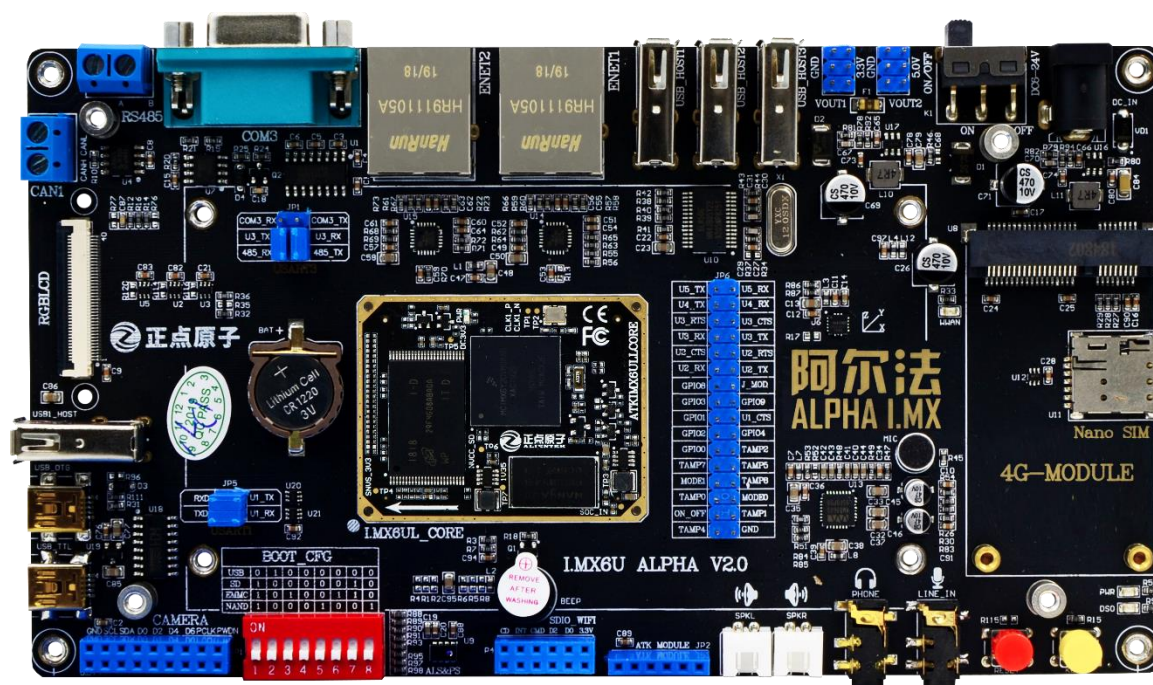


I.MX6U Qt 交叉编译环境搭建 V1.0



正点原子广州市星翼电子科技有限公司

淘宝店铺 1: <http://eboard.taobao.com>

淘宝店铺 2: <http://openedv.taobao.com>

技术支持论坛 (开源电子网) : www.openedv.com

原子哥在线教学: www.yuanzige.com

官方网站: www.alientek.com

最新资料下载链接: <http://www.openedv.com/posts/list/13912.htm>

E-mail: 389063473@qq.com QQ: [389063473](https://www.qq.com/389063473)

咨询电话: [020-38271790](tel:020-38271790)

传真号码: [020-36773971](tel:020-36773971)

团队: [正点原子团队](#)

正点原子, 做最全面、最优秀的嵌入式开发平台软硬件供应商。

友 情 提 示

如果您想及时免费获取“正点原子”最新资讯, 敬请关注正点原子微信公众平台, 我们将及时给您发布最新消息和重要资料。



关注方法:

- (1) 微信“扫一扫”, 扫描右侧二维码, 添加关注
- (2) 微信→添加朋友→公众号→输入“正点原子”→关注
- (3) 微信→添加朋友→输入“alientek_stm32”→关注



文档更新说明






版本	版本更新说明	负责人	校审	发布日期
V1.0	初稿:	正点原子 linux 团队		2019.10.26

前言	V
1. 安装交叉编译工具链	6
2. 安装 Qt Creator	6
2.2.1 下载 Qt	6
2.2.2 安装 Qt	7
3. 搭建 IMX6 Qt Creator 交叉编译环境	14
3.1 查看本机的 Kits	14
3.2 配置交叉编译器	15
3.3 配置 qmake	18
3.4 配置 gdb 调试器	20
3.5 配置 Kits	22
3.6 配置环境变量	23
3.7 测试编译 Qt 工程	24
4. 命令行交叉编译 Qt 工程	29

前言

搭建嵌入式 Qt 的开发环境是我们很重要的一步。正点原子团队将与大家一起搭建嵌入式 Qt 的开发环境。

版本说明:

-  Ubuntu 环境: ubuntu-14.04 64bit 或以上
-  应用程序开发工具: Qt Creator3.5.1 64bit
-  Qt sdk 版本: Qt5.5.1 64bit
-  交叉工具链版本为: arm-poky-linux-gnueabi-gcc (GCC) 5.3.0
-  应用平台: i.MX6 平台

1. 安装交叉编译工具链

注意这里重复了 IMX6 用户体验文档的第四章第 2 小节, 为了让用户不在多个文档中来回参考, 所以还是把 IMX6 用户体验文档的第四章第 2 小节内容写在下面, 内容不多。

把 [开发板光盘\5、开发工具\1、交叉编译器\fs1-imx-x11-glibc-x86_64-meta-toolchain-qt5-cortexa7hf-neon-toolchain-4.1.15-2.1.0.sh](#) (yocto 编译出来的 sdk 工具包) 拷贝到 Ubuntu 虚拟机(可以使用 Winscp、FileZilla、虚拟机共享文件夹或建立 Samba 搭建文件夹共享的方式拷贝到 Ubuntu 虚拟机, 本次认为您已经拷贝了该脚本到 Ubuntu 虚拟机)。

如下图 1.1 本文已经把交叉编译工具拷贝到了 Ubuntu 虚拟机。

```
alientek@ubuntu:~$ ls
fs1-imx-x11-glibc-x86_64-meta-toolchain-qt5-cortexa7hf-neon-toolchain-4.1.15-2.1.0.sh
alientek@ubuntu:~$
```

图 1.1 拷贝交叉编译工具到虚拟机

执行下面的指令修改脚本的权限, 并且可以看到当执行完成下指令后, 脚本在终端所显示的颜色也发生了相应的变化。

```
USER# chmod u+x fs1-imx-x11-glibc-x86_64-meta-toolchain-qt5-cortexa7hf-neon-toolchain-4.1.15-2.1.0.sh
```

```
alientek@ubuntu:~$ chmod u+x fs1-imx-x11-glibc-x86_64-meta-toolchain-qt5-cortexa7hf-neon-toolchain-4.1.15-2.1.0.sh
alientek@ubuntu:~$ ls
fs1-imx-x11-glibc-x86_64-meta-toolchain-qt5-cortexa7hf-neon-toolchain-4.1.15-2.1.0.sh http Pictures QtCourse QW
alientek@ubuntu:~$
```

图 1.2 赋予可执行权限

直接执行脚本安装交叉编译工具, 连续敲下两次回车键确认, 再输入用户密码即可。本次安装的目录为脚本所指定的默认安装的目录, 后面的内核编译环境的交叉编译都是按这个安装目录去操作, 所以建议您也是默认安装到/opt/fs1-imx-x11/4.1.15-2.1.0 这个默认目录。

```
USER# ./fs1-imx-x11-glibc-x86_64-meta-toolchain-qt5-cortexa7hf-neon-toolchain-4.1.15-2.1.0.sh
```

```
alientek@ubuntu:~$ ./fs1-imx-x11-glibc-x86_64-meta-toolchain-qt5-cortexa7hf-neon-toolchain-4.1.15-2.1.0.sh
Freescall i.MX Release Distro SDK installer version 4.1.15-2.1.0
=====
Enter target directory for SDK (default: /opt/fs1-imx-x11/4.1.15-2.1.0):
You are about to install the SDK to "/opt/fs1-imx-x11/4.1.15-2.1.0". Proceed[Y/n]?
[sudo] password for alientek:
Extracting SDK.....done
Setting it up...done
SDK has been successfully set up and is ready to be used.
Each time you wish to use the SDK in a new shell session, you need to source the environment setup script e.g.
$ . /opt/fs1-imx-x11/4.1.15-2.1.0/environment-setup-cortexa7hf-neon-poky-linux-gnueabi
alientek@ubuntu:~$
```

图 1.3 安装交叉编译工具到指定目录

2. 安装 Qt Creator

2.2.1 下载 Qt

安装 Qt Creator 的教程步骤重复了我们 Qt 教程文档里面的第二章第 1.2 小节, 让读者快速搭建环境, 这里简化了一些步骤。

进入 <http://download.qt.io/archive/qt/5.5/5.5.1/>, 这里选择 qt-opensource-linux-x64-5.5.1.run 下载。这里有 32 位与 64 位的版本下载, 作者虚拟机是 64 位的, 所以选择 64 位的 sdk 包下载。如下图 2.2.1 选择下图红色框框的内容进入下载。若正点原子光盘已经提供, 则无需再下载。



Qt Downloads

[Qt Home](#) [Bug Tracker](#) [Code Review](#) [Planet Qt](#)

Name	Last modified	Size	Metadata
↑ Parent Directory		-	
submodules/	14-Oct-2015 12:34	-	
single/	14-Oct-2015 12:31	-	
qt-opensource-windows-x86-wintrt-5.5.1.exe	14-Oct-2015 09:58	914M	Details
qt-opensource-windows-x86-msvc2013_64-5.5.1.exe	14-Oct-2015 09:57	823M	Details
qt-opensource-windows-x86-msvc2013-5.5.1.exe	14-Oct-2015 09:56	804M	Details
qt-opensource-windows-x86-msvc2012-5.5.1.exe	14-Oct-2015 09:55	747M	Details
qt-opensource-windows-x86-msvc2010-5.5.1.exe	14-Oct-2015 09:55	725M	Details
qt-opensource-windows-x86-mingw492-5.5.1.exe	14-Oct-2015 09:54	1.0G	Details
qt-opensource-windows-x86-android-5.5.1.exe	14-Oct-2015 09:53	1.1G	Details
qt-opensource-mac-x64-clang-5.5.1.dmg	14-Oct-2015 09:51	679M	Details
qt-opensource-mac-x64-android-ios-5.5.1.dmg	14-Oct-2015 09:50	2.0G	Details
qt-opensource-mac-x64-android-5.5.1.dmg	14-Oct-2015 09:48	747M	Details
qt-opensource-linux-x86-android-5.5.1.run	14-Oct-2015 09:44	711M	Details
qt-opensource-linux-x86-5.5.1.run	14-Oct-2015 09:43	634M	Details
qt-opensource-linux-x64-android-5.5.1.run	14-Oct-2015 09:42	698M	Details
qt-opensource-linux-x64-5.5.1.run	14-Oct-2015 09:40	622M	Details
md5sums.txt	14-Oct-2015 12:29	1.0K	Details

