

# Purple Pi R1 Qt Creator开发流程

---

Qt Creator开发流程

配置Qt开发环境

Qt Creator添加Qt4.8.7开发套件

Qt Creator添加Qt4.8.7交叉编译

编写Qt 程序在开发板中运行



Purple Pi R1

Qt Creator开发流程

深圳触觉智能科技有限公司

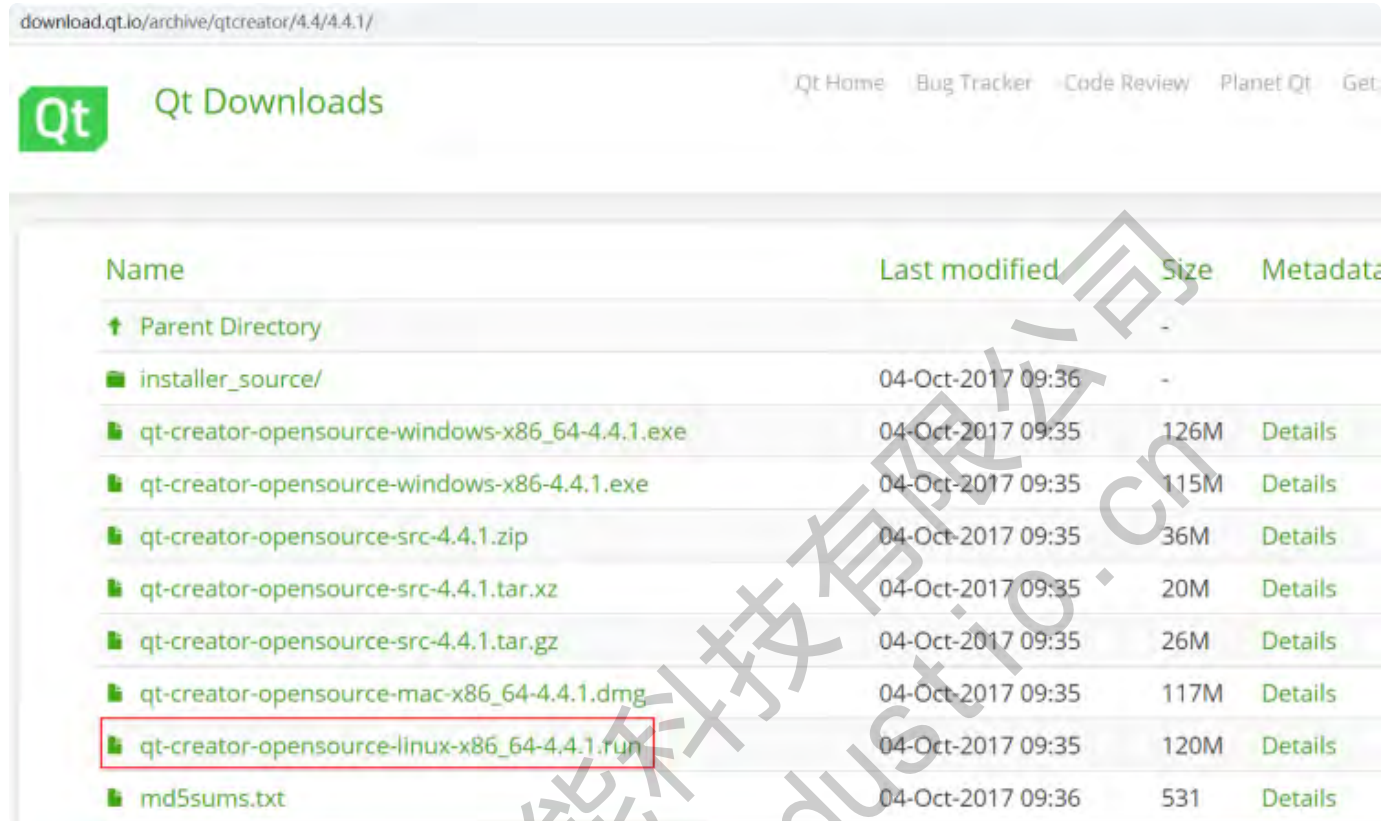
[www.industio.cn](http://www.industio.cn)

---

Qt Creator开发流程

# 配置Qt开发环境

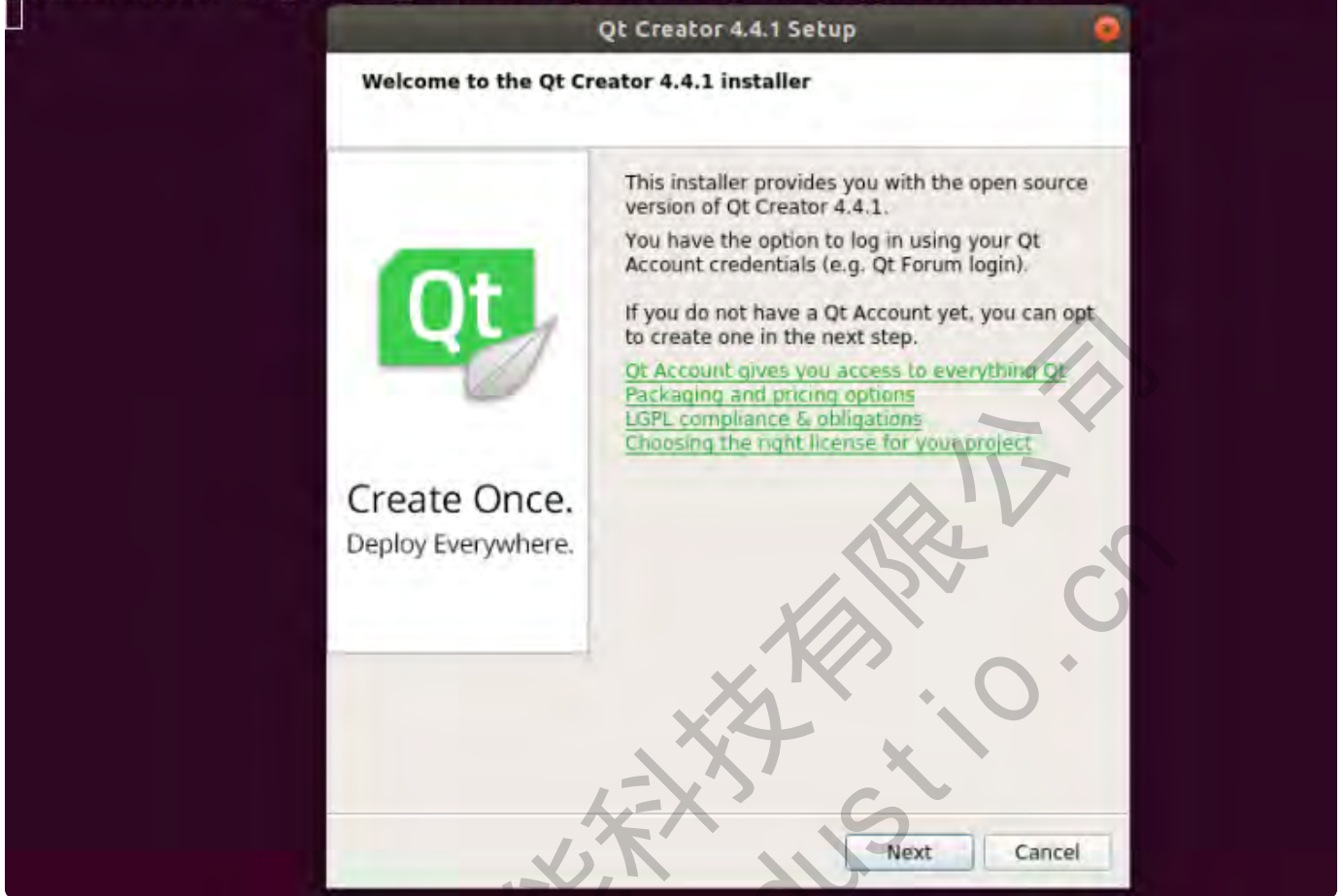
1、下载Qt Creator4.4.1下载地址: <https://download.qt.io/archive/qtcreator/4.4/4.4.1/>



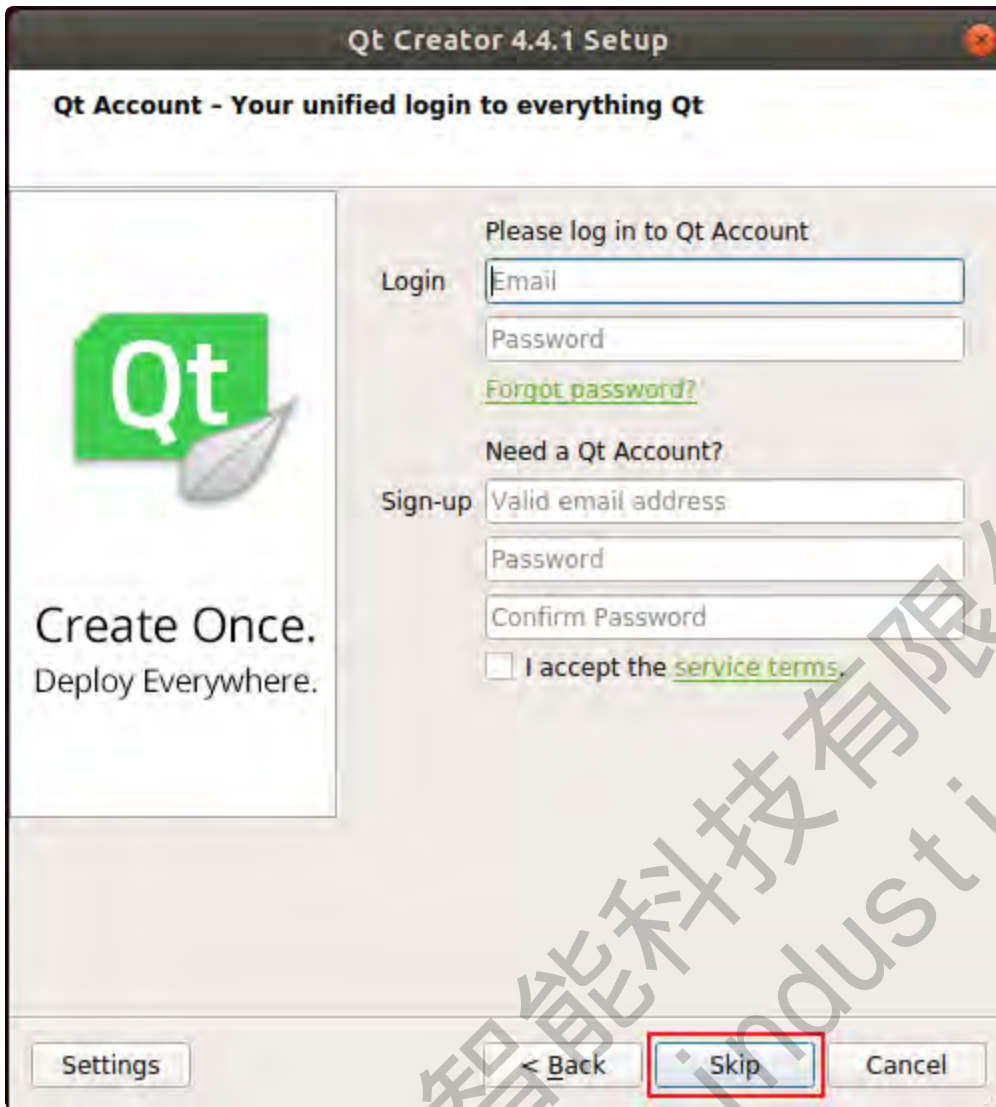
Name	Last modified	Size	Metadata
↑ Parent Directory	-	-	
installer_source/	04-Oct-2017 09:36	-	
qt-creator-opensource-windows-x86_64-4.4.1.exe	04-Oct-2017 09:35	126M	<a href="#">Details</a>
qt-creator-opensource-windows-x86-4.4.1.exe	04-Oct-2017 09:35	115M	<a href="#">Details</a>
qt-creator-opensource-src-4.4.1.zip	04-Oct-2017 09:35	36M	<a href="#">Details</a>
qt-creator-opensource-src-4.4.1.tar.xz	04-Oct-2017 09:35	20M	<a href="#">Details</a>
qt-creator-opensource-src-4.4.1.tar.gz	04-Oct-2017 09:35	26M	<a href="#">Details</a>
qt-creator-opensource-mac-x86_64-4.4.1.dmg	04-Oct-2017 09:35	117M	<a href="#">Details</a>
qt-creator-opensource-linux-x86_64-4.4.1.run	04-Oct-2017 09:35	120M	<a href="#">Details</a>
md5sums.txt	04-Oct-2017 09:36	531	<a href="#">Details</a>

选择下载qt-creator-opensource-linux-x86\_64-4.4.1.run, 并将下载的文件拷贝至Ubuntu虚拟机中。在Ubuntu终端执行sudo ./qt-creator-opensource-linux-x86\_64-4.4.1.run命令打开安装向导界面,点击“Next”按钮进入下一步。如下图所示

```
fu@fu-VirtualBox:/home/industio_work/Qt$ sudo ./qt-creator-opensource-linux-x86_64-4.4.1.run
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
```



