

ARK1668E 搭建 Qt 交叉编译环境详细介绍

目录

ARK1668E 搭建 Qt 交叉编译环境详细介绍	1
第一章 通过 Buildroot 下载编译 Qt5.15.2	2
1.1 配置 Buildroot 选中 Qt5	2
1.2 编译 BSP 源码	3
第二章 编译 BSP 生成交叉编译器	4
2.1 编译生成交叉编译器 SDK	4
2.2 解压交叉编译器 SDK	4
2.3 配置交叉编译器 SDK	5
第三章 配置 QCreator 的构建套件 Kit	5
3.1 下载 PC 版 Linux Qt	5
3.2 配置交叉编译构建套件	6
第四章 新建 Qt 测试工程	9
4.1 新建 Qt 工程	9
4.2 配置工程	9
4.3 添加一个测试界面	11
第五章 在 Arm 平台运行 Qt 编译生成的应用	13
5.1 拷贝测试文件到 TF/USB	13
5.2 在开发板上运行测试文件	13
第六章 通过 Buildroot 裁剪 Qt 模块	15
6.1 重新配置 Buildroot 的 Qt 模块	15
6.2 重新编译 buildroot	16

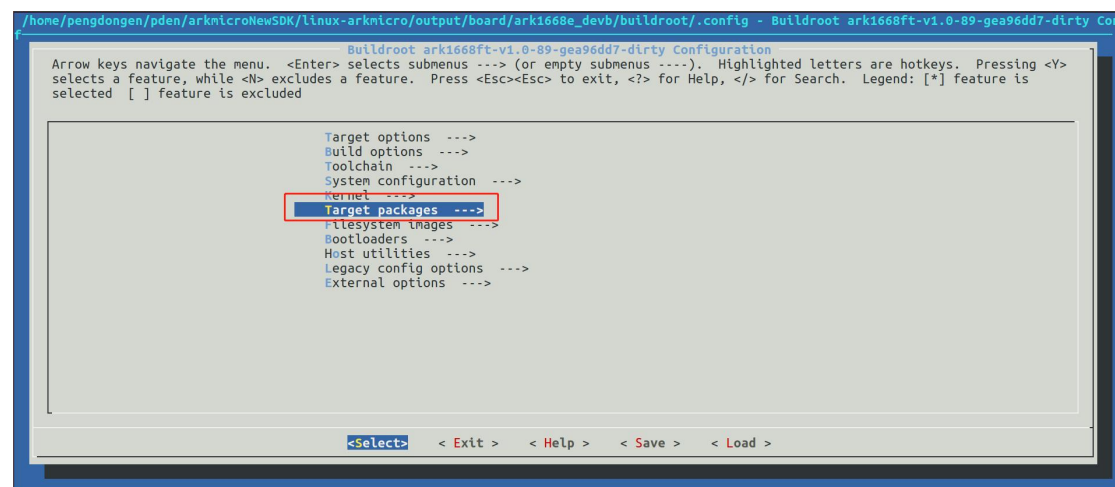
第一章 通过 Buildroot 下载编译 Qt5.15.2

1.1 配置 Buildroot 选中 Qt5

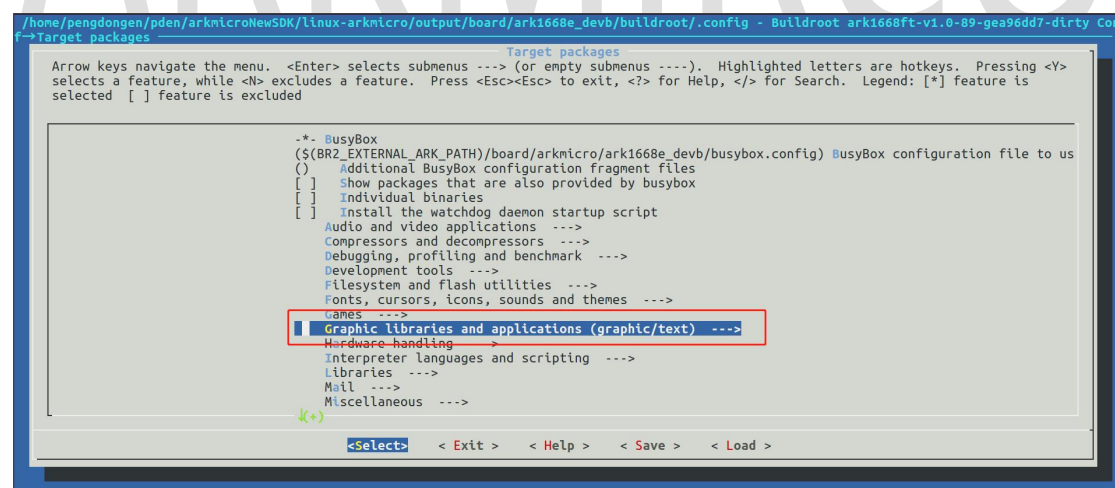
在 linux-arkmicro/build 下执行下图中的指令配置 Buildroot

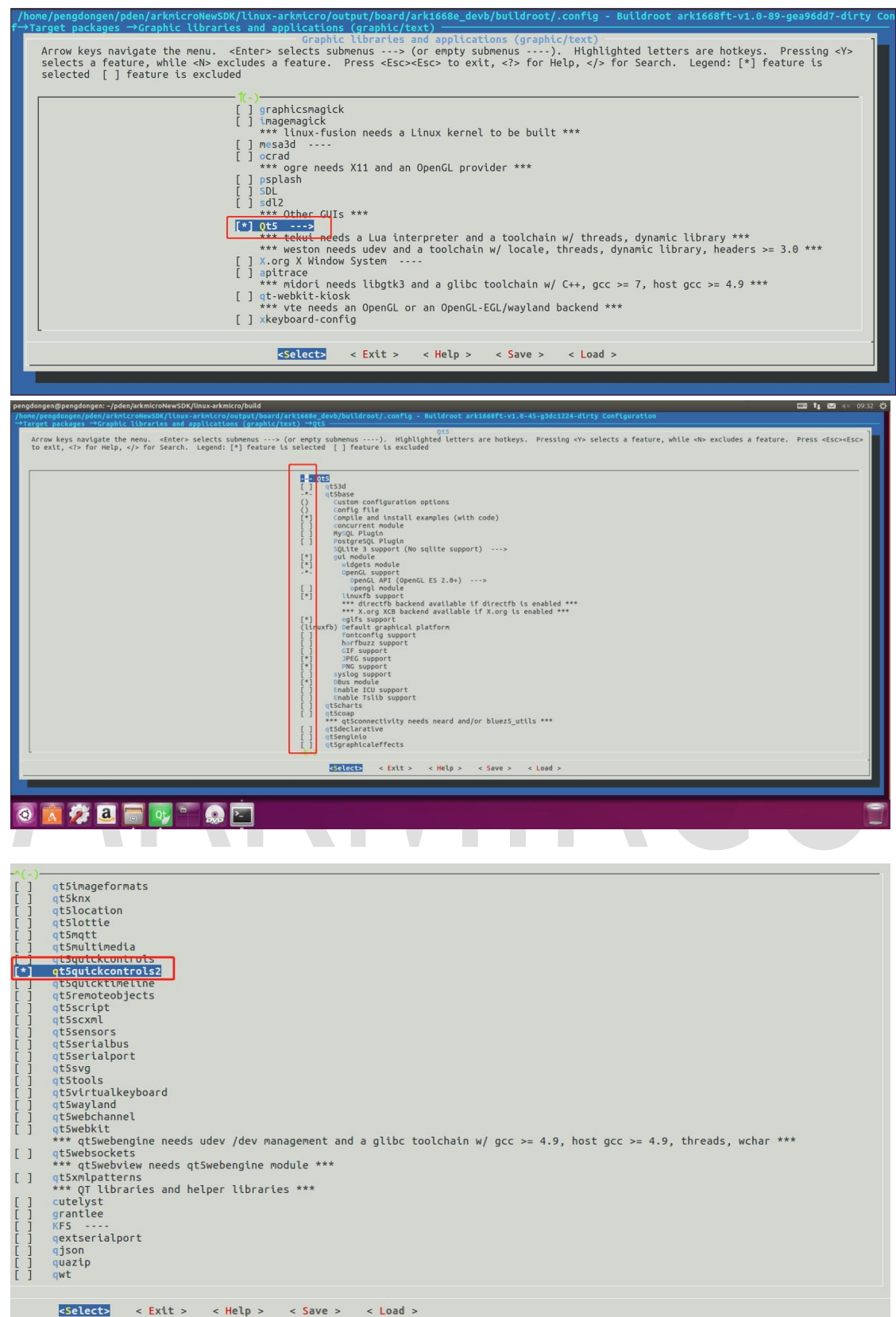
```
pengdongen@pengdongen:~/pden/arkmicroNewSDK/linux-arkmicro/build$ ./build.sh ark1668e_devb config buildroot
config buildroot...
# GEN /home/pengdongen/pden/arkmicroNewSDK/linux-arkmicro/output/board/ark1668e_devb/buildroot/Makefile
# configuration written to /home/pengdongen/pden/arkmicroNewSDK/linux-arkmicro/output/board/ark1668e_devb/buildroot/.config
# GEN /home/pengdongen/pden/arkmicroNewSDK/linux-arkmicro/output/board/ark1668e_devb/buildroot/Makefile
```