图 2.2.1. 1 Qt 下载页面

备注：可在 Ubuntu 里直接下载 Linux 环境下的 Qt，不用在 windows 下载再拷贝过去也是可以的，但是考虑到读者对 Linux 的下载不是那么熟悉和下载时不是那么直观，所以在这里还是在 windows 下载。这里也贴出直接在 Ubuntu 下载的方法。鼠标右键点击上图，复制下载链接，直接在 Ubuntu 终端下执行下面的指令就可以下载 qt-opensource-linux-x64-5.5.1.run。

USER# `wget http://download.qt.io/archive/qt/5.5/5.5.1/qt-opensource-linux-x64-5.5.1.run`

```

alien@ubuntu:~$ wget http://download.qt.io/archive/qt/5.5/5.5.1/qt-opensource-linux-x64-5.5.1.run
--2019-09-03 16:43:29-- http://download.qt.io/archive/qt/5.5/5.5.1/qt-opensource-linux-x64-5.5.1.run
正在解析主机 download.qt.io (download.qt.io)... 77.86.229.90
正在连接 download.qt.io (download.qt.io)[77.86.229.90]:80... 已连接。
已发出 HTTP 请求，正在等待回应... 302 Found
位置：http://mirrors.ustc.edu.cn/qtproject/archive/qt/5.5/5.5.1/qt-opensource-linux-x64-5.5.1.run [跟随至新的 URL]
--2019-09-03 16:43:30-- http://mirrors.ustc.edu.cn/qtproject/archive/qt/5.5/5.5.1/qt-opensource-linux-x64-5.5.1.run
正在解析主机 mirrors.ustc.edu.cn (mirrors.ustc.edu.cn)... 202.141.176.110, 202.38.95.110, 2001:da8:d800:95::110
正在连接 mirrors.ustc.edu.cn (mirrors.ustc.edu.cn)[202.141.176.110]:80... 已连接。
已发出 HTTP 请求，正在等待回应... 200 OK
长度：652064113 (622M) [application/x-makeself]
正在保存至：“qt-opensource-linux-x64-5.5.1.run”

0% [
] 2,977,792 479KB/s 估时 22m 35s
  
```

图 2.2.1. 2 使用 wget 下载 Qt

2.2.2 安装 Qt

按上面的下载完成后，得到的文件为 qt-opensource-linux-x64-5.5.1.run 将上面的 qt-opensou

qt-linux-x64-5.5.1.run 用 WinSCP 或者 FileZilla(免费开源的 FTP 工具)或者 Samba 等工具,上传到虚拟机上 Ubuntu 14.04 上,如下图,作者已经将 qt-opensource-linux-x64-5.5.1.run 上传至 Ubuntu 虚拟机上,安装过程将会十分简单。

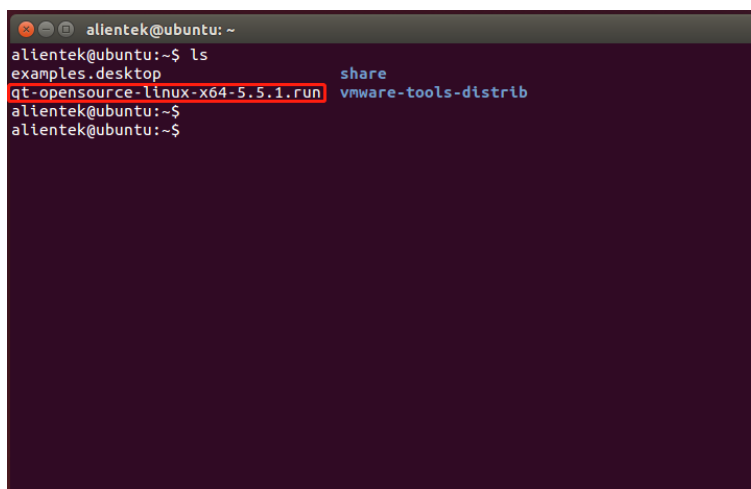


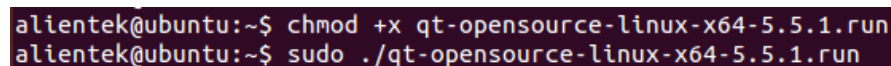
图 2.2.2. 1 拷贝文件到虚拟机

在当前目录直接执行命令,用 `sudo` 表示是当前用户 `alientek` 执行安装 `qt-opensource-linux-x64-5.5.1.run`,然后按提示输入用户密码。注意若不加 `sudo`,会被安装到当前目录下,这里建议读者和本文的环境一样。

先修改该文件的权限,下面为 `qt-opensource-linux-x64-5.5.1.run` 添加可执行权限,再执行安装。

Ubuntu# `chmod +x qt-opensource-linux-x64-5.5.1.run`

Ubuntu# `sudo ./qt-opensource-linux-x64-5.5.1.run`

A terminal window showing the commands 'chmod +x qt-opensource-linux-x64-5.5.1.run' and 'sudo ./qt-opensource-linux-x64-5.5.1.run' being executed. The prompt 'alientek@ubuntu:~\$' is visible at the bottom.

```
alientek@ubuntu:~$ chmod +x qt-opensource-linux-x64-5.5.1.run
alientek@ubuntu:~$ sudo ./qt-opensource-linux-x64-5.5.1.run
```

图 2.2.2. 2 修改权限再安装



图 2.2.2. 3 Qt 安装欢迎页面

直接点 Skip(跳过), 跳过这个输入账号与密码的窗口。到了设置 Qt 5.5.1 窗口, 直接点击下一步。

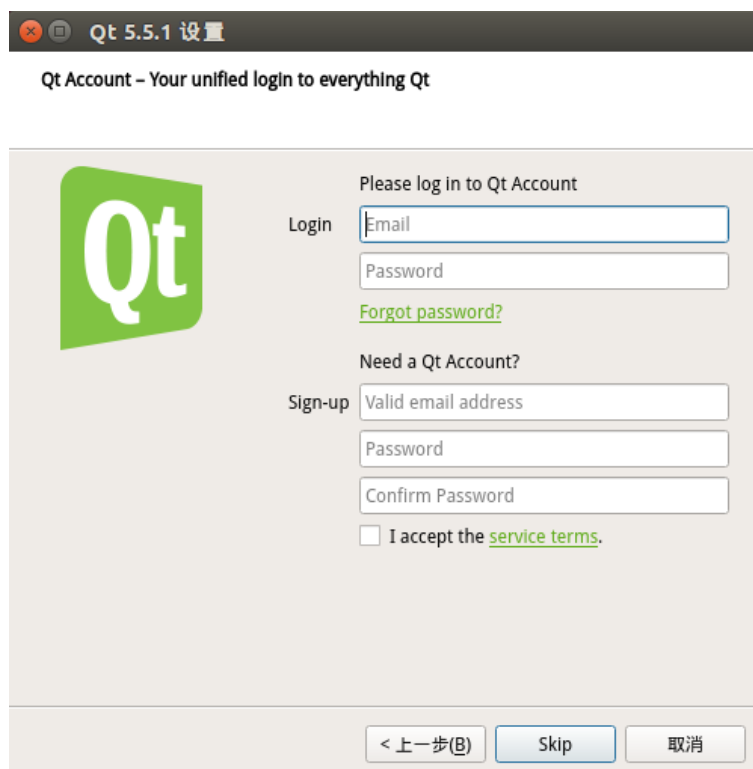


图 2.2.2. 4 设置账号密码窗口



图 2.2.2. 5 欢迎安装页面

这一步是选择安装文件夹，建议与本文一样默认（下面操作的指令都是与这个路径有关的）。点击下一步。



图 2.2.2. 6 选择安装路径

出现的窗口是选择组件，也是默认即可（可根据个人所需要的组件酌情选择），点击下一步。

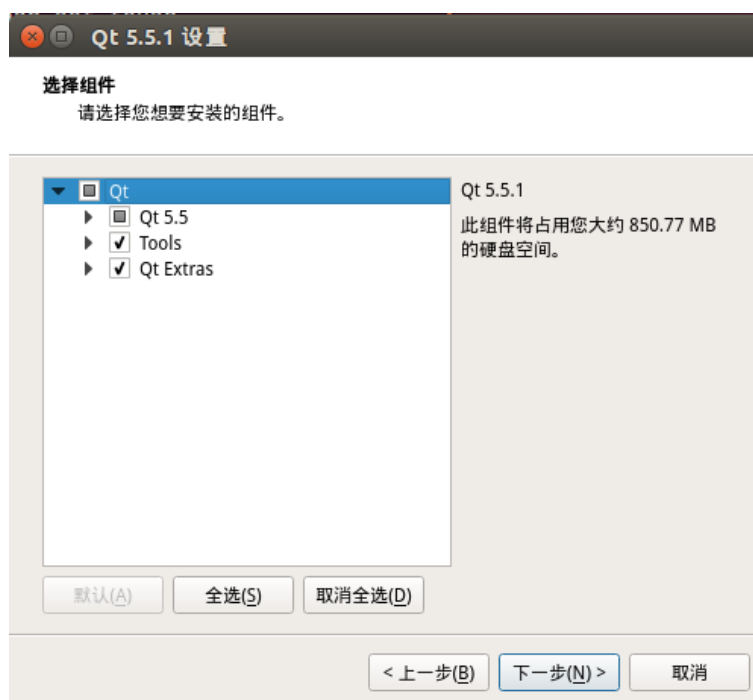


图 2.2.2. 7 选择 Qt 组件

出现一个询问许可协议的窗口，选择第一项已同意并阅读条款。再点击下一步。到了已做好安装准备这一步，直接单击安装即可。



图 2.2.2. 8 接受条款

到此 Qt 已经准备好安装了。



图 2.2.2. 9 Qt 准备安装

可以看到 Qt 5.5.1 正在安装。



图 2.2.2. 10 Qt 正在安装

等待 Qt 5.5.1 安装，单击完成按钮。



图 2.2.2. 11 安装完成

注意：安装完成 Qt 5.5.1 是不会有桌面启动快捷方式的，可以在 `/opt/Qt5.5.1/Tools/QtCreator/bin/` 下直接执行 `qtcreator.sh` 脚本来启动 Qt Creator。

启动 Qt creator，如下执行指令。

Ubuntu# `/opt/Qt5.5.1/Tools/QtCreator/bin/qtcreator.sh &` (&的作用为在后台运行)

