Purple Pi R1 Qt Creator开发流程

Purple Pi R1

Qt Creator开发流程

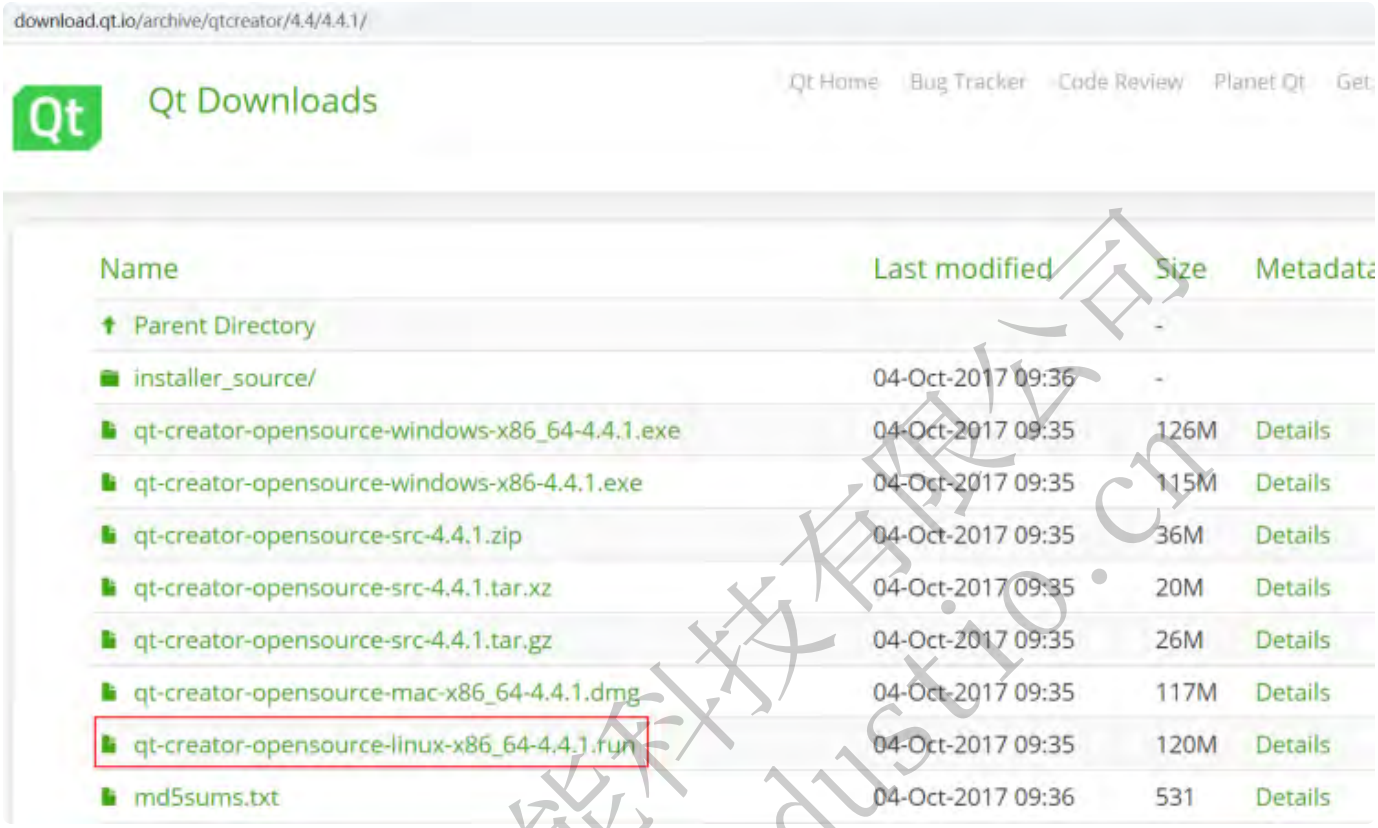
深圳触觉智能科技有限公司

www.industio.cn

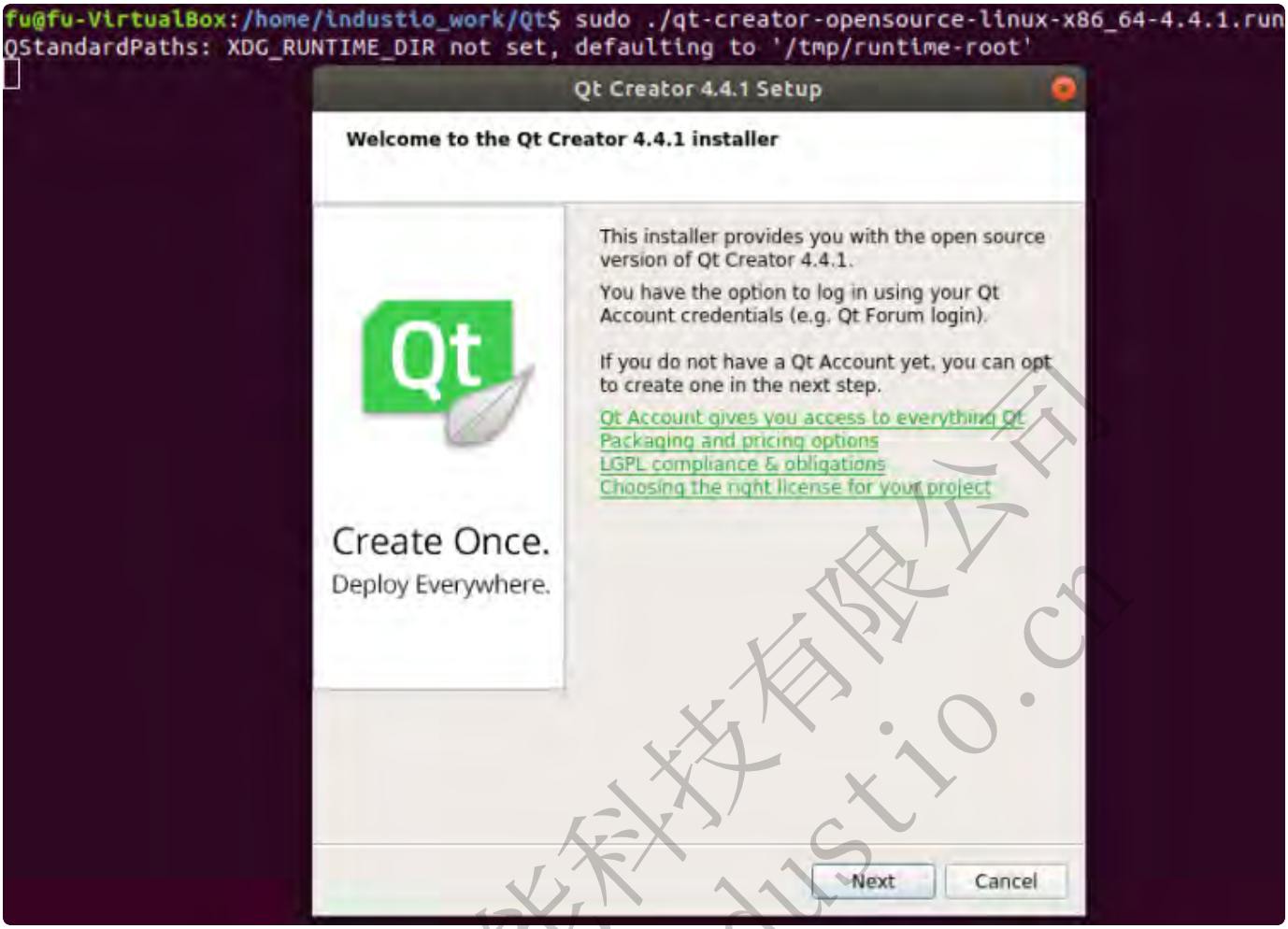
Qt Creator开发流程

配置Qt开发环境

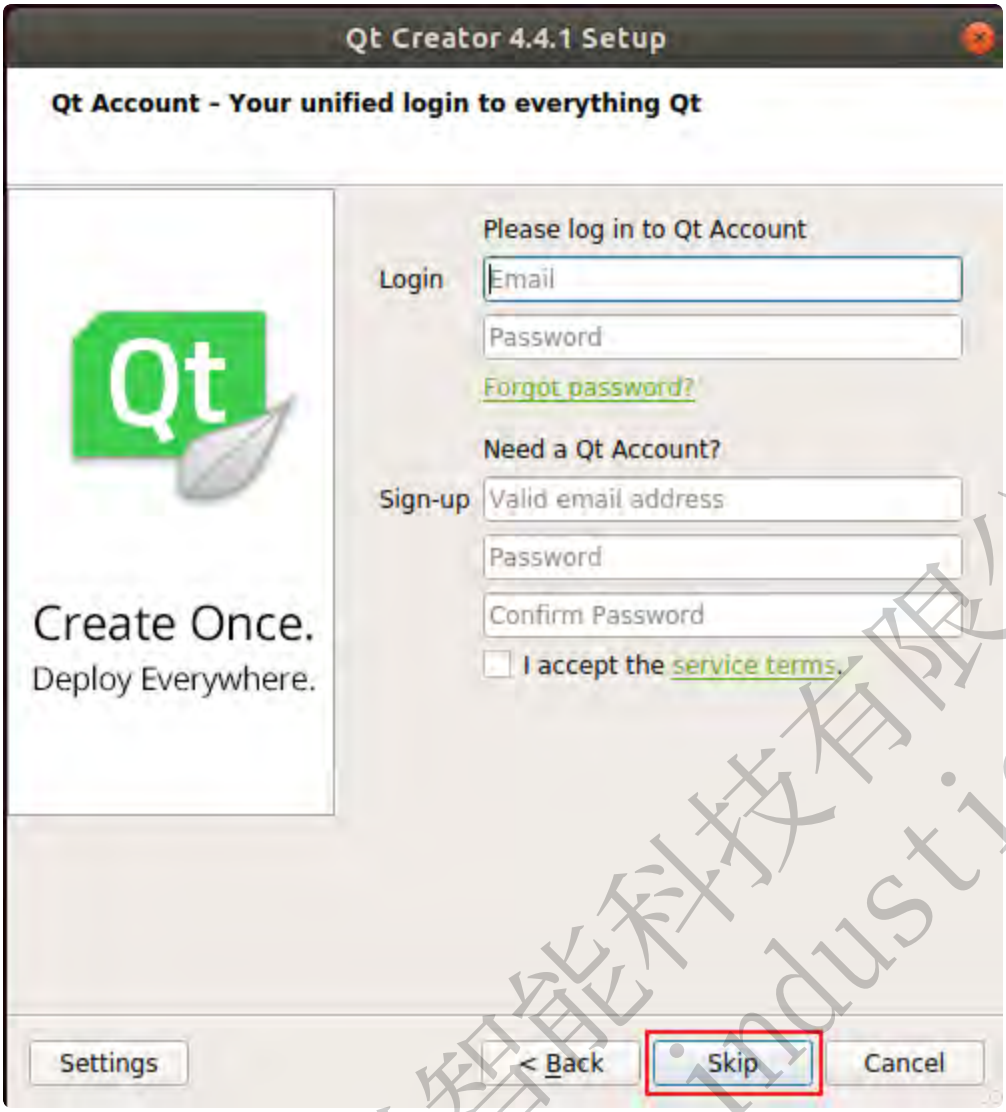
1、下载Qt Creator4.4.1下载地址：https://download.qt.io/archive/qtcreator/4.4/4.4.1/



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”按键进入下一步。如下图所示

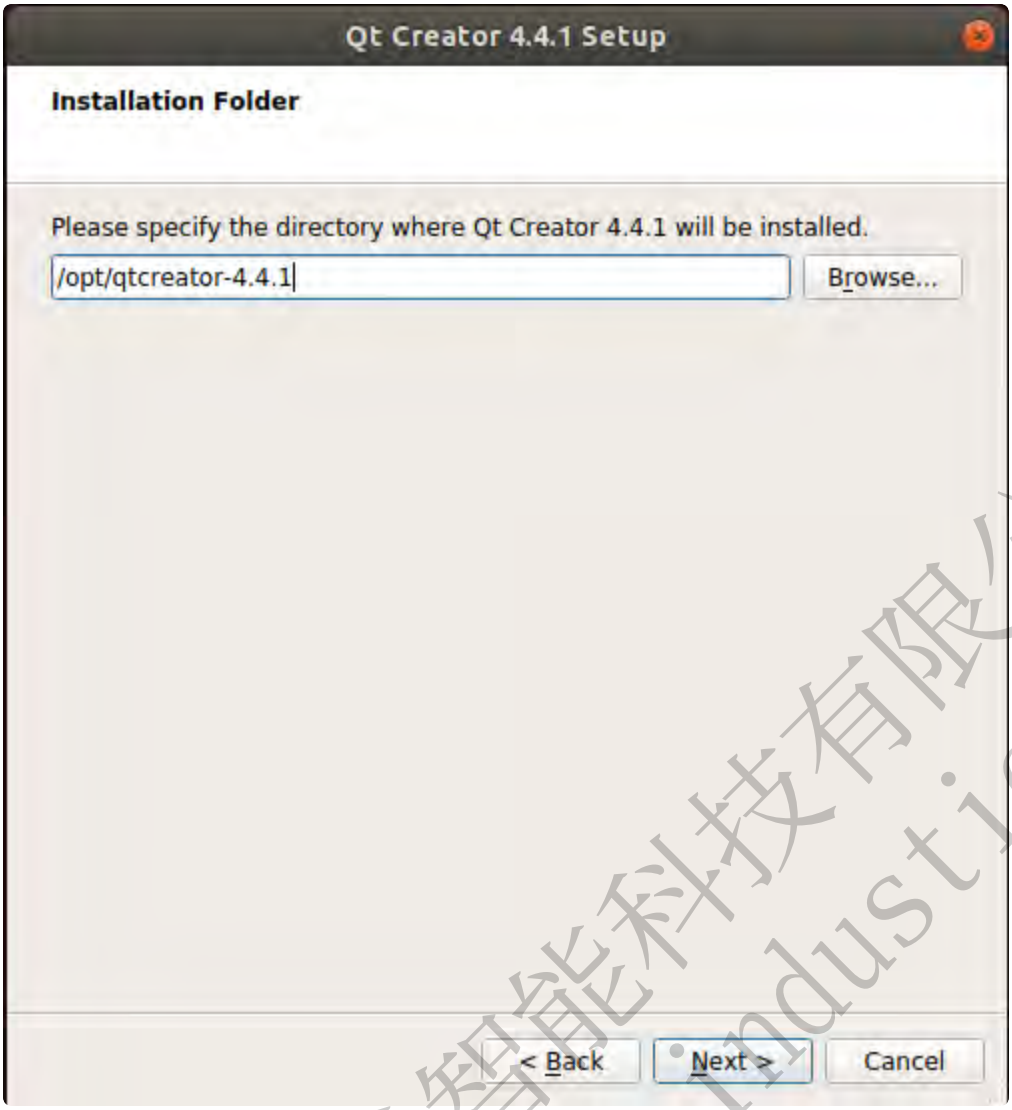


此处如果不需要登陆，点击“skip”跳过即可;