此处如果需要登陆，点击“skip”跳过即可；



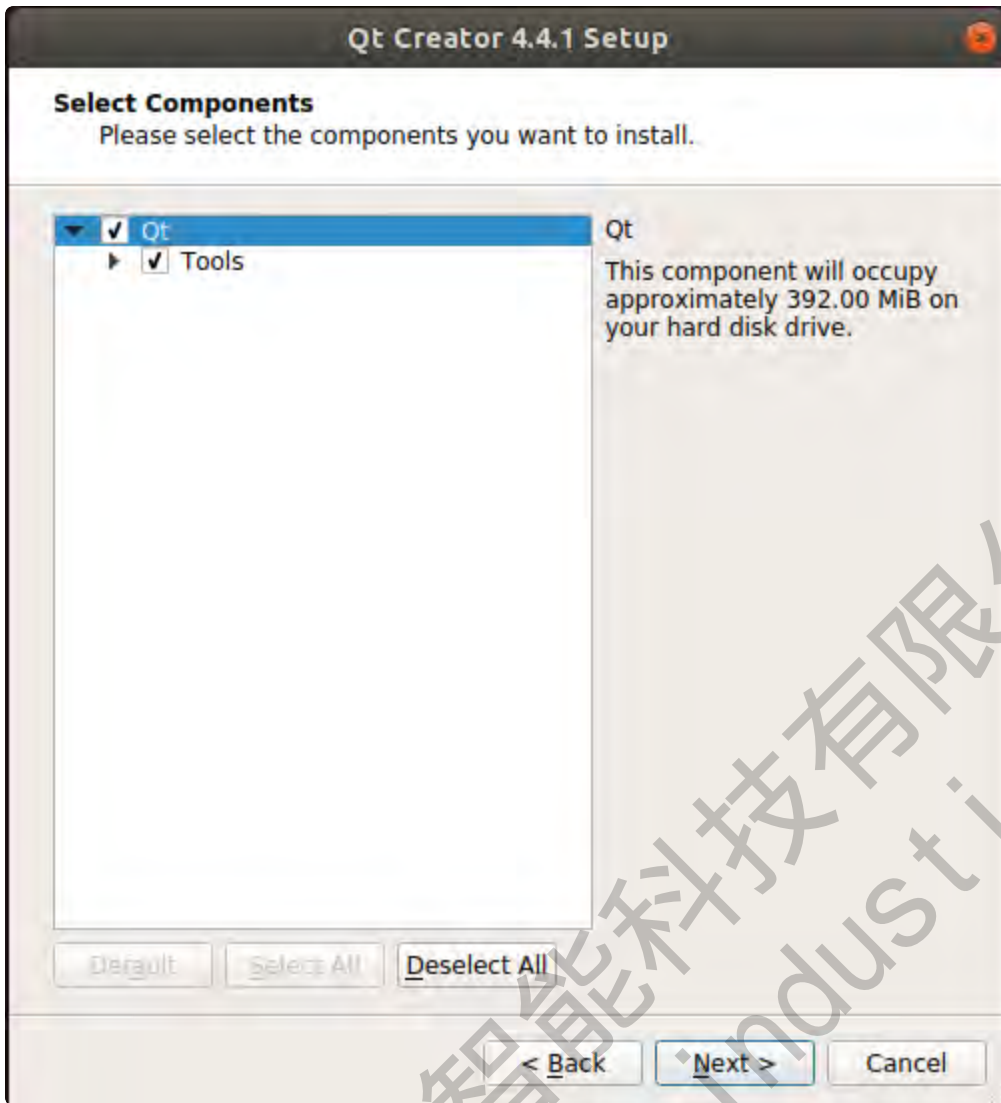
点击“Next”进入下一步；



默认安装目录为“/opt/qtcreator-4.4.1”，点击“Next”进入下一步；

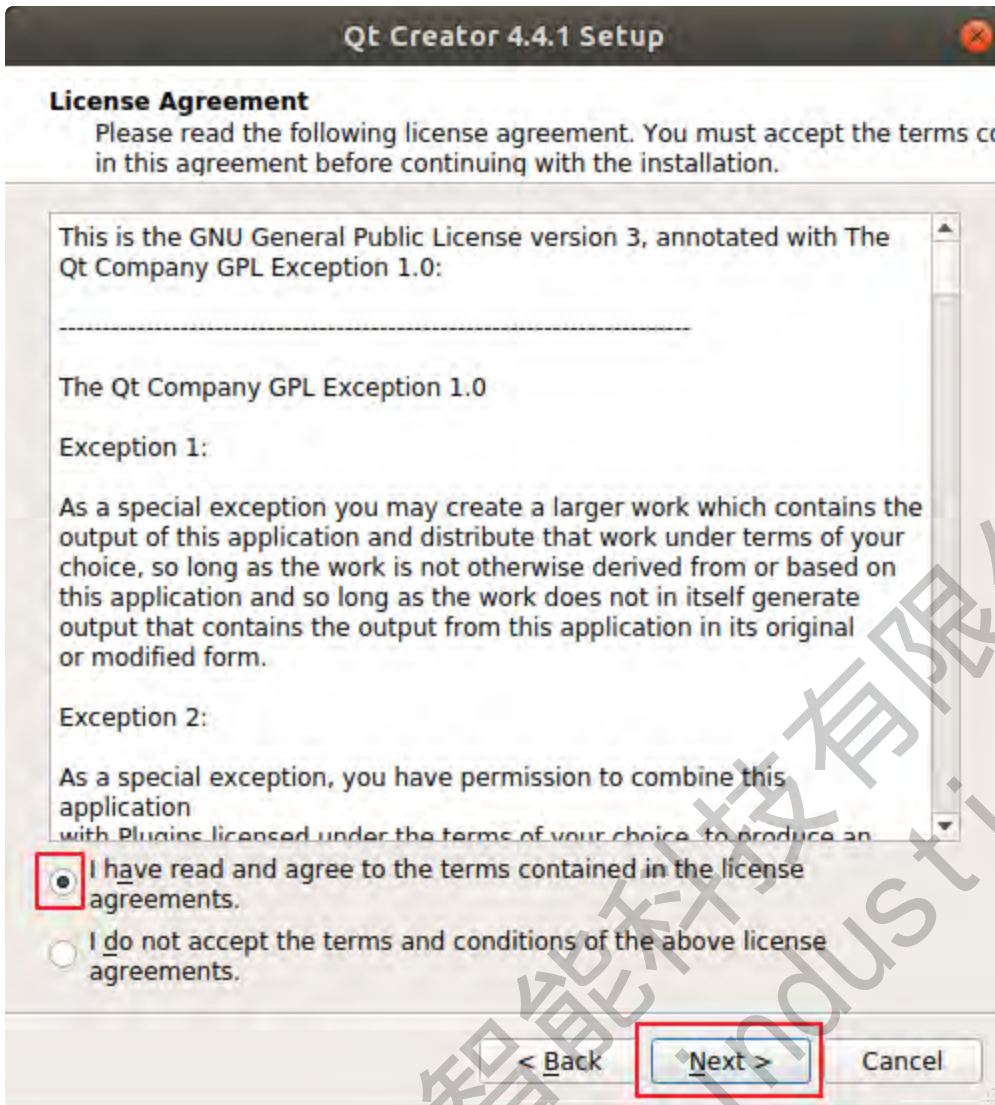


保留默认的勾选安装组件内容，点击“Next”进入下一步；



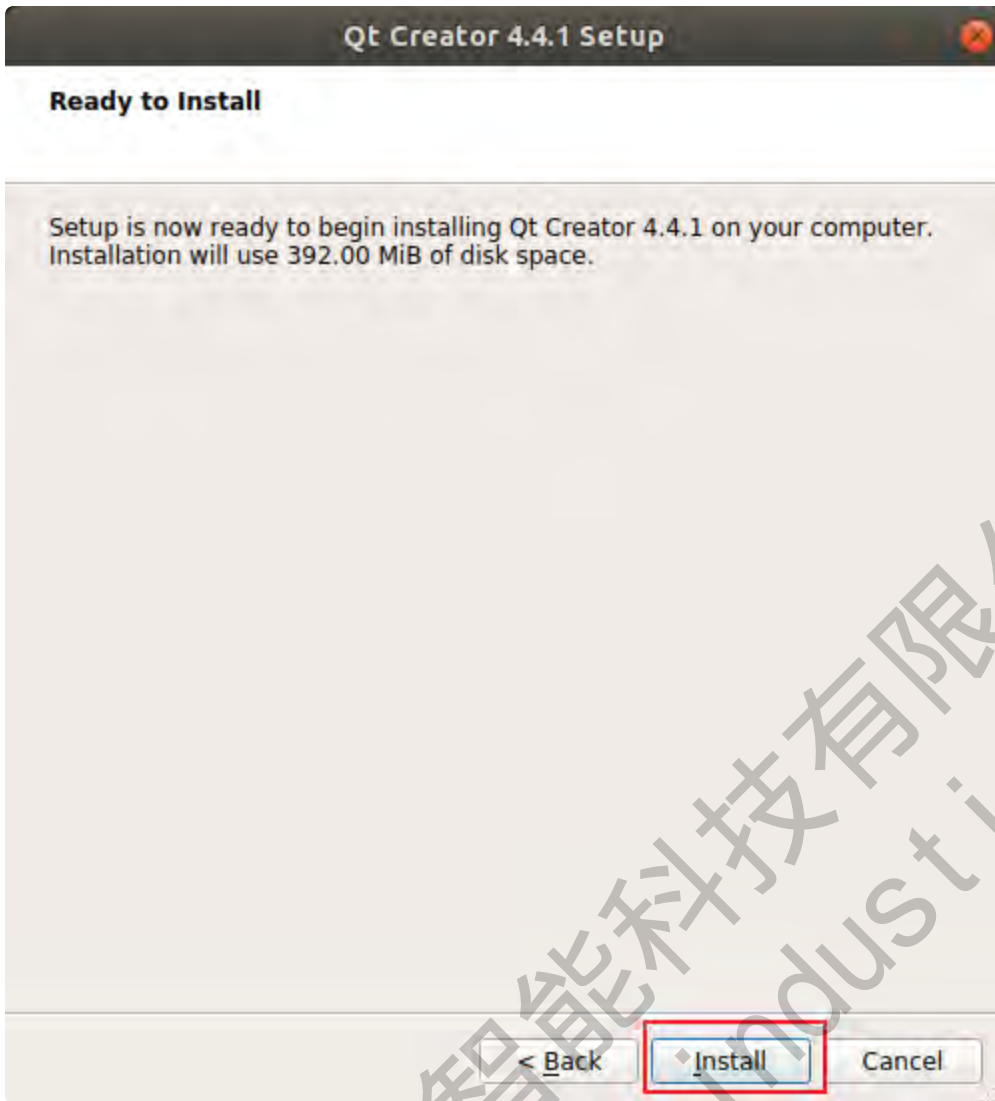
阅读协议条款内容后，选择同意，点击”Next“进入下一步；





点击”Install“开始安装；





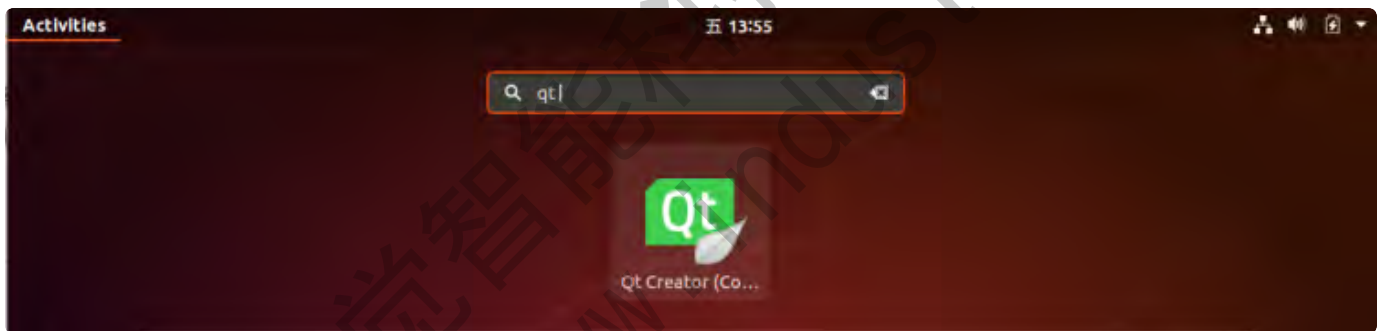
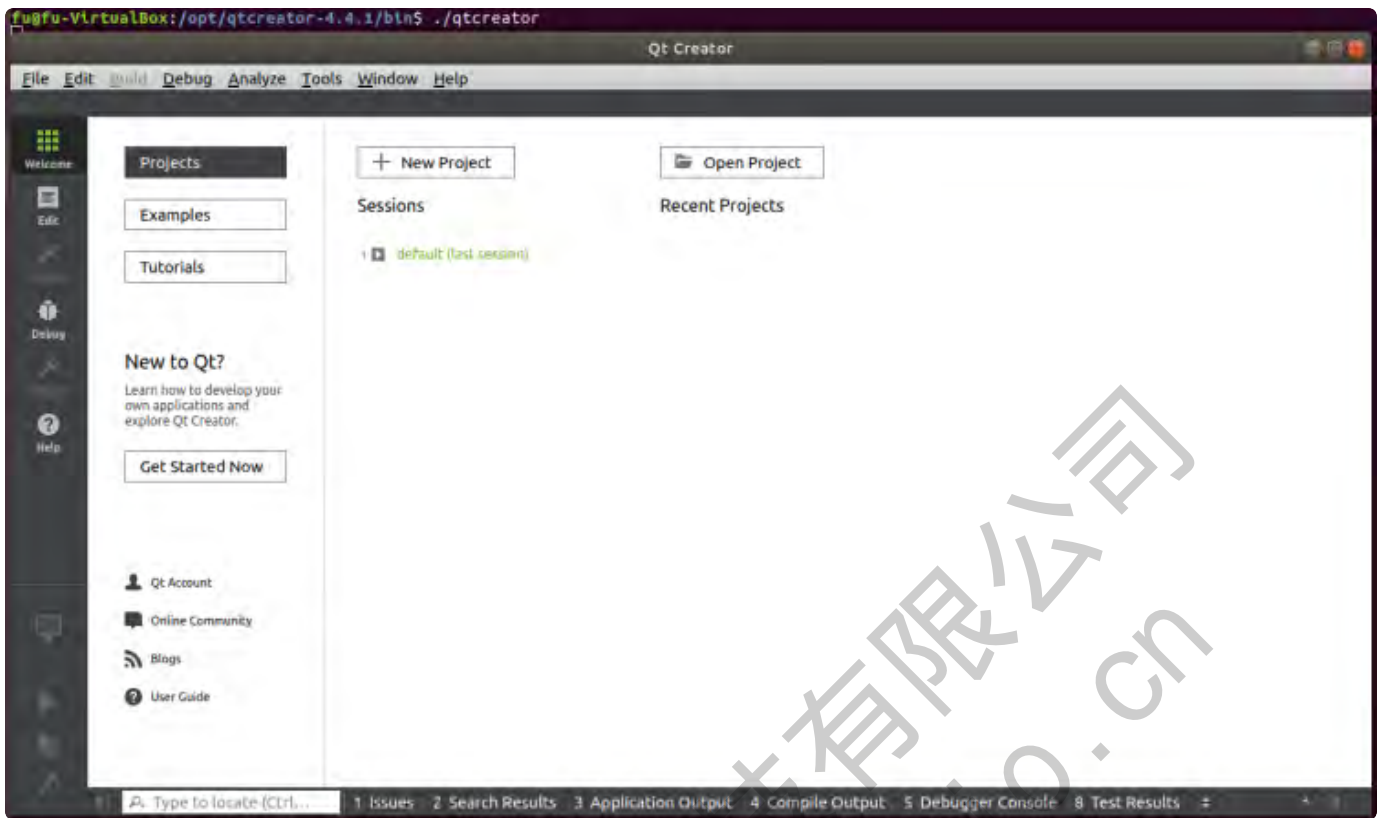
安装进度界面如下图；