或者在桌面顶层左侧资源搜索器处搜索“qt”也可以打开 Qt Creator。（如果需要使能环境变量的情况下，不建议这样打开）。

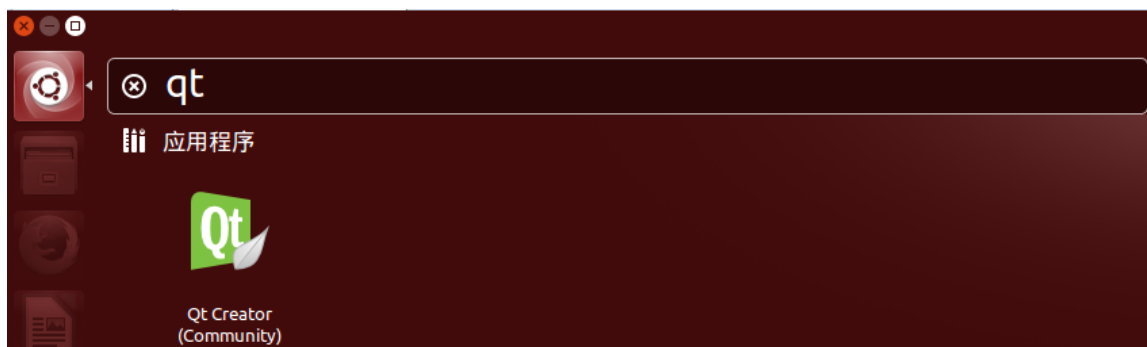


图 2.2.2. 12 在资源搜索器打开 Qt Creator

Qt Creator 启动后出现下图欢迎页面，至此 Qt Creator 安装完毕。

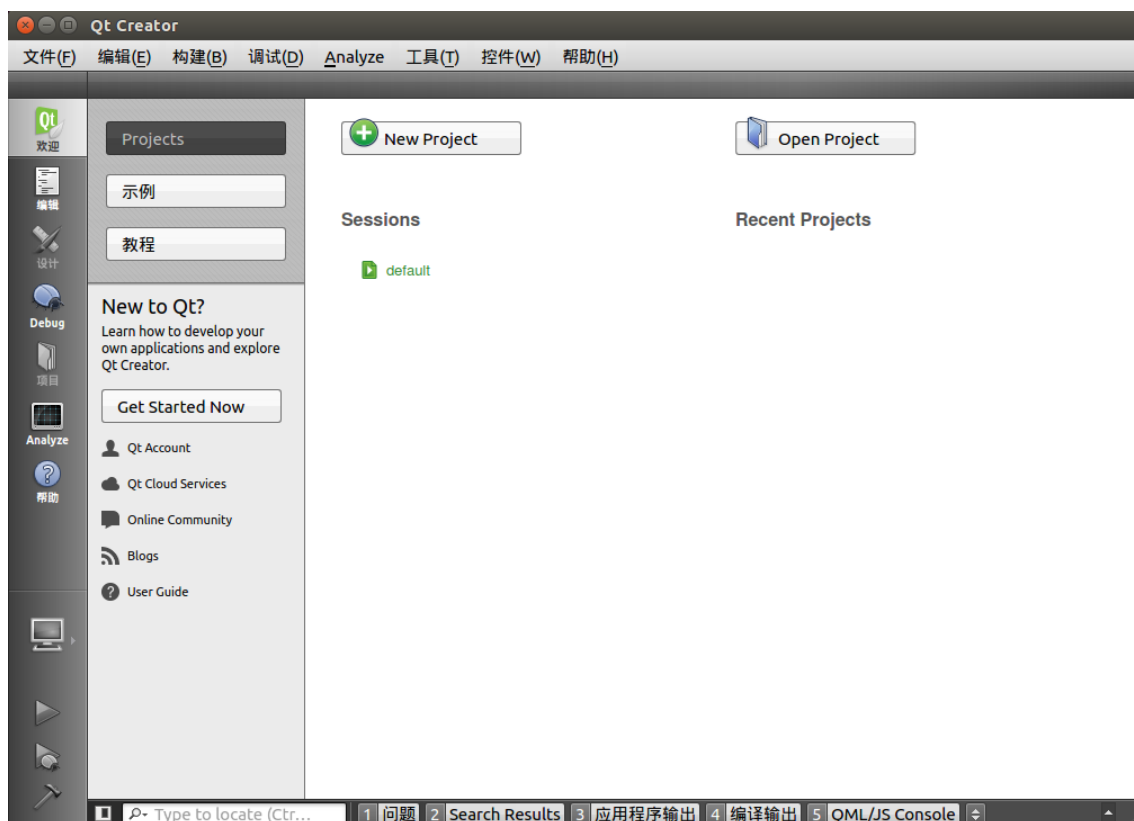


图 2.2.2. 13 Qt 欢迎页面

3.搭建 IMX6 Qt Creator 交叉编译环境

3.1 查看本机的 Kits

按上面的方法启动 Qt Creator，如下图点击工具》选项》构建与运行。



图 3. 1 查看本机的 Kits

可以看到下图已经有一个 Kit(构建套件)，是安装 Qt Creator 自带的，编译出来的 Qt 程序可在 Ubuntu 虚拟机上运行。要想在我们 IMX6 开发板运行我们要搭建自己的 Kit。按照 2.2~2.4 小节的方法搭建自己的 Kit。

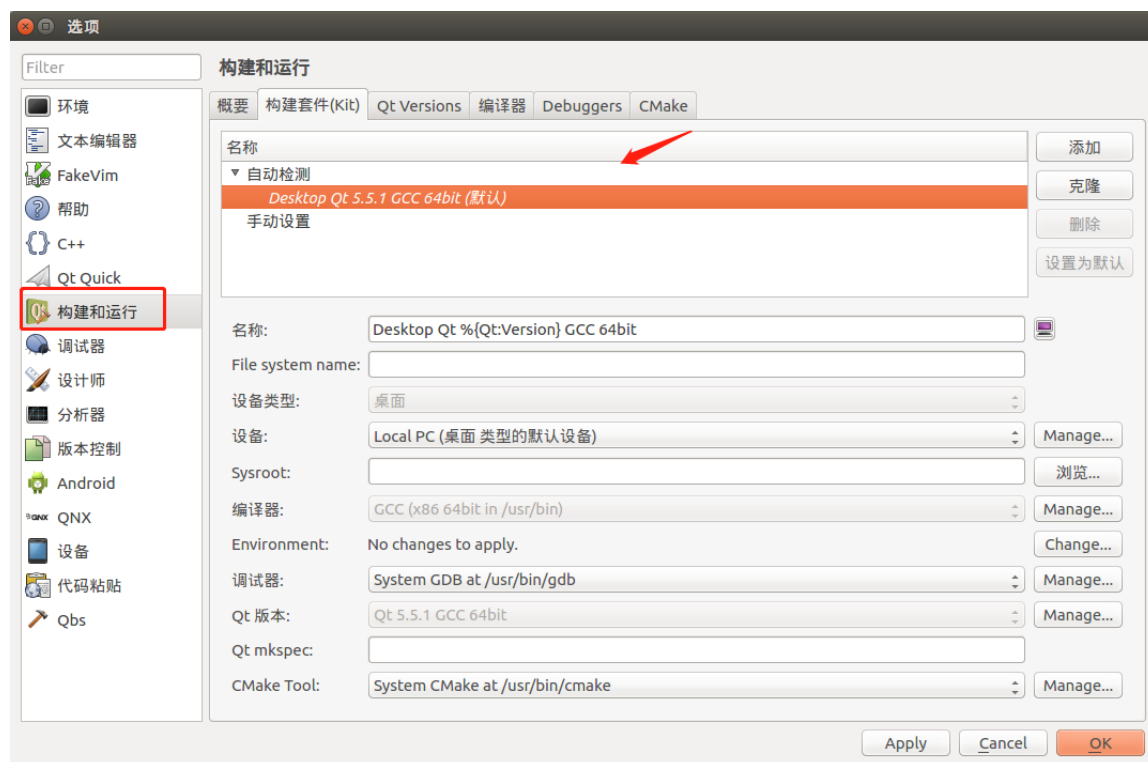


图 3.1.2 本机的 Kits

3.2 配置交叉编译器

按如下的步骤添加 GCC 编译器，（这里我啰嗦一下，我们配置的是 GCC 编译器，Windows 下配置的编译器名称叫 MinGW（Minimalist GNU on Windows，实际上是将经典的开源 C/C++ 语言编译器 GCC 移植到了 Windows 平台下）。在选项中还有更多的编译器名称，这一步也是 Qt 为跨平台做的一步工作。