在界面中选择 Target Packages 选项



依次选择下图中的选项





1.2 编译 BSP 源码

此时，如果是第一次编译 BSP 就整体编译

```

pengdongen@pengdongen:~/pden/arkmicroNewSDK/linux-arkmicro/build$ ./build.sh ark1668e_devb buildroot^C
pengdongen@pengdongen:~/pden/arkmicroNewSDK/linux-arkmicro/build$
pengdongen@pengdongen:~/pden/arkmicroNewSDK/linux-arkmicro/build$
pengdongen@pengdongen:~/pden/arkmicroNewSDK/linux-arkmicro/build$
pengdongen@pengdongen:~/pden/arkmicroNewSDK/linux-arkmicro/build$
pengdongen@pengdongen:~/pden/arkmicroNewSDK/linux-arkmicro/build$
pengdongen@pengdongen:~/pden/arkmicroNewSDK/linux-arkmicro/build$
pengdongen@pengdongen:~/pden/arkmicroNewSDK/linux-arkmicro/build$

```

如果已经执行过上图命令，就只需要单独编译 Buildroot

```

GEN /home/pengdongen/pden/arkmicroNewSDK/linux-arkmicro/output/board/ark1668e_devb/buildroot/Makefile
pengdongen@pengdongen:~/pden/arkmicroNewSDK/linux-arkmicro/build$ ./build.sh ark1668e_devb build buildroot^C
pengdongen@pengdongen:~/pden/arkmicroNewSDK/linux-arkmicro/build$
pengdongen@pengdongen:~/pden/arkmicroNewSDK/linux-arkmicro/build$ ./build.sh ark1668e_devb build buildroot^C
pengdongen@pengdongen:~/pden/arkmicroNewSDK/linux-arkmicro/build$
pengdongen@pengdongen:~/pden/arkmicroNewSDK/linux-arkmicro/build$
pengdongen@pengdongen:~/pden/arkmicroNewSDK/linux-arkmicro/build$

```

第二章 编译 BSP 生成交叉编译器

2.1 编译生成交叉编译器 SDK

在 BSP 源码（linux-arkmicro）编译后生成的 output 目录下编译生成 SDK。

```

pengdongen@pengdongen:~/pden/arkmicroNewSDK/linux-arkmicro$ ls
bootstrap build buildroot-2021.02.2 buildroot-external doc env.source linux output tools u-boot
pengdongen@pengdongen:~/pden/arkmicroNewSDK/linux-arkmicro$ cd output/board/ark1668e_devb/buildroot/
pengdongen@pengdongen:~/pden/arkmicroNewSDK/linux-arkmicro/output/board/ark1668e_devb/buildroot$
pengdongen@pengdongen:~/pden/arkmicroNewSDK/linux-arkmicro/output/board/ark1668e_devb/buildroot$ source ../../../../env.source
pengdongen@pengdongen:~/pden/arkmicroNewSDK/linux-arkmicro/output/board/ark1668e_devb/buildroot$ make sdk
pengdongen@pengdongen:~/pden/arkmicroNewSDK/linux-arkmicro/output/board/ark1668e_devb/buildroot$
>>> Finalizing host directory
>>> Finalizing target directory
mkdir -p /home/pengdongen/pden/arkmicroNewSDK/linux-arkmicro/output/board/ark1668e_devb/buildroot/host/etc/meson
sed -e 's%@TARGET_CROSS%#/home/pengdongen/pden/arkmicroNewSDK/linux-arkmicro/output/board/ark1668e_devb/buildroot/host/bin/arm-
linux-gnueabihf-%g' -e 's%@TARGET_ARCH%#arm%g' -e 's%@TARGET_CPU%#cortex-a7%g' -e 's%@TARGET_ENDIAN%#little%g' -e 's%@TARGET_C%

```

2.2 解压交叉编译器 SDK

编译完成后，SDK 压缩文件（arm-buildroot-linux-gnueabihf_sdk-buildroot.tar.gz）在当前目录的 images 目录下。然后，拷贝生成的 SDK 压缩文件到虚拟机的任意目录下。这里选择拷贝到/home/pengdongen/pden/arkmicroNewSDK/目录。

```

pengdongen@pengdongen:~/pden/arkmicroNewSDK/output/board/ark1668e_devb/buildroot/images$
pengdongen@pengdongen:~/pden/arkmicroNewSDK/output/board/ark1668e_devb/buildroot/images$
pengdongen@pengdongen:~/pden/arkmicroNewSDK/output/board/ark1668e_devb/buildroot/images$ ls
arm-buildroot-linux-gnueabihf_sdk-buildroot.tar.gz  rootfs.tar  rootfs.ubi  rootfs.ubifs
pengdongen@pengdongen:~/pden/arkmicroNewSDK/output/board/ark1668e_devb/buildroot/images$ cp arm-buildroot-linux-
gnueabihf_sdk-buildroot.tar.gz /home/pengdongen/pden/arkmicroNewSDK/
pengdongen@pengdongen:~/pden/arkmicroNewSDK/output/board/ark1668e_devb/buildroot/images$
pengdongen@pengdongen:~/pden/arkmicroNewSDK/output/board/ark1668e_devb/buildroot/images$ 2.
pengdongen@pengdongen:~/pden/arkmicroNewSDK/output/board/ark1668e_devb/buildroot/images$ cd /home/pengdongen/pd
en/arkmicroNewSDK/
pengdongen@pengdongen:~/pden/arkmicroNewSDK$
pengdongen@pengdongen:~/pden/arkmicroNewSDK$ 3.
pengdongen@pengdongen:~/pden/arkmicroNewSDK$ tar xvf arm-buildroot-linux-gnueabihf_sdk-buildroot.tar.gz