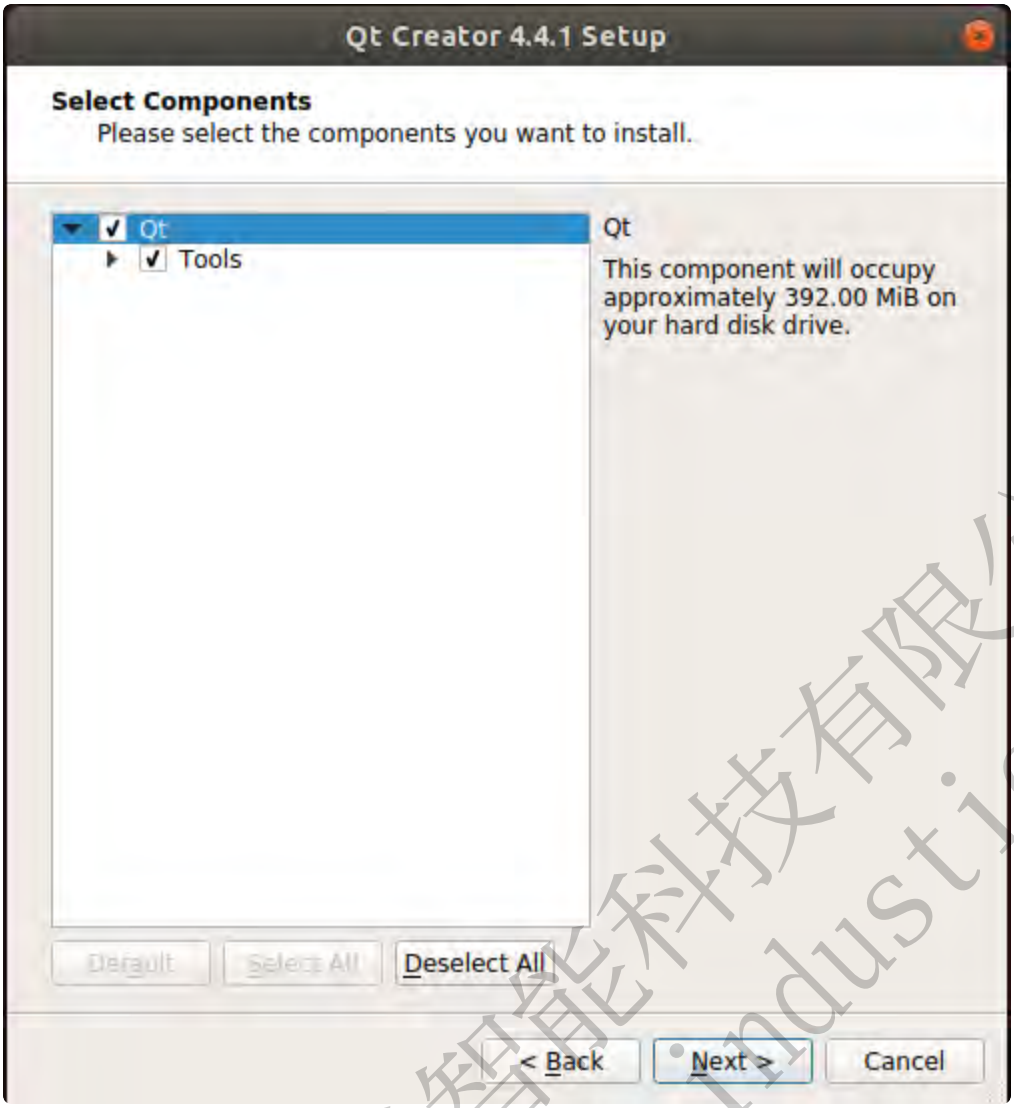


点击“Next”进入下一步；

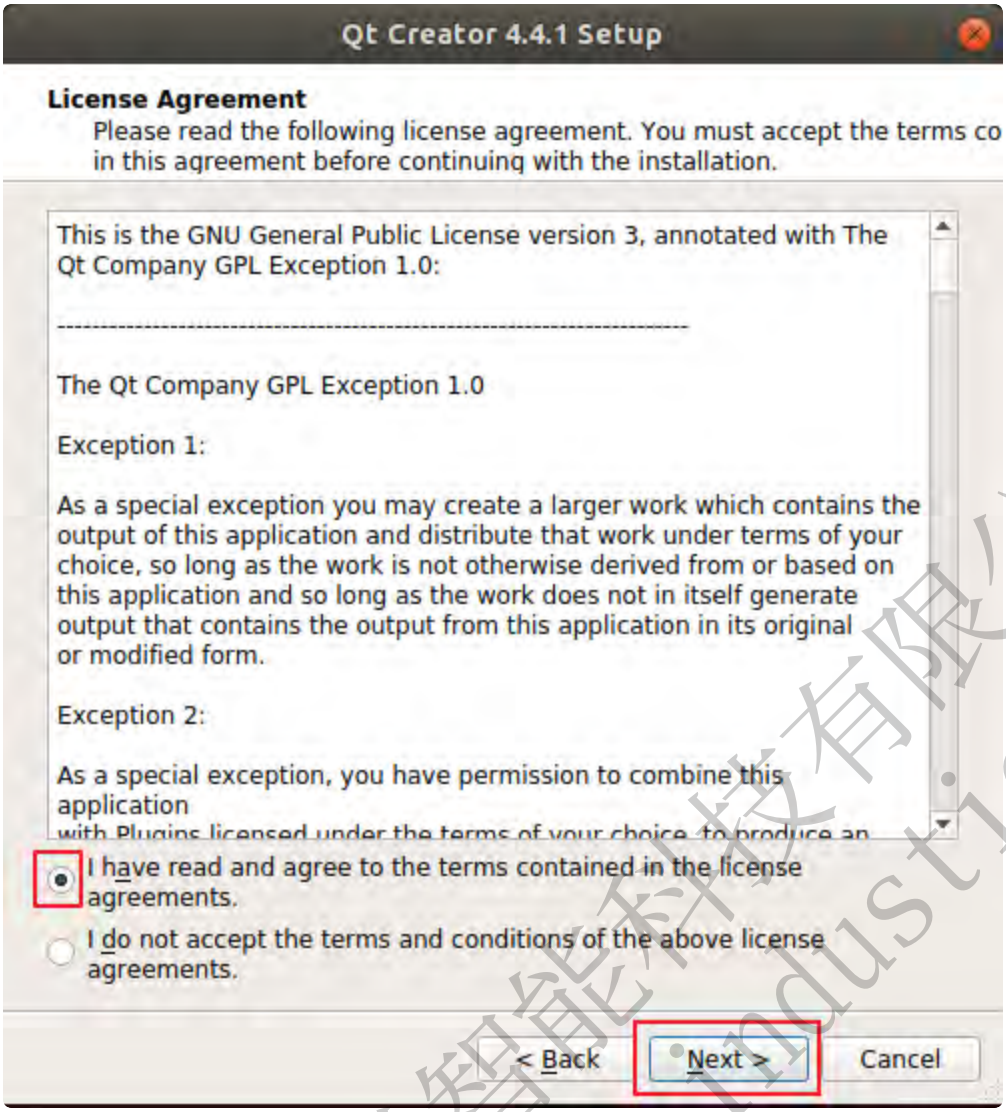
默认安装目录为“/opt/qtcreator-4.4.1”，点击“Next”进入下一步；