安装完成后，即可在安装目录“/opt/qtcreator-4.4.1”的bin目录下找到可执行程序`qtcreator`

```
fu@fu-VirtualBox:/opt/qtcreator-4.4.1/bin$ ls
qbs          qbs-config-ui      qbs-qmltypes      qbs-setup-qt      qt.conf          qtcreator.sh
qbs-config   qbs-create-project qbs-setup-android  qbs-setup-toolchains qtcreator
```

可以在命令行的“/opt/qtcreator-4.4.1/bin”目录下，执行`./qtcreator`开启Qt Creator4.4.1集成开发环境；或者点击Ubuntu18.04界面的左上角“Activities”，在弹出的搜索框中输入“Qt”，在查找结果中找到Qt Creator的图标，双击图标开启Qt Creator程序。



## 安装Qt4.8.7

在Linux平台安装Qt4.8.7需要从官网下载源代码，通过编译源码获得可用的Qt4.8.7 sdk。

官方源码下载地址：<https://download.qt.io/archive/qt/4.8/4.8.7/>



Name	Last modified	Size	Metadata
↑ Parent Directory	-	-	-
qt-opensource-windows-x86-vs2010-4.8.7.exe	04-Jun-2018 17:26	236M	<a href="#">Details</a>
qt-opensource-windows-x86-vs2008-4.8.7.exe	04-Jun-2018 17:27	235M	<a href="#">Details</a>
qt-opensource-windows-x86-mingw482-4.8.7.exe	04-Jun-2018 17:27	329M	<a href="#">Details</a>
qt-opensource-mac-4.8.7.dmg	04-Jun-2018 17:27	185M	<a href="#">Details</a>
qt-opensource-mac-4.8.7-debug-libs.dmg	04-Jun-2018 17:27	476M	<a href="#">Details</a>
qt-everywhere-opensource-src-4.8.7.zip	04-Jun-2018 17:26	268M	<a href="#">Details</a>
qt-everywhere-opensource-src-4.8.7.tar.gz	04-Jun-2018 17:27	230M	<a href="#">Details</a>
md5sums-4.8.7	04-Jun-2018 17:26	517	<a href="#">Details</a>
changes-4.8.7	04-Jun-2018 17:27	12K	<a href="#">Details</a>

选择下载“qt-everywhere-opensource-src-4.8.7.tar.gz”，将下载的文件拷贝至Ubuntu虚拟机中。

```
fu@fu-VirtualBox:/home/industio_work/Qt$ ls
qt-creator-opensource-linux-x86_64-4.4.1.run qt-everywhere-opensource-src-4.8.7.tar.gz
```

在终端执行下方命令，解压源码，并切换至解压后的源码目录。

```
1  industio@industio$:tar zxvf qt-everywhere-opensource-src-4.8.7.tar.gz
2  industio@industio$:cd qt-everywhere-opensource-src-4.8.7
```

```
fu@fu-VirtualBox:/home/industio_work/Qt/qt-everywhere-opensource-src-4.8.7$ ls
bin                configure.exe      include            LICENSE.GPL3       projects.pro       tools
changes-4.8.7      demos             INSTALL            LICENSE.LGPL        qmake              translations
config.profiles    doc               LGPL_EXCEPTION.txt LICENSE.LGPLv21     README             util
config.tests        examples           lib                LICENSE.LGPLv3      src
configure           imports           LICENSE.FDL        mkspecs             templates
```

安装编译Qt需要的库



```

1  industio@industio$:sudo apt-get install gcc g++ g++-multilib make automake
2  industio@industio$:sudo apt-get install zlib1g-dev lib32ncurses5 lib32z1 li
    bpng-dev autoconf libtool
3  industio@industio$:sudo apt-get install libxext-dev libx11-dev libxext-dev
    libxtst-dev
4  industio@industio$:sudo apt install libgstreamer1.0-dev libgstreamer1-perl
    libgstreamer-plugins-base0.10-dev libgstreamerd-3-dev
5  industio@industio$:sudo apt install freeglut3-dev mesa-utils

```

修改源码gcc/g++编译配置，指定标准为“gnu++98”，修改源码“修改源码“mkspecs/common/gcc-base.conf”文件中的“QMAKE\_CXXFLAGS”参数，在参数后面添加`-std=gnu++98`

```

root@ubuntu: /home/wxq/Desktop/qt-everywhere-opensource-src-4.8.7
File Edit View Search Terminal Help
QMAKE_CFLAGS_DEPS      += -M
QMAKE_CFLAGS_WARN_ON   += -Wall -W
QMAKE_CFLAGS_WARN_OFF  += -w
QMAKE_CFLAGS_RELEASE   += -O2
QMAKE_CFLAGS_DEBUG     += -g
QMAKE_CFLAGS_SHLIB     += -fPIC
QMAKE_CFLAGS_STATIC_LIB += -fPIC
QMAKE_CFLAGS_YACC       += -Wno-unused -Wno-parentheses
QMAKE_CFLAGS_HIDESYMS  += -fvisibility=hidden

QMAKE_CXXFLAGS         += $$QMAKE_CFLAGS -std=gnu++98
QMAKE_CXXFLAGS_DEPS    += $$QMAKE_CFLAGS_DEPS
QMAKE_CXXFLAGS_WARN_ON += $$QMAKE_CFLAGS_WARN_ON
QMAKE_CXXFLAGS_WARN_OFF += $$QMAKE_CFLAGS_WARN_OFF
QMAKE_CXXFLAGS_RELEASE += $$QMAKE_CFLAGS_RELEASE

```

修改源码“mkspecs/common/g++-base.conf”文件中的“QMAKE\_CXX”参数，在参数后面添加`-std=gnu++98`

```

root@ubuntu: /home/wxq/Desktop/qt-everywhere-opensource-src-4.8.7
File Edit View Search Terminal Help
#
# Qmake configuration for the GNU C++ compiler
#
# Before making changes to this file, please read the comment in
# gcc-base.conf, to make sure the change goes in the right place.
#
# To verify that your change has the desired effect on the final configuration
# you can use the manual test in tests/manual/mkspecs.
#
QMAKE_CC = gcc

QMAKE_LINK_C      = $$QMAKE_CC
QMAKE_LINK_C_SHLIB = $$QMAKE_CC

QMAKE_CFLAGS_RELEASE_WITH_DEBUGINFO += -O2 -g

QMAKE_CXX = g++ -std=gnu++98

QMAKE_LINK      = $$QMAKE_CXX
QMAKE_LINK_SHLIB = $$QMAKE_CXX

```

## 源码编译配置

```
1  industio@industio$ ./configure --static --debug-and-release -nomake demos -nomake examples -no-openssl -no-exceptions
```

```
root@ubuntu:/home/wxq/Desktop/qt-everywhere-opensource-src-4.8.7# ./configure --static --debug-and-release -nomake demos -nomake examples -no-openssl -no-exceptions
which edition of Qt do you want to use ?

Type 'c' if you want to use the Commercial Edition.
Type 'o' if you want to use the Open Source Edition.
o

This is the Open Source Edition.

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.
yes

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

如果没有报错，配置结束后提示内容如下：

执行”make”编译源码，执行”make install”在编译完成后，将Qt4.8.7安装到默认路径”/usr/local/Trolltech/Qt-4.8.7”目录。

```
for /home/industio_work/Qt/qt-everywhere-opensource-src-4.8.7/src/

Qt is now configured for building. Just run 'make'.
Once everything is built, you must run 'make install'.
Qt will be installed into /usr/local/Trolltech/Qt-4.8.7

