



PET-R16 开发板/全功能板 开发手册

一、编译环境搭建指南

- 安装 Ubuntu 14.04 64 位。
- 安装 openjdk-7-jdk。

```
sudo apt-get install openjdk-7-jdk
```

- 安装完成后修改 `/etc/profile` 中的环境变量，将 `openjdk` 加入到 `JAVA_HOME` 中。
- 输入命令 `java -version` 检查 `java` 的主版本号是否为 `1.7`。
在编译过程中可能会由于系统缺少依赖包而报错，可以通过百度或谷歌查找相关解决方案，一般通过 `sudo apt-get install <需要安装的包名>` 这个命令进行在线安装。

二、解压源代码

将源代码压缩文件全部复制到 Ubuntu 系统下，保证所在磁盘剩余空间要大于 50G，使用以下命令解压源代码：

```
tar xvJf PET_R16_6.0.1_Source.tar.xz
```

三、编译安卓 Android

首次编译请严格按照步骤进行内核、uboot、android 的编译，否则编译可能会出现错误。

1、编译内核

```
cd lichee
./build.sh -p sun8iw5p1_android
```

编译完成后正确提示如下：

```
regenerate rootfs cpio
11282 blocks
11990 blocks
build_ramfs
Copy boot.img to output directory ...
Copy modules to target ...

sun8iw5p1 compile kernel successful

INFO: build kernel OK.
INFO: build rootfs ...
INFO: skip make rootfs for android
INFO: build rootfs OK.
INFO: -----
INFO: build lichee OK.
INFO: -----
[root@SeKeDe lichee]#
```

2、编译 uboot

首次编译或修改 uboot 代码后需要执行这一步骤。

```
cd lichee/brandy
./build.sh -p sun8iw5p1
编译完成后正确提示如下
/root/work/A33_5.1.1_SKD_BASE/lichee/brandy/u-boot-2011.09/./gcc-linaro/bin/arm-linux-gnueabi-objcopy --gap-fill=0xff -O binary /root/work/A33_5.1.1_SKD_BASE/lichee/brandy/u-boot-2011.09/sunxi_spl/boot0/boot0_sdcard.axf /root/work/A33_5.1.1_SKD_BASE/lichee/brandy/u-boot-2011.09/sunxi_spl/boot0/boot0_sdcard.bin
make[1]: Leaving directory `/root/work/A33_5.1.1_SKD_BASE/lichee/brandy/u-boot-2011.09/sunxi_spl/boot0'
â€ˆboot0_nand_sun8iw5p1.binâ€™ -> â€ˆ~/root/work/A33_5.1.1_SKD_BASE/lichee/brandy/u-boot-2011.09/././tools/pack/chips/sun8iw5p1/bin/boot0_nand_sun8iw5p1.binâ€™
â€ˆboot0_sdcard_sun8iw5p1.binâ€™ -> â€ˆ~/root/work/A33_5.1.1_SKD_BASE/lichee/brandy/u-boot-2011.09/././tools/pack/chips/sun8iw5p1/bin/boot0_sdcard_sun8iw5p1.binâ€™
root@SeKeDe brandy1#
```

3、编译 android

```
cd android
source build/envsetup.sh
lunch astar_d7-eng
extract-bsp
make -j4
pack
编译完成后正确提示如下
config.fex Len: 0xa2e0
split_xxxx.fex Len: 0x200
sys_partition.fex Len: 0x1251
boot0_nand.fex Len: 0x8000
boot0_sdcard.fex Len: 0x8000
u-boot.fex Len: 0xc4000
fes1.fex Len: 0x23a0
usbtool.fex Len: 0x23000
aultools.fex Len: 0x26ead
aultls32.fex Len: 0x238dd
cardtool.fex Len: 0x41400
cardscript.fex Len: 0x6ea
sunxi_mbr.fex Len: 0x10000
dlinfo.fex Len: 0x4000
arisc.fex Len: 0x35520
boot-resource.fex Len: 0x4dc400
Vboot-resource.fex Len: 0x4
env.fex Len: 0x20000
Venv.fex Len: 0x4
boot.fex Len: 0xe75800
Vboot.fex Len: 0x4
system.fex Len: 0x1eb0c6b0
Vsystem.fex Len: 0x4
recovery.fex Len: 0xf1a000
Vrecovery.fex Len: 0x4
BuildImg 0
Dragon execute image.cfg SUCCESS !
-----image is at-----
/root/work/PET_A33_6.0.1/lichee/tools/pack/sun8iw5p1_android_d7_uart0.img
pack finish
```