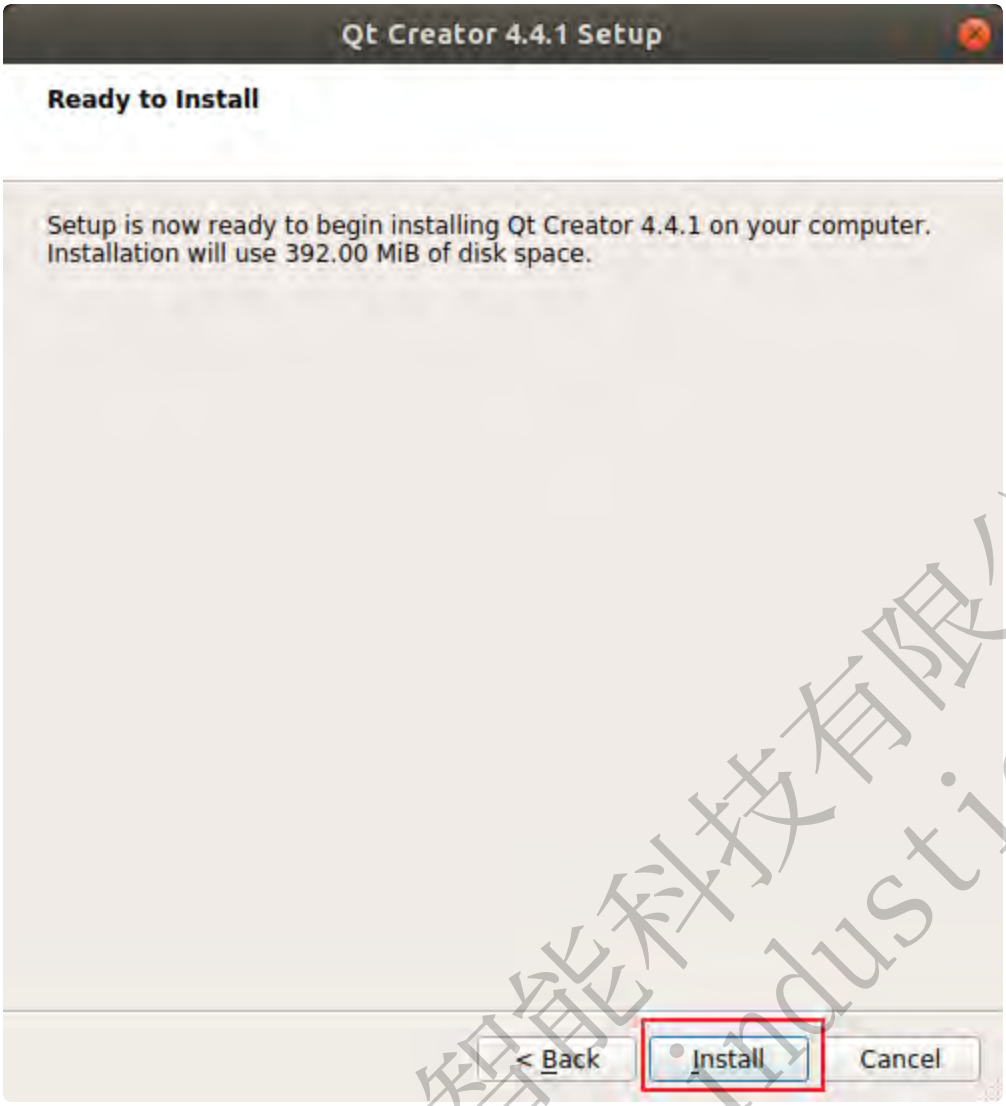
保留默认的勾选安装组件内容，点击“Next”进入下一步；



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；

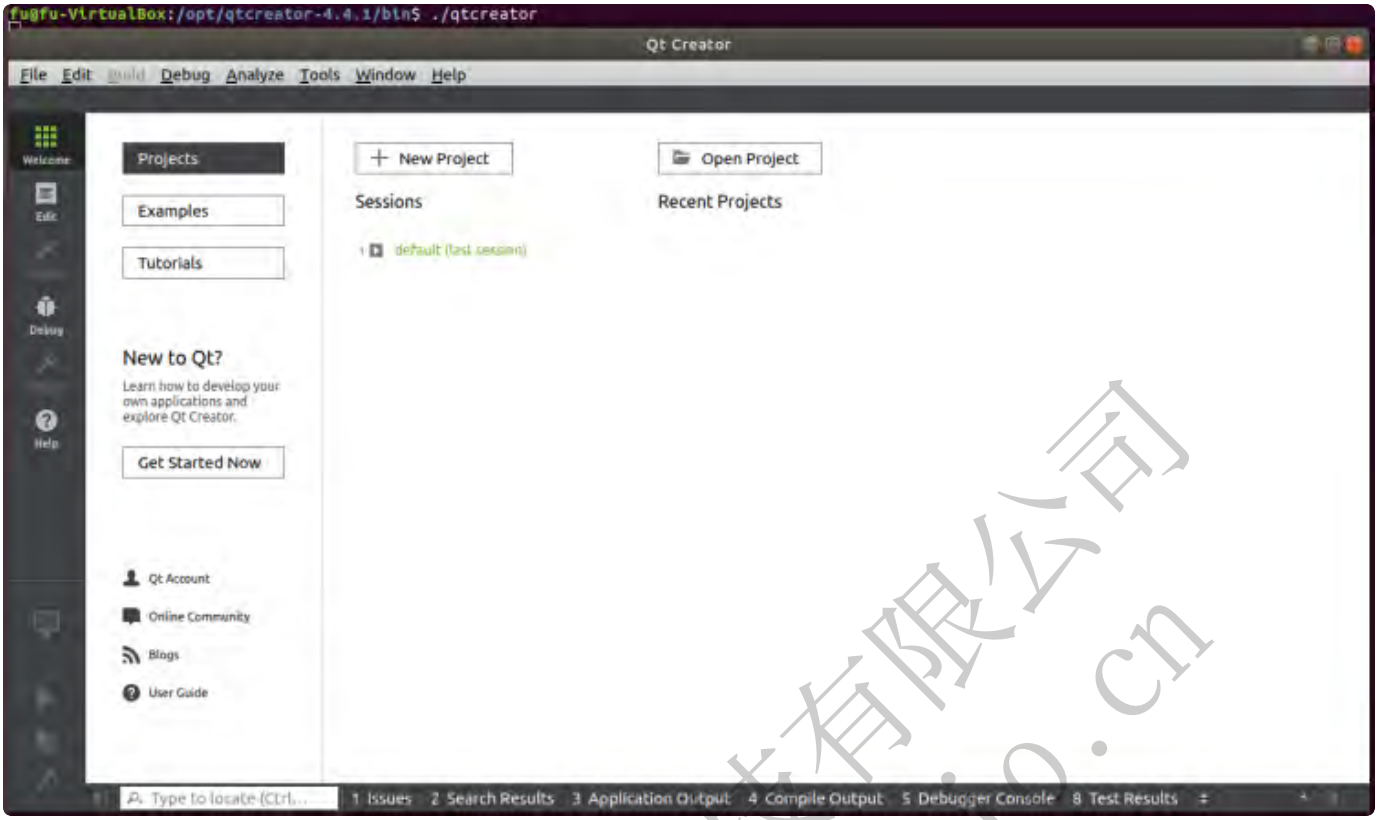


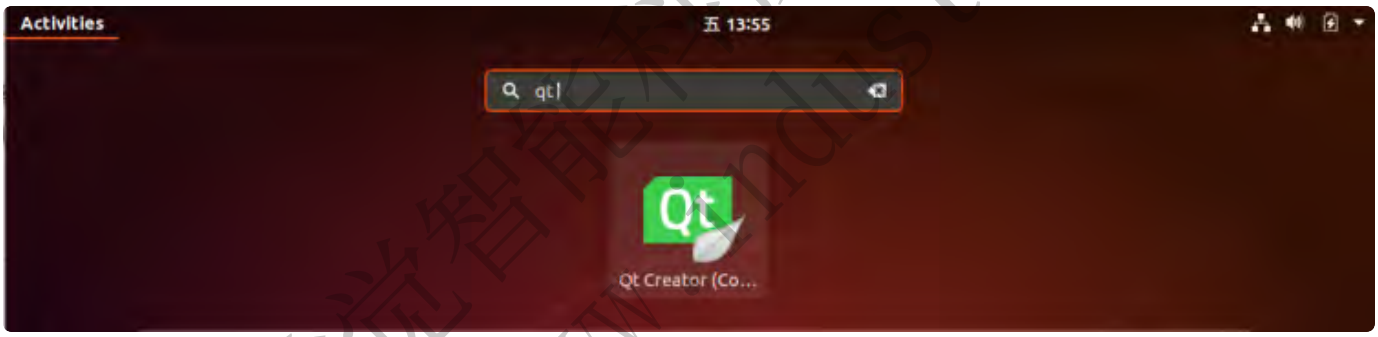
安装进度界面如下图；

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。

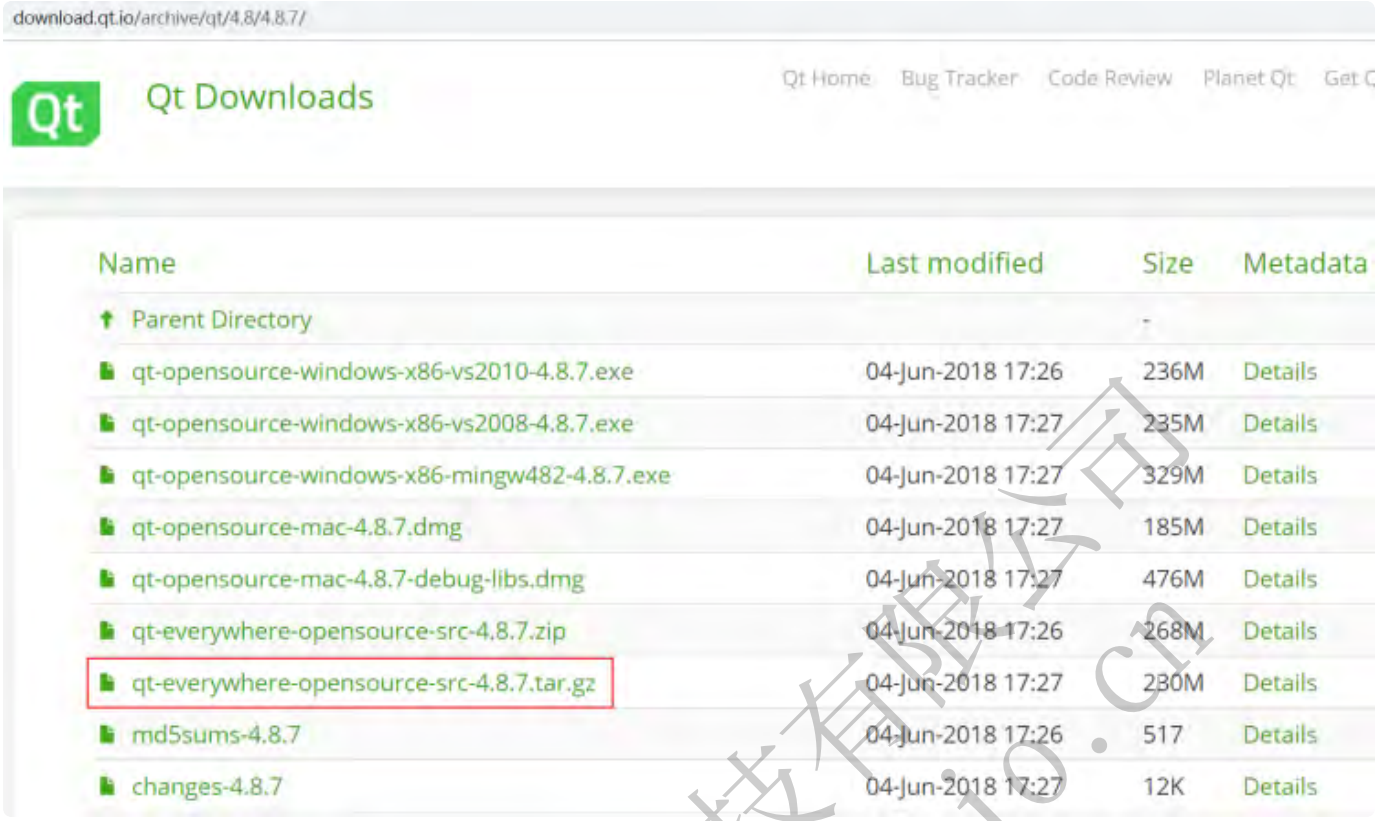




安装Qt4.8.7

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/



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。

Plain Text | 复制代码

```
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
```

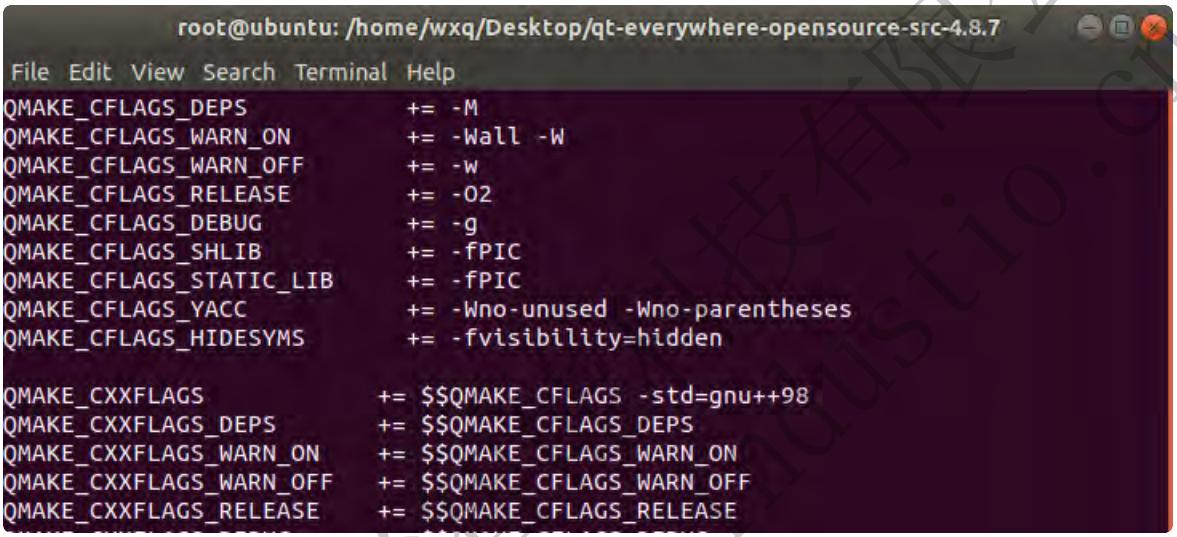
```
fu@fu-VirtualBox:/home/industio_work/Qt/qt-everywhere-opensource-src-4.8.7$ ls
bin              configure.exe  include            LICENSE.GPL3     projects.pro  tools
changes-4.8.7    demos          INSTALL            LICENSE.LGPL     qmake         translations
config.profiles  doc            LGPL_EXCEPTION.txt 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

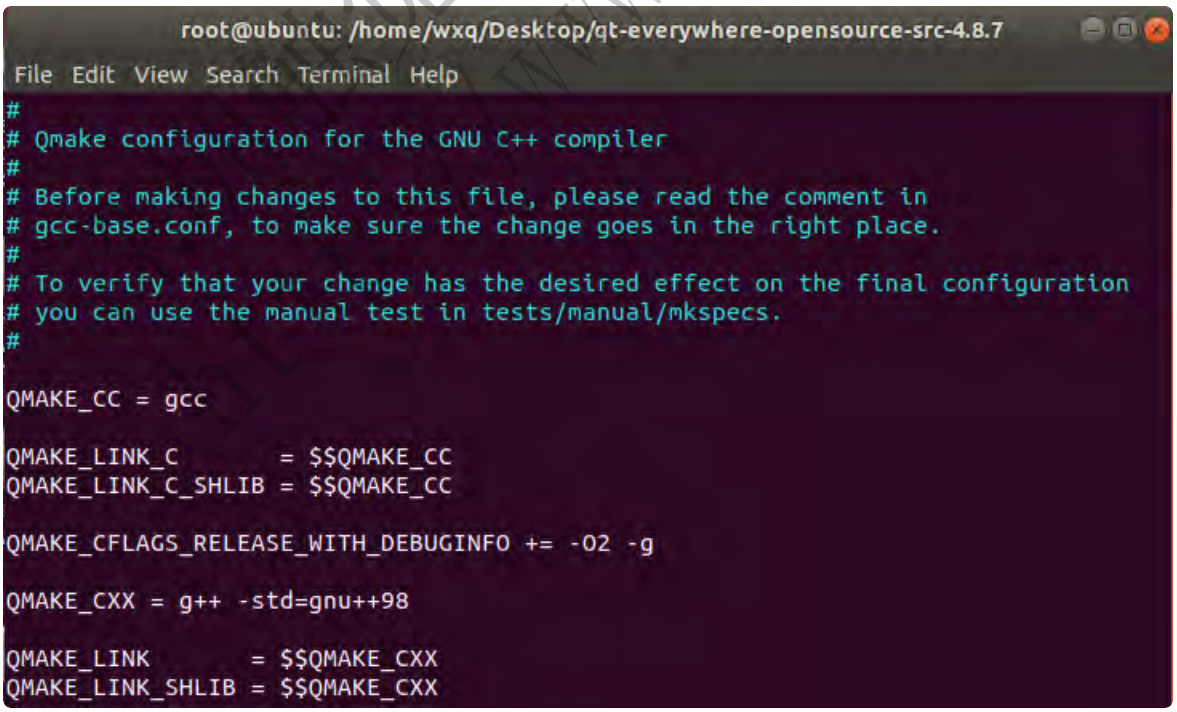
Plain Text | 复制代码

```
industio@industio$:sudo apt-get install gcc g++ g++-multilib make automake
industio@industio$:sudo apt-get install zlib1g-dev lib32ncurses5 lib32z1 libpng-dev autoconf libtool
industio@industio$:sudo apt-get install libxext-dev libx11-dev libxext-dev libxtst-dev
industio@industio$:sudo apt install libgstreamer1.0-dev libgstreamer1-perl libgstreamer-plugins-base0.10-dev libgstreamerd-3-dev
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