```

2.3 配置交叉编译器 SDK

解压后生成目录（arm-buildroot-linux-gnueabihf_sdk-buildroot），然后在这个目录下执行脚本 relocate-sdk.sh

```

pengdongen@pengdongen: ~/pden/arkmicroNewSDK/arm-buildroot-linux-gnueabihf_sdk-buildroot
pengdongen@pengdongen:~/pden/arkmicroNewSDK$ ls
arkapp arm-buildroot-linux-gnueabihf_sdk-buildroot arm-buildroot-linux-gnueabihf_sdk-buildroot.tar.gz linux-arkmicro
pengdongen@pengdongen:~/pden/arkmicroNewSDK$ cd arm-buildroot-linux-gnueabihf_sdk-buildroot/
pengdongen@pengdongen:~/pden/arkmicroNewSDK/arm-buildroot-linux-gnueabihf_sdk-buildroot$ ls
arm-buildroot-linux-gnueabihf bin doc etc include lib lib64 libexec mkspecs opt relocate-sdk.sh sbin share usr
pengdongen@pengdongen:~/pden/arkmicroNewSDK/arm-buildroot-linux-gnueabihf_sdk-buildroot$ ./relocate-sdk.sh
Relocating the buildroot SDK from /home/pengdongen/pden/arkmicroNewSDK/linux-arkmicro/output/board/ark1668e_devb/buildroot/host
to /home/pengdongen/pden/arkmicroNewSDK/arm-buildroot-linux-gnueabihf_sdk-buildroot ...

```

qmake、GCC 和 G++编译器在当前目录下的 bin 目录下，在此要留意这三个文件的路径，后面 QtCreator 配置构建套件会用到这 3 个工具。如下图：

```

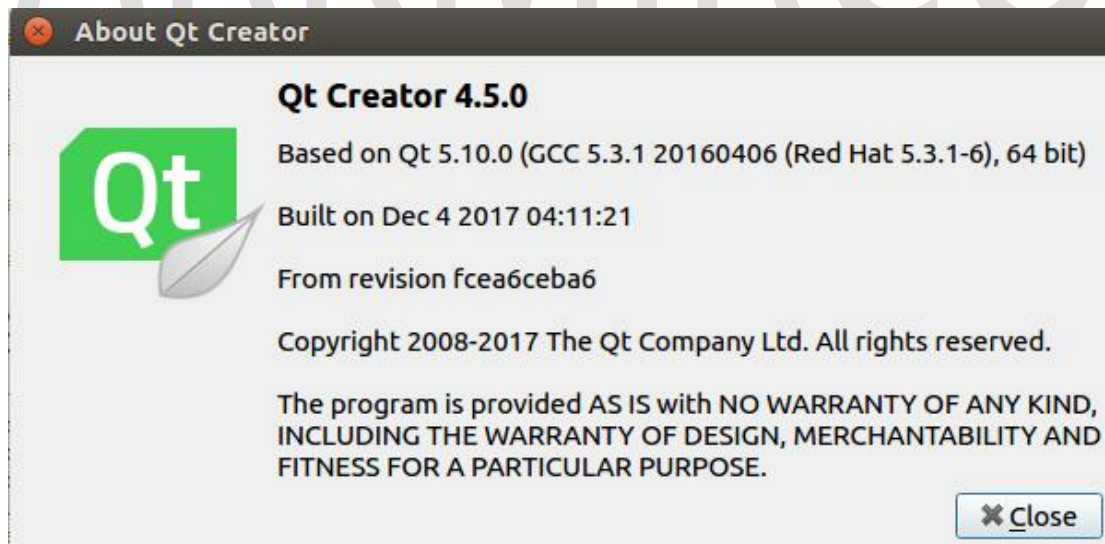
arm-linux-gnueabihf-gcc arm-linux-gnueabihf-g++ arm-linux-gnueabihf-qmake
pengdongen@pengdongen:~/pden/arkmicroNewSDK/arm-buildroot-linux-gnueabihf_sdk-buildroot/bin$ ls
qmake arm-linux-gnueabihf-gcc arm-linux-gnueabihf-g++
arm-linux-gnueabihf-g++ arm-linux-gnueabihf-gcc qmake
pengdongen@pengdongen:~/pden/arkmicroNewSDK/arm-buildroot-linux-gnueabihf_sdk-buildroot/bin$

```

第三章 配置 QtCreator 的构建套件 Kit

3.1 下载 PC 版 Linux Qt

从官网下载 pc 版本的 Linux Qt，只要是 Qt5 以上版本都可以，这里下载的是：
qt-opensource-linux-x64-5.10.0.run

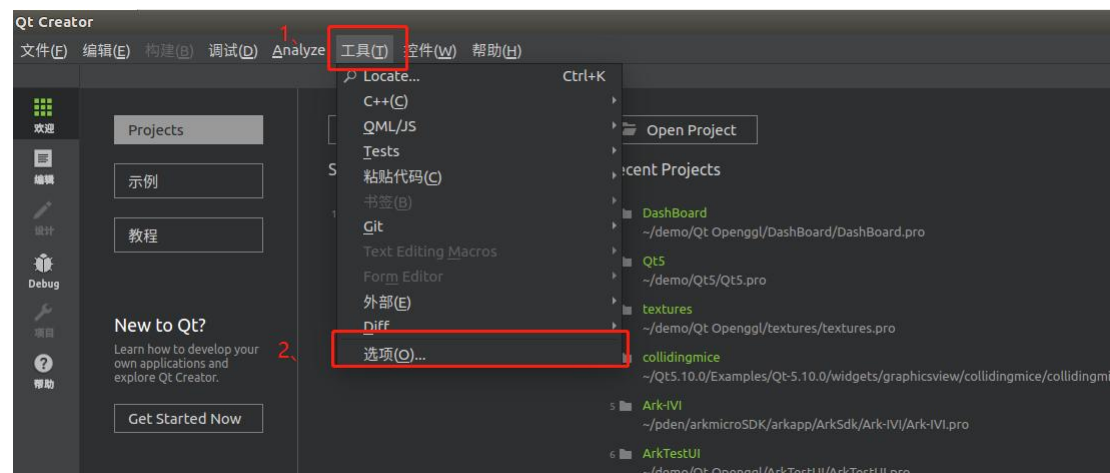


官方网址：http://download.qt.io/official_releases/qt/

下载完成后直接运行 ./qt-opensource-linux-x64-5.10.0.run 全选傻瓜式安装即可！安装完成后接下来就是配置 qt 交叉编译环境了，如下：