编译完成后会在 lichee/tools/pack 目录下生成 sun8iw5p1_android_d7_uart0.img 系统烧写镜像文件。

四、编译 Linux + QT5.8

请首先新开一个控制台进行编译操作。

首次编译请严格按照步骤进行内核、uboot、Rootfs 的编译，否则编译可能会出现错误。

1、编译内核

```

cd lichee
./build.sh -p sun8iw5p1_dragonboard
编译完成后正确提示如下
build_ramfs
Copy boot.img to output directory ...
sun8iw5p1 compile kernel successful

INFO: build kernel OK.
INFO: build rootfs ...
Regenerating dragonboard Rootfs...
event num 4
/root/work/A33_5.1.1_SKD_BASE/lichee/buildroot/target/dragonboard/rootfs/etc/profile
generating rootfs...
blocks: 551M -> 768M
Creating filesystem with parameters:
  Size: 805306368
  Block size: 4096
  Blocks per group: 32768
  Inodes per group: 8192
  Inode size: 256
  Journal blocks: 3072
  Label:
  Blocks: 196608
  Block groups: 6
  Reserved block group size: 47
Created filesystem with 10196/49152 inodes and 147561/196608 blocks
e2fsck 1.42.12 (29-Aug-2014)
fsck.ext4: Bad magic number in super-block while trying to open rootfs.ext4
success in generating rootfs
build at: Mon May  8 17:56:23 CST 2017
INFO: build rootfs OK.
INFO: -----
INFO: build lichee OK.
INFO: -----
[root@SeKeDe lichee]#
    
```

2、编译 uboot

首次编译或修改 uboot 代码后需要执行这一步骤。

```

首先切换到 uboot 目录
cd lichee/brandy
./build.sh -p sun8iw5p1
编译完成后正确提示如下
/root/work/A33_5.1.1_SKD_BASE/lichee/brandy/u-boot-2011.09/./gcc-linaro/bin/arm-linux-gnueabi-objc
opy --gap-fill=0xff -O binary /root/work/A33_5.1.1_SKD_BASE/lichee/brandy/u-boot-2011.09/sunxi_sp1
/boot0/boot0_sdcard.axf /root/work/A33_5.1.1_SKD_BASE/lichee/brandy/u-boot-2011.09/sunxi_sp1/boot0/
boot0_sdcard.bin
make[1]: Leaving directory `./root/work/A33_5.1.1_SKD_BASE/lichee/brandy/u-boot-2011.09/sunxi_sp1/bo
ot0'
â€ˆboot0_nand_sun8iw5p1.binâ€™ -> â€ˆ/root/work/A33_5.1.1_SKD_BASE/lichee/brandy/u-boot-2011.09/./
./tools/pack/chips/sun8iw5p1/bin/boot0_nand_sun8iw5p1.binâ€™
â€ˆboot0_sdcard_sun8iw5p1.binâ€™ -> â€ˆ/root/work/A33_5.1.1_SKD_BASE/lichee/brandy/u-boot-2011.09/
./tools/pack/chips/sun8iw5p1/bin/boot0_sdcard_sun8iw5p1.binâ€™
[root@SeKeDe brandy]#
    
```