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

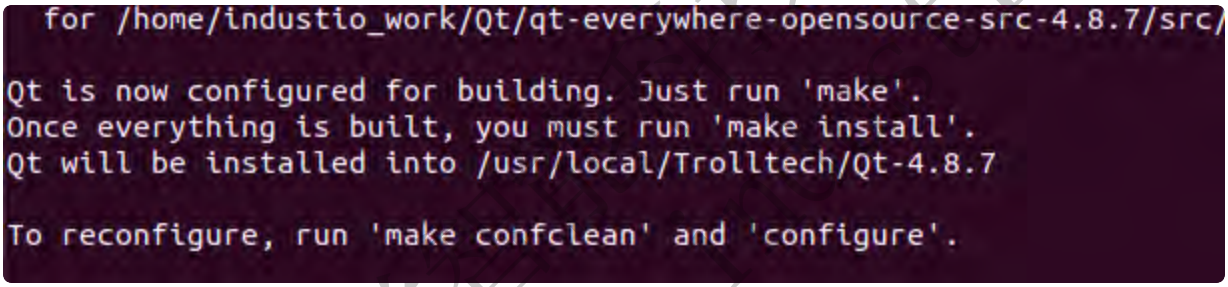


源码编译配置

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

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。



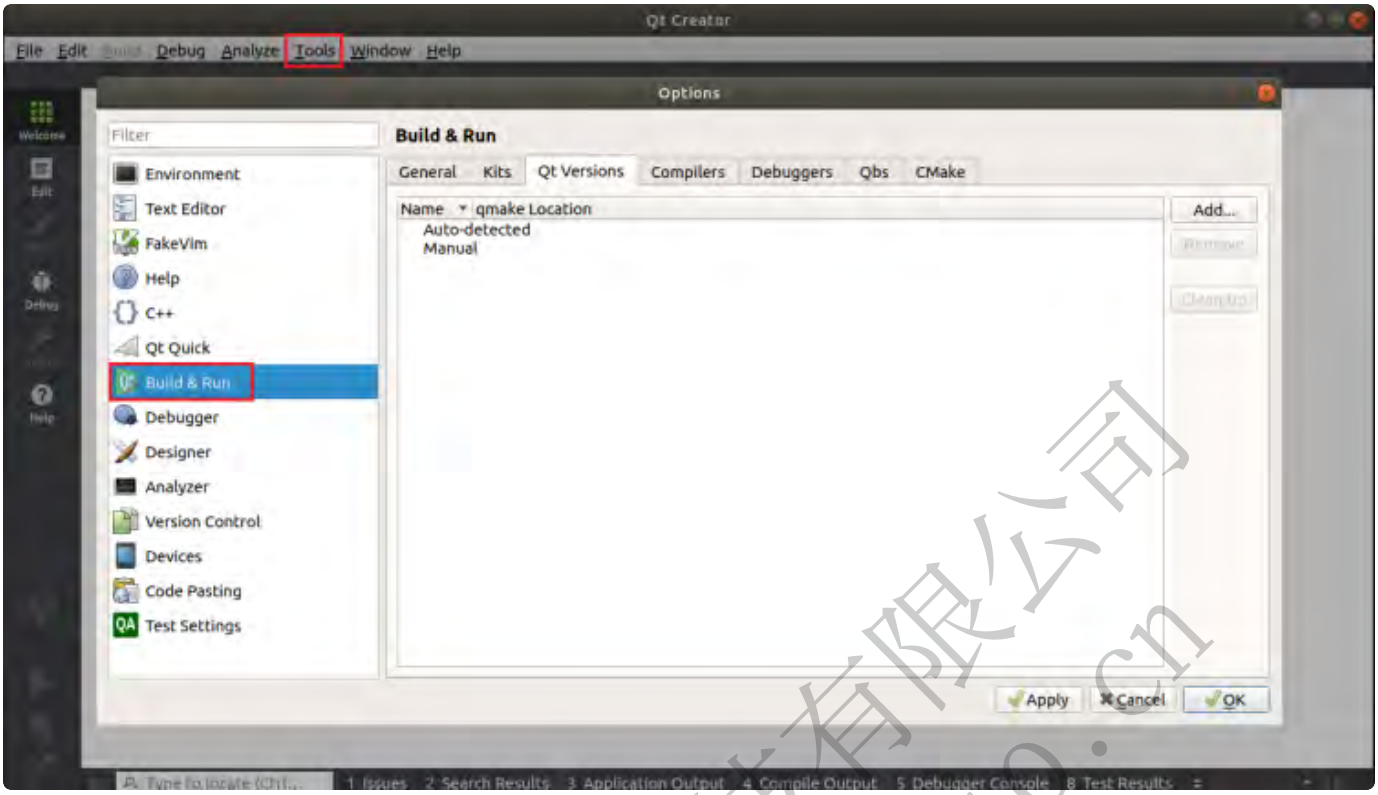
编译安装

```
industio@industio$:make && make install
```

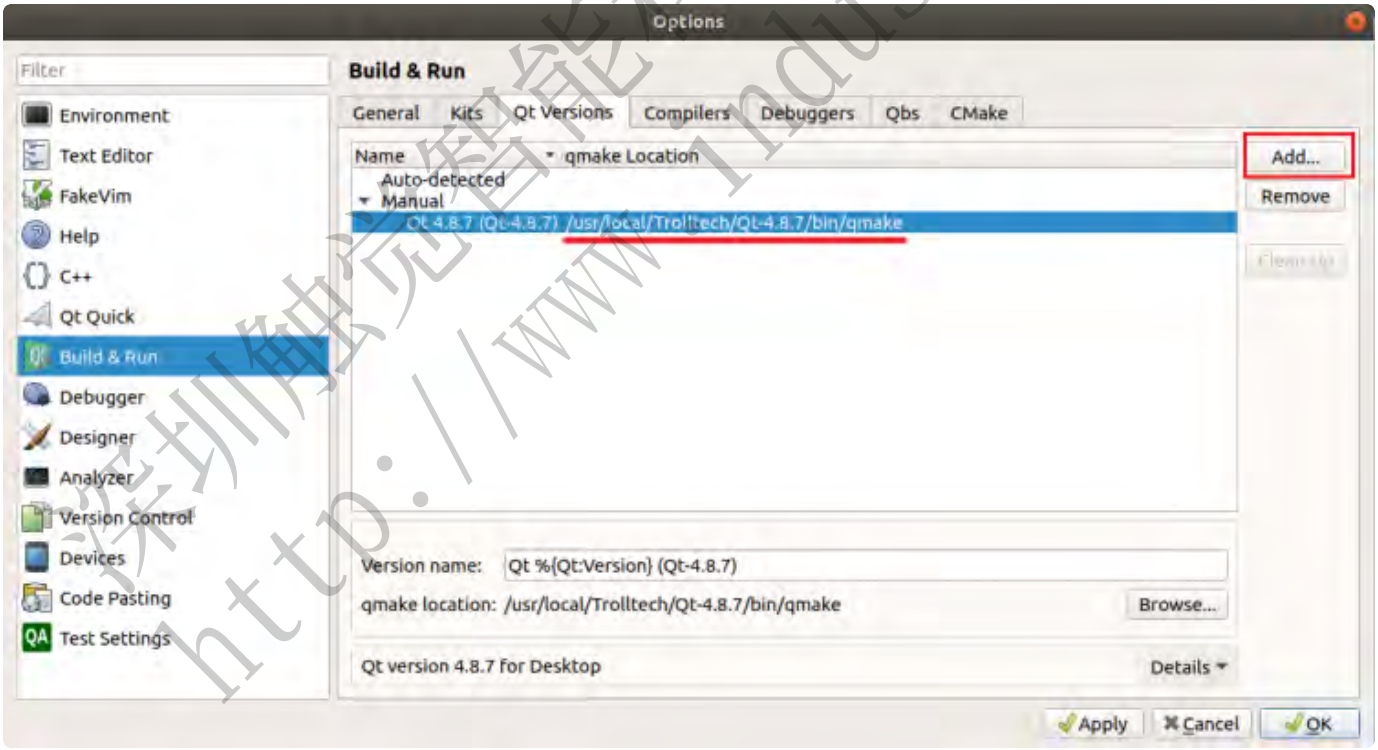
安装完成后，安装的文件位于”/usr/local/Trolltech/Qt-4.8.7“目录，进入sdk的bin目录，执行./qmake -v即可查看安装的qmake和Qt的版本。

Qt Creator添加Qt4.8.7开发套件

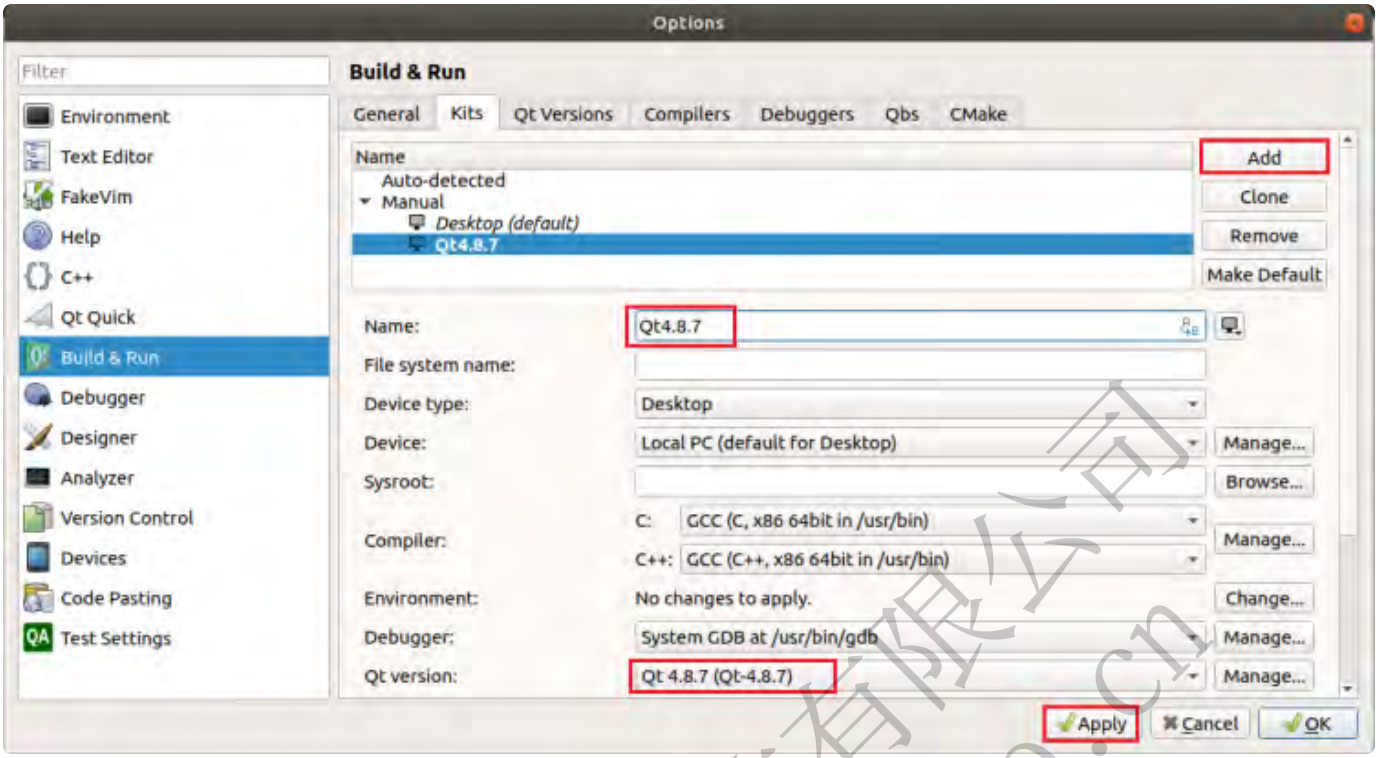
Qt Creator 4.4.1需要添加Qt4.8.7开发套件才可能在Ubuntu18.04 PC端编译、运行和调试Qt代码。打开Qt Creator 4.4.1软件，选择”Tool“->”Options”，在弹出的界面中选择”Build & Run”。



在”Qt Versions”栏点击右边的”Add..”按键，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“按键生效添加的Qt Version。



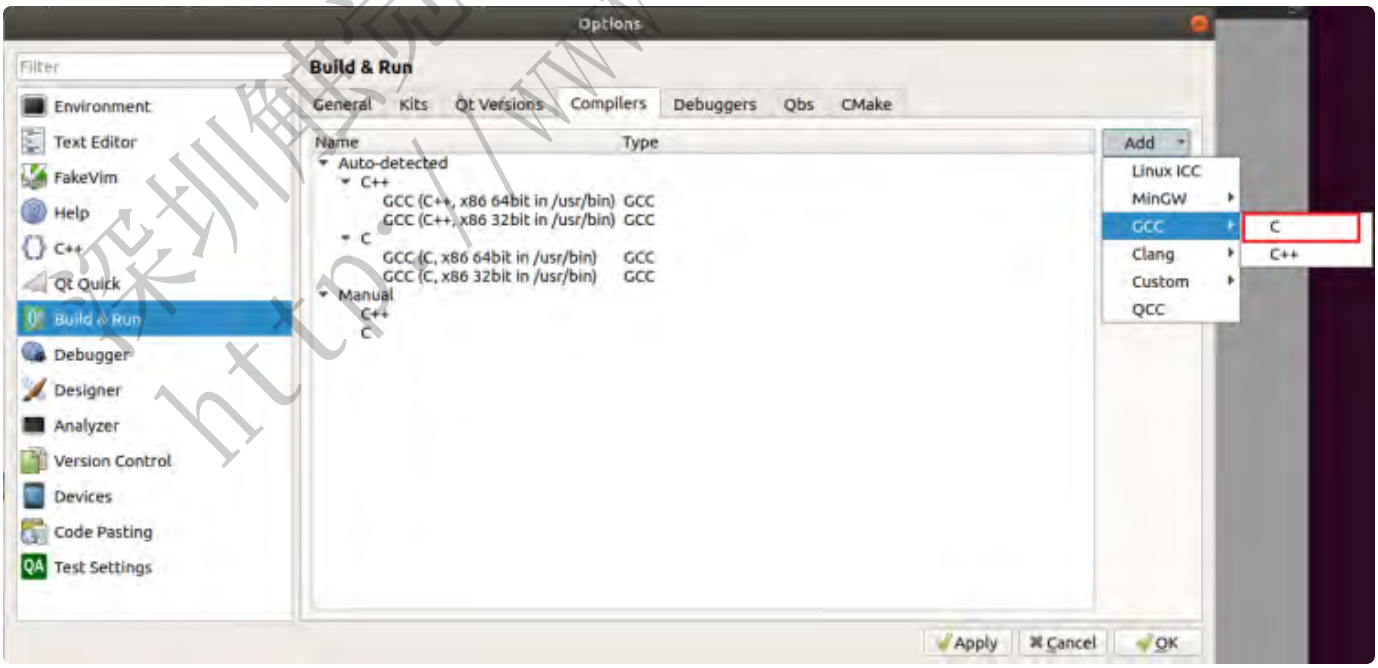
点击进入Kits栏，点击”add“按键新增一项kit，修改名字为”Qt4.8.7“，Qt Version选择步骤2中添加的Qt-4.8.7，点击”Apply“按键生效，点击”OK”按键关闭窗口。