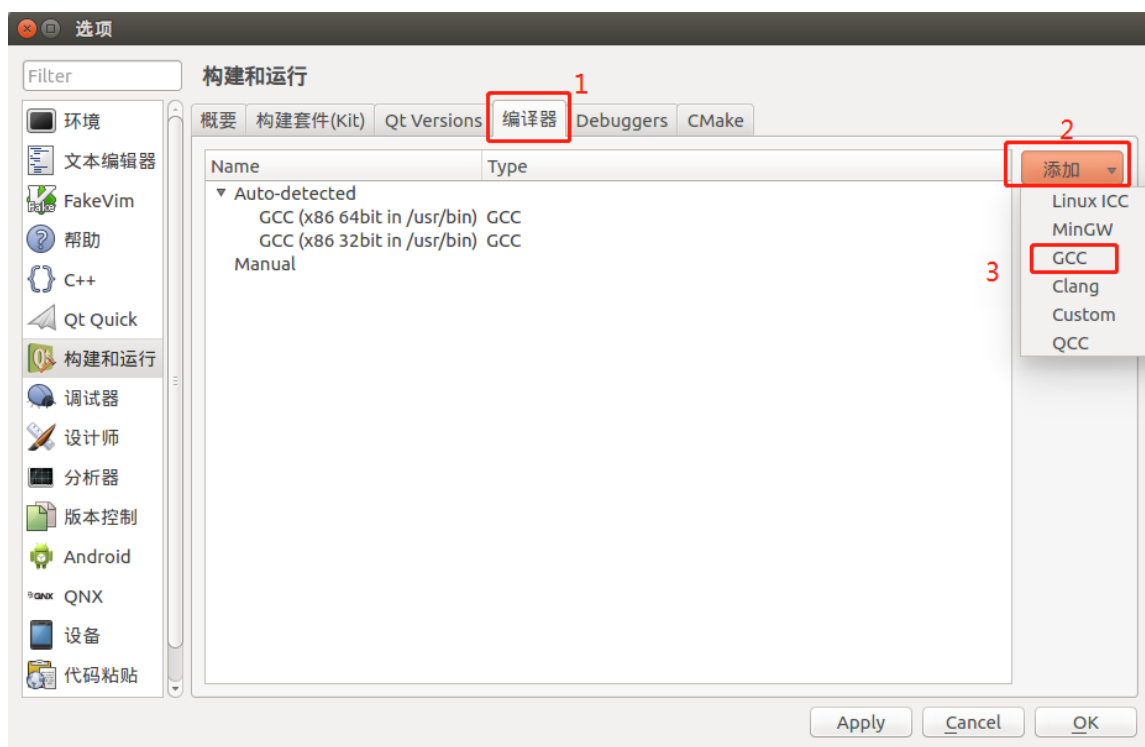


图 3.2.1 添加 GCC 编译器

按如下步骤浏览交叉编译器的路径

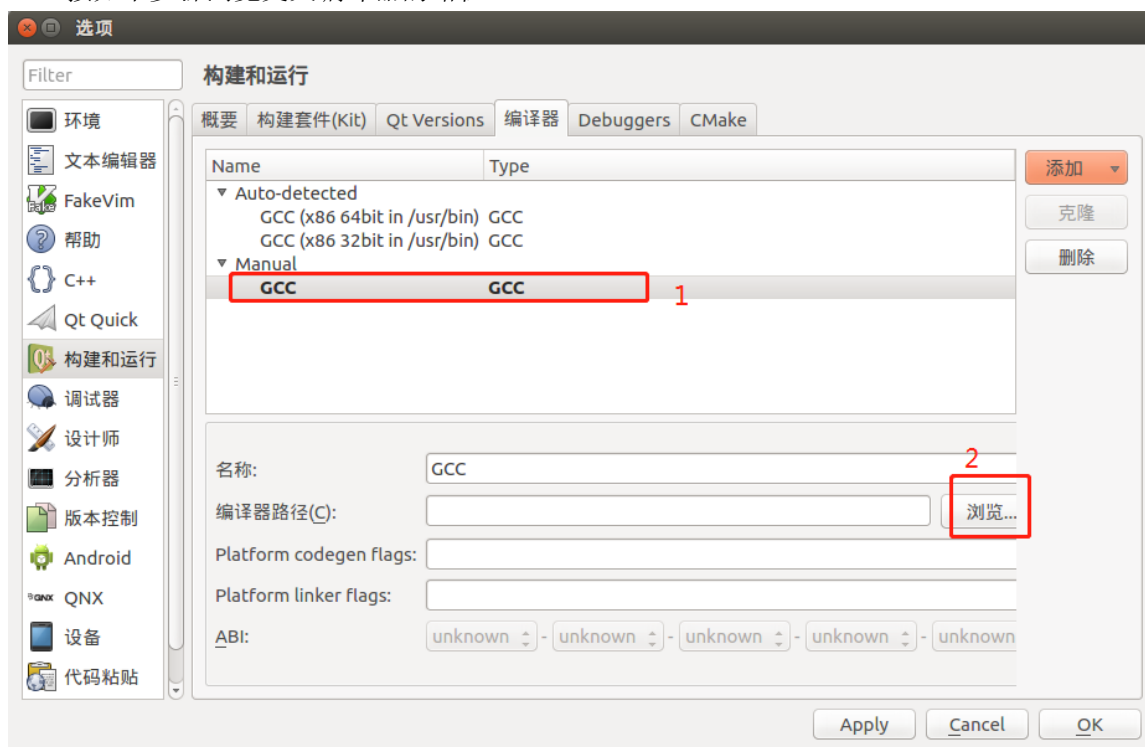


图 3.2.2 点击浏览按钮

浏览这个路径/opt/fsl-imx-x11/4.1.15-2.1.0/sysroots/x86_64-pokysdk-linux/usr/bin/arm-poky-linux-gnueabi/arm-poky-linux-gnueabi-g++找到我们第 1 节里安装的交叉编译器 arm-poky-linux-gnueabi-g++。(题外话这里为什么不是 arm-poky-linux-gnueabi-gcc 呢? gcc 是编译 c 语言的, g++

+可以编译 c 语言还包括 c++，我们 Qt 编程语言就是 c++）。

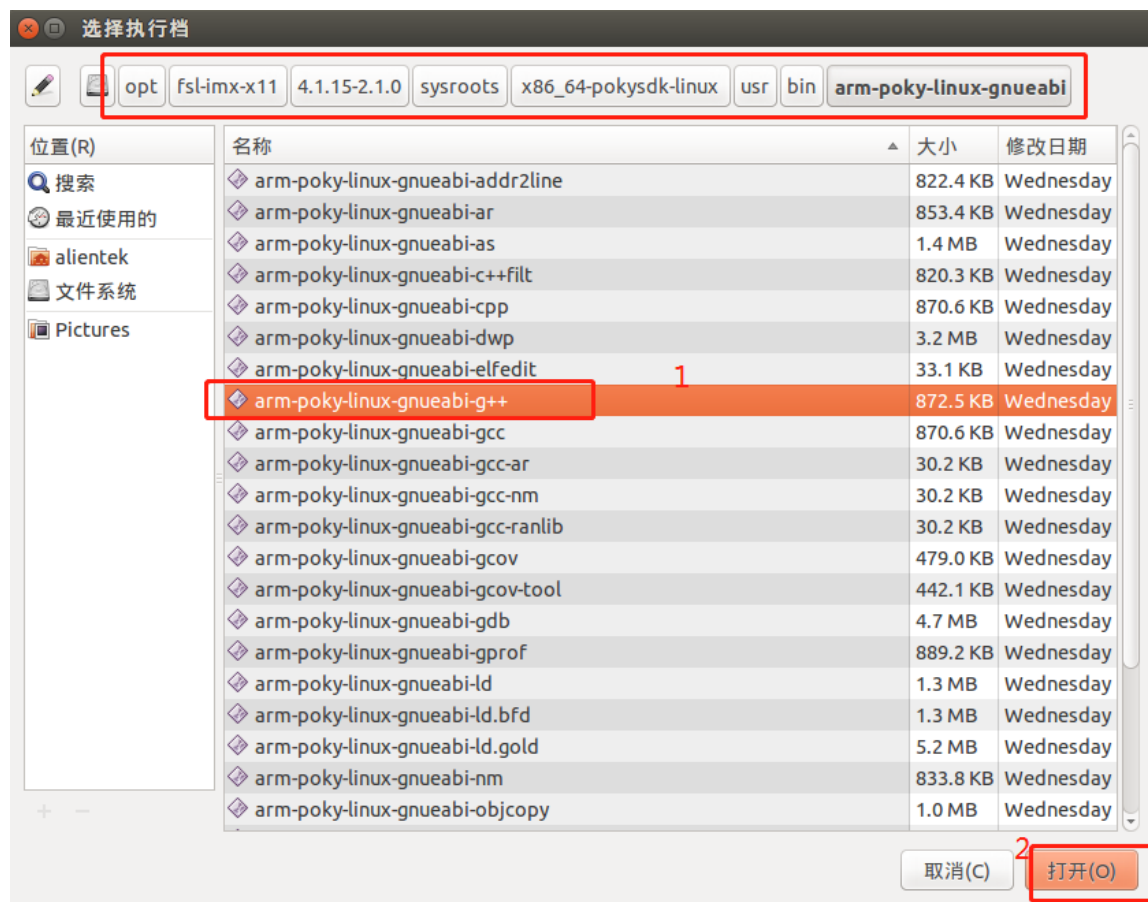


图 3.2.3 选择 g++编译器

选择好了后，修改名称，修改为 imx6gcc，再点击应用（Apply）。再到 3.3 小节配置 qmake。

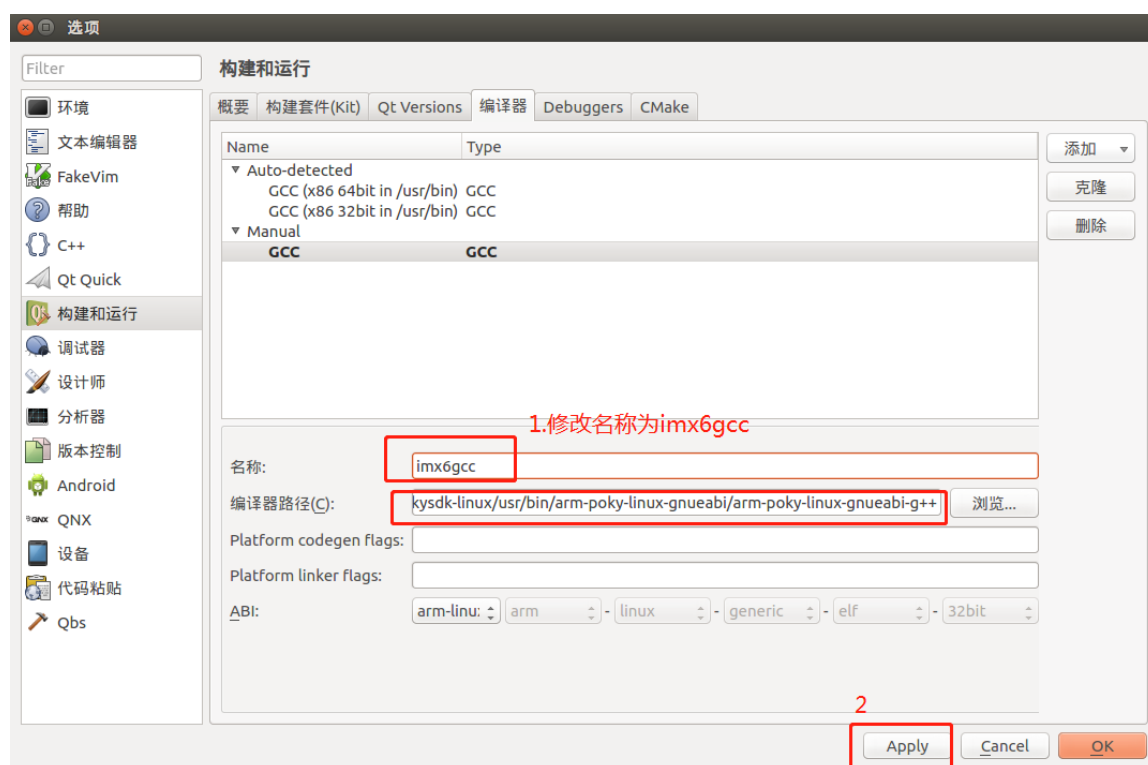


图 3.2. 4 修改名称

3.3 配置 qmake

点击 Qt Versions 这一项，可以 Qt Creator 有一个 qmake，这个 qmake 是安装 Qt Creator 时候自带的（qmake 的作用是生成用于编译的 Makefile 文件）。和配置交叉编译器一样，我们需要配置我们 IMX6 开发板所使用的 qmake。点击右上角的添加，如下图 3.3. 1。

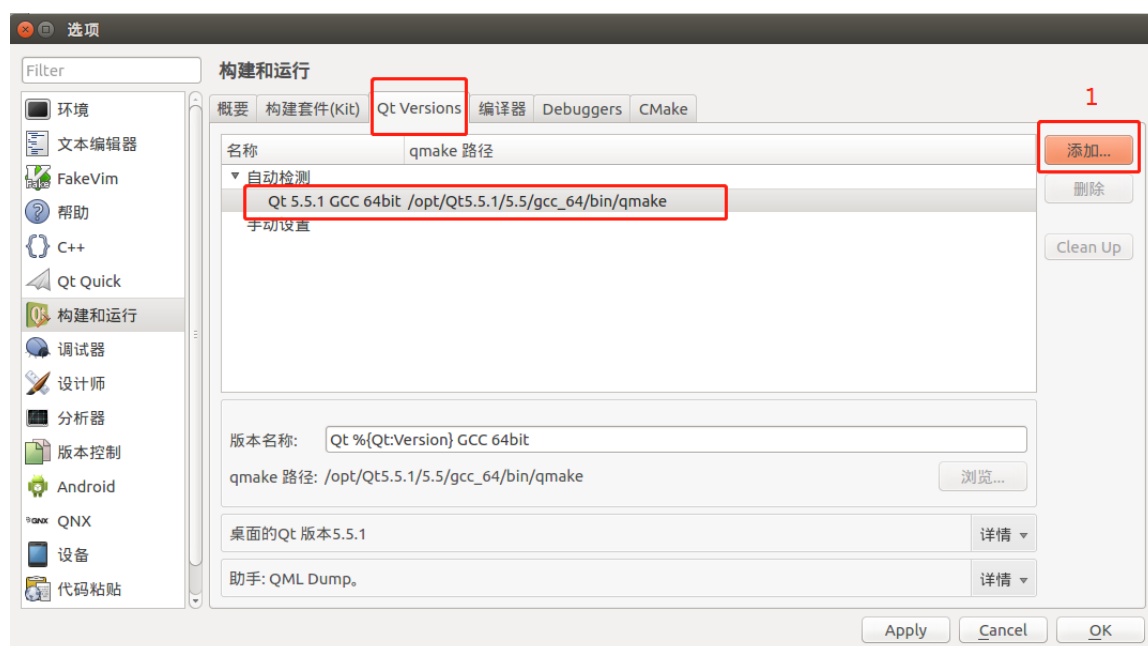


图 3.3. 1 点击添加 qmake

我们 IMX6 开发板所用的 qmake 在交叉编译工具的安装路径/opt/fsl-imx-x11/4.1.15-2.1.0/sysroots/x86_64-pokysdk-linux/usr/bin/qt5/qmake 下。