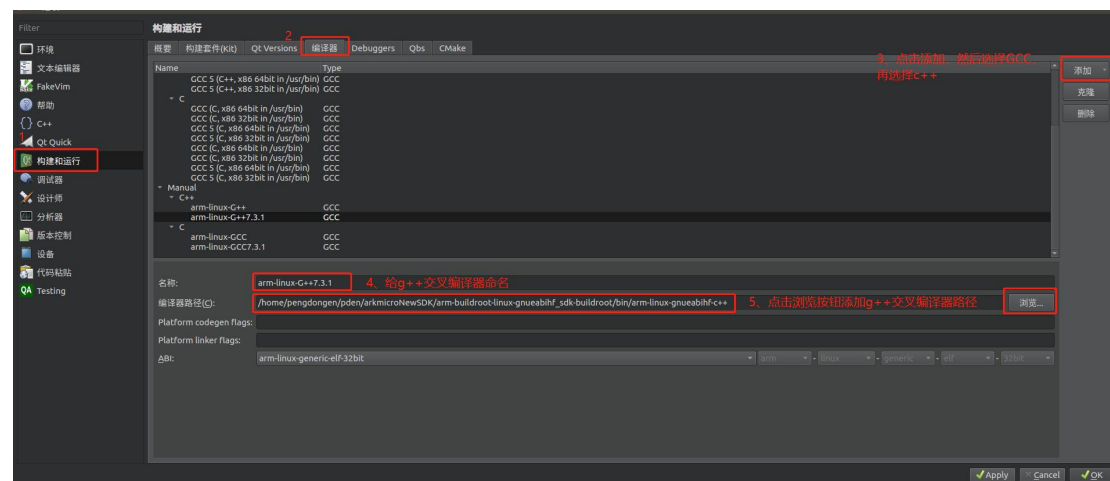
3.2 配置交叉编译构建套件

打开 QtCreator 的工具选项配置界面,步骤如下



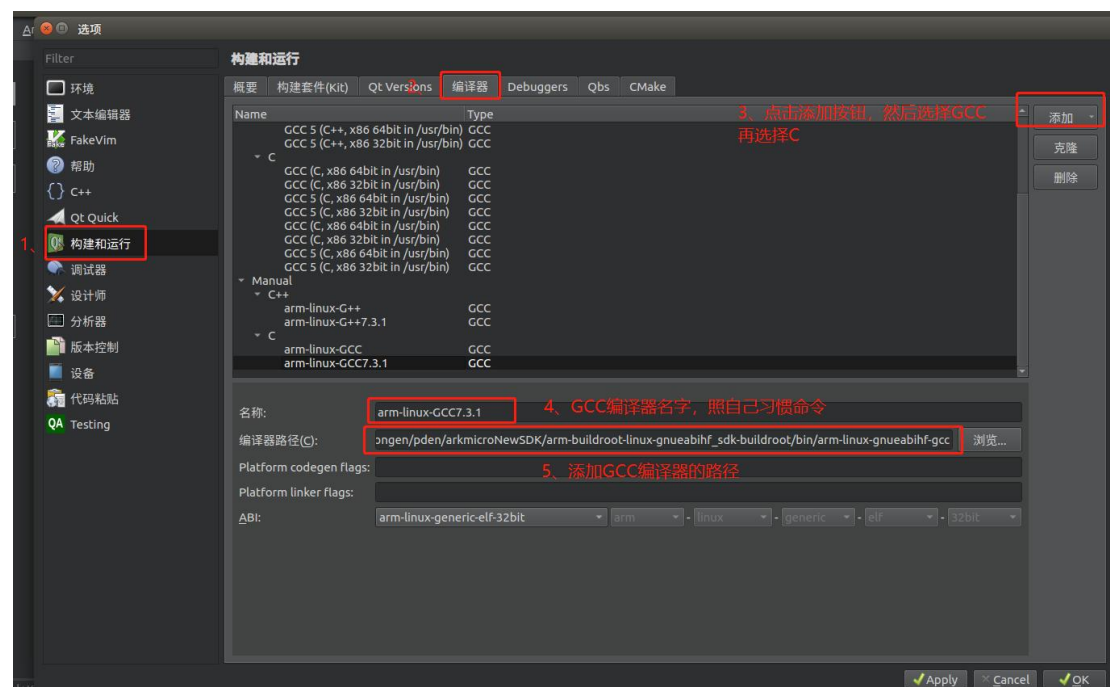
3.2.1 添加 G++交叉编译器

配置如下图:

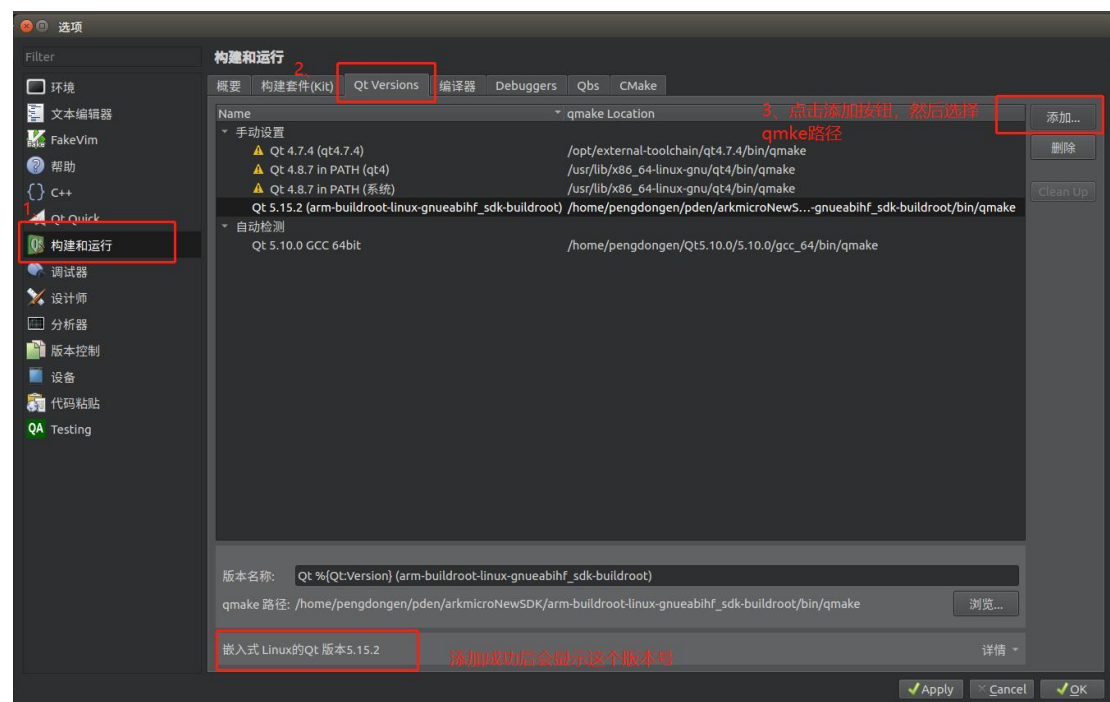


3.2.2 添加 GCC 交叉编译器

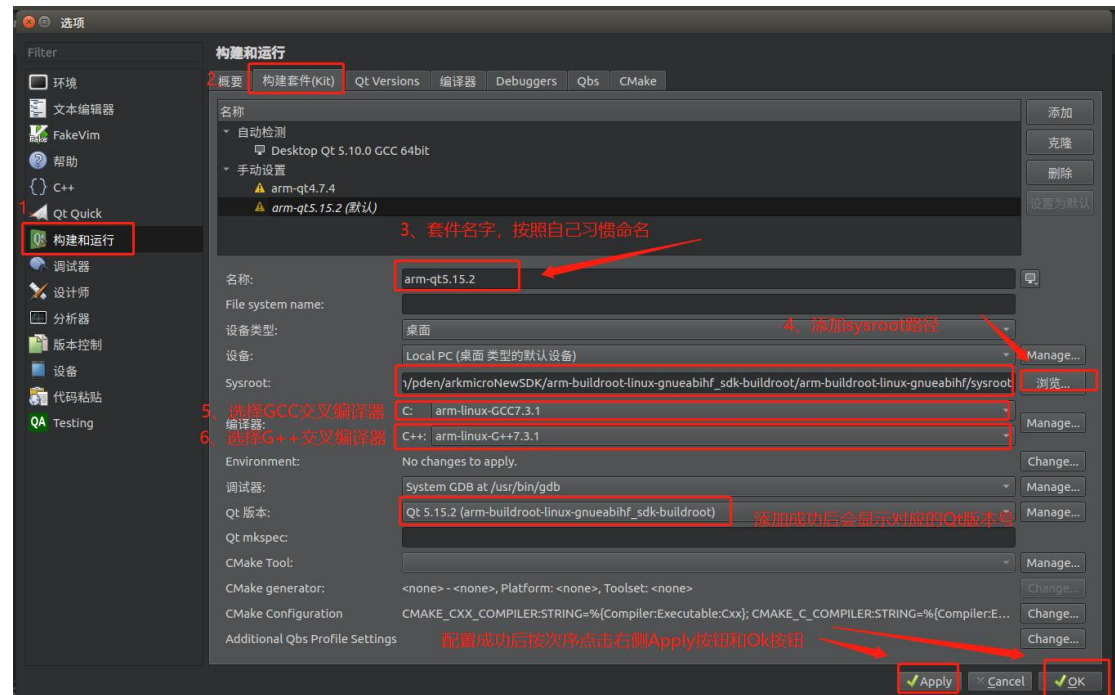
配置如下:



3.3.3 添加 qmake



3.3.4 配置构建套件



在此要注意 sysroot 路径是第一章中 1.2 步骤解压后生成的目录下。如下图:

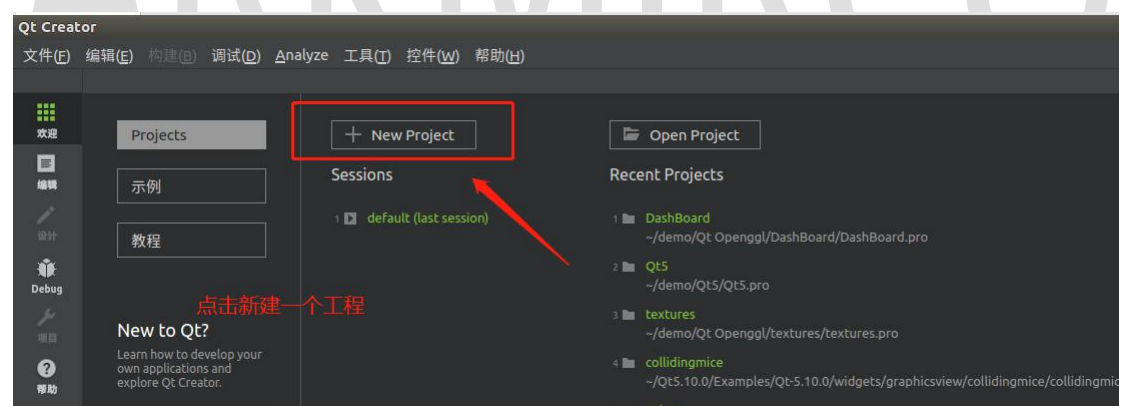
```
pengdongen@pengdongen:~/pden/arkmicroNewSDK/arm-buildroot-linux-gnueabihf_sdk-buildroot/arm-buildroot-linux-gnueabihf$ ls
sysroot
pengdongen@pengdongen:~/pden/arkmicroNewSDK/arm-buildroot-linux-gnueabihf_sdk-buildroot/arm-buildroot-linux-gnueabihf$ pwd
/home/pengdongen/pden/arkmicroNewSDK/arm-buildroot-linux-gnueabihf_sdk-buildroot/arm-buildroot-linux-gnueabihf
pengdongen@pengdongen:~/pden/arkmicroNewSDK/arm-buildroot-linux-gnueabihf_sdk-buildroot/arm-buildroot-linux-gnueabihf$
```

到此, Qt 交叉编译环境搭建完成!