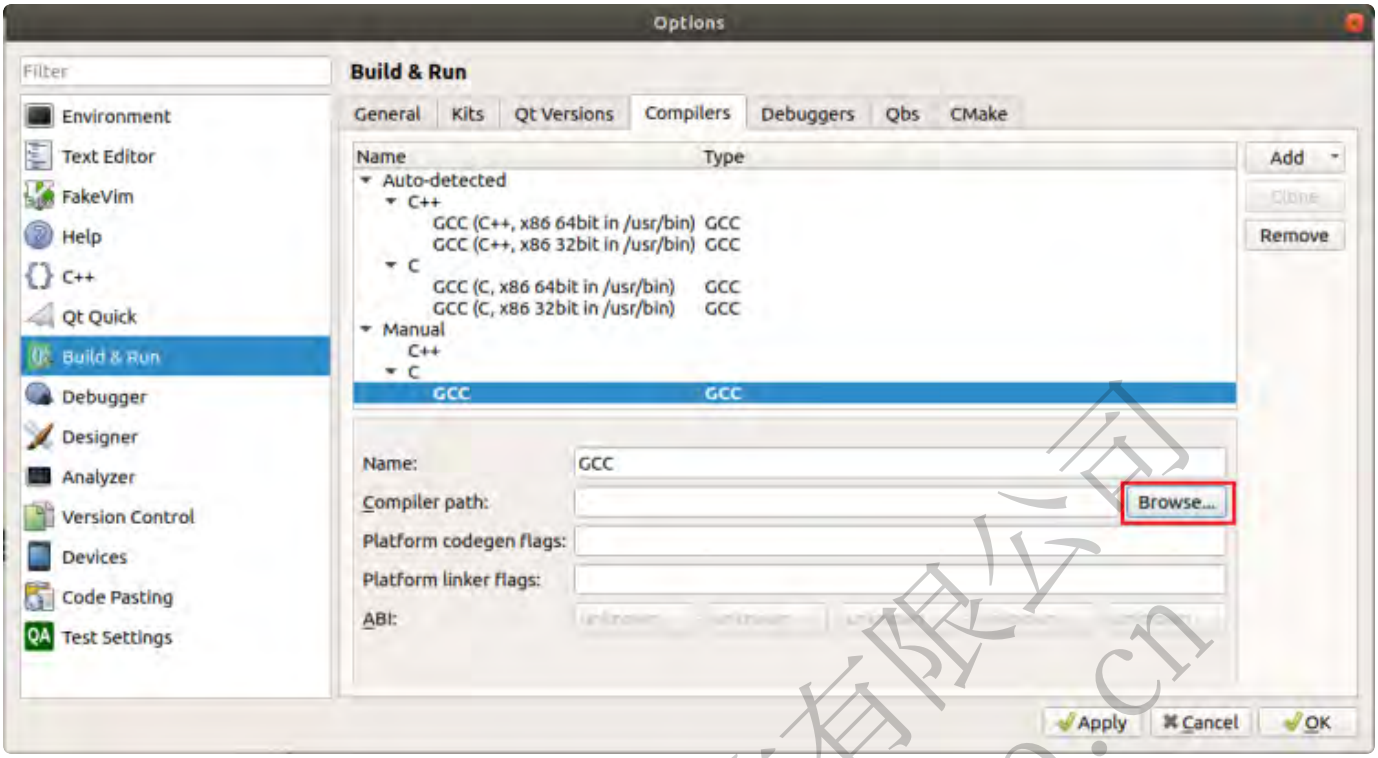
Qt Creator添加Qt4.8.7交叉编译

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。

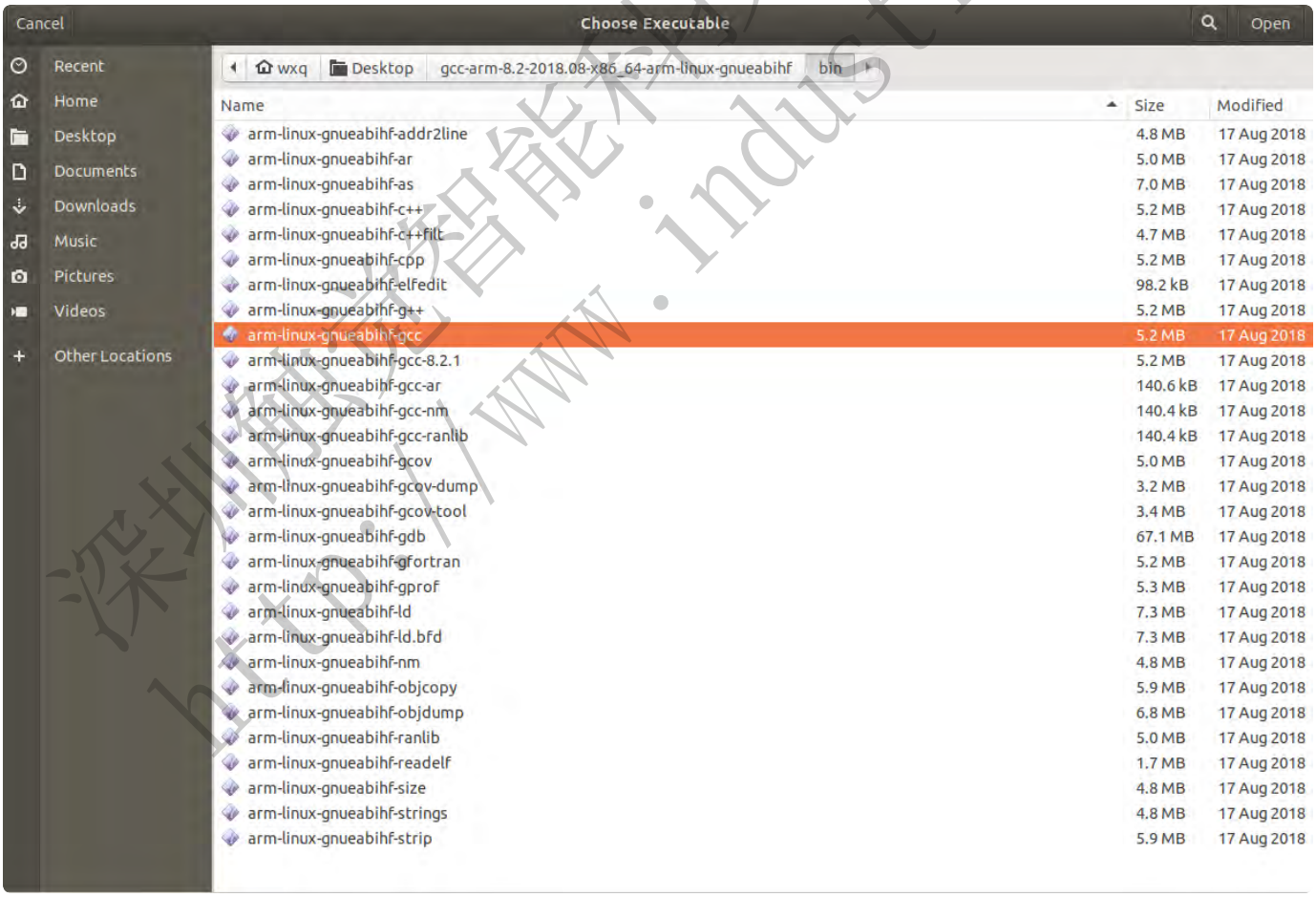
添加gcc在界面的右侧点击Add–>GCC–>C，添加一个GCC编译器。



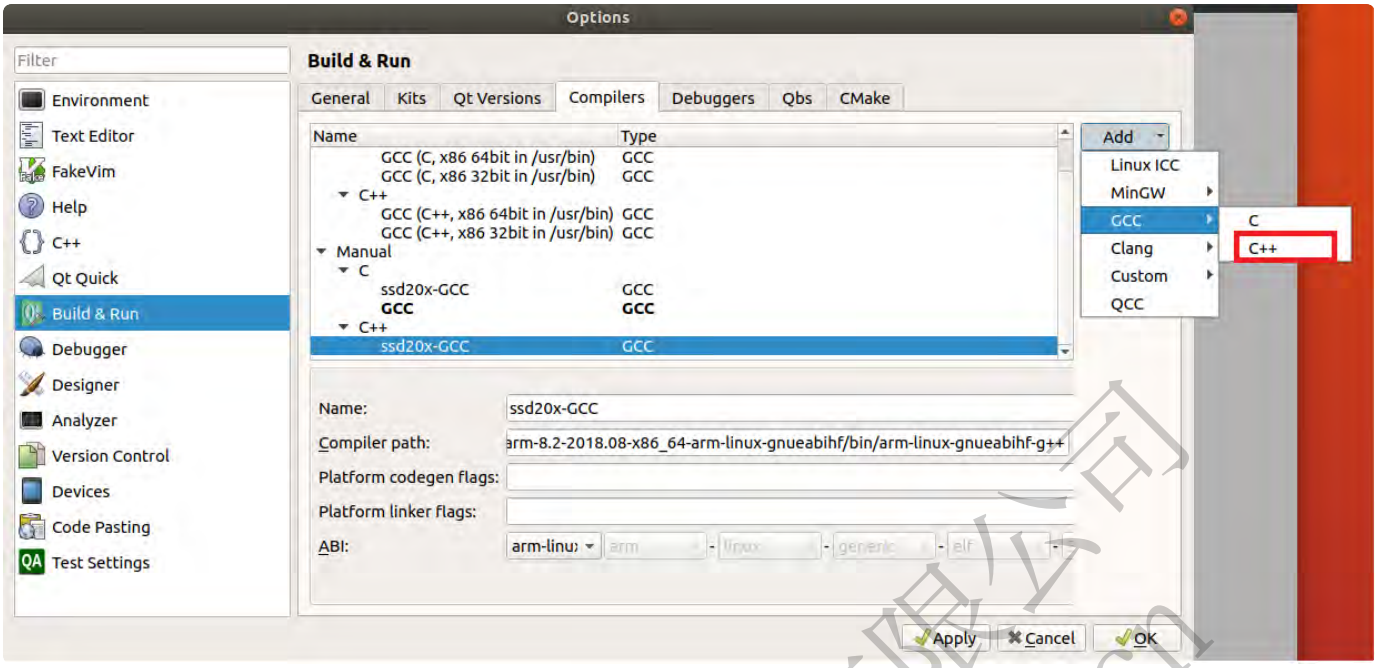
选中新添加的GCC编译器，然后点击下方的“Browse”按键开启文件资源浏览器；



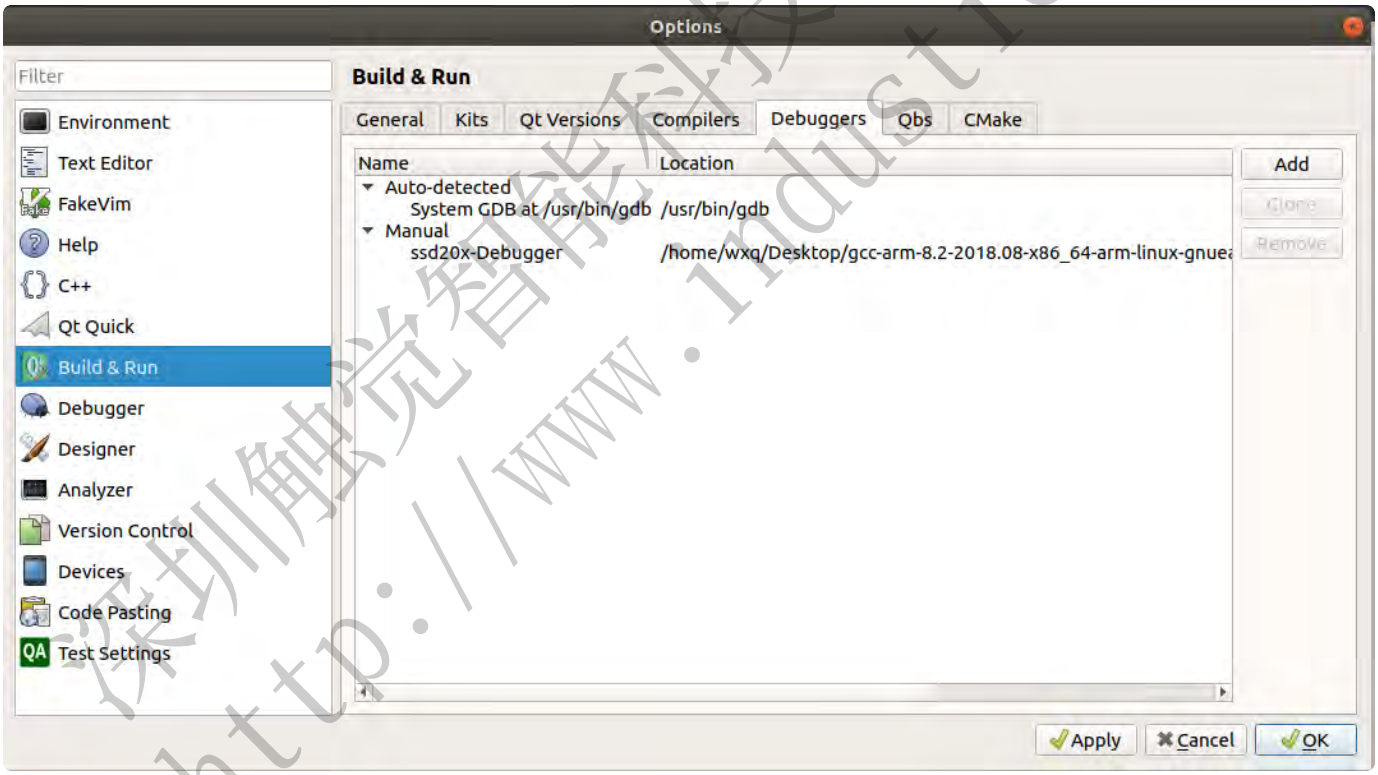
选中gcc编译工具“arm-linux-gnueabihf-gcc”，添加完成点击“Apply”生效设置