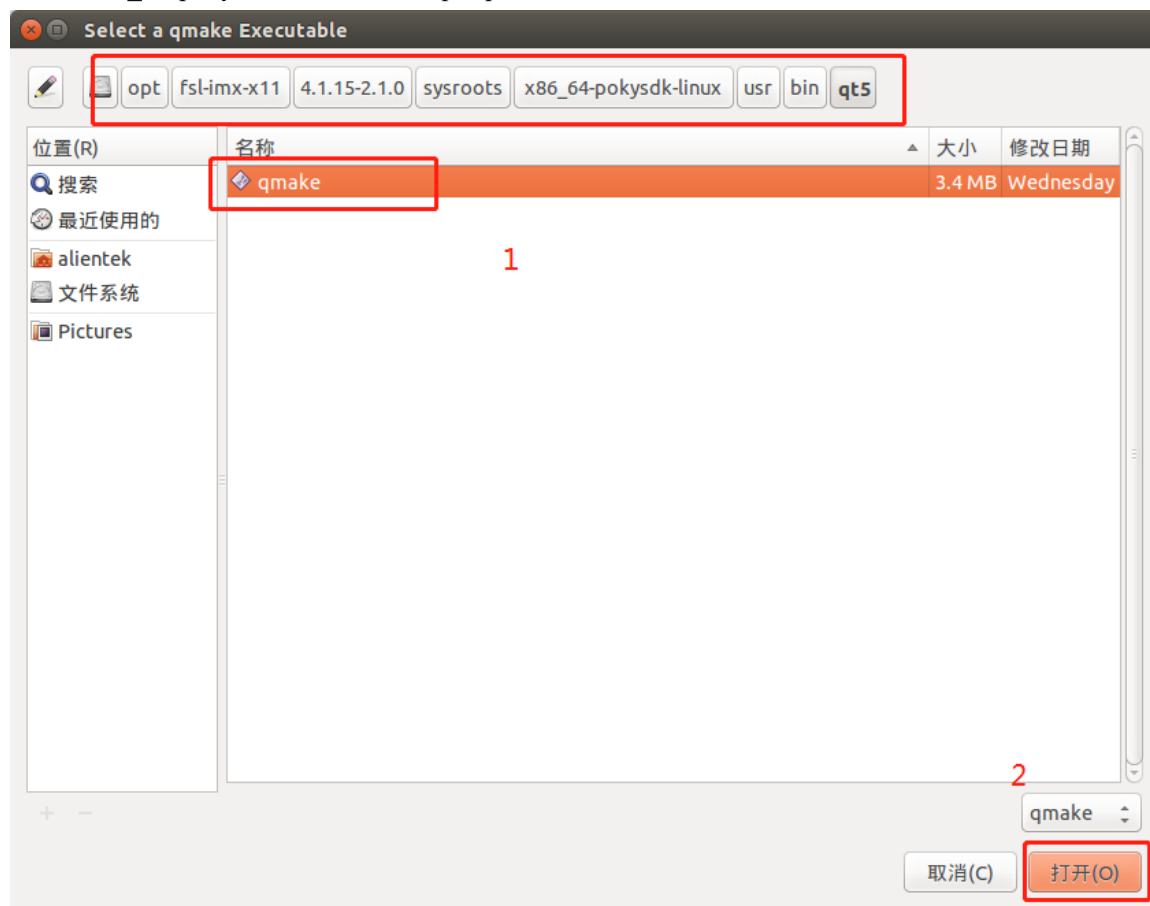


图 3.3. 2 选择 qmake

为了好区分，在版本名称处把 Qt %Qt:Version} (qt5)改为 Qt %Qt:Version} (imx6)，点击应用后（Apply）后，就完成了 qmake 的配置。

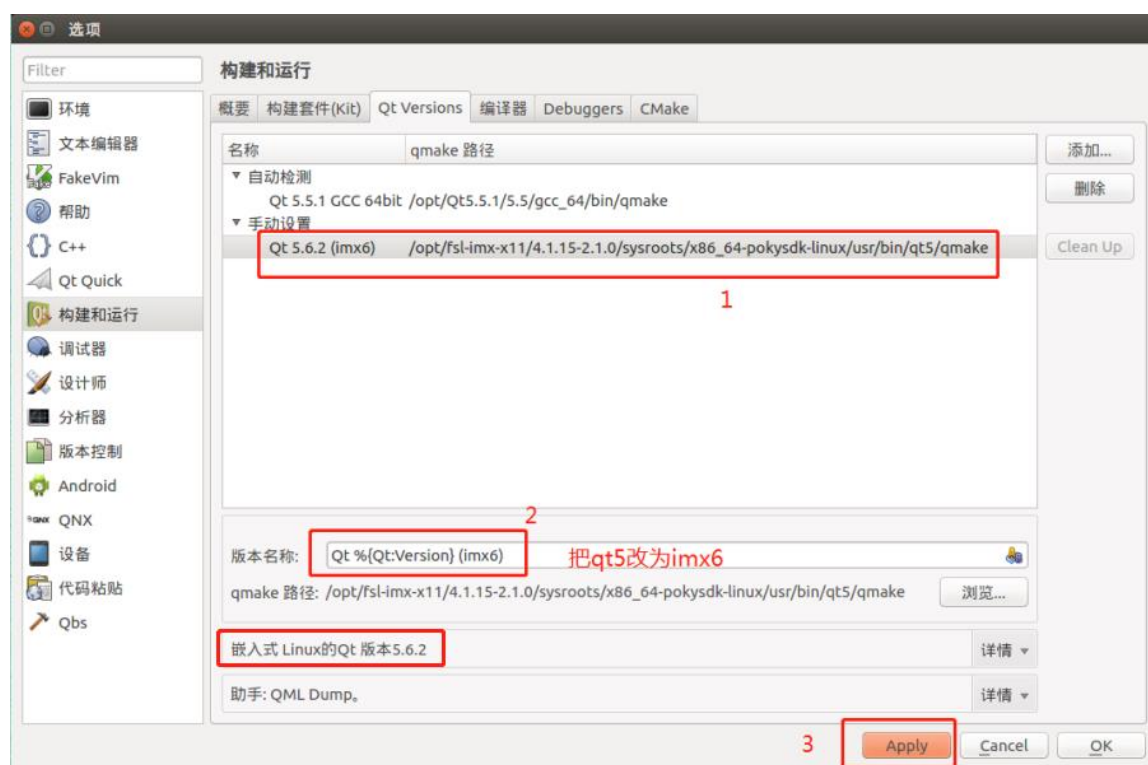


图 3.3.3 修改版本名称

备注：在这里可以看到我们交叉编译工具链里的 Qt 版本是 Qt5.6.2，这个与 Qt5.5.1 版本的 Qt Creator 没有太大关系（这里要啰嗦一下，为什么我们不用 Qt5.6.2 的 Qt Creator 呢？官方自从 Qt5.6 以后就没有 QWebKit（用于网络编程的模块），对于一些想用 QWebKit 编程而又不想升级 Qt 版本库的读者，所以我们就在交叉编译工具链里的 Qt5.6.2 版本额外添加了 QWebKit 模块，同时我们 I.MX6 文件系统也添加了 QWebKit 相关的库）。我们只需要使用交叉编译器里的 Qt5.6.2 版本的来编译我们 I.MX6 开发板上跑的 Qt 应用程序，现在，我们可以把 Qt Creator 看作一个具有界面的程序编辑器。

3.4 配置 gdb 调试器

点击 Add，准备添加 gdb 调试器。（注：gdb 调试器可不用配置亦可）

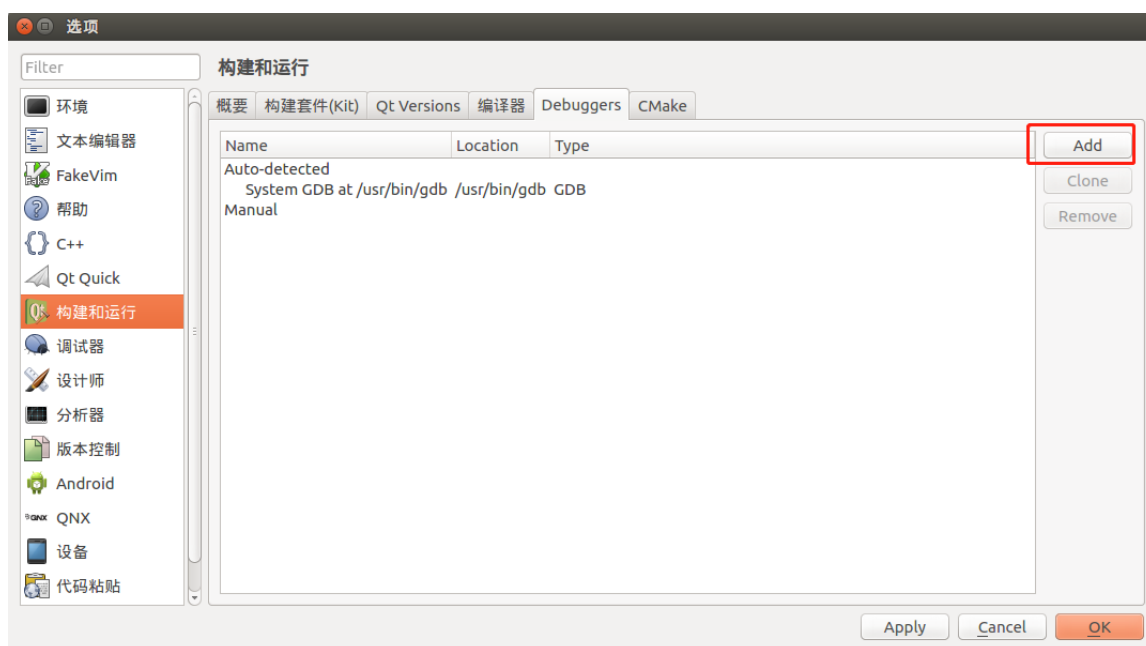


图 3.4. 1 点击 Add

找到我们交叉编译工具链安装路径/opt/fsl-imx-x11/4.1.15-2.1.0/sysroots/x86_64-pokysdk-linux/usr/bin/arm-poky-linux-gnueabi/下的 arm-poky-linux-gnueabi-gdb。选择后点击打开。