3、编译 QT Rootfs

首先切换到 pack 目录
 cd lichee/tools/pack/
 ./pack -c sun8iw5p1 -p dragonboard -b d7 -d uart0 -s none
 编译完成后正确提示如下

```

Filepath: rootfs.fex
FileLength=1d8c268sys_config.fex Len: 0x1072b
config.fex Len: 0xa2e0
split_xxxx.fex Len: 0x200
sys_partition.fex Len: 0xbce
boot0_nand.fex Len: 0x8000
boot0_sdcard.fex Len: 0x8000
u-boot.fex Len: 0xc4000
fes1.fex Len: 0x23a0
usbtool.fex Len: 0x23000
aultools.fex Len: 0x26ead
aultls32.fex Len: 0x238dd
cardtool.fex Len: 0x41400
cardscript.fex Len: 0x6ea
sunxi_mbr.fex Len: 0x10000
dlinfo.fex Len: 0x4000
arisc.fex Len: 0x35520
boot-resource.fex Len: 0x4e0c00
vboot-resource.fex Len: 0x4
env.fex Len: 0x20000
Venv.fex Len: 0x4
boot.fex Len: 0xf6a800
vboot.fex Len: 0x4
rootfs.fex Len: 0x1d8c268
vrootfs.fex Len: 0x4
BuildImg 0
Dragon execute image.cfg SUCCESS !
-----image is at-----
/root/work/PET_A33_6.0.1/lichee/tools/pack/sun8iw5p1_dragonboard_d7_uart0.img
pack finish
    
```

编译完成后会在 lichee/tools/pack 目录下生成 sun8iw5p1_dragonboard_d7_uart0.img 系统烧写镜像文件。

4、修改 Rootfs

完成首次编译后，rootfs 的所有文件位于 lichee\buildroot\target\dragonboard\rootfs 目录下。
 如果需要修改或添加文件，需要将文件复制到 lichee\buildroot\target\dragonboard\extra 相同目录下，
 然后再修改，重新编译内核、uboot、rootfs 即可。

```

例如需要修改 rootfs/etc/init.d/S00peite 这个系统初始化设置脚本文件
cd lichee/buildroot/target/dragonboard
mkdir -p extra/etc/init.d
cp -rf rootfs/etc/init.d/S00peite extra/etc/init.d/S00peite
修改 extra/etc/init.d/S00peite 后重新编译即可生成新的烧写镜像文件
    
```

5、更换 Rootfs 为 Linux 或 Linux + QT

首先删除 lichee\buildroot\target\dragonboard\rootfs 目录。
 将开发资料《源代码》目录下 rootfs 文件更名为 rootfs.tar.xz

复制 rootfs.tar.xz 到 lichee\buildroot\target\dragonboard 覆盖同名文件

重新编译即可

Rootfs 类型有:

Linux_Full --- Linux 全功能版, 不包含 QT

Linux_Lite --- Linux 部分功能版, 不包含 QT

QT_Full --- Linux+QT 全功能版

QT_Lite --- Linux+QT 部分功能版

6、交叉编译其他应用

系统使用的交叉编译器为开发资料《源代码/QT_Source》目录下:

gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi.tar.xz

编译所需的其他库文件是 sysroot_peite.tar.xz 可根据需要进行解压使用, 客户可自行编译其他未包含的支持库、应用程序等。

五、修改 Linux 内核编译选项

首先切换到 linux 内核目录

```
cd lichee/linux-3.4/
```

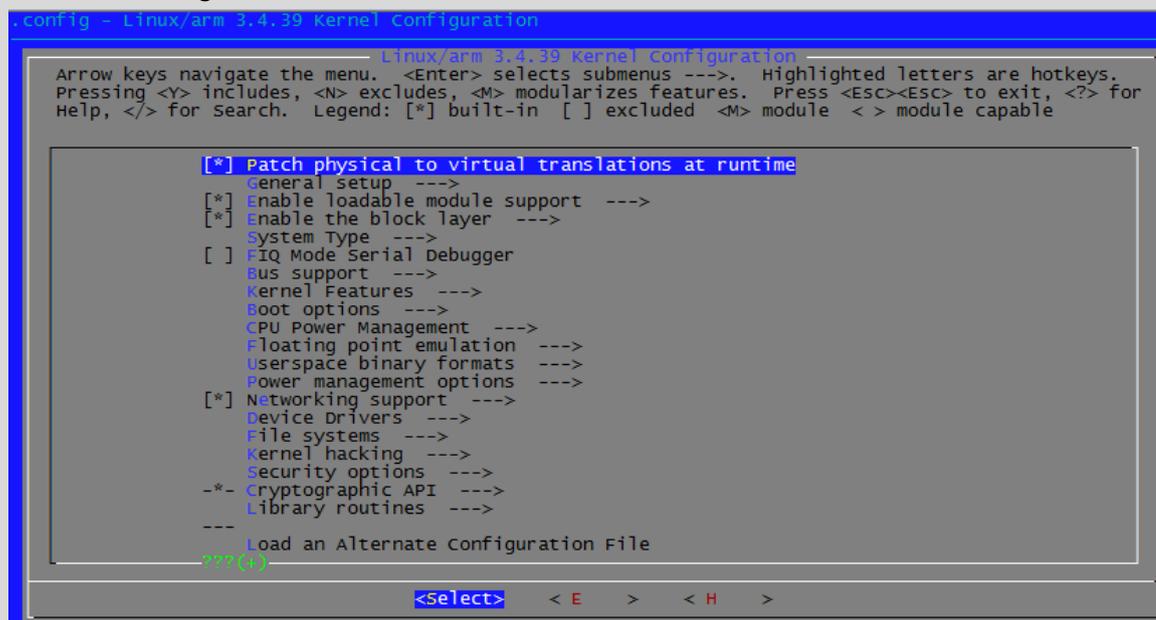
加载默认配置

```
make sun8iw5p1smp_linux_defconfig (Linux+QT 系统)
```

```
make sun8iw5p1smp_android_defconfig (Android 系统)
```

启动内核配置

```
make menuconfig
```



```
.config - Linux/arm 3.4.39 Kernel Configuration
Linux/arm 3.4.39 Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for
Help, </> for search. Legend: [*] built-in [ ] excluded <M> module <> module capable

[*] Patch physical to virtual translations at runtime
  general setup --->
[*] Enable loadable module support --->
[*] Enable the block layer --->
  System Type --->
[ ] FIQ Mode Serial Debugger
  Bus support --->
  Kernel Features --->
  Boot options --->
  CPU Power Management --->
  Floating point emulation --->
  Userspace binary formats --->
  Power management options --->
[*] Networking support --->
  Device Drivers --->
  File systems --->
  Kernel hacking --->
  Security options --->
  *- Cryptographic API --->
  Library routines --->
---
Load an Alternate Configuration File
???(*)
<Select> < E > < H >
```

修改内核选项时不要选择编译成模组文件，可以选择直接编译进内核。

完成配置后保存退出，

将内核根目录下的 .config 文件复制保存为

```
arch/arm/config/sun8iw5p1smp_android_defconfig (Android 系统)
```

```
arch/arm/config/sun8iw5p1smp_linux_defconfig (Linux+QT 系统)
```

，此步骤非常重要，如果不执行的话会自动恢复为默认配置。

完成内核配置修改后，从新编译 android 或 linux 即可。

六、镜像文件烧写

开发过程中，一般使用 PhoenixSuit 进行镜像文件的烧写，具体操作方式请参考开发工具目录下的《PhoenixSuit 使用说明文档.pdf》，除了 Android 系统我司的 Linux+QT 系统也支持这种烧写方式。

将开发板的 MicroUSB 接口连接到系统主机后，Linux+QT 系统检测到的设备信息如下：



The screenshot shows the PhoenixSuit interface with the following content:

- PhoenixSuit 一键刷机工具** (PhoenixSuit One-click Bricking Tool)
- Navigation icons: 首页 (Home), 一键刷机 (One-click Bricking), 设备管理 (Device Management), 资讯页面 (Information Page)
- Message: 欢迎使用 PhoenixSuit 刷机工具 (Welcome to use PhoenixSuit bricking tool)
- Device Model: 型号: PET
- Firmware Version: 固件版本号: Linux-PET_V1.0 SEKEDE
- Compilation Time: 编译时间: 中国标准时间 1970-1-1 8: 0: 0
- Android Version: Android版本: Linux-PET
- Chip Model: 芯片型号: www.gzpeite.com
- Kernel Version: 内核版本: Linux version 3.4.39 (root@SeKeDe) (gcc version 4.6.3 20120201 (prerelease) (crosstool-NG linaro-1.13.1-2012.02-20120222 - Linaro GCC 2012.02)) #67 SMP PREEMPT Thu Mar 30 17:59:27 CST 2017
- Status: 设备已经连接成功 (Device connected successfully)
- Action: 现在可以开始进行刷机和安装游戏应用等操作 (Now you can start bricking and installing game applications)
- Footer: 设备已经连接成功 (Device connected successfully) | PhoenixSuit V1.10

烧写操作需要首先通过 Micro USB 数据线连接主机的开发板，在进行烧写时如果出现主机识别到新的设备没有正常安装驱动的情况时，需要手动安装设备驱动程序，驱动程序位于开发工具文件夹内。

注意，在点击烧写镜像后，设备会重启黑屏，此时需要在 PC 端的设备管理区中再手动安装镜像烧写设备驱动 (AW_Driver)。

七、建立 QT 应用程序编译环境

所需工具位于开发资料的《开发工具/QT》目录下，以下所有操作需要有 **ROOT** 权限:

- 1、解压交叉编译器 gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi.tar.xz

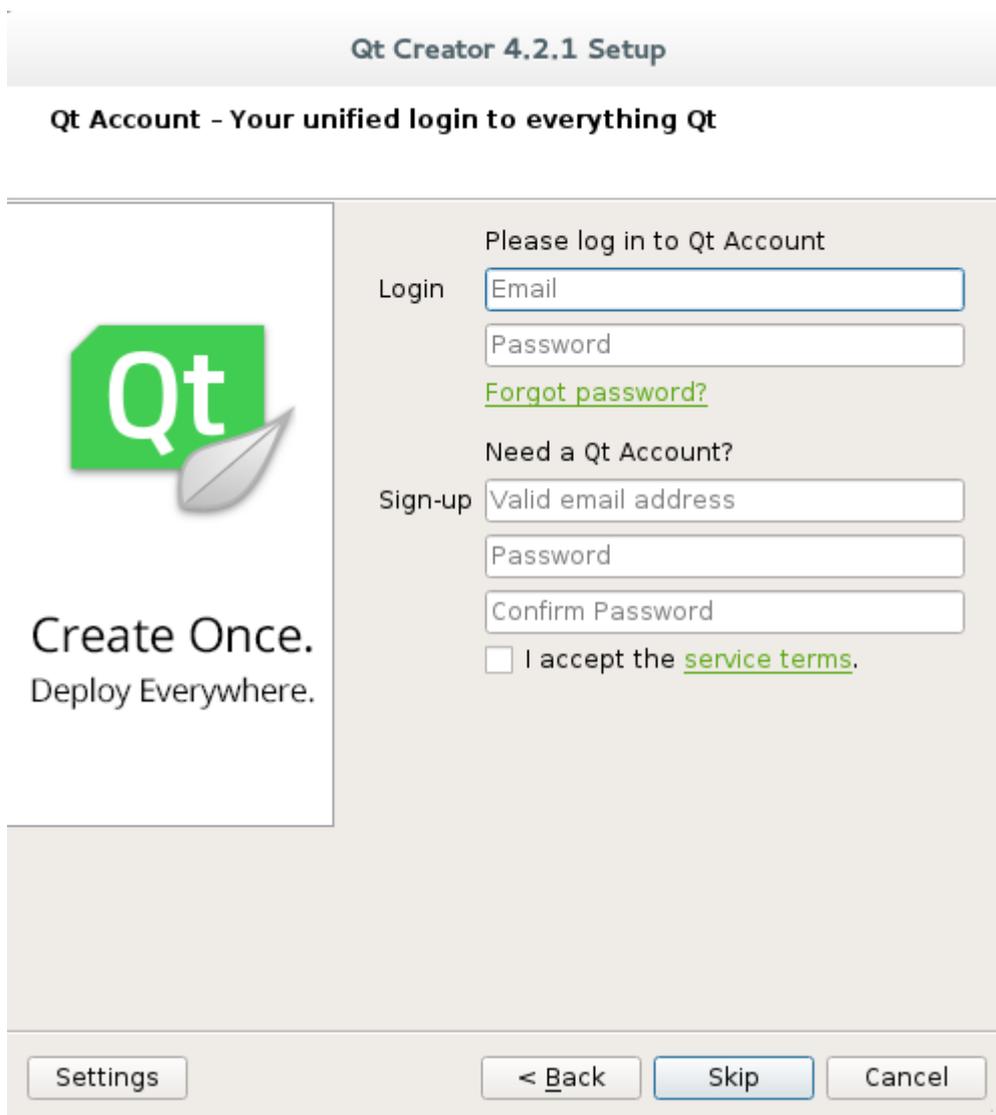
```
tar -xJf gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi.tar.xz -C /usr/local
```

