

搭建 Qt 开发环境简要介绍

开发环境：

Ubuntu 14.04 64bit 或以上(其他 Linux 发行版请自行解决遇到的问题)

(1) 安装依赖库

Ubuntu14.04、16.04 64bit 版本已不支持 ia32-libs 的软件包，而是使用了

lib32ncurses5、lib32z1 软件包做为替代，所以在 Ubuntu16.04 版本当中应该安装执行：

```
sudo apt install lib32ncurses5
sudo apt install lib32z1
sudo apt install lib32stdc++6
```

注：具体是哪个不太记得了，也可以自行根据报错网上搜索解决。

(2) 解压交叉编译工具

将 external-toolchain.tar 类似的包解压到/opt 目录下，之后打开终端 Terminal，运

行/opt/external-toolchain/bin/arm-none-linux-gnueabi-g++ -v，如果正确将会看到

如下类似输出：

```
user@user:/opt/external-toolchain/bin$ /opt/external-toolchain/bin/arm-none-linux-gnueabi-
g++ -v
Using built-in specs.
COLLECT_GCC=/opt/external-toolchain/bin/arm-none-linux-gnueabi-g++
COLLECT_LTO_WRAPPER=/opt/external-toolchain/bin/../libexec/gcc/arm-none-linux-gnueabi/4.5.
1/lto-wrapper
Target: arm-none-linux-gnueabi
Configured with: /scratch/nathan/arm-lite/src/gcc-4.5-2010.09/configure --build=i686-pc-li
nux-gnu --host=i686-pc-linux-gnu --target=arm-none-linux-gnueabi --enable-threads --disabl
e-libmudflap --disable-libssp --disable-libstdcxx-pch --enable-extra-xxl-lite-multilibs --
with-arch=armv5te --with-gnu-as --with-gnu-ld --with-specs='%{save-temps: -fverbose-asm} %
{funwind-tables|fno-unwind-tables|mabi=*|ffreestanding|nostdlib:;-funwind-tables} -D CS
SOURCERYGXX MAJ =2010 -D CS SOURCERYGXX MIN =9 -D CS SOURCERYGXX REV =50 %{02:%{!fno-
remove-local-statics: -fremove-local-statics}} %0*:%{0|00|01|02|0s;;%{!fno-remove-local-
statics: -fremove-local-statics}}}' --enable-languages=c,c++ --enable-shared --enable-lto
--enable-symvers=gnu --enable-__cxa_atexit --with-pkgversion='Sourcery G++ Lite 2010.09-50
' --with-bugurl=https://support.codesourcery.com/GNUToolchain/ --disable-nls --prefix=/opt
/codesourcery --with-sysroot=/opt/codesourcery/arm-none-linux-gnueabi/libc --with-build-sy
sroot=/scratch/nathan/arm-lite/install/arm-none-linux-gnueabi/libc --with-gmp=/scratch/nat
han/arm-lite/obj/host-libs-2010.09-50-arm-none-linux-gnueabi-i686-pc-linux-gnu/usr --with-
mpfr=/scratch/nathan/arm-lite/obj/host-libs-2010.09-50-arm-none-linux-gnueabi-i686-pc-linu
x-gnu/usr --with-mpc=/scratch/nathan/arm-lite/obj/host-libs-2010.09-50-arm-none-linux-gnue
abi-i686-pc-linux-gnu/usr --with-ppl=/scratch/nathan/arm-lite/obj/host-libs-2010.09-50-arm
-none-linux-gnueabi-i686-pc-linux-gnu/usr --with-host-libstdcxx='-static-libgcc -Wl,-Bstat
ic,-lstdc++,-Bdynamic -lm' --with-cloog=/scratch/nathan/arm-lite/obj/host-libs-2010.09-50-
arm-none-linux-gnueabi-i686-pc-linux-gnu/usr --with-libelf=/scratch/nathan/arm-lite/obj/ho
st-libs-2010.09-50-arm-none-linux-gnueabi-i686-pc-linux-gnu/usr --disable-libgomp --enable
-poison-system-directories --with-build-time-tools=/scratch/nathan/arm-lite/install/arm-no
ne-linux-gnueabi/bin --with-build-time-tools=/scratch/nathan/arm-lite/install/arm-none-lin
ux-gnueabi/bin
Thread model: posix
gcc version 4.5.1 (Sourcery G++ Lite 2010.09-50)
```