To reconfigure, run 'make confclean' and 'configure'.
```

## 编译安装

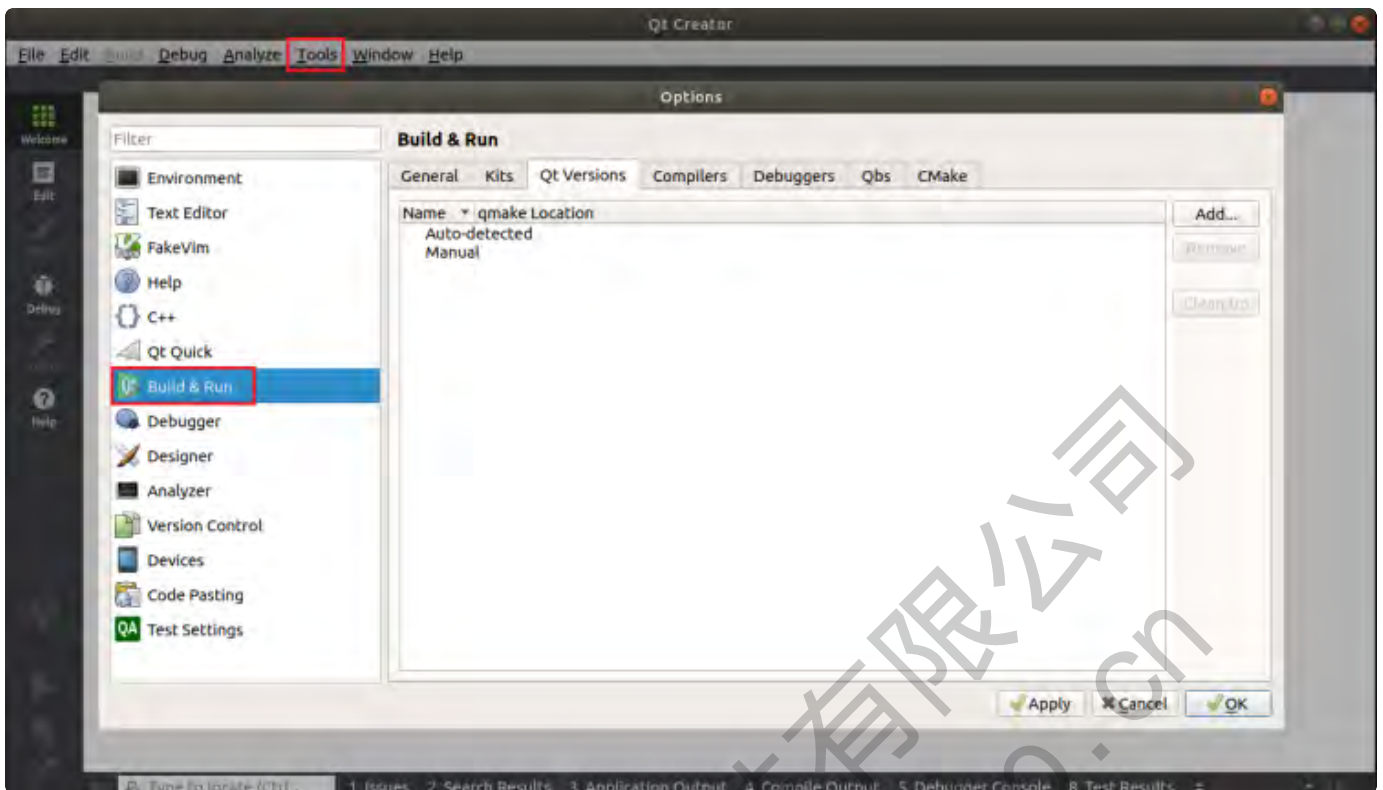
```
1  industio@industio$ make && make install
```

安装完成后，安装的文件位于”/usr/local/Trolltech/Qt-4.8.7”目录，进入sdk的bin目录，执行`./qmake -v`即可查看安装的qmake和Qt的版本。

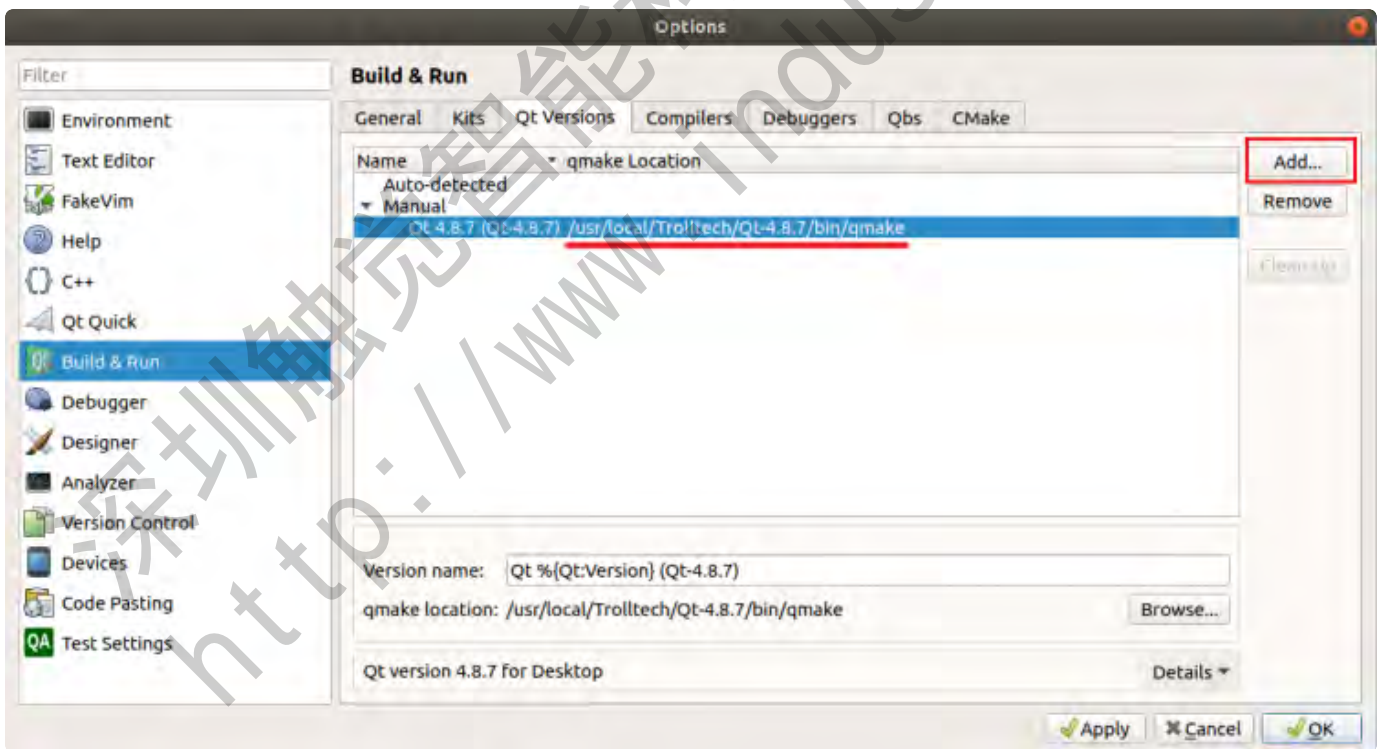
```
root@ubuntu:/usr/local/Trolltech/Qt-4.8.7/bin# ./qmake -v
QMake version 2.01a
Using Qt version 4.8.7 in /usr/local/Trolltech/Qt-4.8.7/lib
root@ubuntu:/usr/local/Trolltech/Qt-4.8.7/bin#
```

## Qt Creator添加Qt4.8.7开发套件

Qt Creator 4.4.1需要添加Qt4.8.7开发套件才可能在Ubuntu18.04 PC端编译、运行和调试Qt代码。打开Qt Creator 4.4.1软件，选择”Tool“->”Options”，在弹出的界面中选择”Build & Run”。

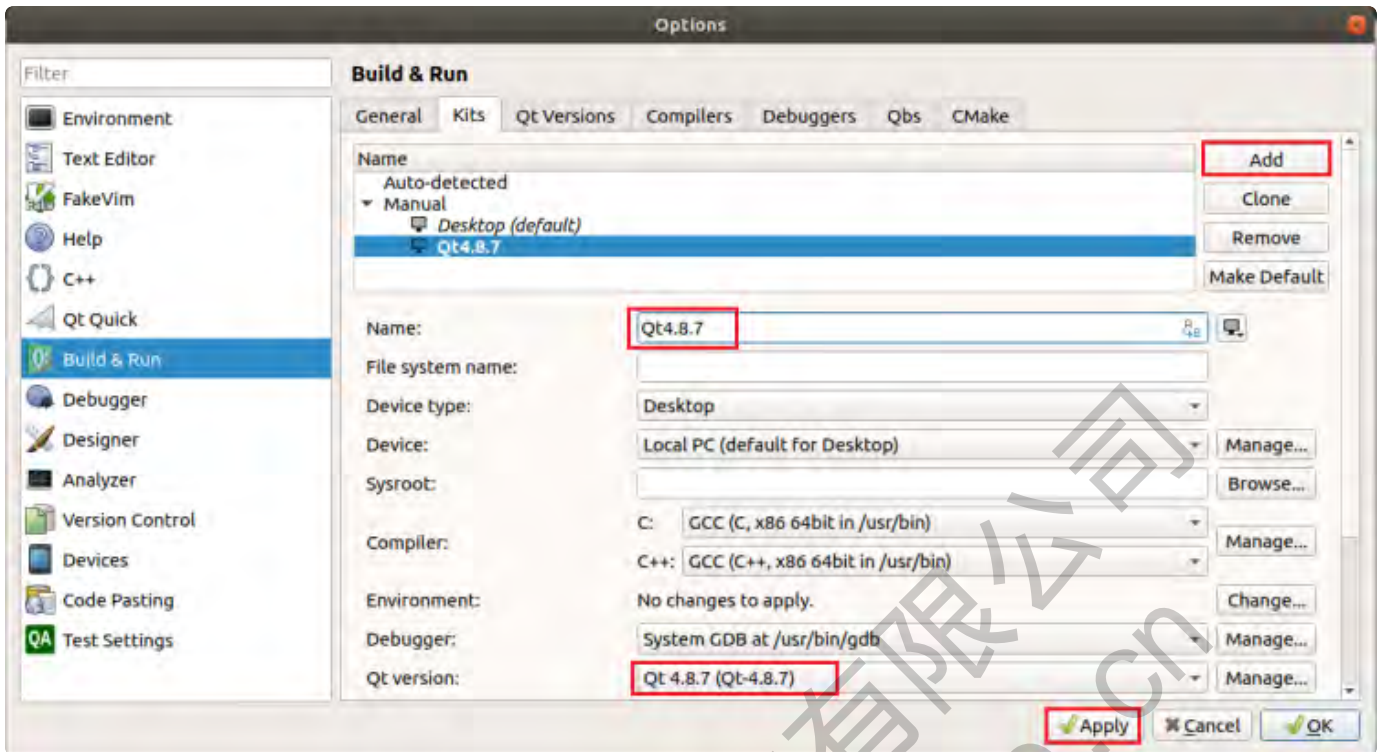