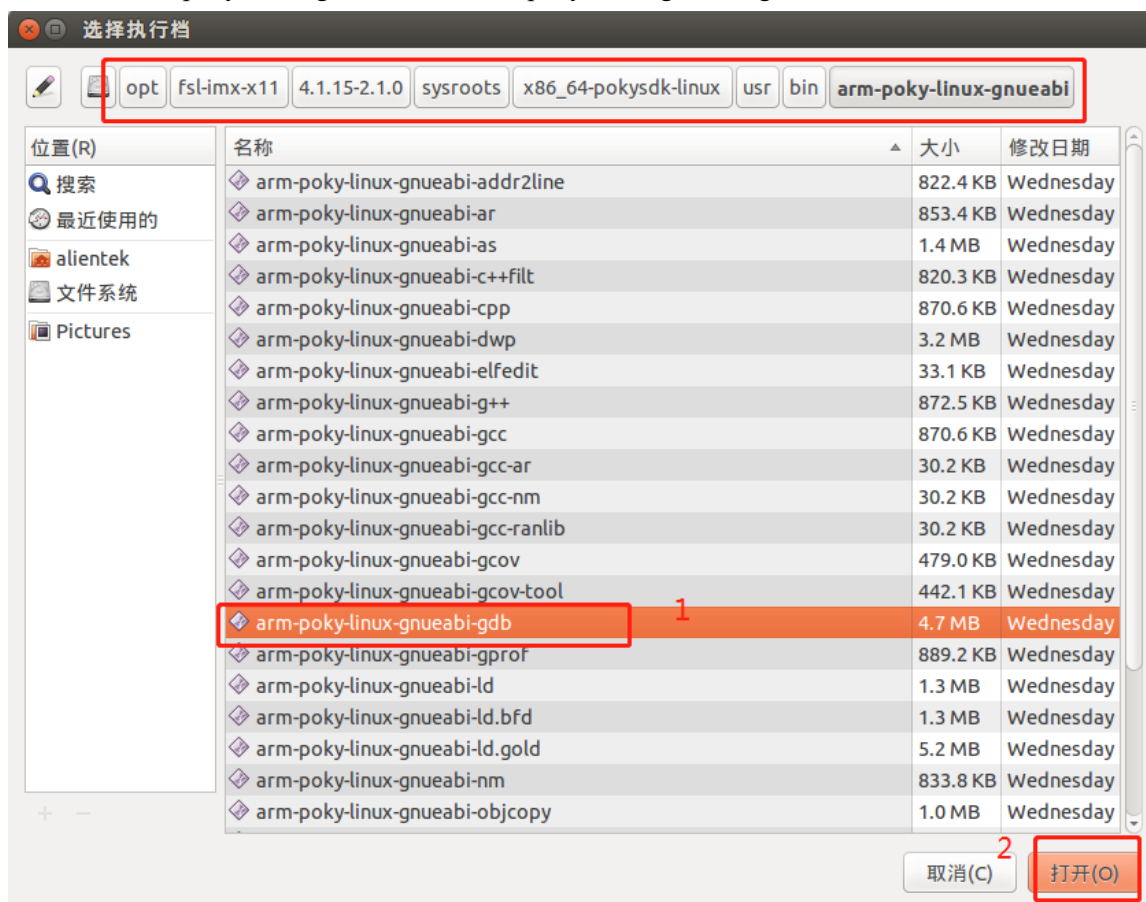


图 3.4. 2 选择 gdb 调试工具

如下图 3.4.3 图修改 Name 处名称为 imx6gdb。

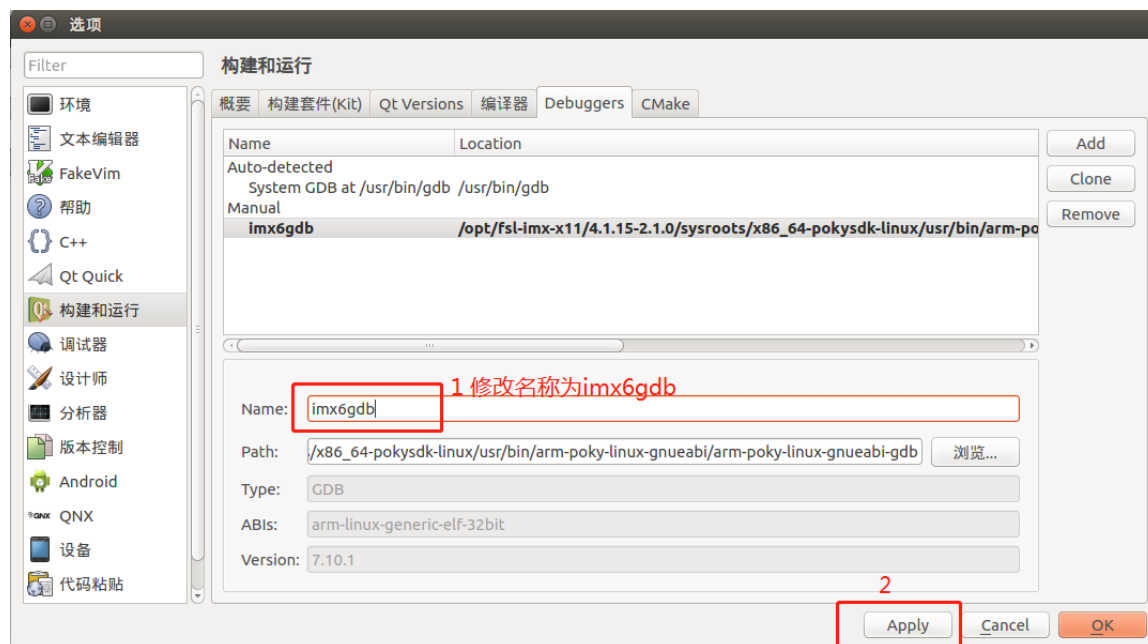


图 3.4.3 修改调试器的 “Name”

3.5 配置 Kits

按如下的步骤配置 Kits，点击 OK 确认即可。



图 3.5.1 Kits 配置

这里说明一下为什么要在 Qt mkspec 填上 linux-oe-g++。实质这里指定了../mkspec/linux-o

e-g++下的 qmake.conf 文件(用于执行 qmake 时生成 Makefile)。

用 env 指令查看使能工具链后的环境变量, 查看我们用了哪个 mkspecs。如下图

#Ubuntu env

```
llbc:/opt/fsl-imx-x11/4.1.15-2.1.0/sysroots/x86_64-pokysdk-linux/usr/bin/arm-poky-linux-musl:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
OE_QMAKE_PATH_HOST_BINS=/opt/fsl-imx-x11/4.1.15-2.1.0/sysroots/x86_64-pokysdk-linux/usr/bin/Qt5
QT_IN_MODULE=FcItz
QT_QPA_PLATFORMTHEME=appmenu-qt5
LD=arm-poky-linux-gnueabi-ld --sysroot=/opt/fsl-imx-x11/4.1.15-2.1.0/sysroots/cortexa7hf-neon-poky-linux-gnueabi
LC_IDENTIFICATION=en_US.UTF-8
PWD=/home/allientek
JOB=gnome-session
STRIP=arm-poky-linux-gnueabi-strip
XNOFFICE=gnome-fcItz
GNOME_KEYRING_PID=1925
LANG=zh_CN.UTF-8
OE_QMAKE_LINK=arm-poky-linux-gnueabi-g++ -arch=armv7ve -fpu=neon -float=abi=hard -mcpu=cortex-a7 --sysroot=/opt/fsl-imx-x11/4.1.15-2.1.0/sysroots/cortexa7hf-neon-poky-linux-gnueabi
GOM_LANG=zh_CN
MANDATORY_PATH=/usr/share/occonf/ubuntu.mandatory.path
QMAKESPEC=/opt/fsl-imx-x11/4.1.15-2.1.0/sysroots/cortexa7hf-neon-poky-linux-gnueabi/usr/lib/Qt5/mkspecs/linux-oe-g++
QT_CONF_PATH=/opt/fsl-imx-x11/4.1.15-2.1.0/sysroots/x86_64-pokysdk-linux/usr/bin/Qt5/Qt.conf
LC_MEASUREMENT=en_US.UTF-8
GCCORE_NATIVE_SYSROOT=/opt/fsl-imx-x11/4.1.15-2.1.0/sysroots/x86_64-pokysdk-linux
COMPIZ_CONFIG_PROFILE=ubuntu
IN_CONFIG_PHASE=1
CONFIGURE_FLAGS=-target=arm-poky-linux-gnueabi --host=arm-poky-linux-gnueabi --build=x86_64-linux --with-libtool-sysroot=/opt/fsl-imx-x11/4.1.15-2.1.0/sysroots/cortexa7hf-neon-poky-linux-gnueabi
PAPERSIZE=letter
COMSESSION=ubuntu
OE_QMAKE_QT_CONFIG=/opt/fsl-imx-x11/4.1.15-2.1.0/sysroots/cortexa7hf-neon-poky-linux-gnueabi/usr/lib/Qt5/mkspecs/qconfig.pri
OE_QMAKE_CC=arm-poky-linux-gnueabi-gcc -arch=armv7ve -fpu=neon -float=abi=hard -mcpu=cortex-a7 --sysroot=/opt/fsl-imx-x11/4.1.15-2.1.0/sysroots/cortexa7hf-neon-poky-linux-gnueabi
KCLFLAGS=-sysroot=/opt/fsl-imx-x11/4.1.15-2.1.0/sysroots/cortexa7hf-neon-poky-linux-gnueabi
CXX=arm-poky-linux-gnueabi-g++ -arch=armv7ve -fpu=neon -float=abi=hard -mcpu=cortex-a7 --sysroot=/opt/fsl-imx-x11/4.1.15-2.1.0/sysroots/cortexa7hf-neon-poky-linux-gnueabi
OBJCOPY=arm-poky-linux-gnueabi-objcopy
SESSIONTYPE=gnome-session
SHLVL=1
HOME=/home/allientek
XDG_SEAT=seat0
LANG=zh_CN.UTF-8
```

图 3.5.2 查看使用的 mkspecs

3.6 配置环境变量

要想正确使用安装的编译工具链, 还需要向 Qt Creator 加入环境变量。

在 Qt creator 的启动脚本里使能环境变量, 在 /opt/Qt5.5.1/Tools/QtCreator/bin/qtcreator.sh 的第一行插入使能交叉编译工具链的指令 source /opt/fsl-imx-x11/4.1.15-2.1.0/environment-setup-cortexa7hf-neon-poky-linux-gnueabi (注意这里不能在#! /bin/sh 以下行插入)。

Ubuntu# sudo vi /opt/Qt5.5.1/Tools/QtCreator/bin/qtcreator.sh

```
alientek@ubuntu: ~
source /opt/fsl-imx-x11/4.1.15-2.1.0/environment-setup-cortexa7hf-neon-poky-linux-gnueabi
#!/bin/sh
makeAbsolute() {
    case $1 in
        /*)
            # already absolute, return it
            echo "$1"
            ;;
        *)
            # relative, prepend $2 made absolute
            echo `makeAbsolute "$2" "$PWD"/"$1" | sed 's,/\.,$,,'`
            ;;
    esac
}

me=`which "$0" # Search $PATH if necessary
if test -L "$me"; then
    # Try readlink(1)
    readlink=`type readlink 2>/dev/null` || readlink=
    if test -n "$readlink"; then
        # We have readlink(1), so we can use it. Assuming GNU readlink (for -f).
        me=`readlink -nf "$me"`
    else
        # No readlink(1), so let's try ls -l
        me=`ls -l "$me" | sed 's/^.*-> //'`
        base=`dirname "$me"`
        me=`makeAbsolute "$me" "$base"`
    fi
fi
```

图 3.6.1 在启动脚本添加环境变量

按:wq 保存退出后, 先关闭当前的 Qt Creator。再在终端使用指令启动 Qt Creator。注意要想这个环境变量生效, 必需通过脚本来启动 Qt Creator。