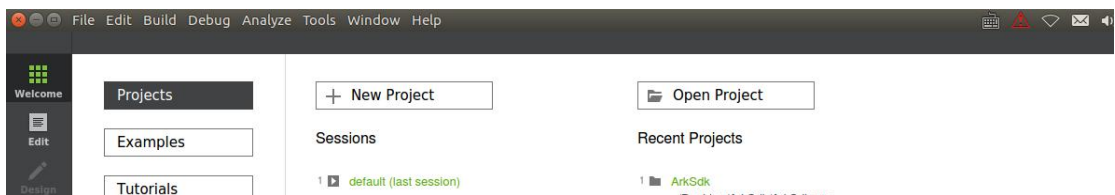
(3) 下载 Qt Creator IDE

去 Qt 官网下载 Qt Creator IDE，版本可自行根据喜好



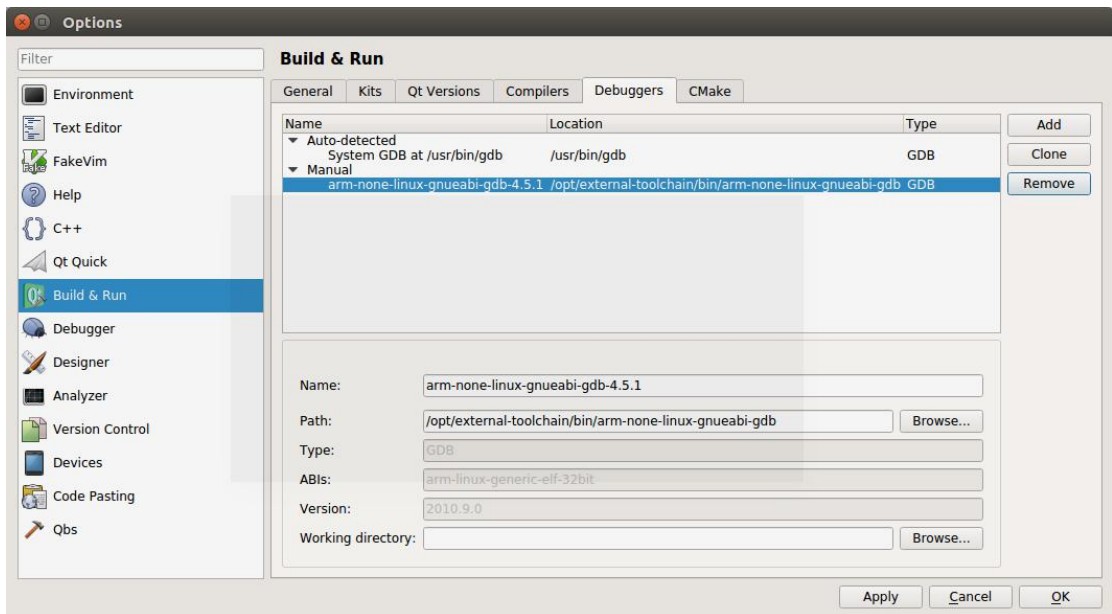
(4) 搭建 Qt 开发环境

1. 打开 Qt Creator



2. 点击菜单栏上的 Tools，再弹出的下拉列表中选择 Options

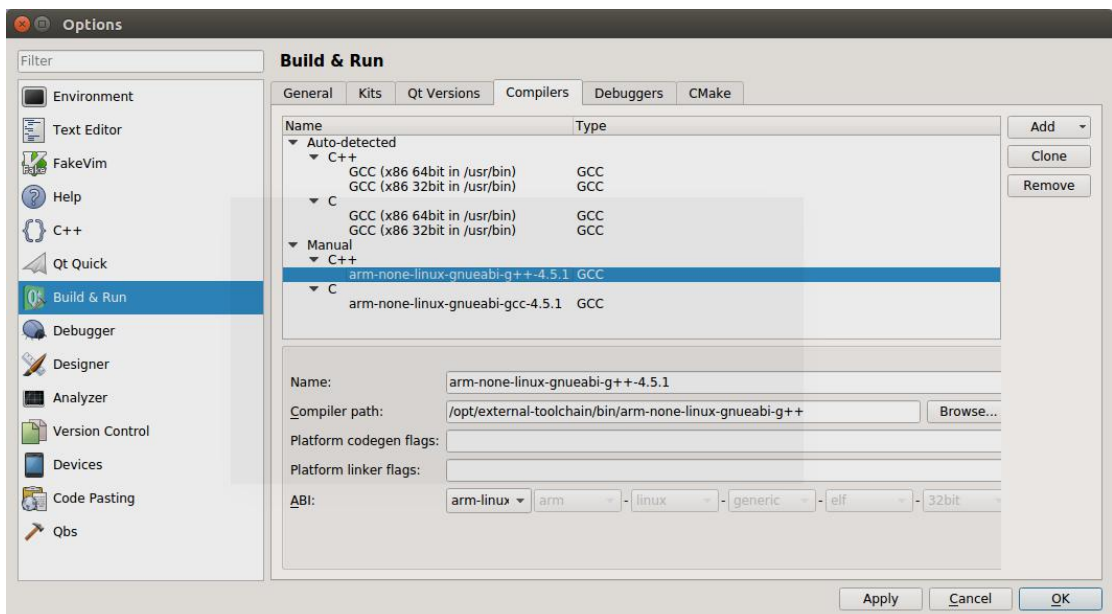
3. 在 Options 设置框中点击左侧的 Build & Run 来配置交叉编译环境