在”Qt Versions”栏点击右边的”Add..”按钮，进入文件资源浏览界面，选择”/usr/local/Trolltech/Qt-4.8.7/bin/qmake“，点击右下方的”Apply“按钮生效添加的Qt Version。



点击进入Kits栏，点击”add“按钮新增一项kit，修改名字为”Qt4.8.7“，Qt Version选择步骤2中添加的Qt-4.8.7，点击”Apply“按钮生效，点击”OK“按钮关闭窗口。

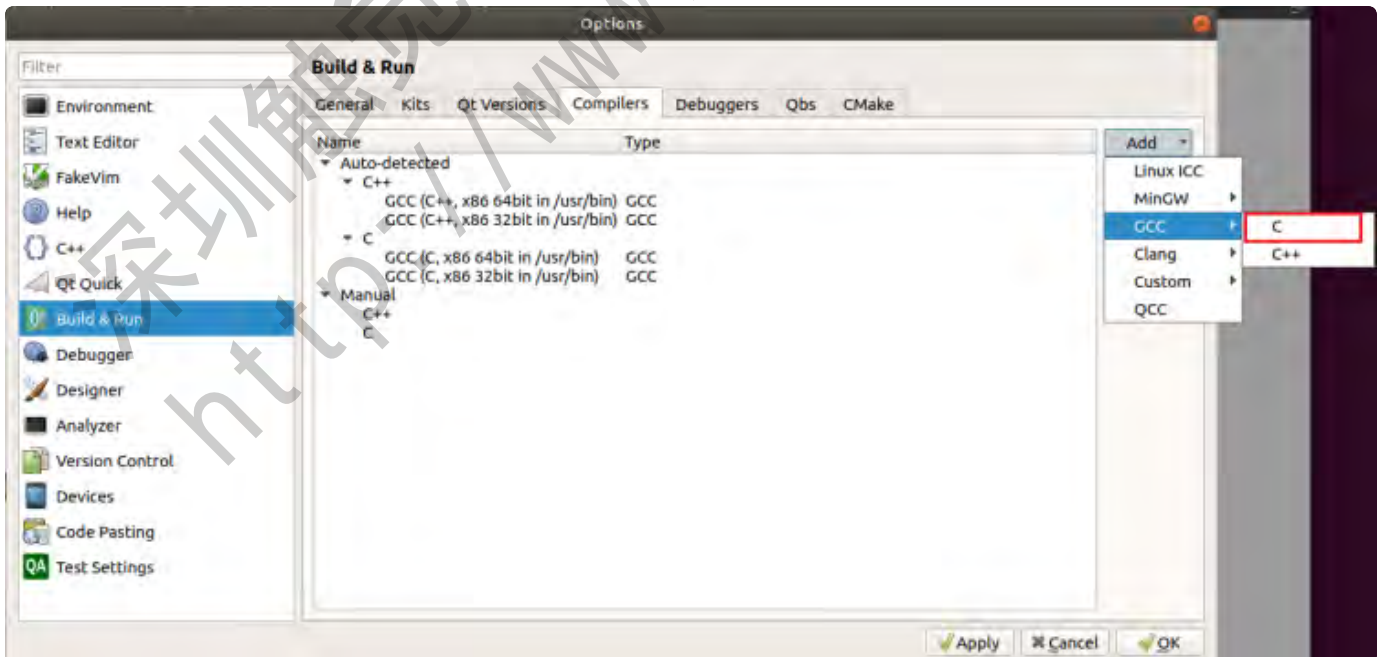




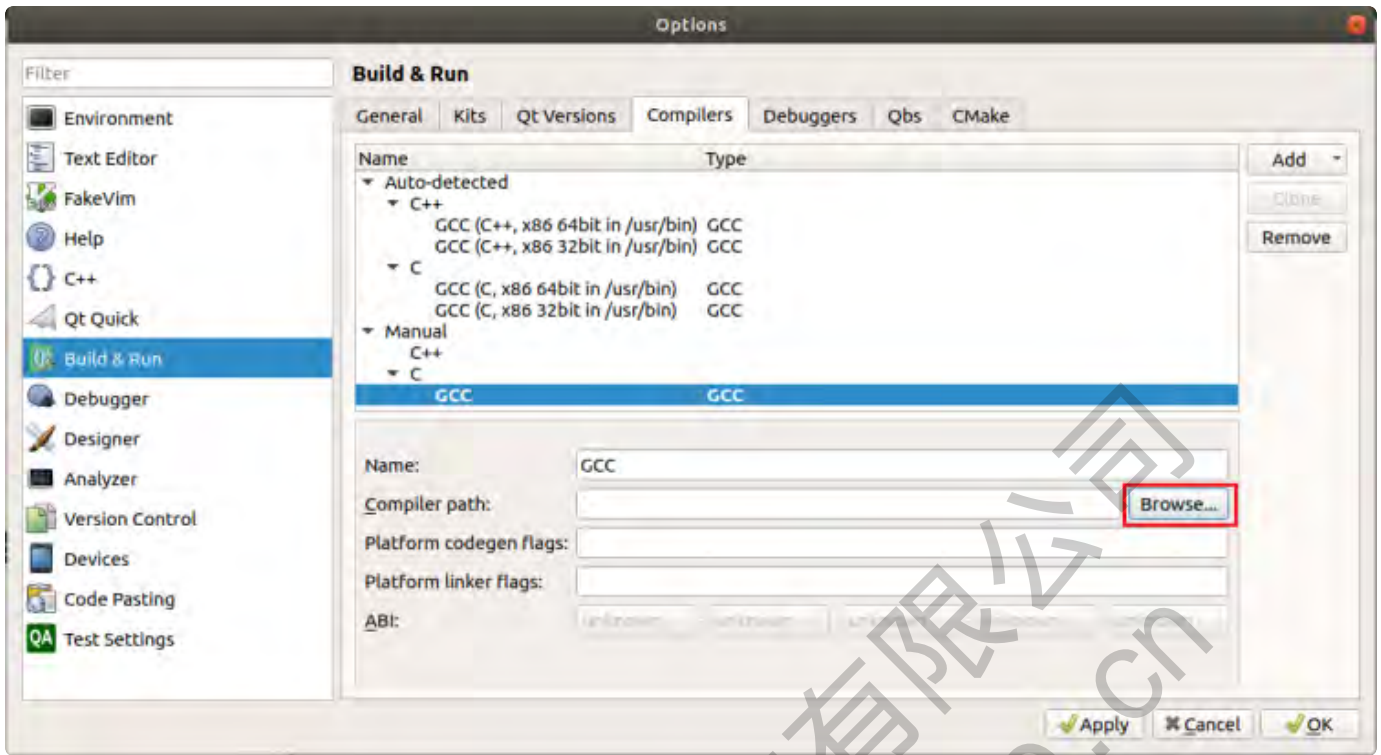
## Qt Creator添加Qt4.8.7交叉编译

开发的Qt程序需要经过交叉编译工具链编译后，才可以在SSD20x开发板的Linux ARM平台上运行。QT Creator交叉编译工具链的设置界面位于的“Tools”->“Options”->“Build & Run”，需要在Compilers、Debuggers、QtVersions和Kits四配置中添加对应的文件。

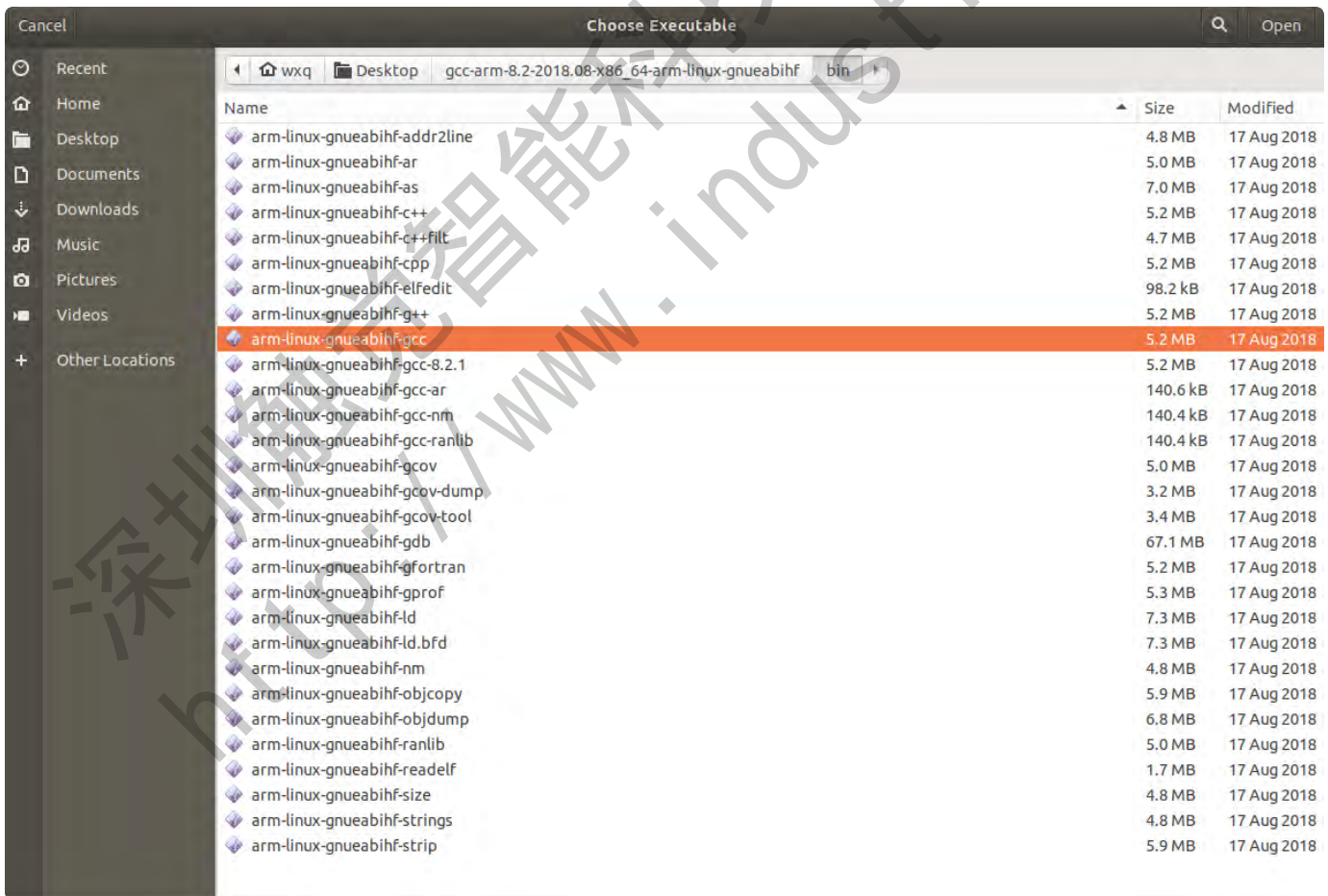
添加gcc在界面的右侧点击Add->GCC->C，添加一个GCC编译器。



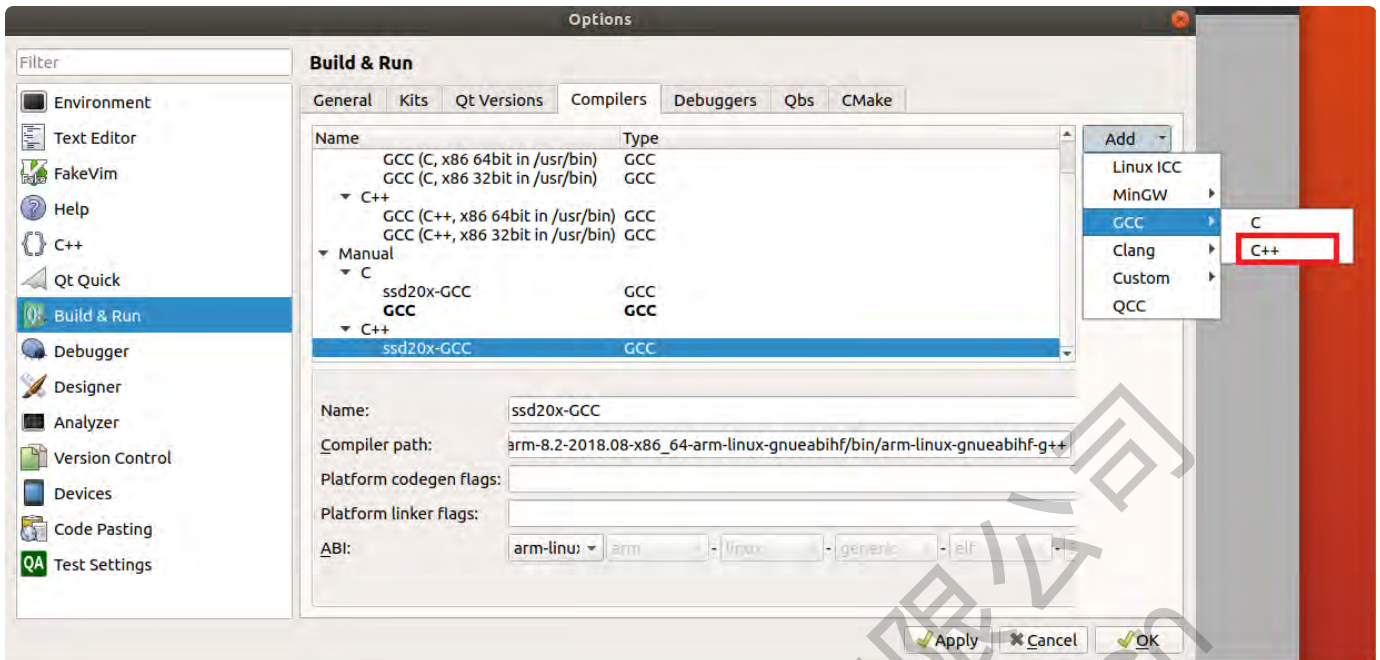
选中新添加的GCC编译器，然后点击下方的“Browse”按钮开启文件资源浏览器；