第四章 新建 Qt 测试工程

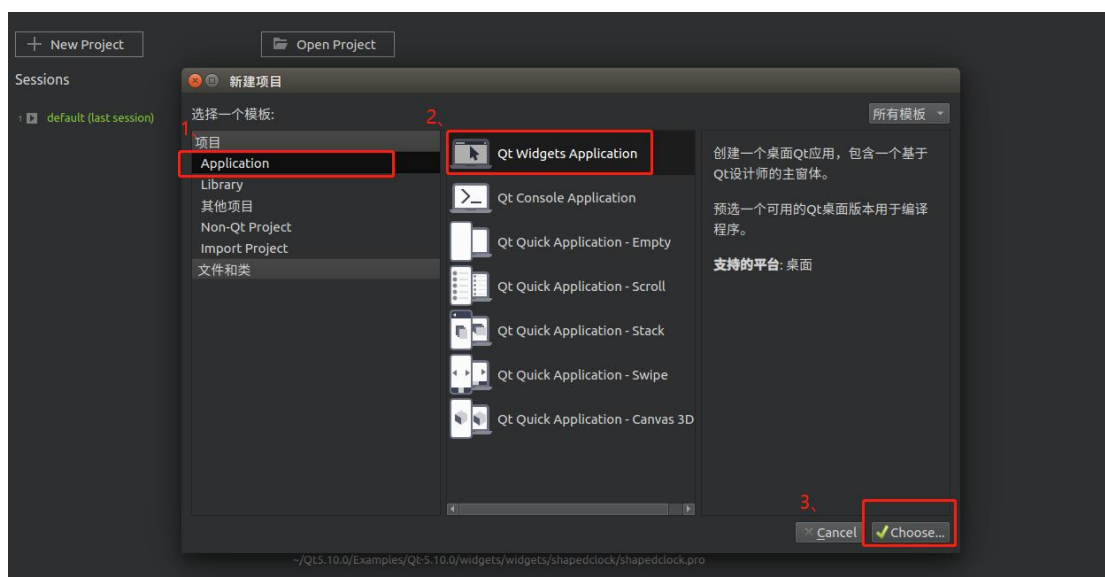
4.1 新建 Qt 工程

点击 New Project 按钮新建一个工程



4.2 配置工程

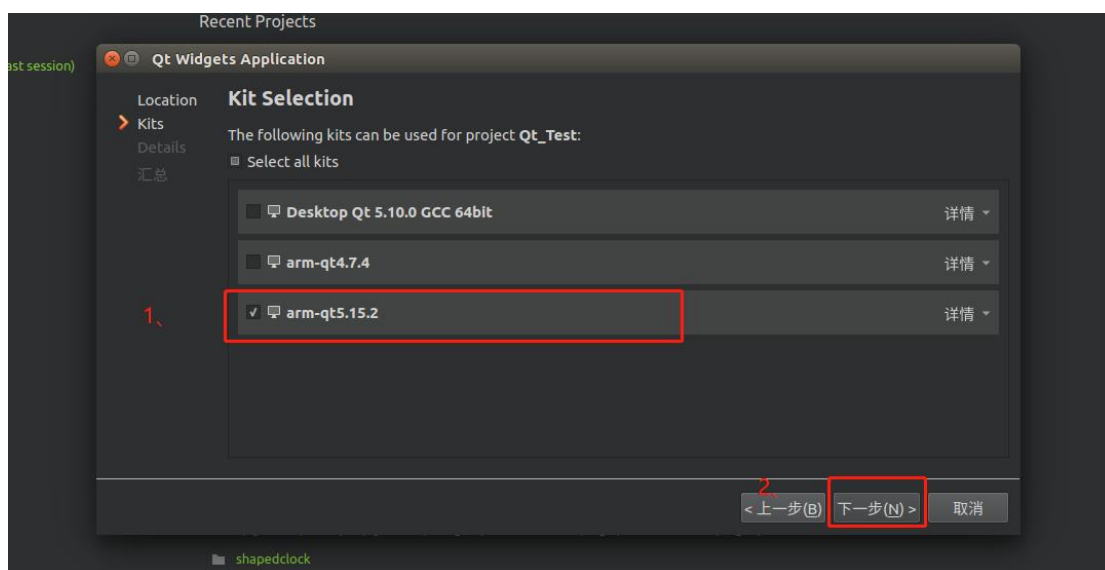
这里选择新建一个 QWidget 的应用



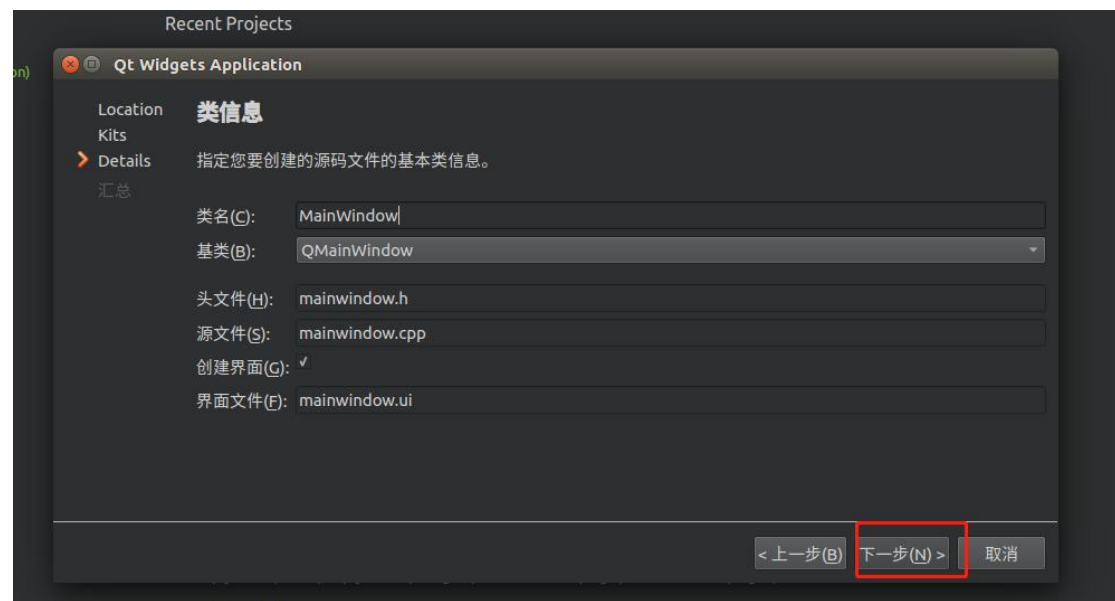
然后设置工程名字和工程路径。如下图：



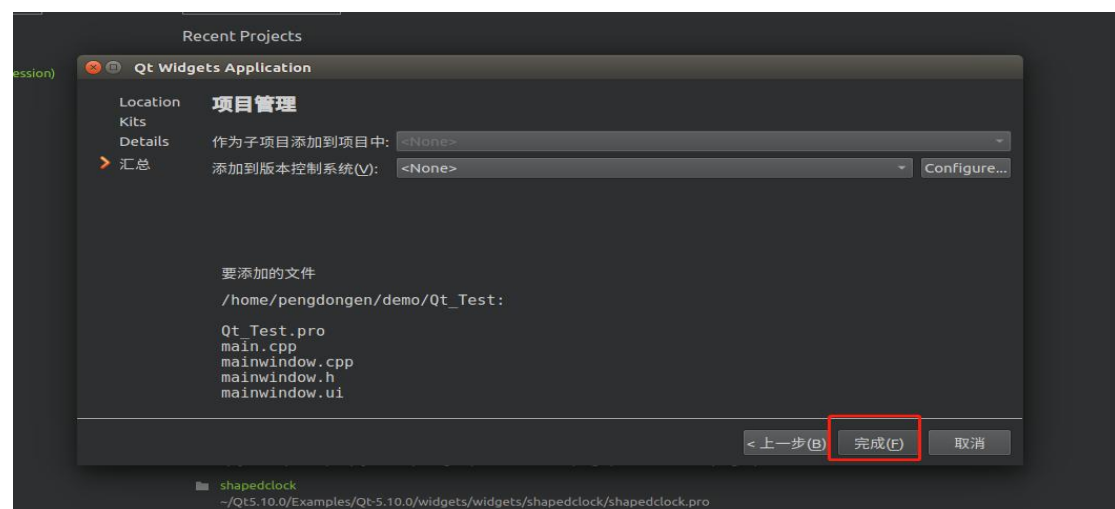
选择构建套件为上面配置的交叉编译器套件



下面直接点击下一步

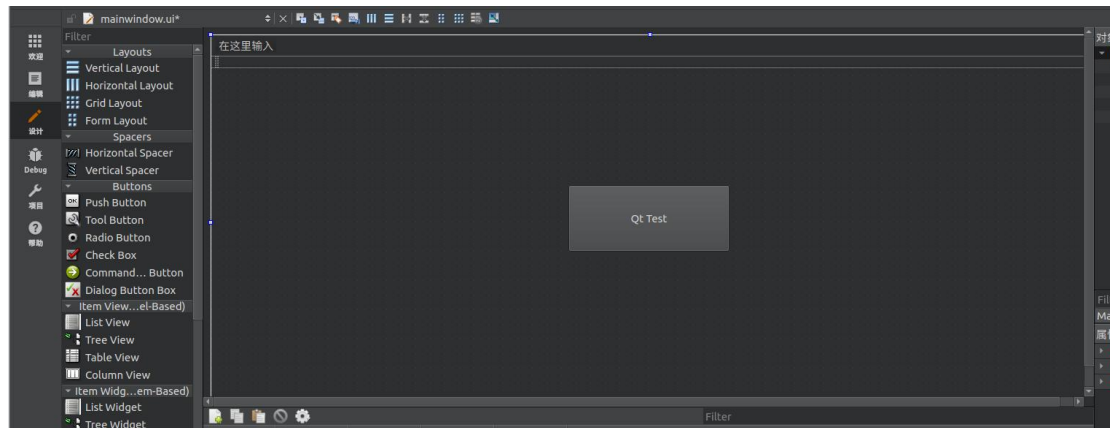


点击完成

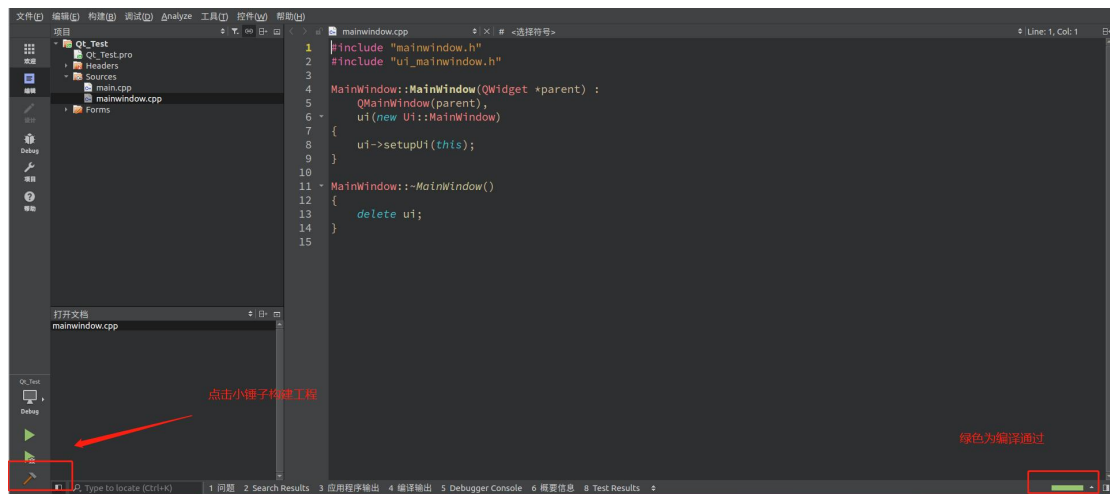


4.3 添加一个测试界面

可以随便设计一个界面，我这里如下



3.4 编译工程



若有报错提示没有安装 LGL，解决如下：

这是由于 Qt5.0 以上的版本默认将 OpenGL 加入了工程，但是在机器上没有安装 OpenGL，所以只需要在机器系统安装 OpenGL 即可

`sudo apt-get install libgl1-mesa-dev`

如果你不知道当前 Linux 系统中 libGL.so 的具体路径，可以使用 `locate libGL` 命令或 `find /usr -name libGL*` 命令查找，然后使用 `ln -s` 创建链接。请看下面的演示：

```
#查找 libGL 所在位置
[root@localhost ~]# locate libGL
/usr/lib64/libGL.so
/usr/lib64/libGL.so.1
/usr/lib64/libGL.so.1.2.0
/usr/share/doc/mesa-libGL-9.2.5
/usr/share/doc/mesa-libGL-9.2.5/COPYING

#创建链接
[root@localhost ~]# ln -s /usr/lib64/libGL.so.1 /usr/lib/libGL.so
```

Linux 系统中可能存在多个版本的 libGL.so，为任意一个版本创建链接即可。普通用户没有权限创建链接，所以我使用了 root 用户。

完成以上操作，再次启动 Qt，然后编译或者运行，就不会出现“cannot find -lGL”错误了。

第五章 在 Arm 平台运行 Qt 编译生成的应用

5.1 拷贝测试文件到 TF/USB

切换到 Qt_Test 工程目录，拷贝编译生成的应用 Qt_Test 到 TF/USB

```
pengdongen@pengdongen:~/demo/build-Qt_Test-arm_qt5_15_2-Debug$ ls
main.o  mainwindow.o  Makefile  moc_mainwindow.cpp  moc_mainwindow.o  moc_predefs.h  Qt_Test  ui_mainwindow.h