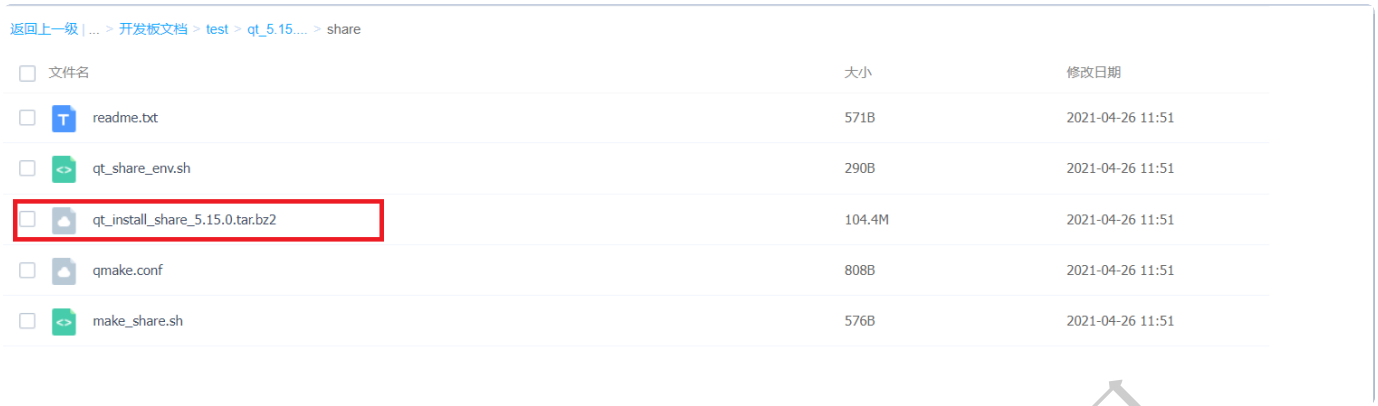
按照GCC的添加方法，添加g++



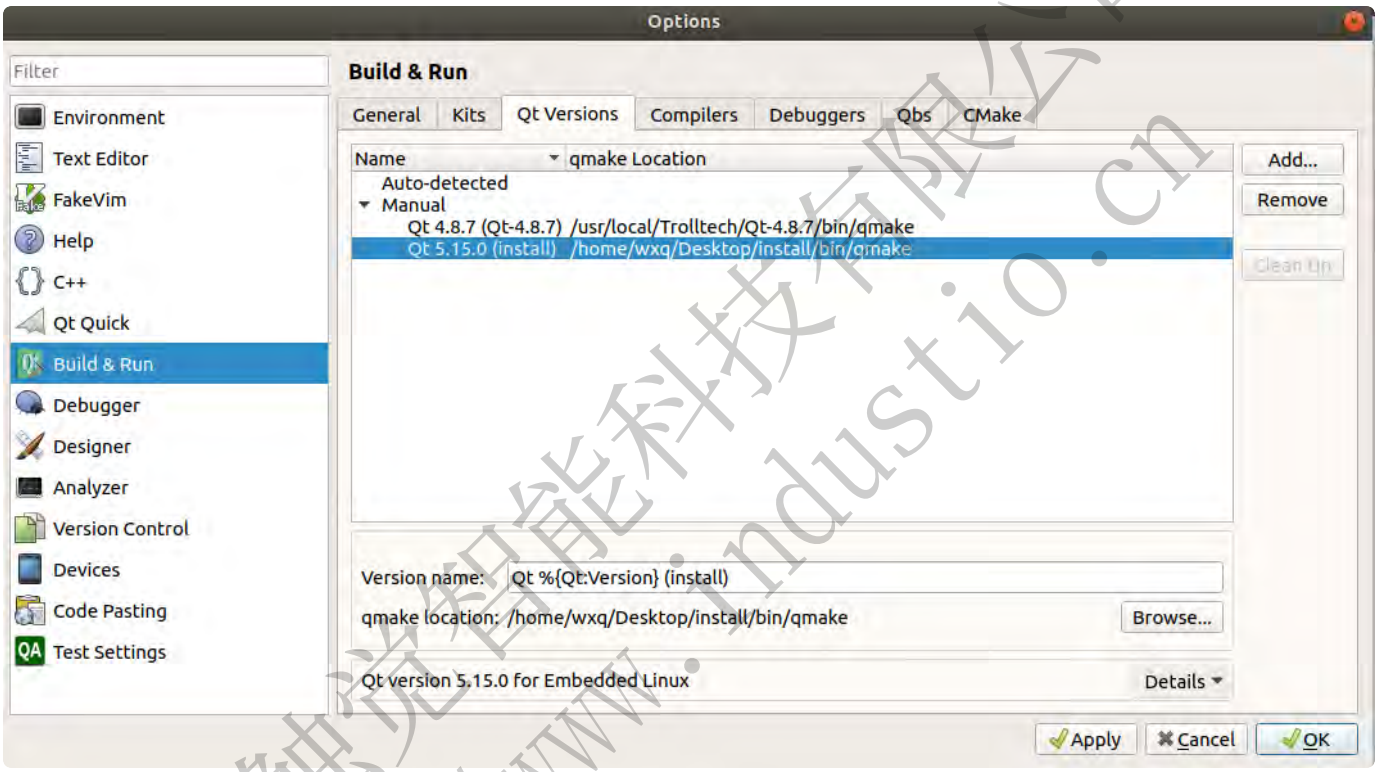
在Debuggers栏点击“Add”按键新增一个Debugger，选择“arm-linux-gnueabihf-gdb”并修改名字为“ssd20x-Debugger”，点击“Apply”按键生效设置。



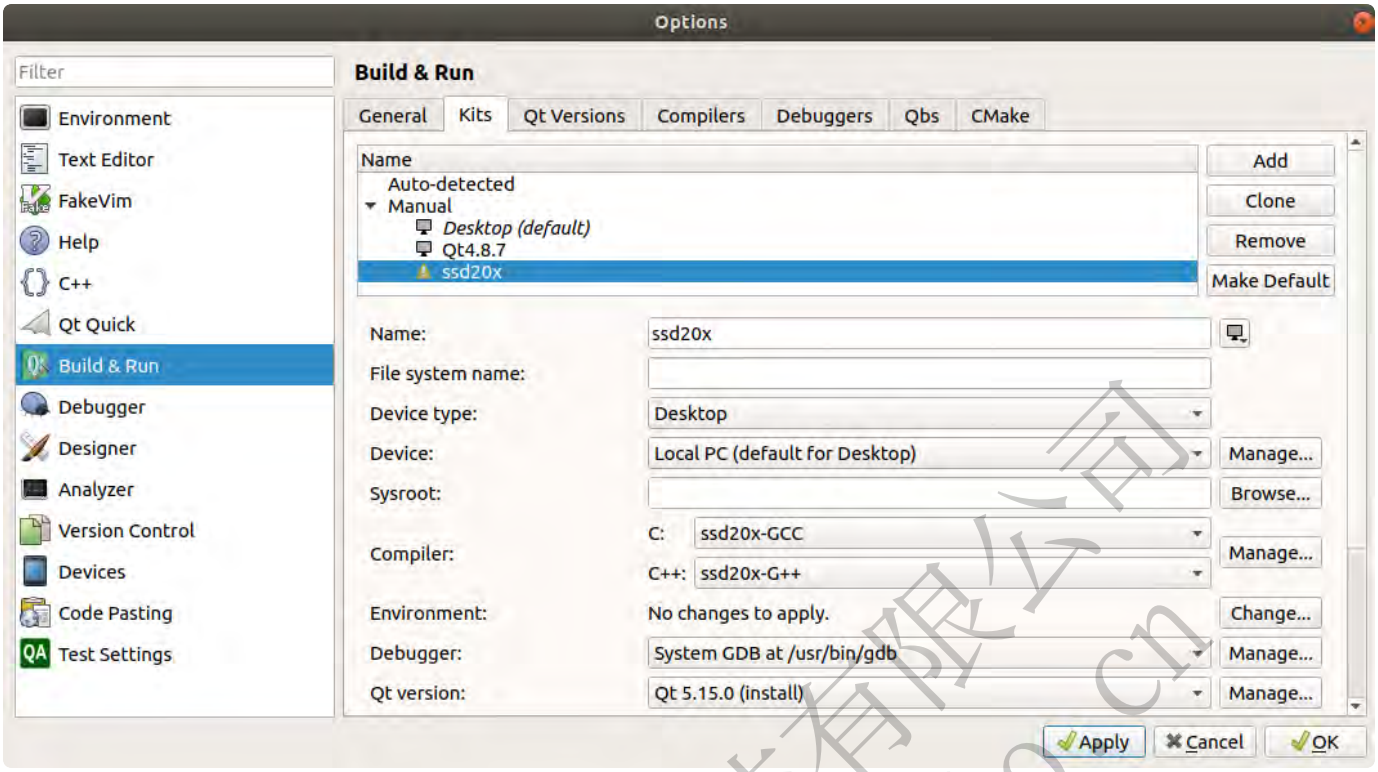
在配置Qt Versions时把网盘中的Qt5.15解压到Ubuntu中



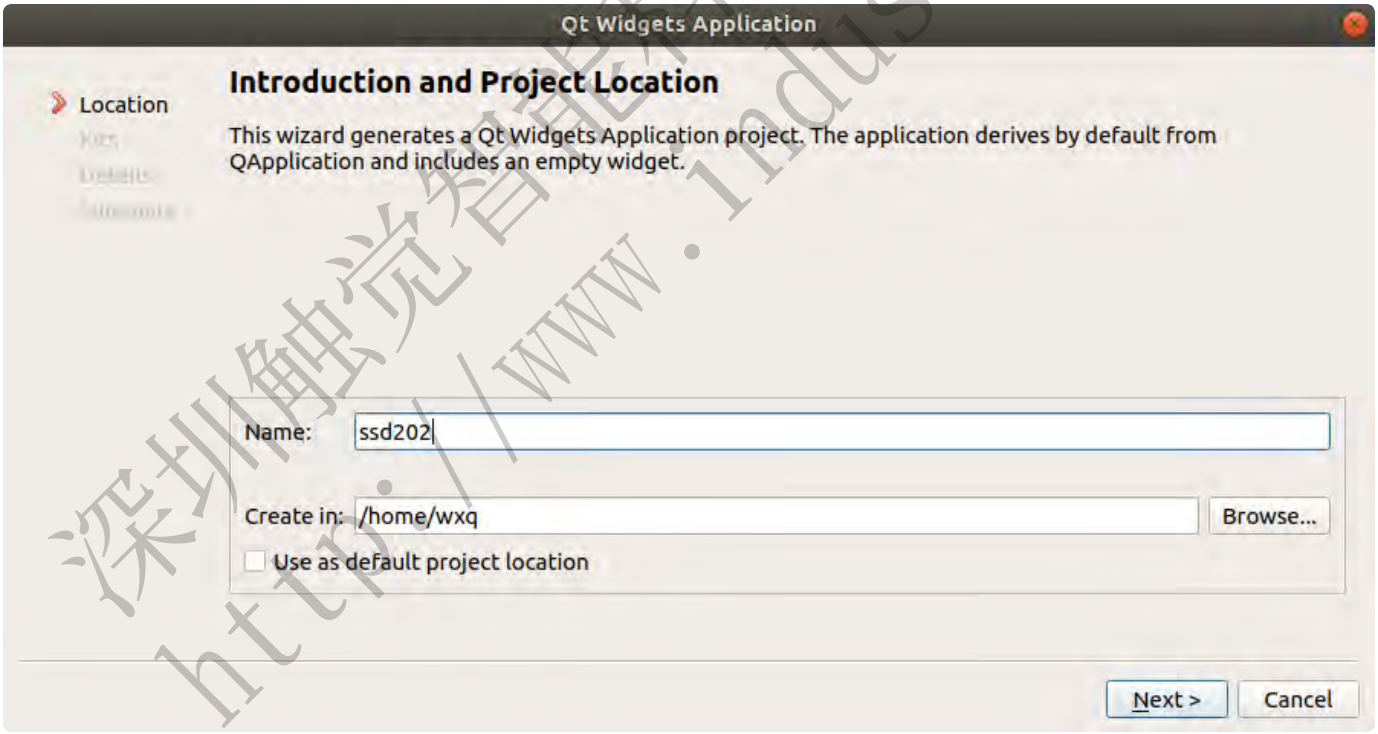
在Qt Versions栏点击“add”打开文件资源浏览器，选择“/install/bin/qmake”，点击“Apply”按键生效设置。



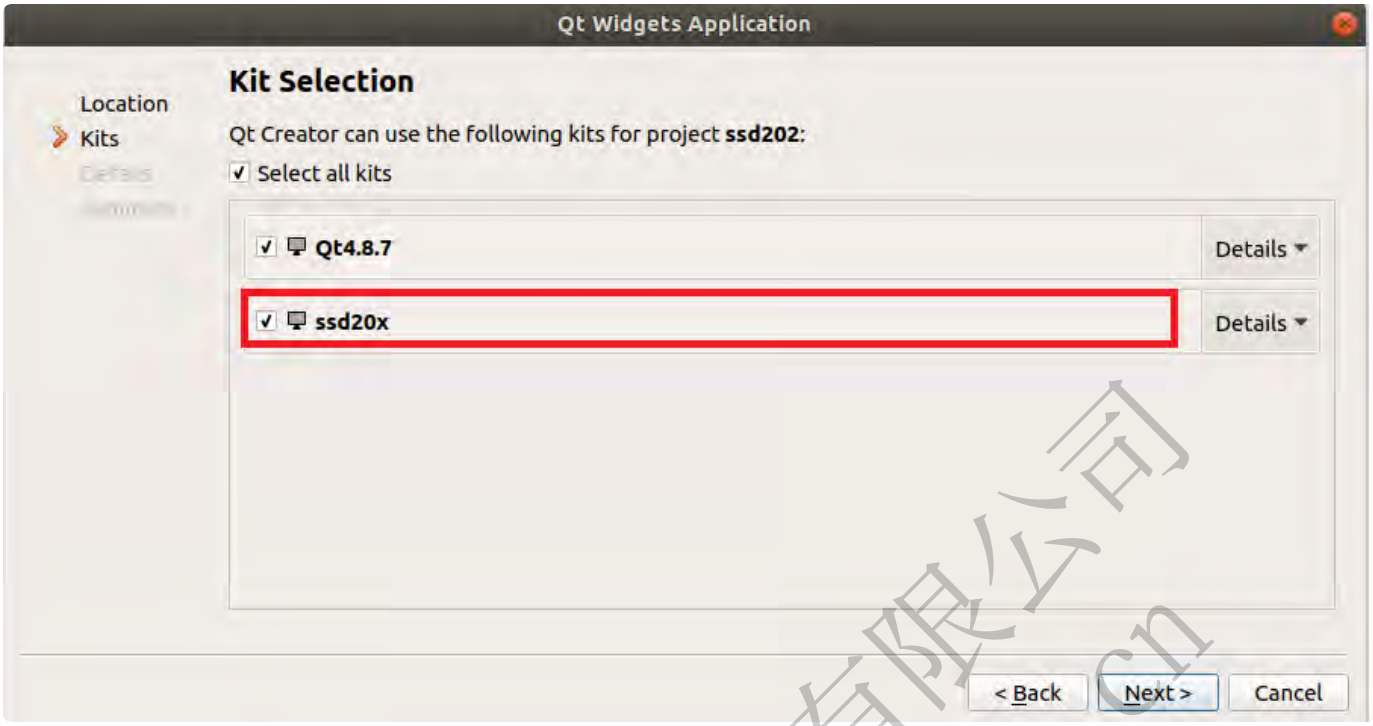
在Kits栏，点击“Add”按键添加一个kit，并修改名字为“ssd20x”，选择Compiler C: ssd20x-GCC；Compiler C++: ssd20x-G++；Qt Version :“Qt5.15.0（install）”，最后点击“Apply”生效配置。



