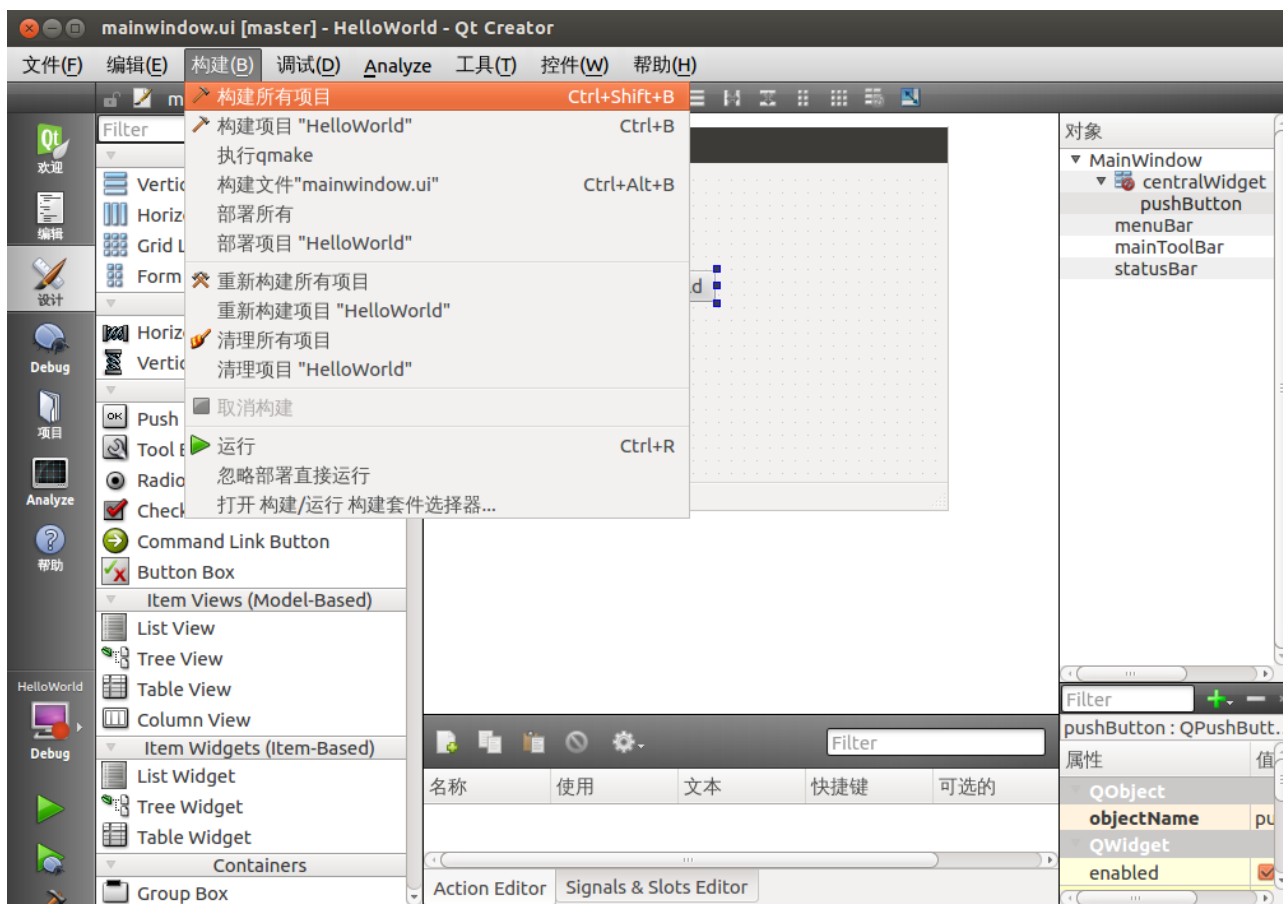
修改界面

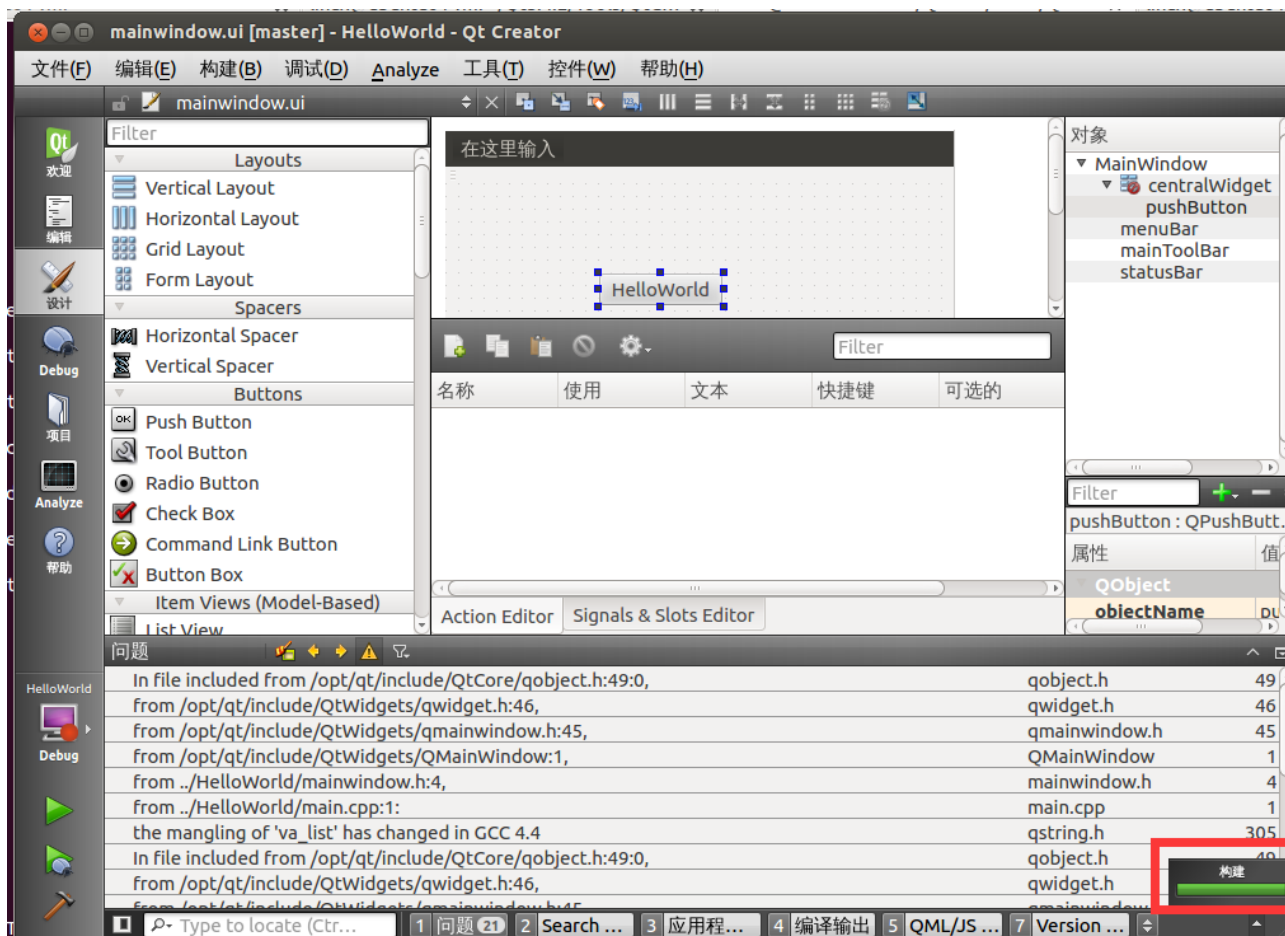






## 构建工程





编译完成后

在工程同级目录下有如下目录“build-HelloWorld-ARM-Debug”

```
linux@ubuntu64-vm:~$ ls build-HelloWorld-ARM-Debug/
HelloWorld main.o mainwindow.o Makefile moc_mainwindow.cpp moc_mainwindow.o ui_mainwindow.h
linux@ubuntu64-vm:~$
```