pengdongen@pengdongen:~/demo/build-Qt_Test-arm_qt5_15_2-Debug$ pwd
/home/pengdongen/demo/build-Qt_Test-arm_qt5_15_2-Debug
pengdongen@pengdongen:~/demo/build-Qt_Test-arm_qt5_15_2-Debug$
pengdongen@pengdongen:~/demo/build-Qt_Test-arm_qt5_15_2-Debug$ cp Qt_Test /media/pengdongen/PENGDE/
```

拷贝编译生成的Qt应用到tf卡

5.2 在开发板上运行测试文件

在开发板上运行 Qt_Test 应用，如果显示出你设计的界面的样式，则证明交叉编译环境搭建成功。

```
#
# mount /dev/mmcblk1 /media/ 1. 挂载卡到Arm
FAT-fs (mmcblk1): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.
# ./media/Qt_Test 2. 运行Qt应用
Unable to query physical screen size, defaulting to 100 dpi.
To override, set QT_OPA_EGLFS_PHYSICAL_WIDTH and QT_OPA_EGLFS_PHYSICAL_HEIGHT (in millimeters).
eudevtouch: cannot open input device /dev/input/event0 (No such file or directory)
Created context for format QSurfaceFormat(version 2.0, options QFlags<QSurfaceFormat::FormatOption>(), depthBufferSize 24, redBufferSize 5, greenBufferS
ize 0, stencilBufferSize 8, samples 0, swapBehavior QSurfaceFormat::DefaultSwapBehavior, swapInterval 1, colorSpace QSurfaceFormat::DefaultColorSpace, p
th config:
    EGL_BUFFER_SIZE: 16
    EGL_ALPHA_SIZE: 0
    EGL_BLUE_SIZE: 5
    EGL_GREEN_SIZE: 6
    EGL_RED_SIZE: 5
    EGL_DEPTH_SIZE: 24
    EGL_STENCIL_SIZE: 8
    EGL_CONFIG_CAVEAT: 12344
```