- 2、解压库文件 sysroot_peite_qt.tar.xz

```
tar -xJf sysroot_peite_qt.tar.xz -C /usr/local
```

- 3、解压安装 qt-creator-opensource-linux-x86_64-4.4.0.tar.xz

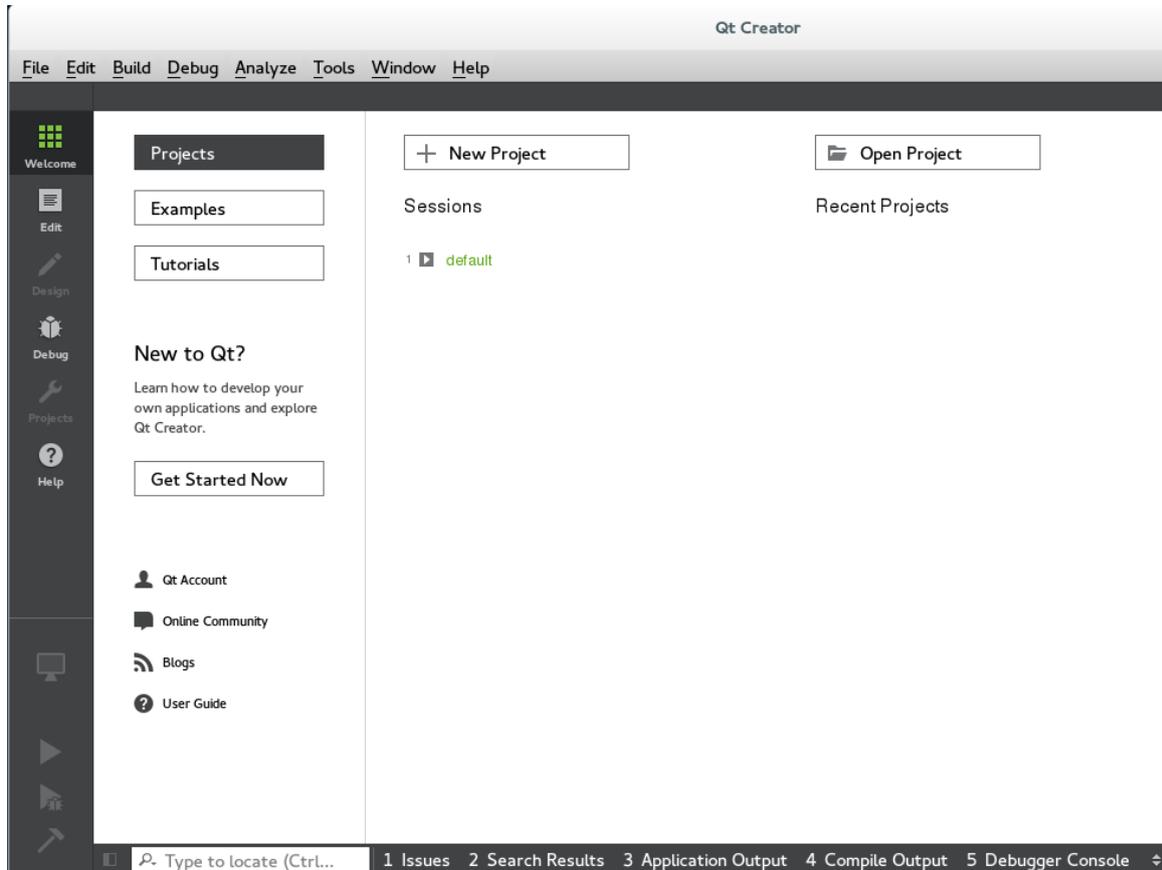
```
tar -xJf qt-creator-opensource-linux-x86_64-4.4.0.tar.xz
./qt-creator-opensource-linux-x86_64-4.4.0.run
```