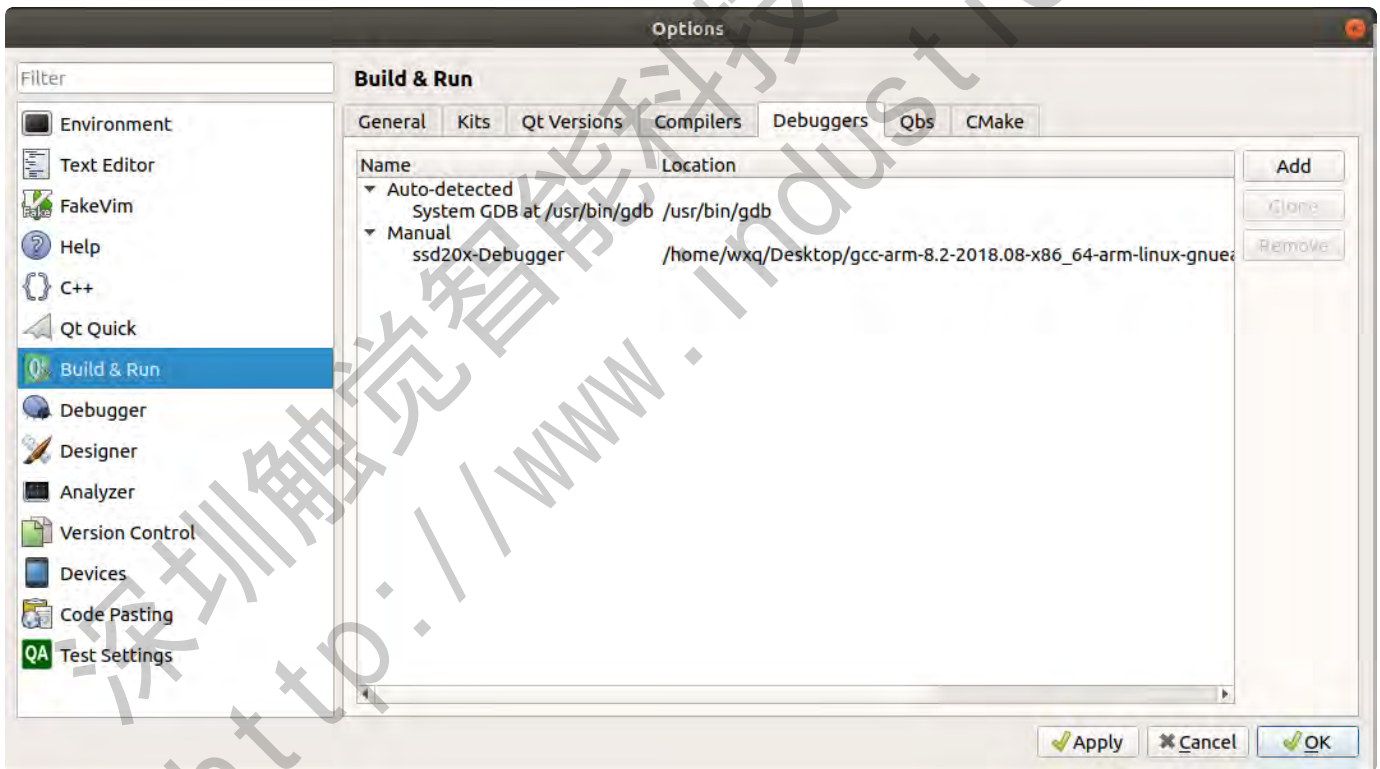
选中gcc编译工具“arm-linux-gnueabi-hf-gcc”，添加完成点击“Apply”生效设置



按照GCC的添加方法，添加g++



在Debuggers栏点击“Add”按钮新增一个Debugger，选择“arm-linux-gnueabi-gdb”并修改名字为“ssd20x-Debugger”，点击“Apply”按钮生效设置。



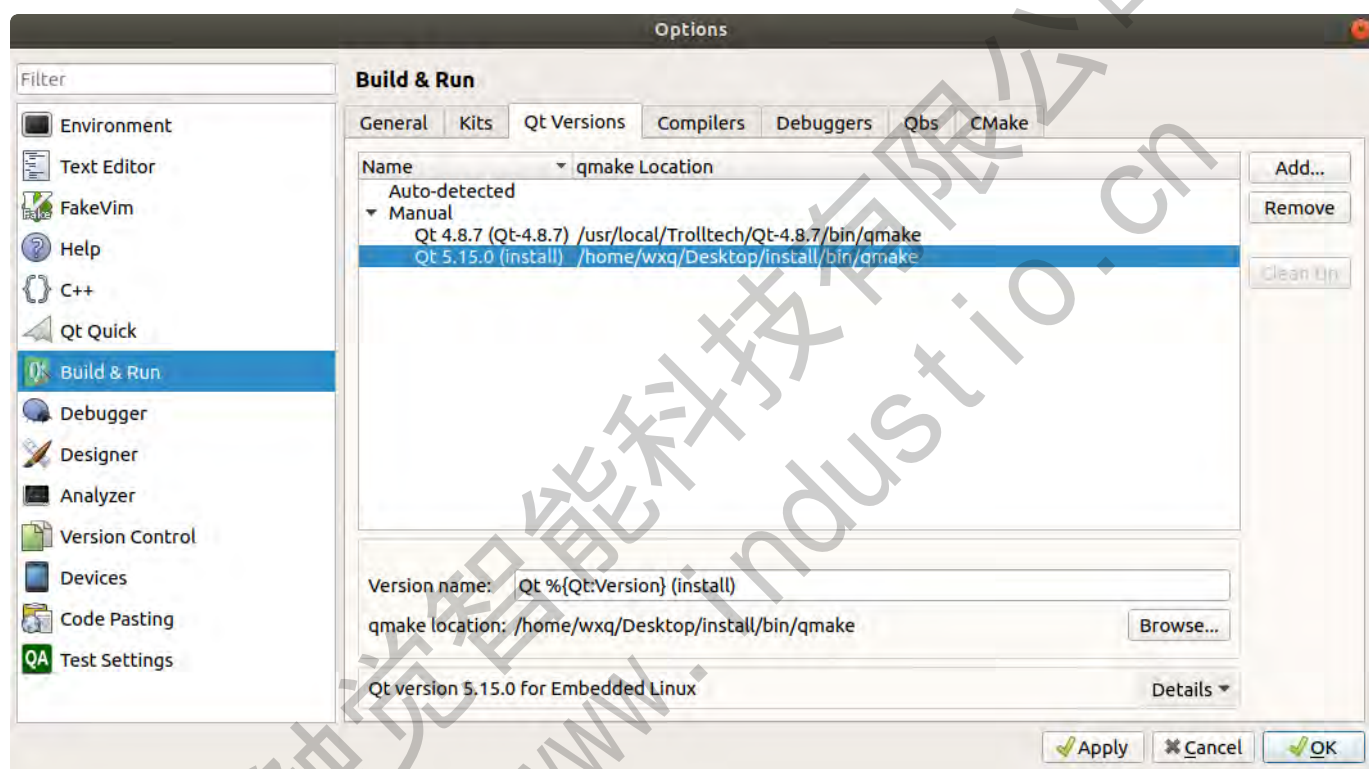
在配置Qt Versions时把网盘中的Qt5.15解压到Ubuntu中



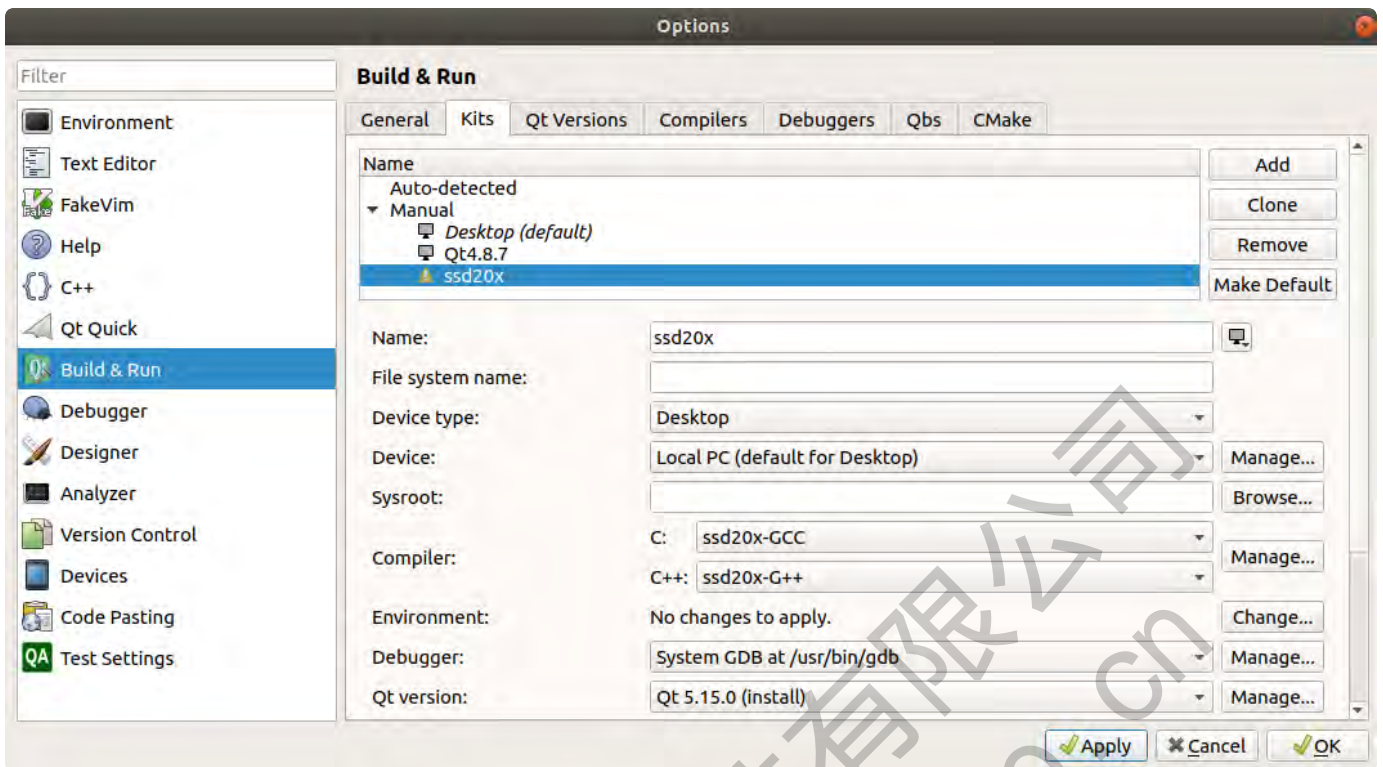
返回上一级 | ... > 开发板文档 > test > qt\_5.15... > share

文件名	大小	修改日期
readme.txt	571B	2021-04-26 11:51
qt_share_env.sh	290B	2021-04-26 11:51
qt_install_share_5.15.0.tar.bz2	104.4M	2021-04-26 11:51
qmake.conf	808B	2021-04-26 11:51
make_share.sh	576B	2021-04-26 11:51

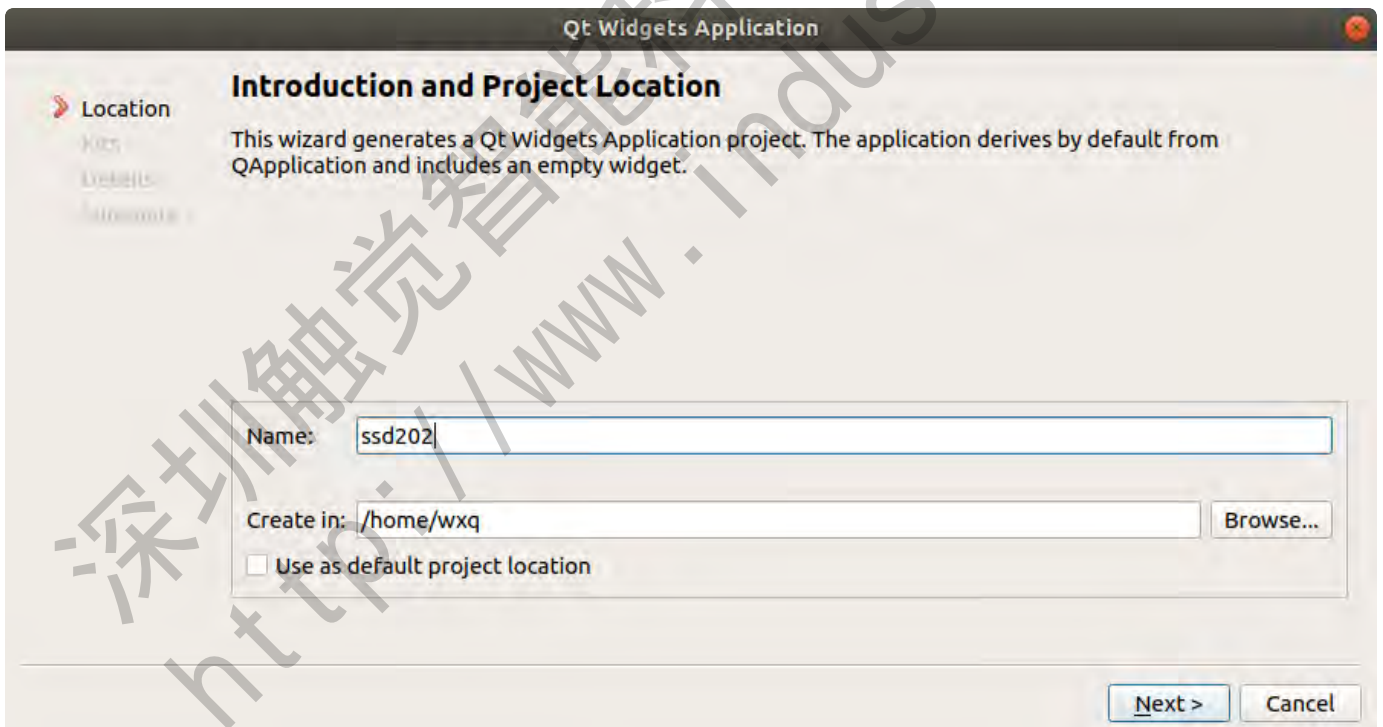