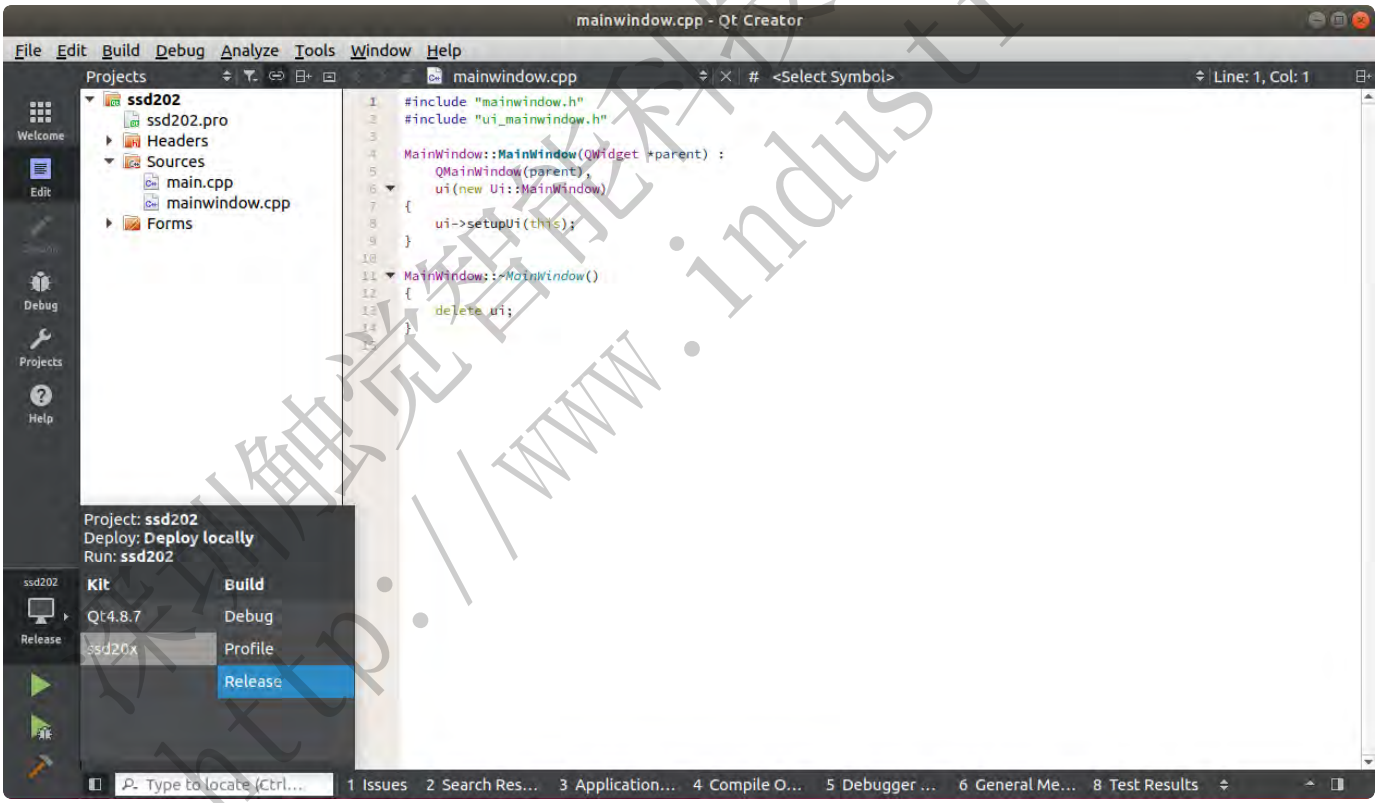
验证Qt Creator交叉编译配置，点击Qt Creator软件的“File–>New File or Project”，新建一个“Qt widget Application”工程。



在“Kit Selection”界面，选择添加的“ssd202x”



在Qt Creator左侧选择“ssd20x->Release”

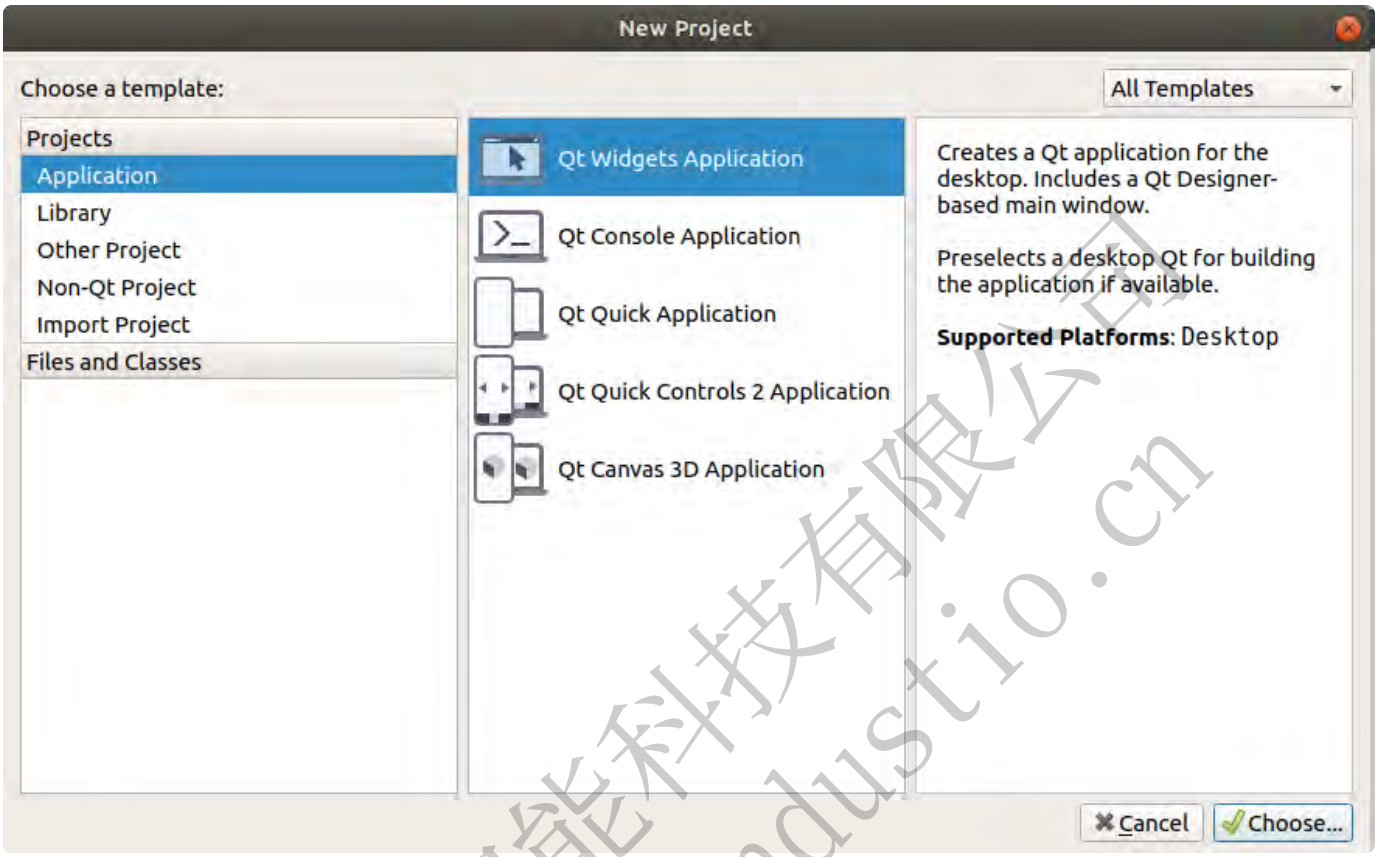


点击”Build”🔨按键编译；正常编译完，在新建工程的“/home/wxq”目录下，产生一个“build-ssd202-ssd20x-Release”文件夹，文件夹中包含编译的中间文件和开发板上运行的可执行文件“ssd202”。

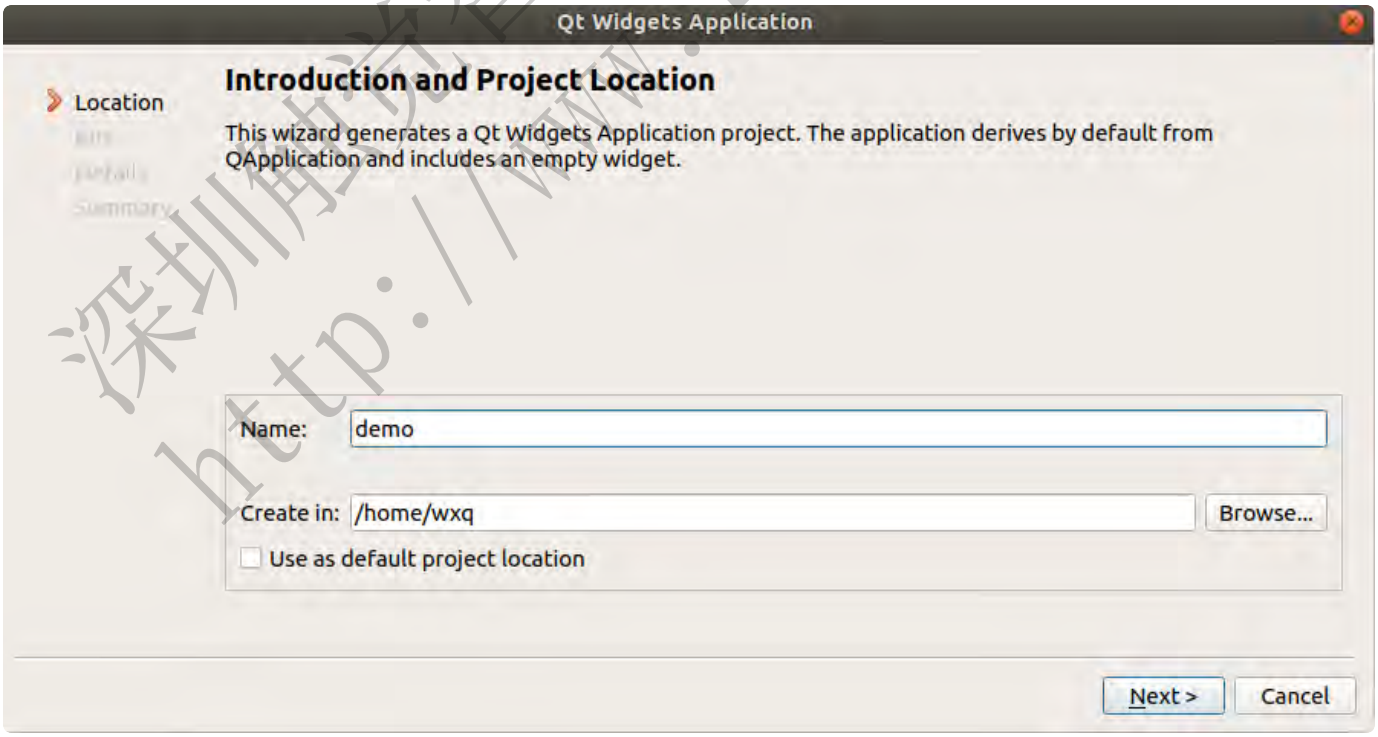
```
wxq@ubuntu:~/build-ssd202-ssd20x-Release$ ls
main.o          Makefile             moc_mainwindow.o  ssd202
mainwindow.o    moc_mainwindow.cpp   moc_predefs.h     ui_mainwindow.h
wxq@ubuntu:~/build-ssd202-ssd20x-Release$
```

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”