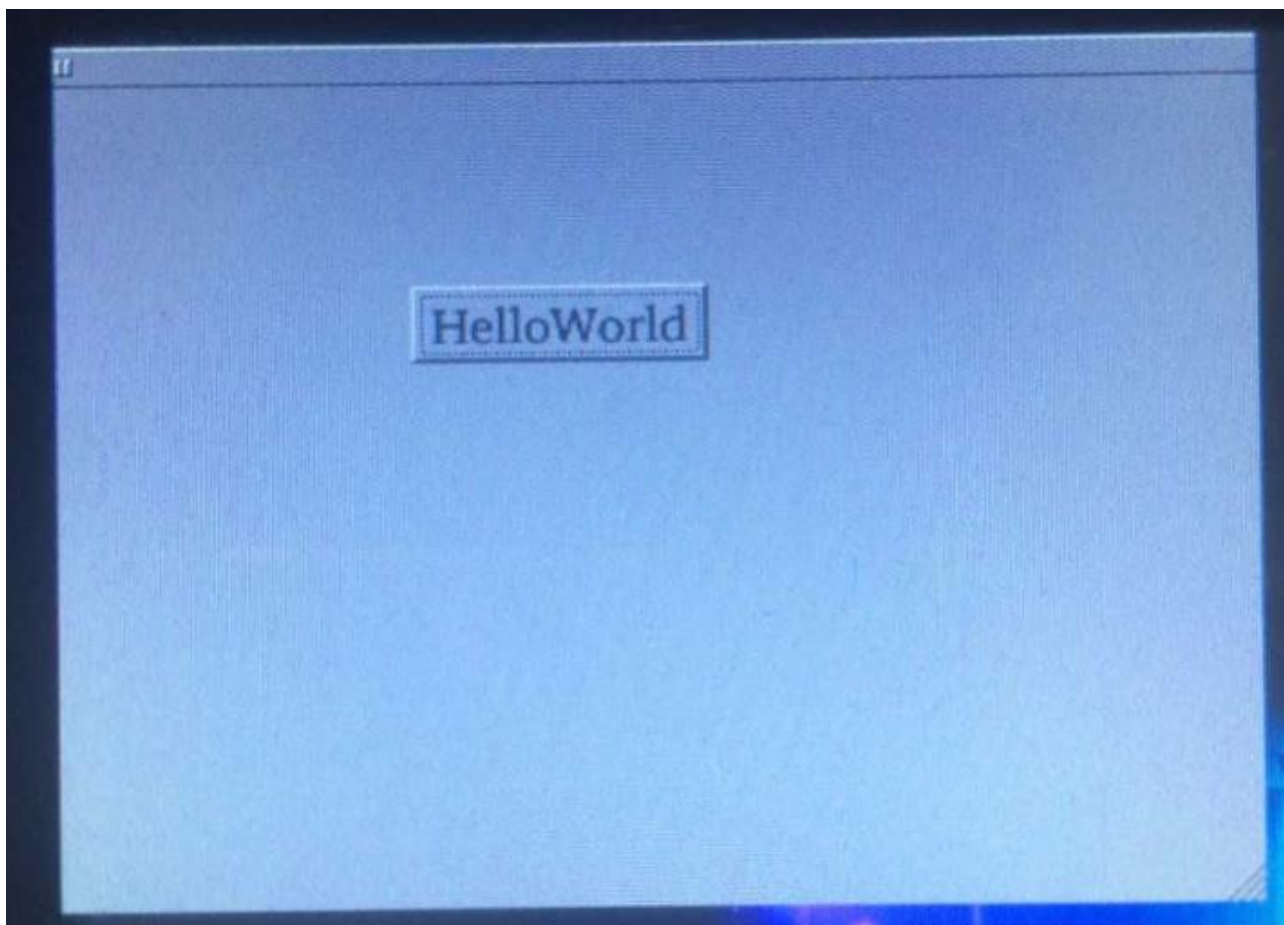
将目录中 HelloWorld 中拷贝到根文件系统中

在开发板上执行如下操作：

```
./HelloWorld -qws
```

这时屏幕显示如下





## 版本历史:

版本	修改人	时间	备注
<b>V1.0</b>	曹忠明	2015-06-26	初始版本