Ubuntu# /opt/Qt5.5.1/Tools/QtCreator/bin/qtcreator.sh & (&的作用为在后台运行)


```
alientek@ubuntu:~$ /opt/Qt5.5.1/Tools/QtCreator/bin/qtcreator.sh &
[1] 5600
alientek@ubuntu:~$
```

图 3.6. 2 使用命令行启动 Qt Creator

3.7 测试编译 Qt 工程

配置好了 IMX6 Qt Creator 的交叉编译环境以后，使用脚本打开 Qt Creator。

打开 Qt 工程文件，本文就以我们 Qt 网络例程为例，拷贝 1、例程源码->8、Qt 综合例程源码->QWebBrowser 文件夹，演示如何编译。在 Qt Creator 工具栏文件》打开文件或者项目。

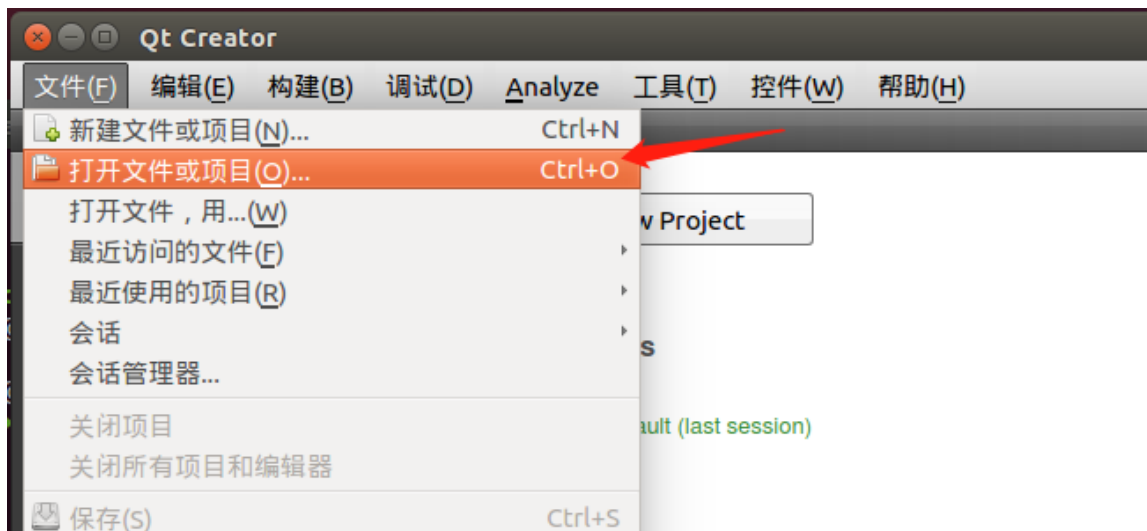


图 3.7. 1 打开文件或项目页面

打开我们的例程 QWebBrowser.pro 文件。

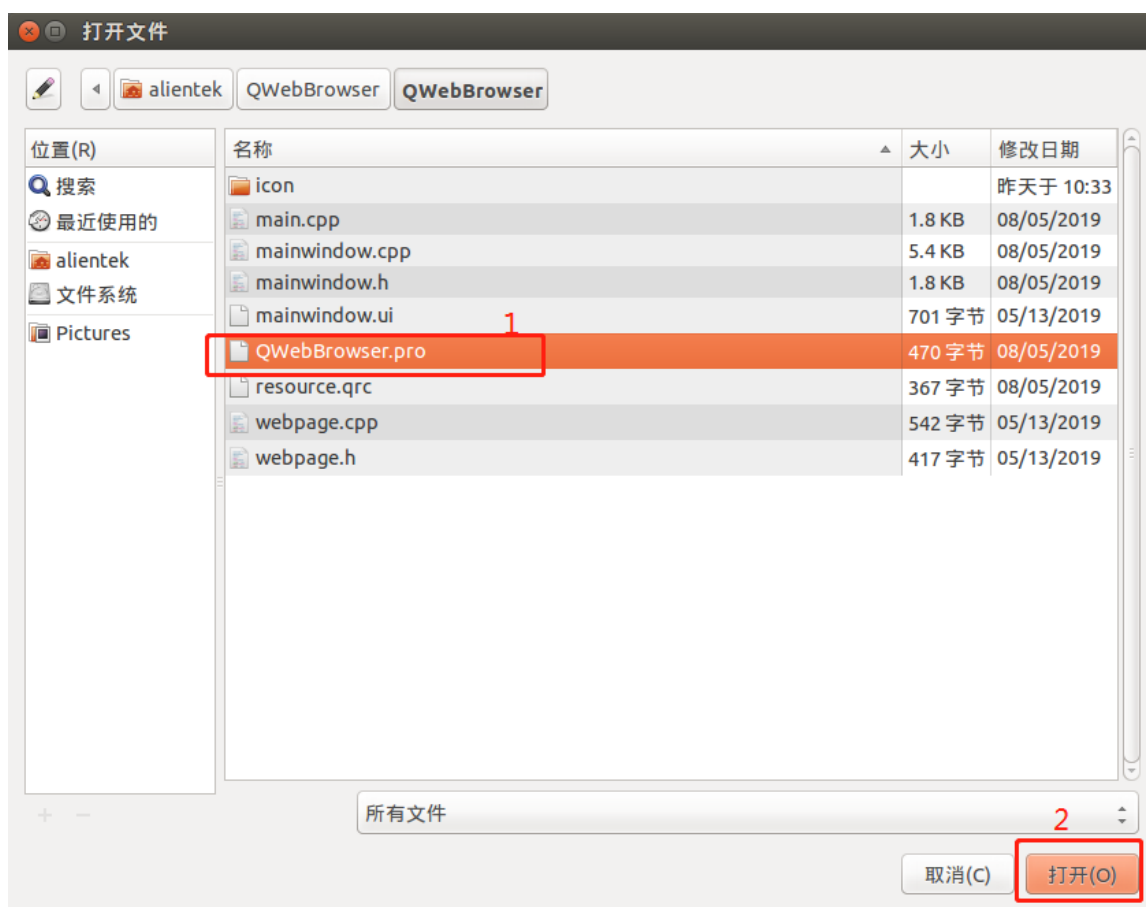


图 3.7.2 选择项目打开

打开后选择我们搭建的套件 Kits，我们选择 imx6 Kits。选择好了就点击 Configure Project。

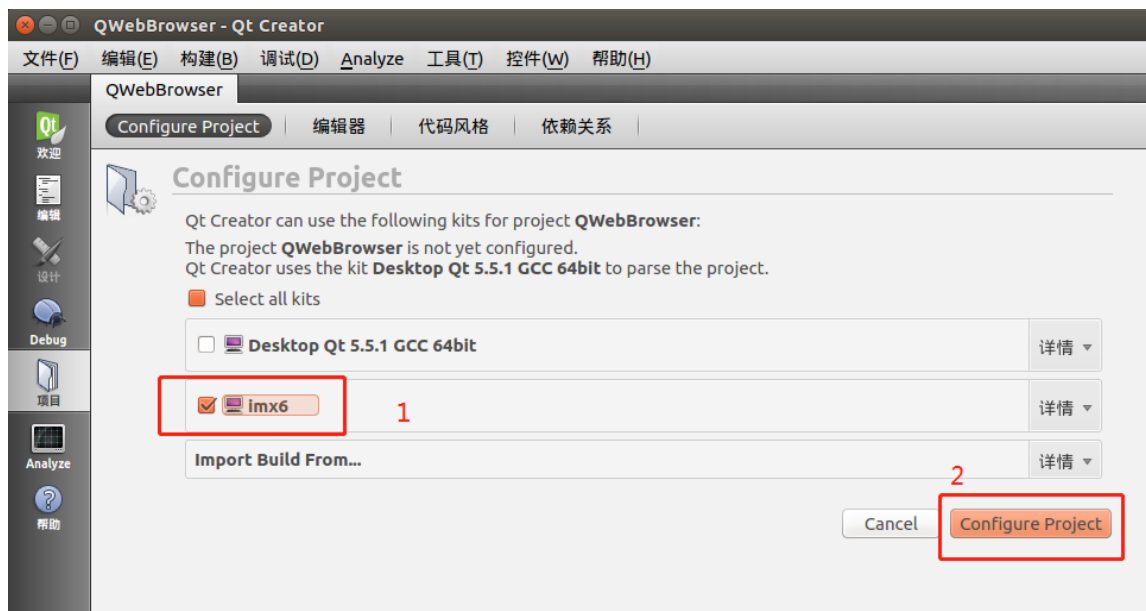


图 3.7.3 选择我们搭建的 imx6 套件

右键工程项目，选择构建，不要点击运行，因为我们编译出来的可执行文件（程序）是 IMX6（ARM）平台的。不能在 Ubuntu 上运行。

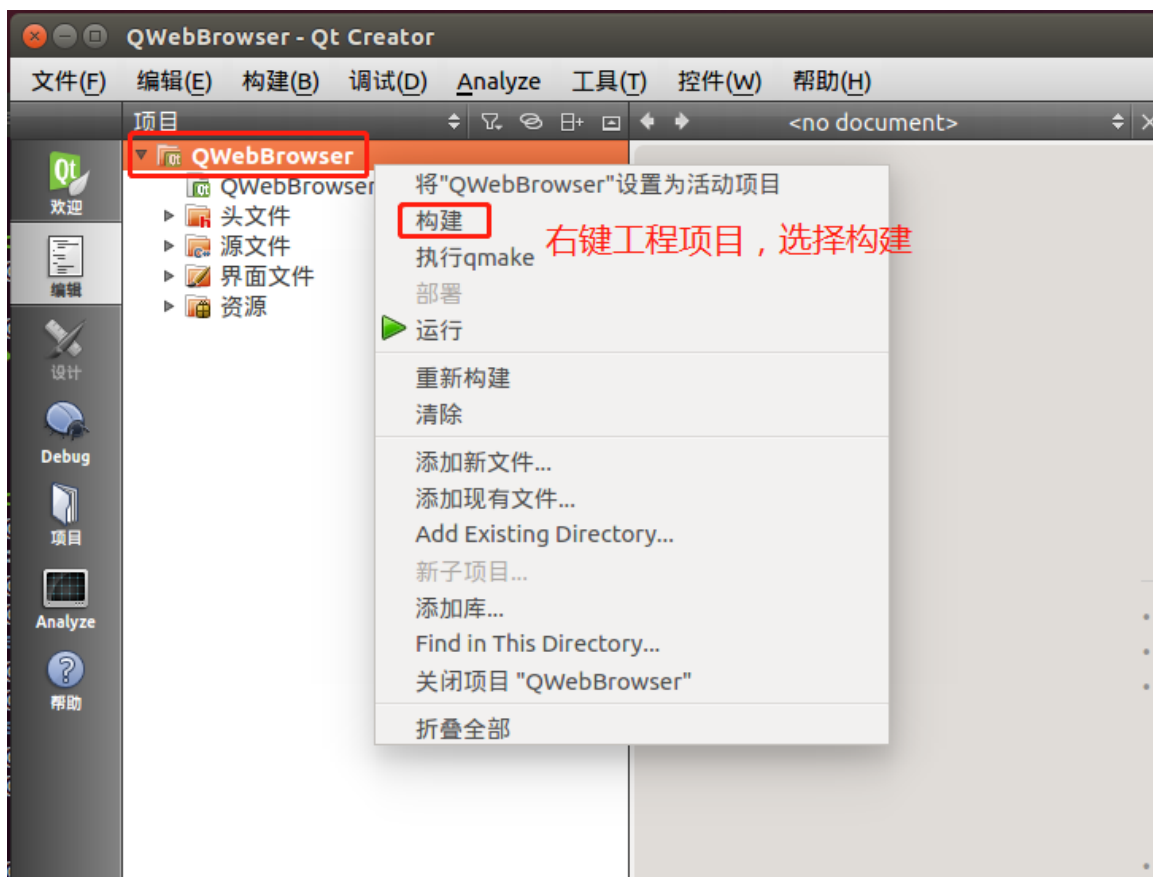


图 3.7.4 点击构建项目

可以看到右下角正在构建，并且构建完成后是绿色的进度条，表示没有错误。

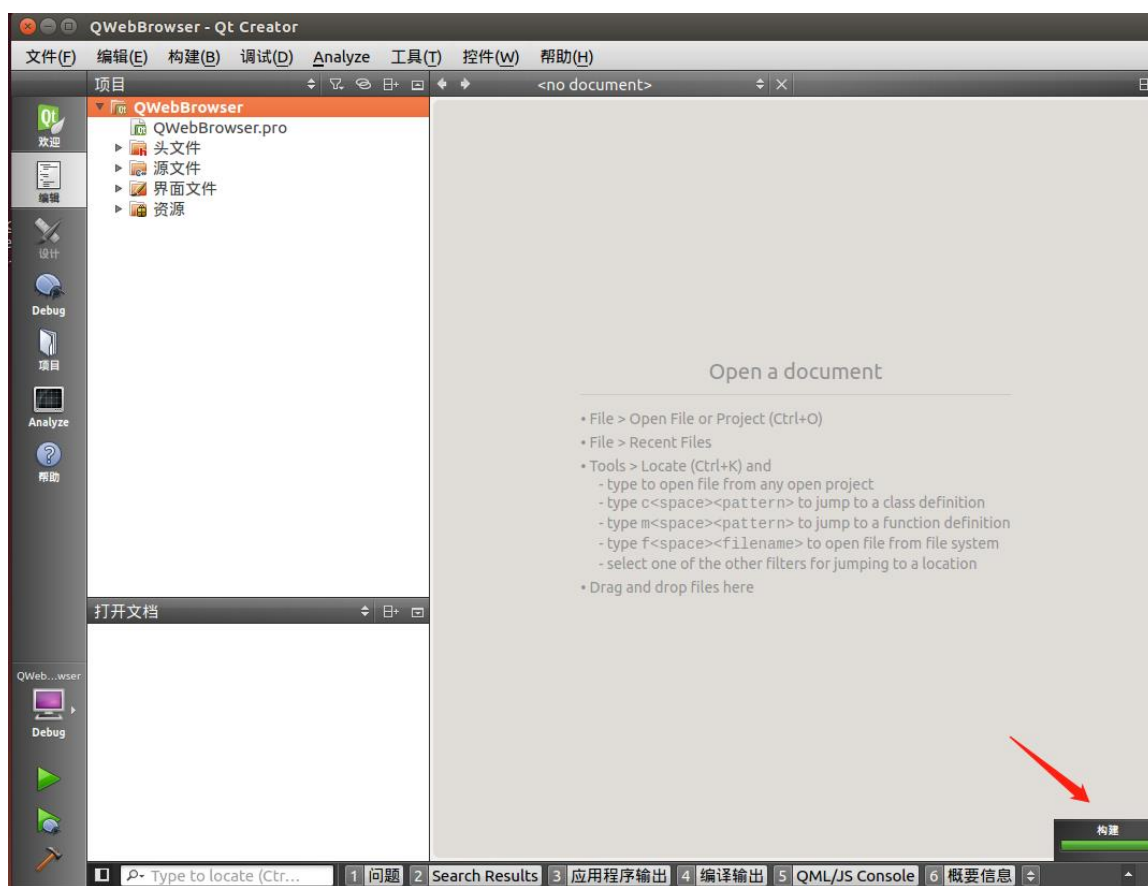


图 3.7.5 构建项目的过程

并且点击编译输出这一栏可以看到没有出现输出错误，表明我们的程序已经正常编译通过了。

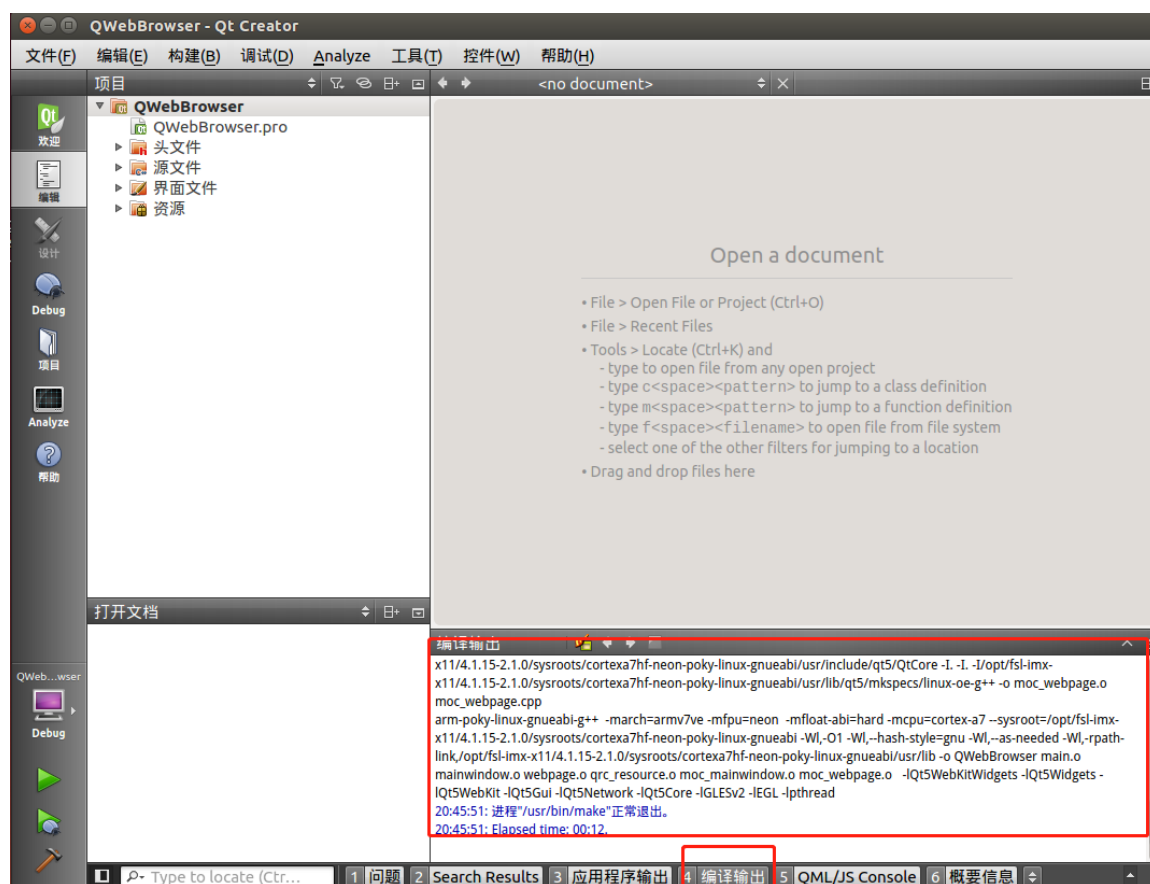


图 3.7.6 编译输出

在终端查看我们的编译出来的可执行文件，进入工程项目文件夹下，可以看到生成一个 build-QWebBrowser-imx6-Debug 文件夹，Qt Creator 编译输出的文件都在这个文件夹下。其中我们可以看到生成的可执行文件 QWebBrowser。

```
alientek@ubuntu:~/QWebBrowser/build-QWebBrowser-imx6-Debug$ ls
main.o      moc_mainwindow.cpp  moc_webpage.o  QWebBrowser
mainwindow.o moc_mainwindow.o    qrc_resource.cpp ui_mainwindow.h
Makefile    moc_webpage.cpp     qrc_resource.o  webpage.o
alientek@ubuntu:~/QWebBrowser/build-QWebBrowser-imx6-Debug$
```

图 3.7.7 编译后最终的可执行文件

将其拷贝到我们 IMX6 开发板上直接执行即可。

```