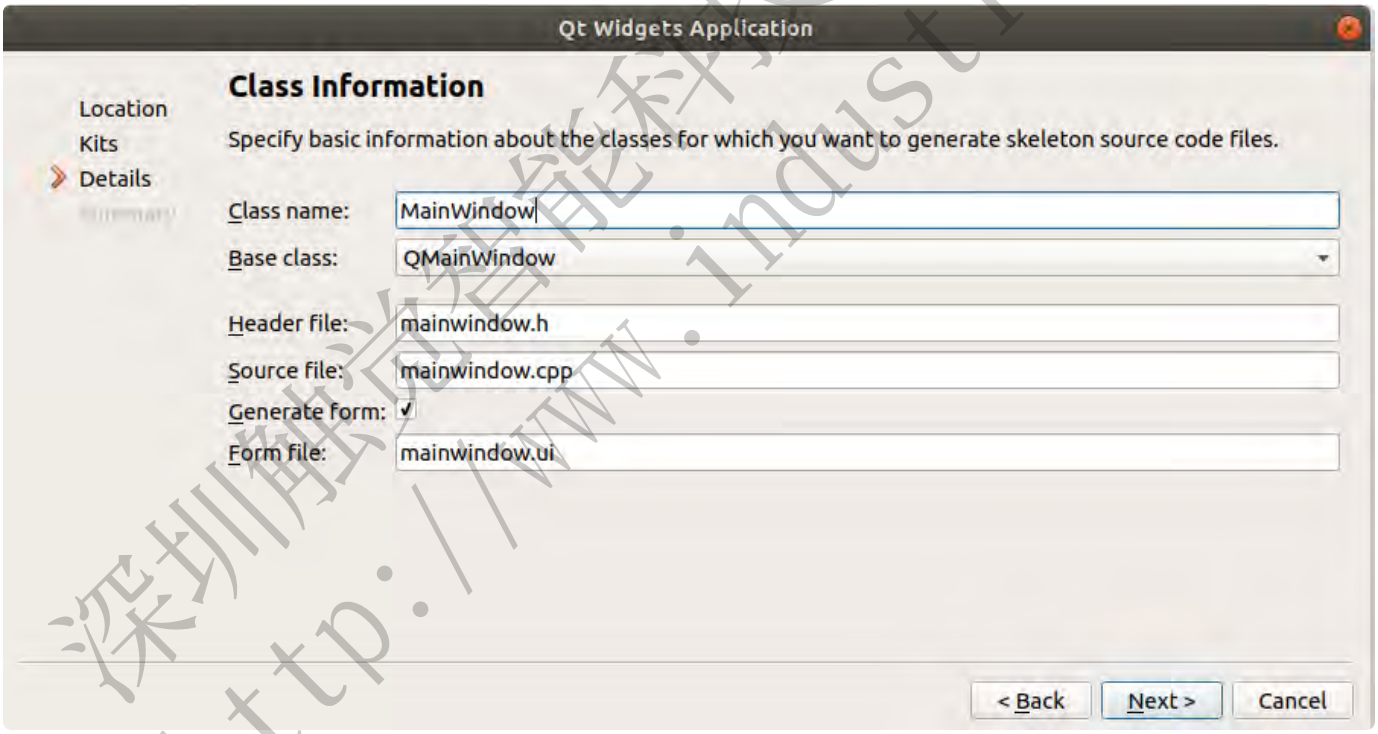


名字可以随便一个

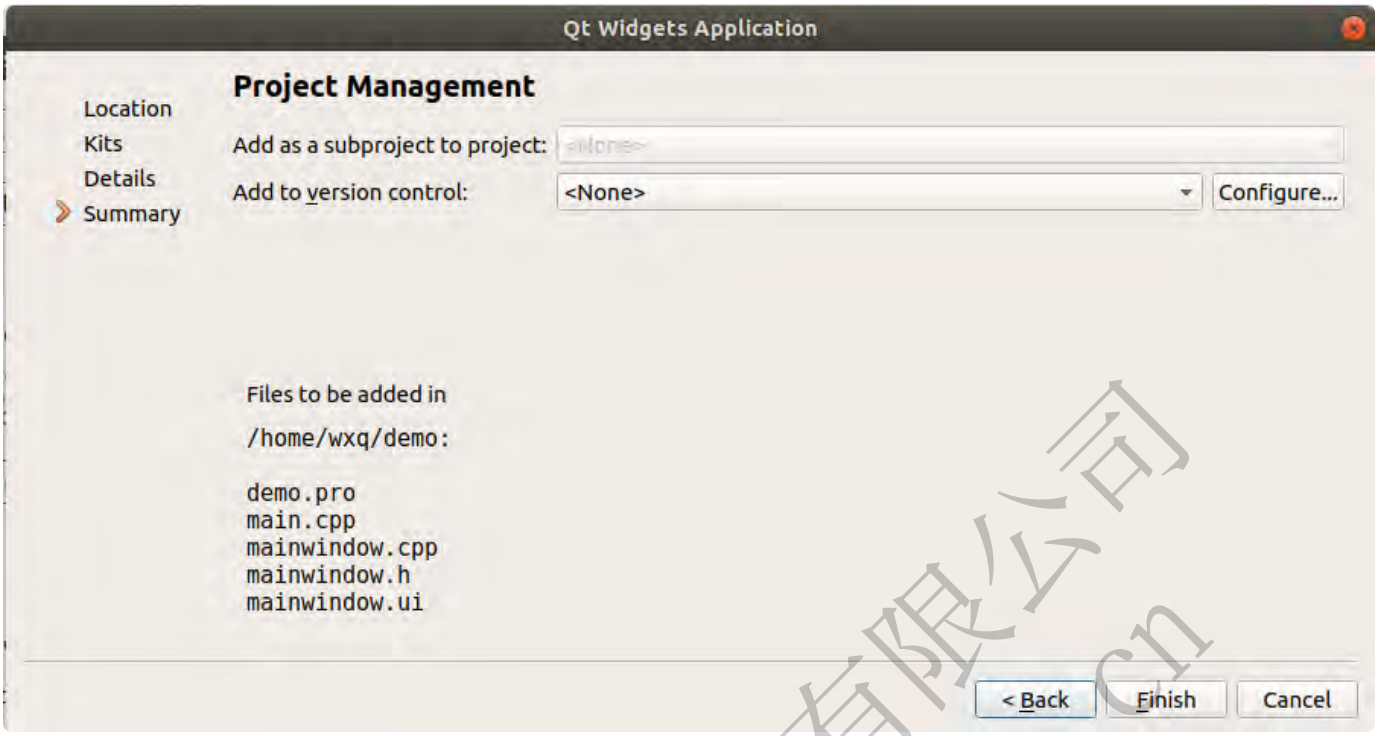


添加ssd20x

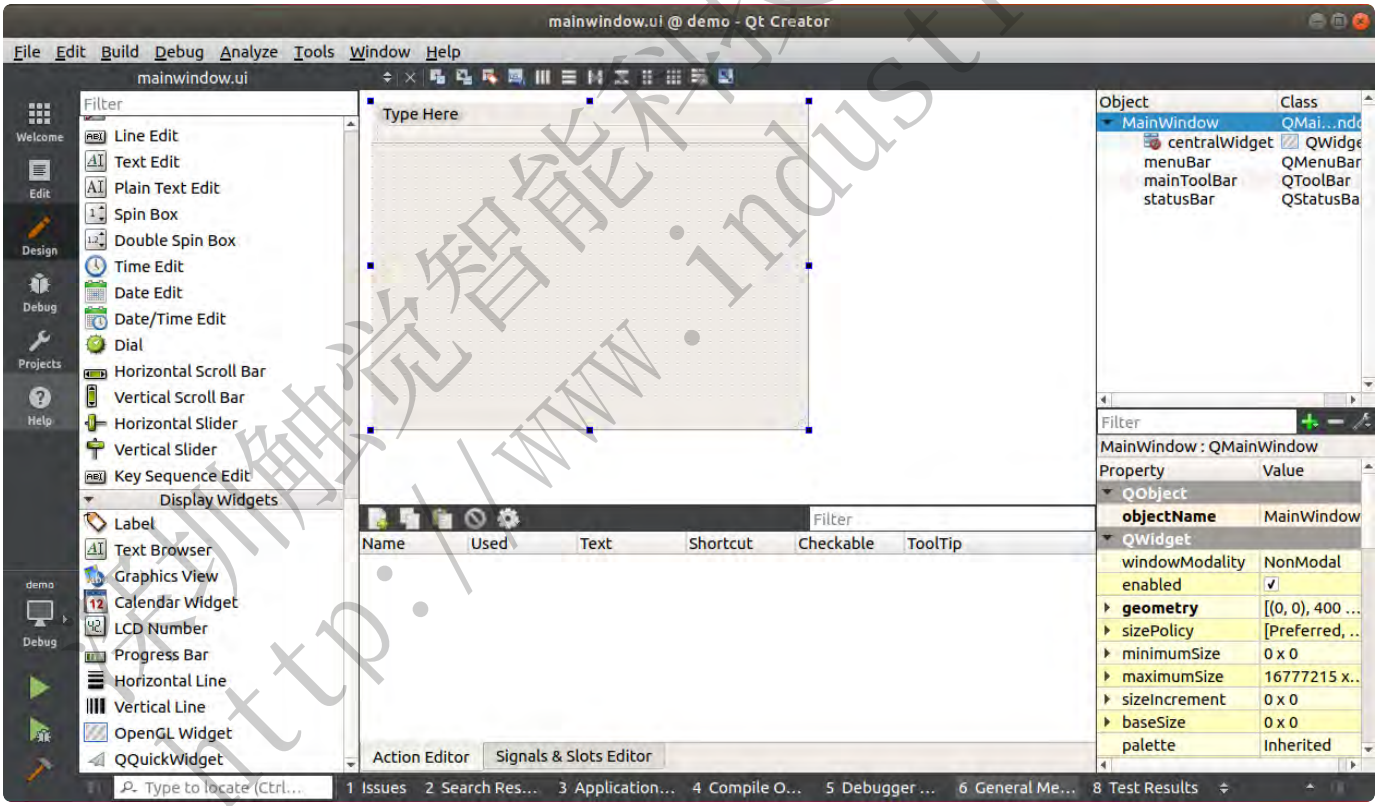
默认即可



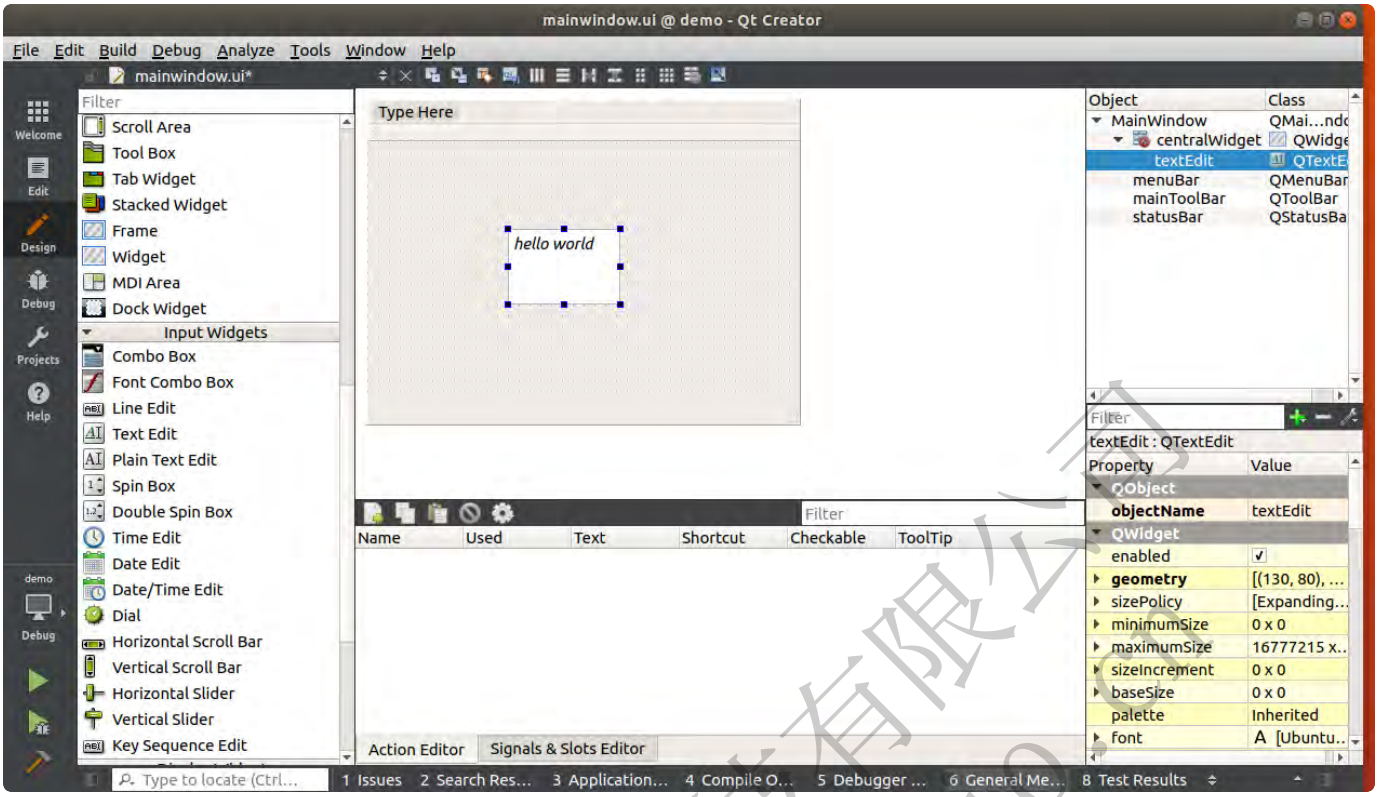
点击“Finish”



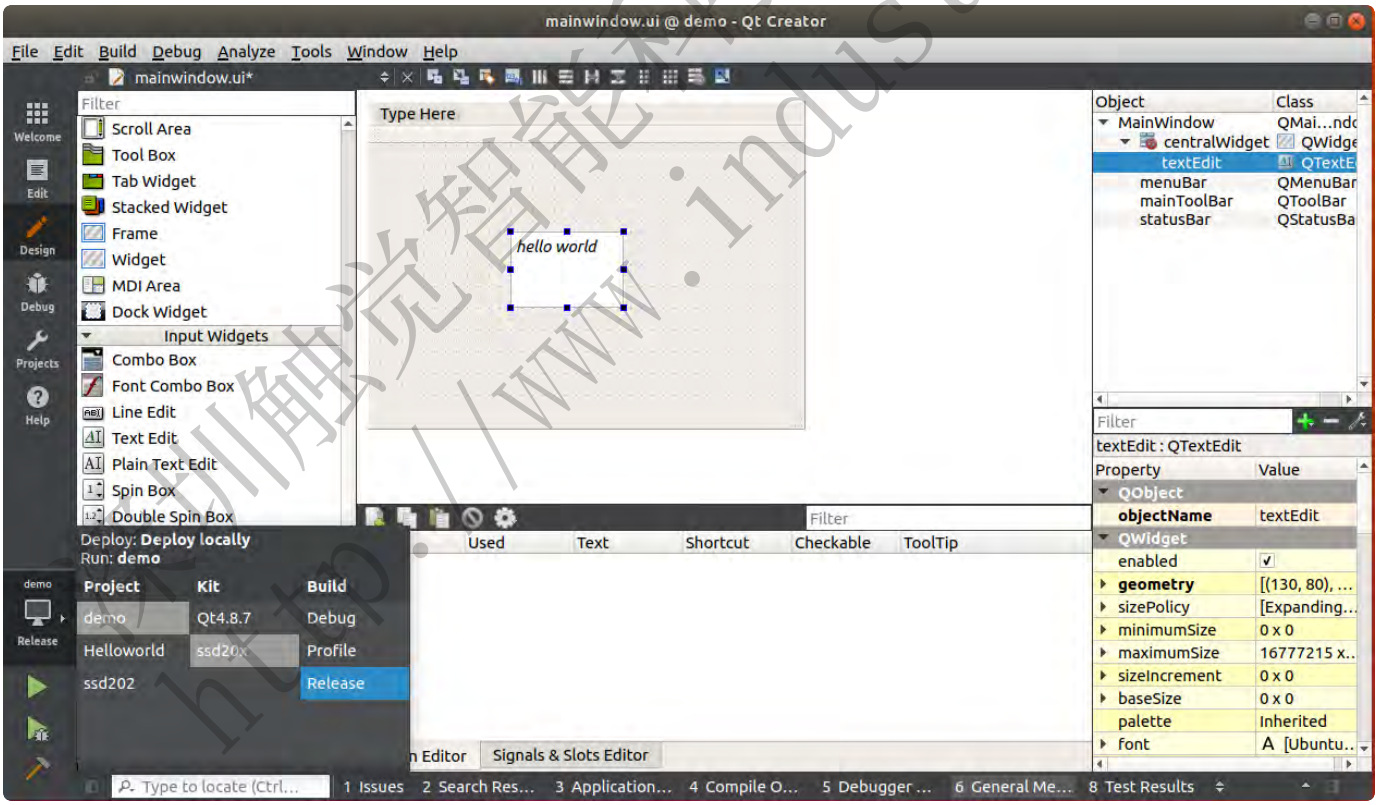
点击“Forms–>mainwindow.ui”



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



debug选择“ssd20x–>Release”



点击”Build”🔨按键编译；生产可执行文件，把文件拷贝到开发中运行