Debuggers: Add -> Browse... -> 选择路径 -> 更改名字 -> Apply

路径: /opt/external-toolchain/bin/arm-none-linux-gnueabi-gdb

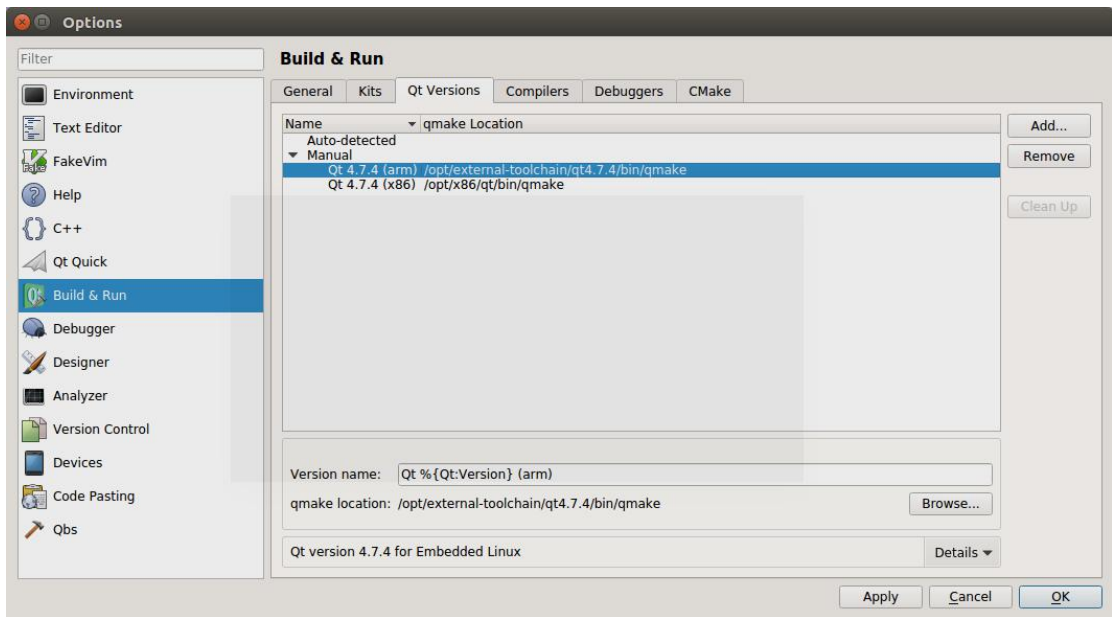
名字: arm-none-linux-gnueabi-gdb-4.5.1



Compilers: Add -> Gcc -> Browse... -> 选择路径 -> 更改名字 -> Apply

路径:/opt/external-toolchain/bin/arm-none-linux-gnueabi-g++

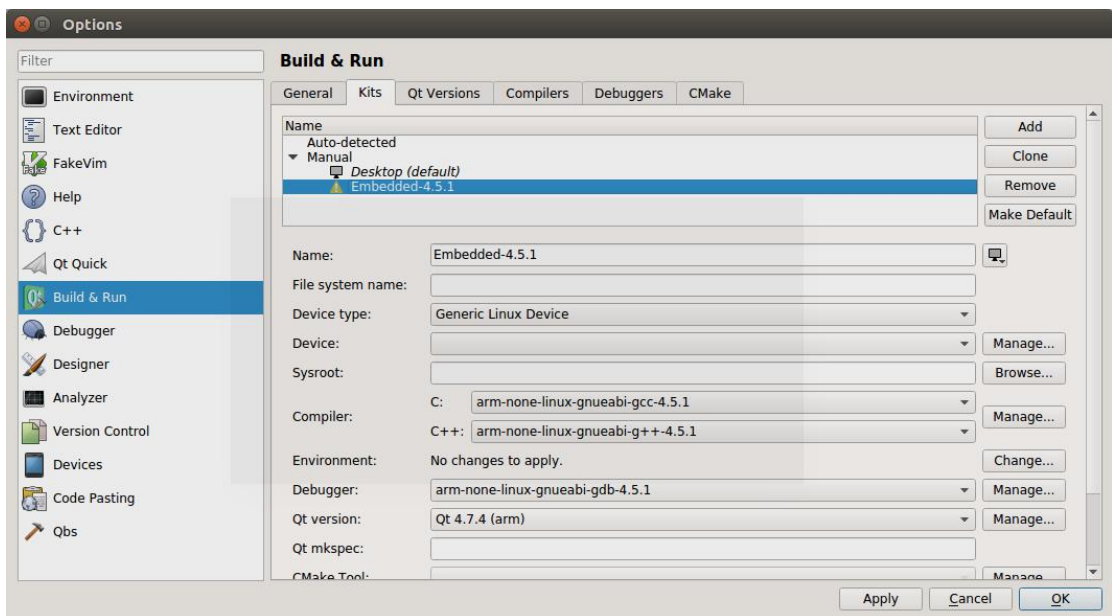
名字:arm-none-linux-gnueabi-g++-4.5.1



Qt Versions: Add -> 选择路径 -> 更改名字 -> Apply

路径: /opt/external-toolchain/qt4.7.4/bin/qmake

名字: Qt % {Qt:Version} (arm)



Kits: Add -> 更改名字 -> 更改 Device Type -> 更改 C/C++ 编译器 -> 选择

Debugger -> 选择 Qt Version

名字: Embedded-4.5.1

Device Type: Generic Linux Device

C 编译器: arm-none-linux-gnueabi-gcc-4.5.1

C++ 编译器: arm-none-linux-gnueabi-g++-4.5.1

Debugger 调试器: arm-none-linux-gnueabi-gdb-4.5.1

Qt version: Qt 4.7.4(arm)

配置好后点击 Ok 保存。

注: 以上只是参考, 名字可自行定, 虽然配置了调试器, 但这边目前还不支持在线调试。

(5)运行测试程序

1.回到 Qt Creator 的 Welcome, 点击 Project s -> New Project -> Choose... ->

Next -> 选择之前建立的, 例如这边的是 Embedded-4.5.1 -> next -> next -> Finish.

2.点击左下角的小斧头编译按钮

3.编译完成后, 将编译生成的可执行文件拷贝到 U 盘

4.将 U 盘挂在到板子上, 进入挂在目录, 执行编译出来的可执行文件./untitled -qws.