root@ALIENTEK-IMX6U:~# ./QWebBrowser
ts_open() failed (No such file or directory)
libpng warning: iccp: known incorrect sRGB profile
```

图 3.7.8 拷贝到开发板上运行

可以看到 LCD 屏幕上运行了 QWebBrowser 程序，下图为 4.3 寸屏 480*272 分辨率所展示 QWebBrowser 界面。



图 3.7.9 开发板 LCD 屏幕演示的 QWebBrowser 程序

4. 命令行交叉编译 Qt 工程

说明：命令行编译 Qt 工程，这里只编译已经调试好的 Qt 工程，生成可以在我们 IMX6 开发板上可以执行的文件。

拷贝 Qt 工程到 Ubuntu 虚拟机，例如本人拷贝 Qt 浏览器工程在光盘路径 1、例程源码->8、Qt 综合例程源码->QwebBrowser 整个文件夹到本人的虚拟机里，使用 cd 命令进入到这个工程目录里，可以看到这一级目录有个 QWebBrowser.pro 工程项目文件。

```
allientek@ubuntu:~/QWebBrowser/QWebBrowser$ ls
icon main.cpp mainwindow.cpp mainwindow.h mainwindow.ui QWebBrowser.pro resource.qrc webpage.cpp webpage.h
allientek@ubuntu:~/QWebBrowser/QWebBrowser$
```

图 4.1 在终端找到 Qt 工程

使用 source 指令使能当前终端交叉编译环境变量（切换终端和用户需要重新执行下面这条指令），如果您在上文已经使能过环境变量就不用再执行下面这一步了。

USER# source /opt/fsl-imx-x11/4.1.15-2.1.0/environment-setup-cortexa7hf-neon-poky-linux-gnueabi

```
allientek@ubuntu:~/QWebBrowser/QWebBrowser$ source /opt/fsl-imx-x11/4.1.15-2.1.0/environment-setup-cortexa7hf-neon-poky-linux-gnueabi
allientek@ubuntu:~/QWebBrowser/QWebBrowser$
```

图 4.2 使能环境变量

执行指令 qmake 生成 Makefile 文件，可以直接在当前目录执行 qmake，或者写成 qmake + 工程名称。如 qmake QWebBrowser.pro

USER# qmake

```
allientek@ubuntu:~/QWebBrowser/QWebBrowser$ qmake
allientek@ubuntu:~/QWebBrowser/QWebBrowser$ ls
icon main.cpp mainwindow.cpp mainwindow.h mainwindow.ui Makefile QWebBrowser.pro resource.qrc webpage.cpp webpage.h
allientek@ubuntu:~/QWebBrowser/QWebBrowser$
```

图 4.3 执行 qmake

直接执行 make，编译这个工程。

USER# make