在Qt Versions栏点击“add”打开文件资源浏览器，选择“/install/bin/qmake”，点击“Apply”按钮生效设置。



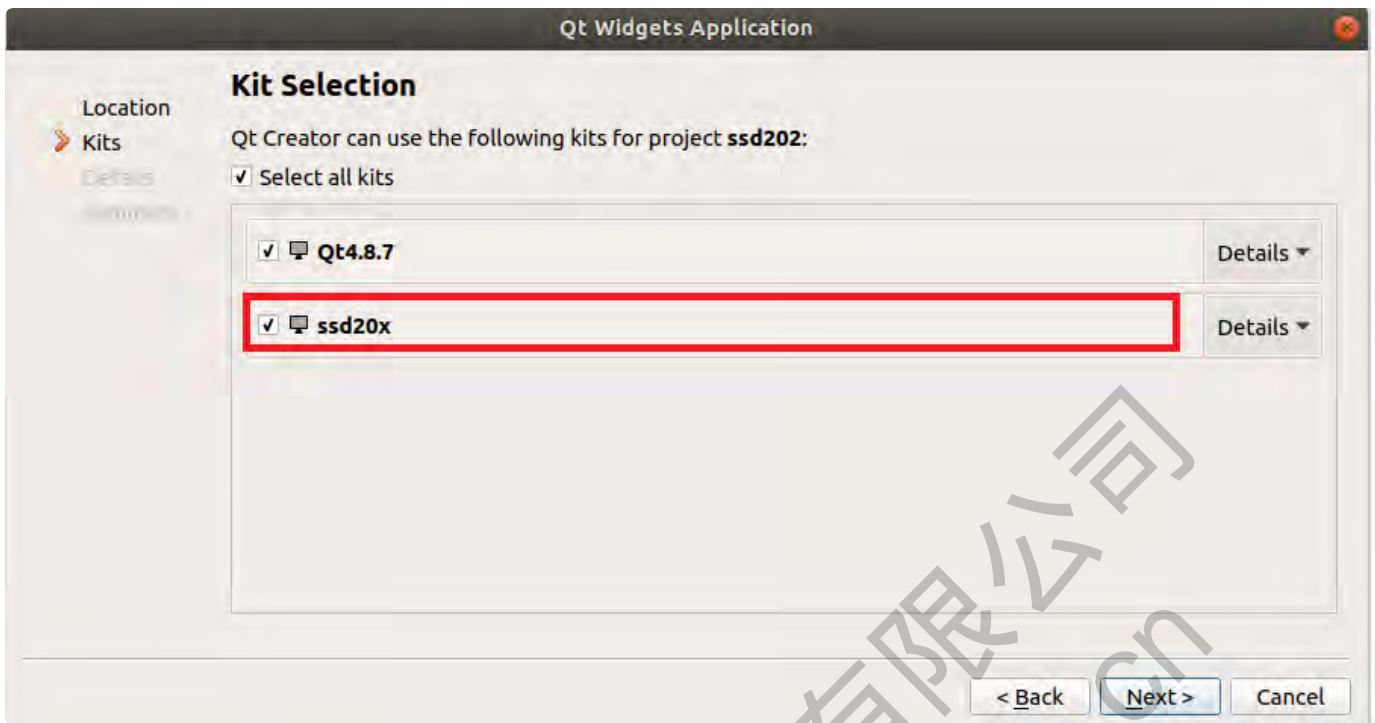
在Kits栏，点击“Add”按钮添加一个kit，并修改名字为“ssd20x”，选择Compiler C: ssd20x-GCC；Compiler C++: ssd20x-G++；Qt Version :“Qt5.15.0 (install)”，最后点击“Apply”生效配置。



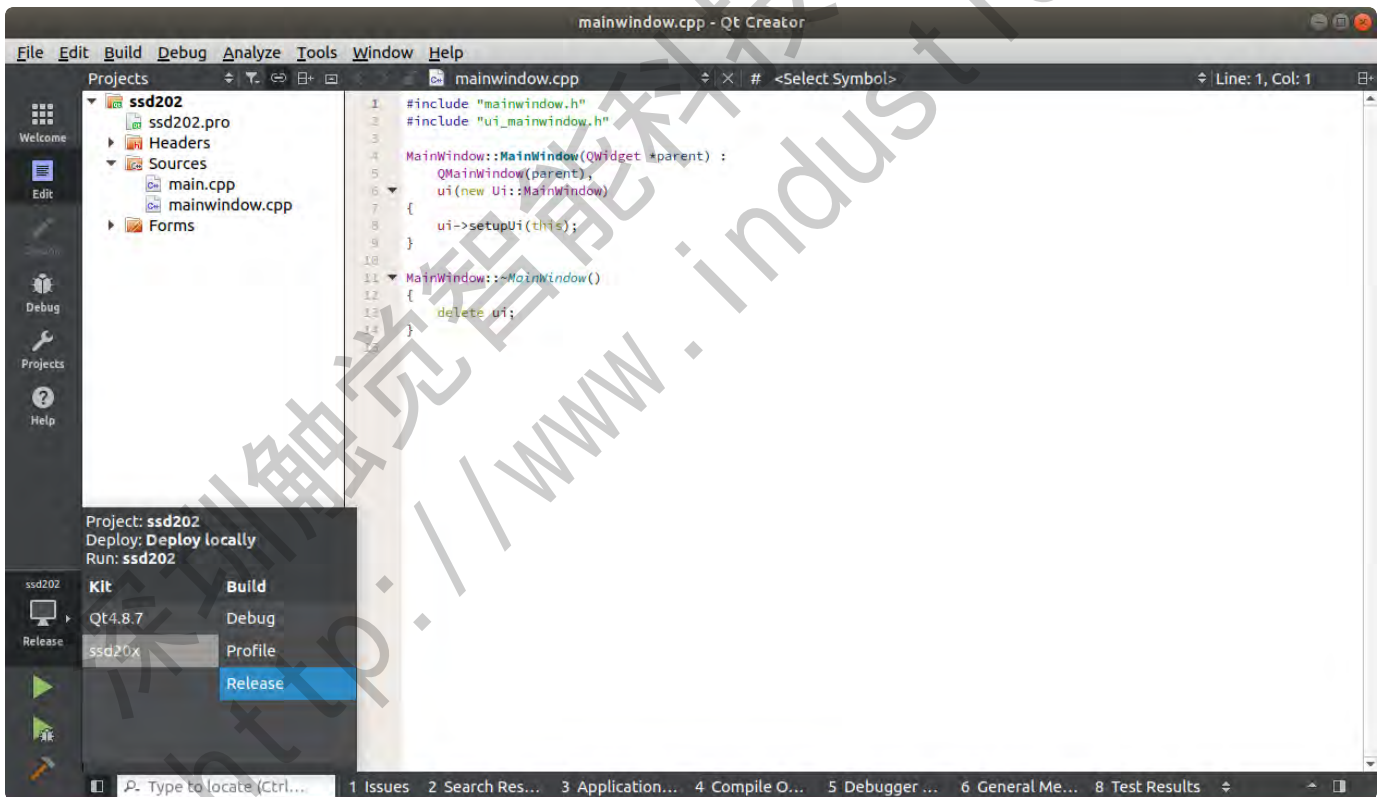
验证Qt Creator交叉编译配置，点击Qt Creator软件的“File→New File or Project”，新建一个“Qt widget Application”工程。



在“Kit Selection”界面，选择添加的“ssd202x”



在Qt Creator左侧选择“ssd20x->Release”

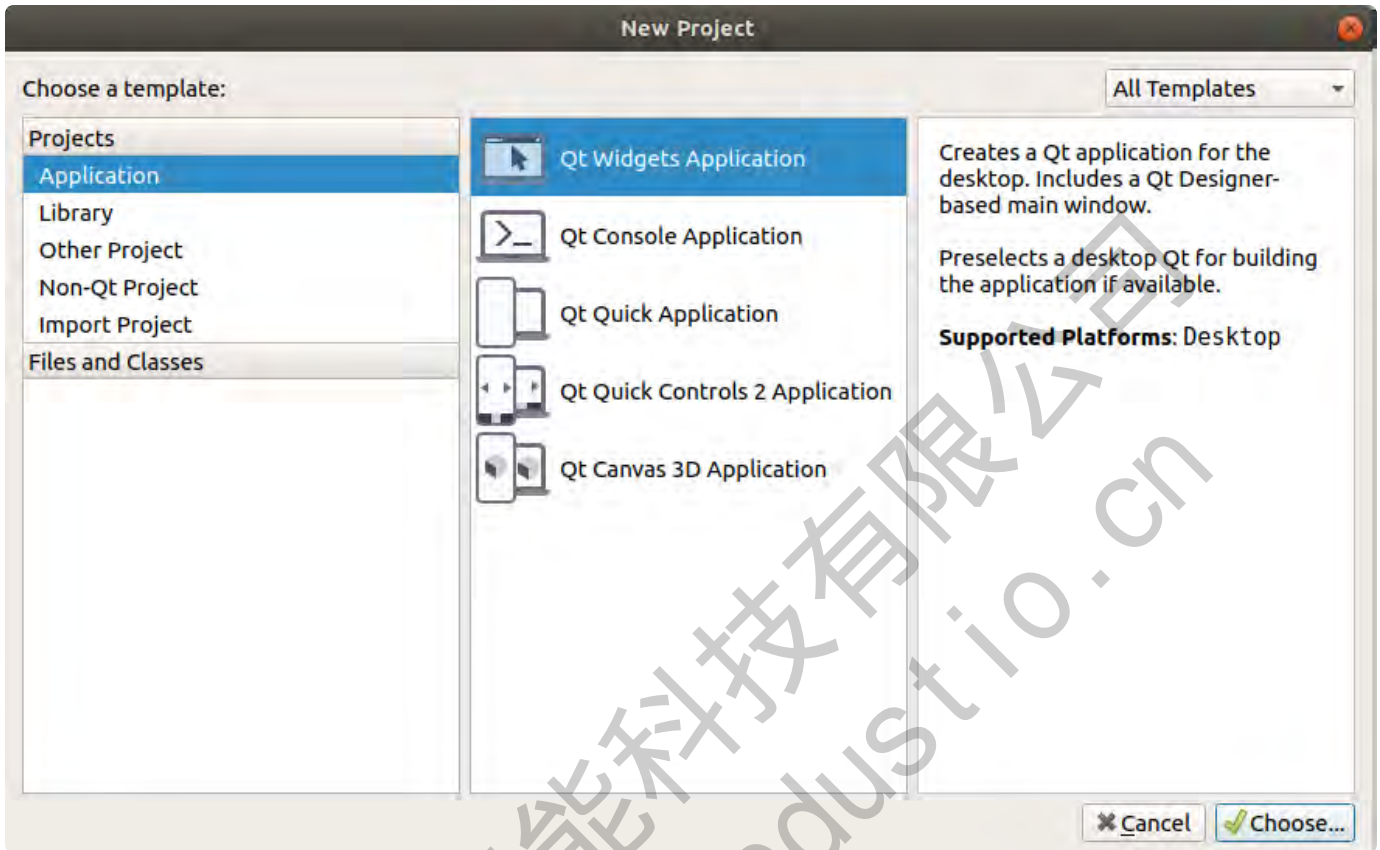


点击“Build”按钮编译；正常编译完，在新建工程的“/home/wxq”目录下，产生一个“build-ssd202-ssd20x-Release”文件夹，文件夹中包含编译的中间文件和开发板上运行的可执行文件“ssd202”。

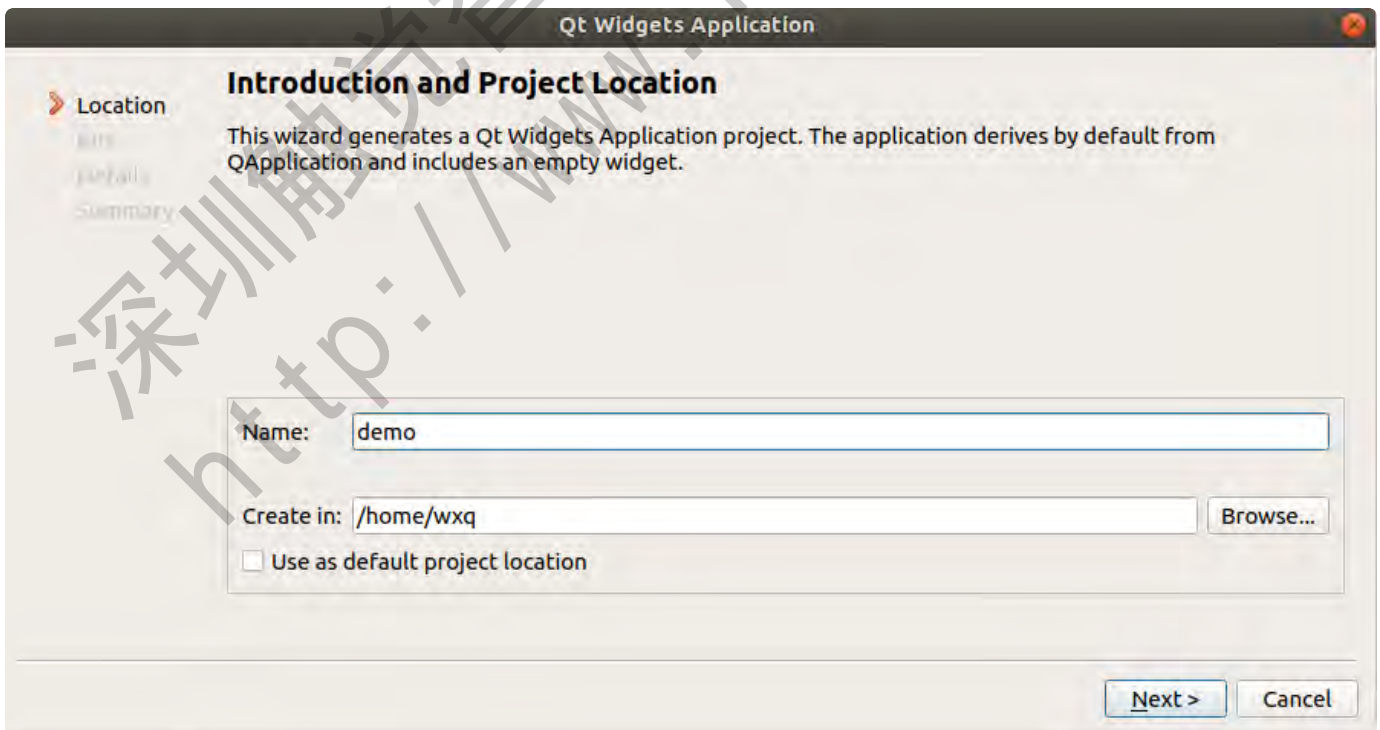
```
wxq@ubuntu:~/build-ssd202-ssd20x-Release$ ls