ARKMIRCO

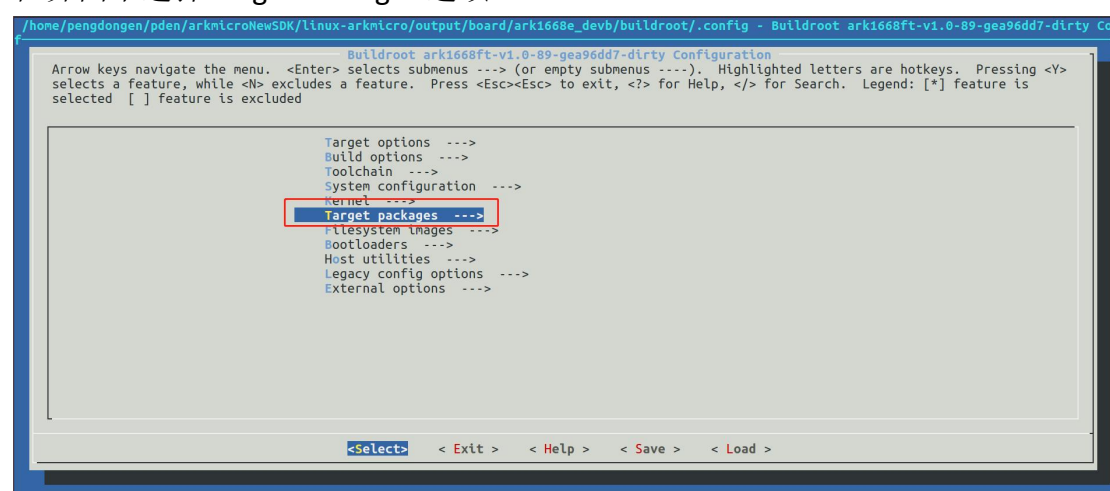
第六章 通过 Buildroot 裁剪 Qt 模块

6.1 重新配置 Buildroot 的 Qt 模块

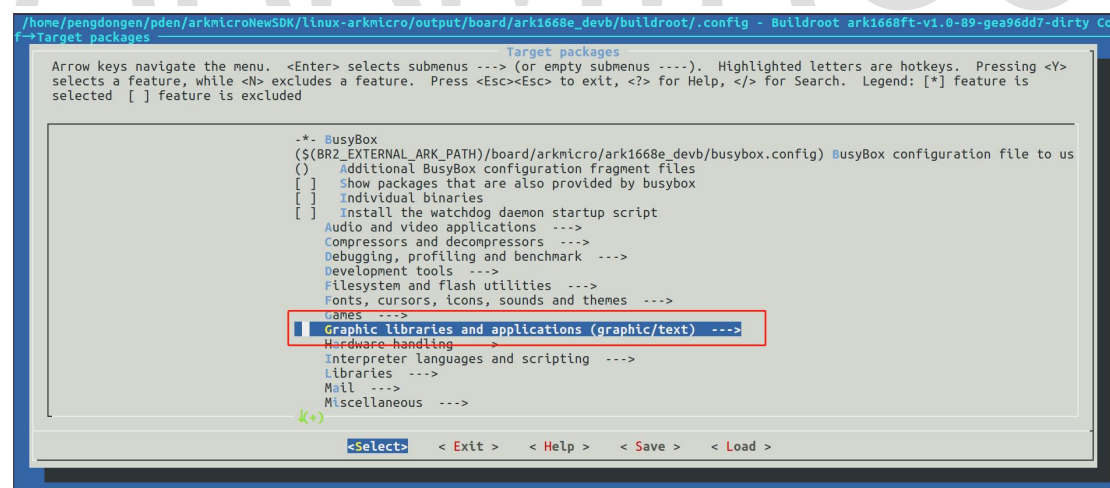
在 BSP 的 build 目录下打开终端执行下图中的指令配置 buildroot

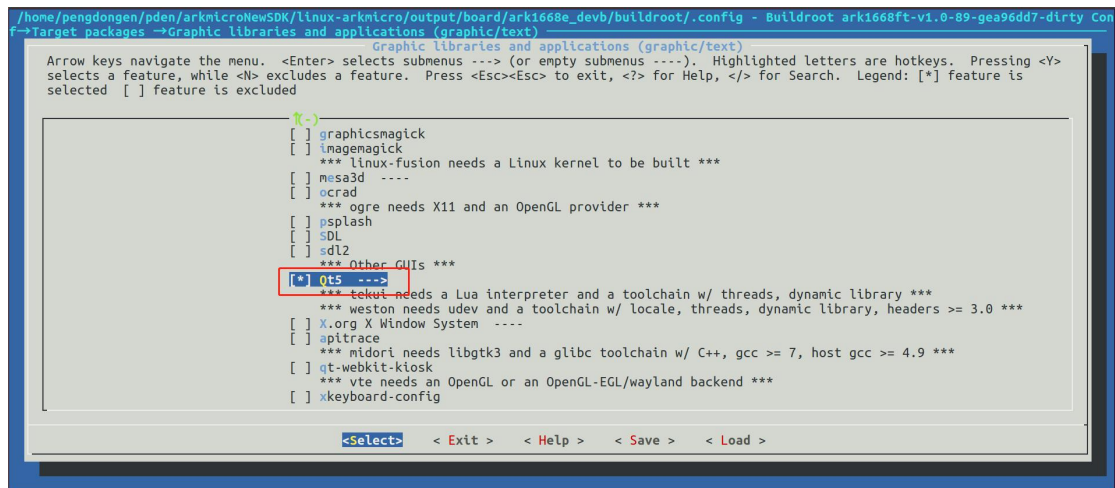
```
pengdongen@pengdongen:~/pden/arkmicroNewSDK/linux-arkmicro/build$ ./build.sh ark1668e_devb config buildroot
config buildroot...
# GEN      /home/pengdongen/pden/arkmicroNewSDK/linux-arkmicro/output/board/ark1668e_devb/buildroot/Makefile
# configuration written to /home/pengdongen/pden/arkmicroNewSDK/linux-arkmicro/output/board/ark1668e_devb/buildroot/.config
# GEN      /home/pengdongen/pden/arkmicroNewSDK/linux-arkmicro/output/board/ark1668e_devb/buildroot/Makefile
```

在界面中选择 Target Packages 选项

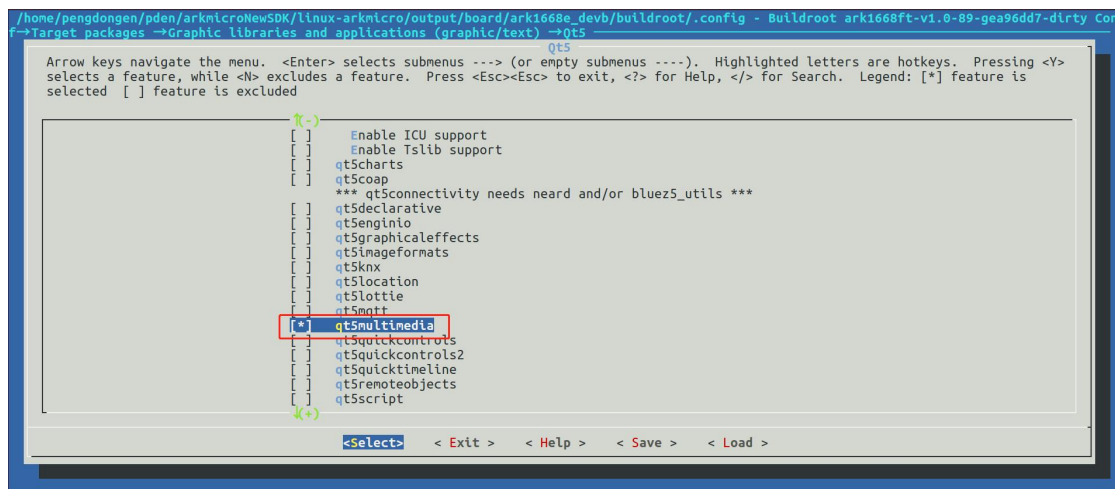


依次选择下图中的选项





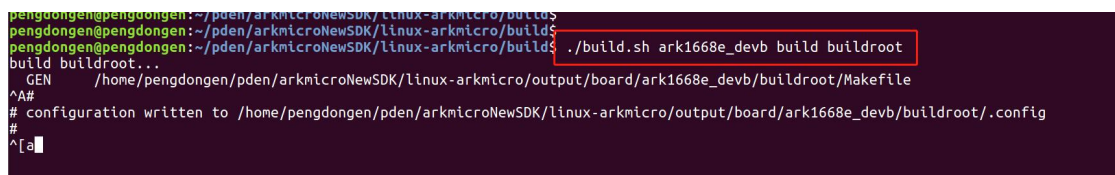
在 Qt5 选项中选择想要裁剪的 Qt 模块，这里选择添加多媒体模块，如下图：



添加完成后保存配置退出。

6.2 重新编译 buildroot

在 BSP 的 build 目录下执行如下图指令编译 buildroot



编译完成后拷贝升级文件到 SD 卡后给平台重新升级。