main.o      Makefile      moc_mainwindow.o  ssd202
mainwindow.o  moc_mainwindow.cpp  moc_predefs.h      ui_mainwindow.h
wxq@ubuntu:~/build-ssd202-ssd20x-Release$
```

## 编写Qt 程序在开发板中运行

这里以编写hello world为例，点击新建工程，选择“Qt Widgets Application”，之后点击“choose”



名字可以随便一个

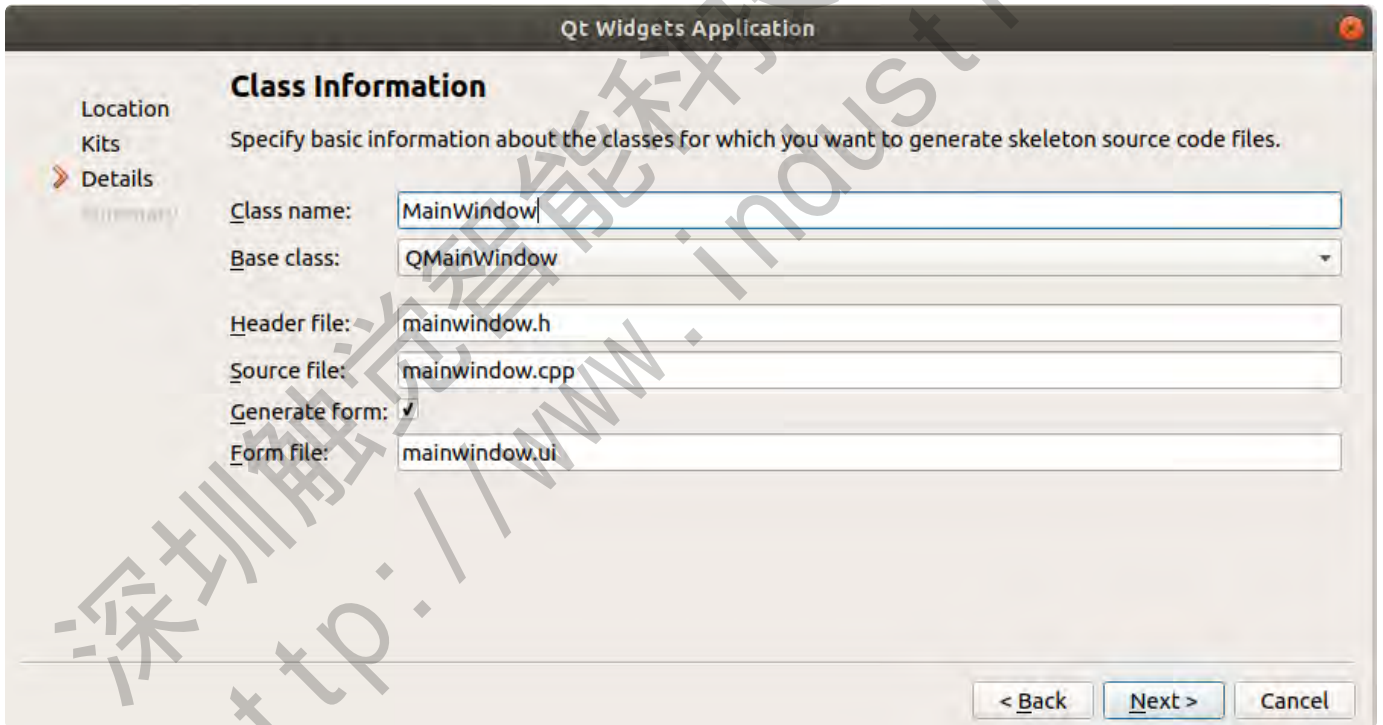


添加ssd20x

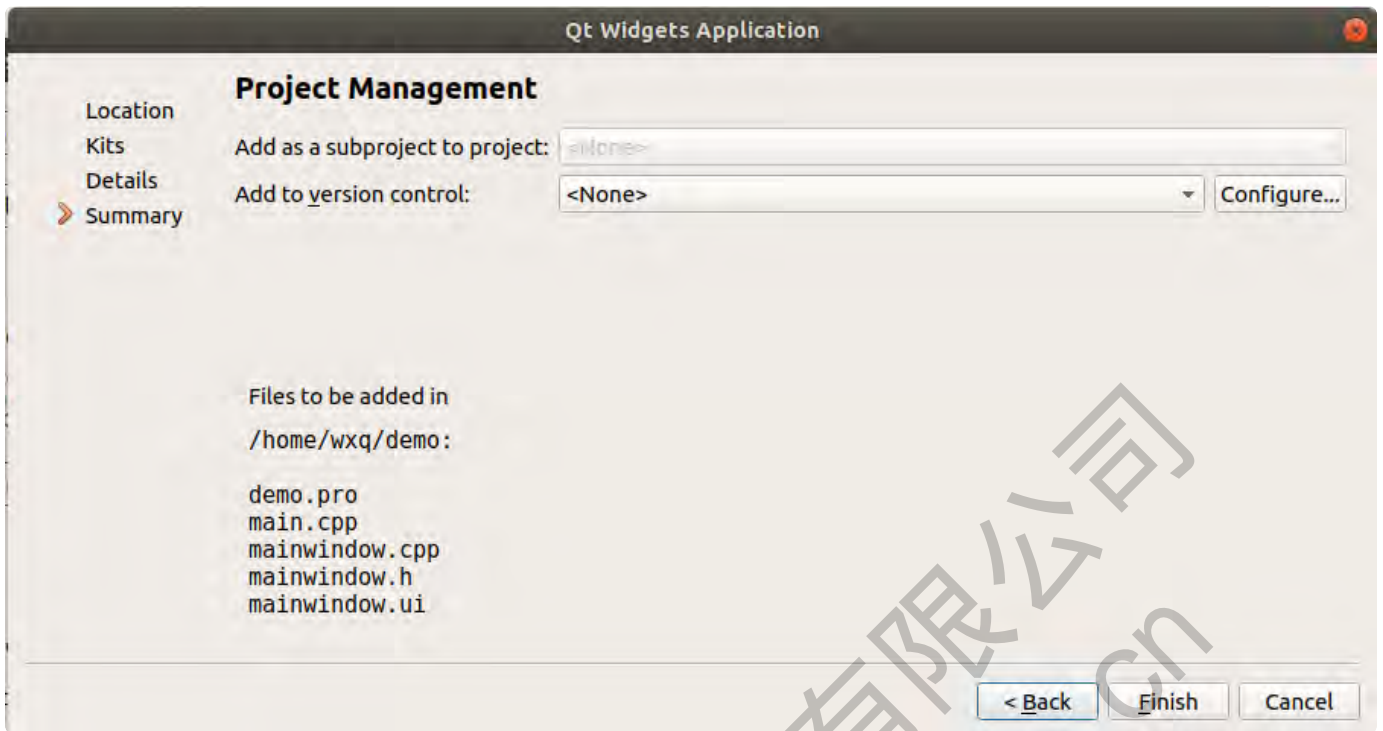




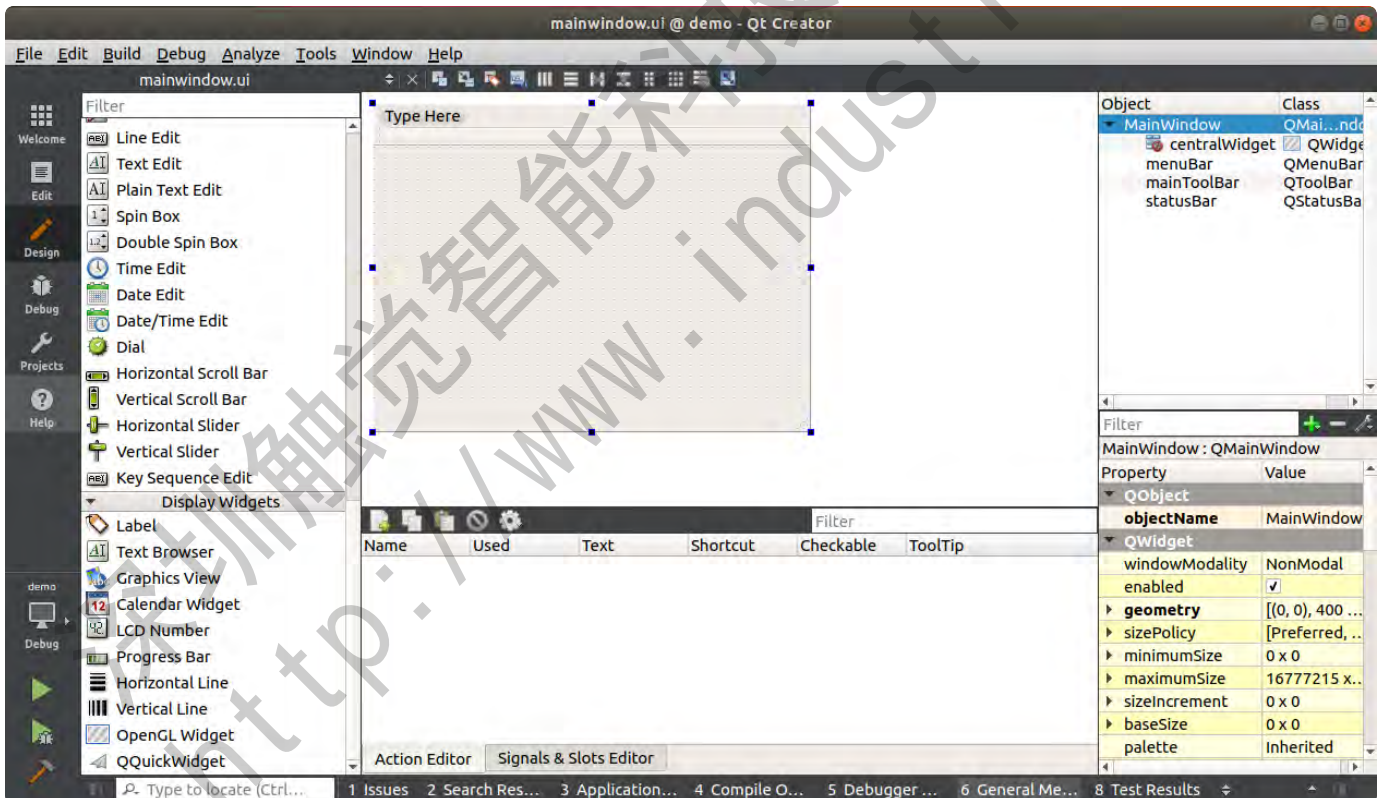
默认即可



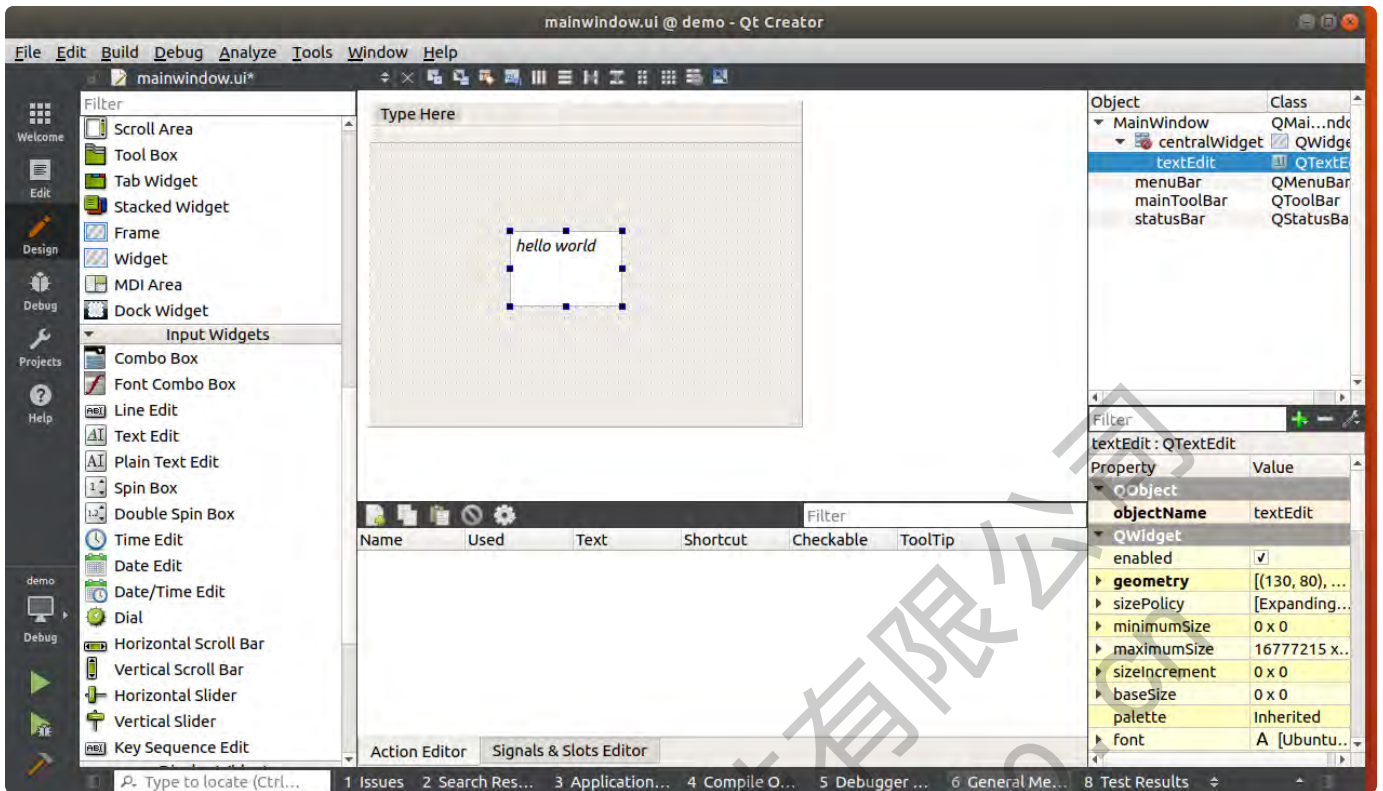
点击“Finish”



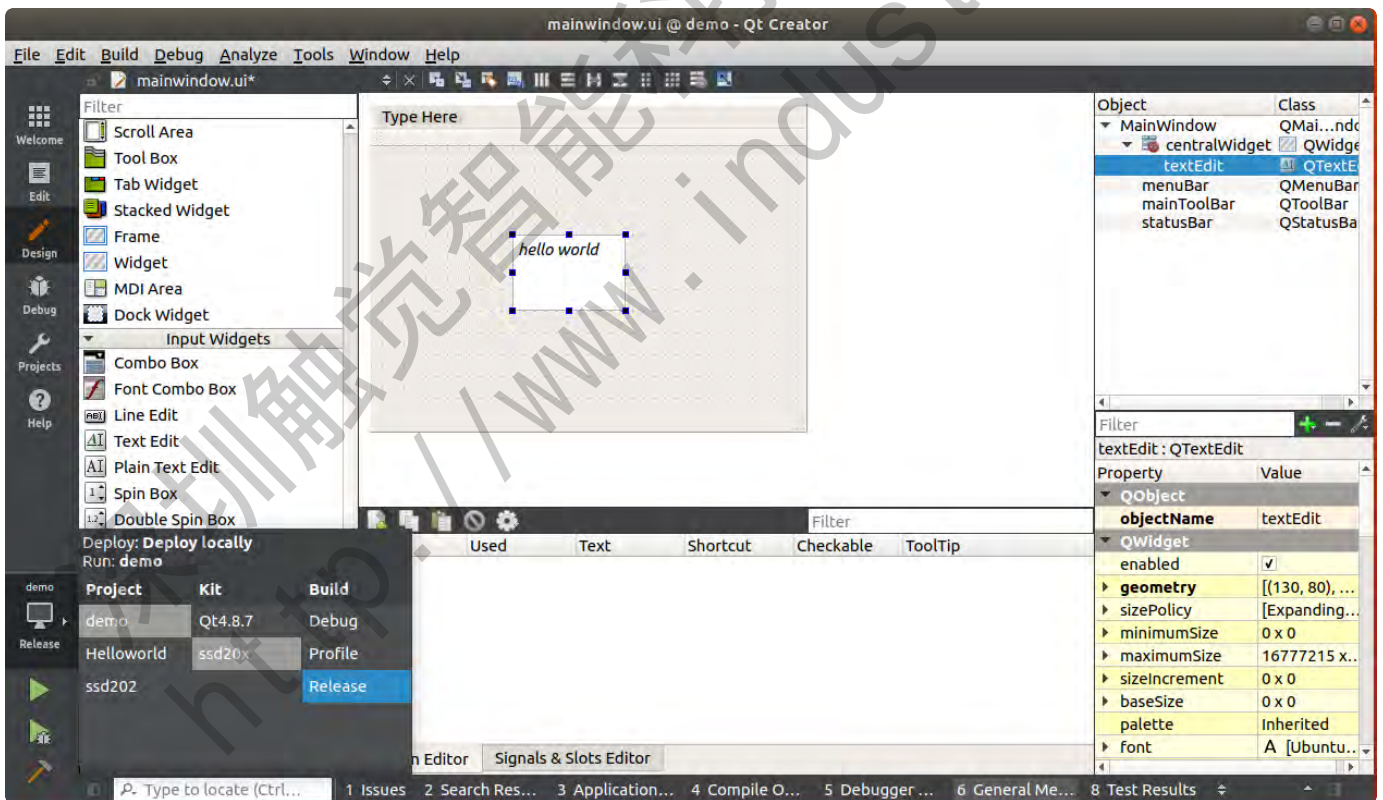
点击“Forms->mainwindow.ui”



选择“Text Edit”拉到“Type Here”中，写入“hello world”



debug选择“ssd20x->Release”



点击”Build”按钮编译；生产可执行文件，把文件拷贝到开发中运行