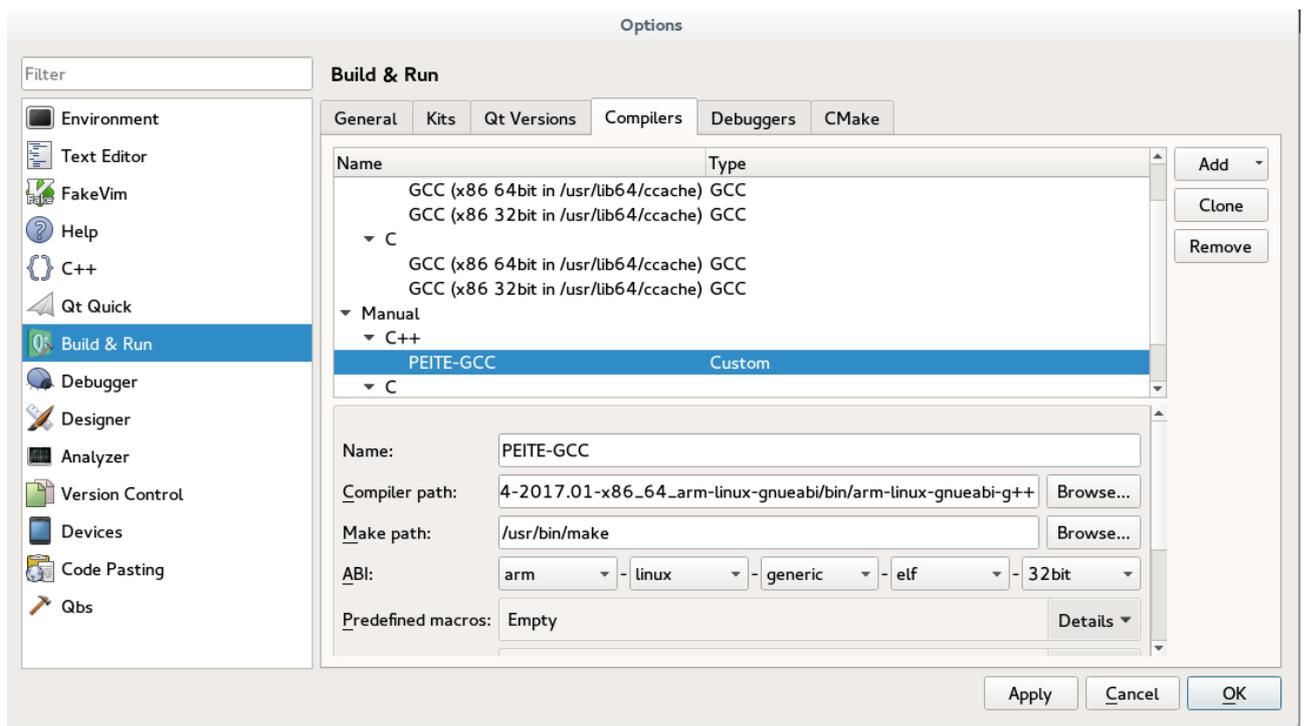
注意在上面这一步选择 **Skip**，其他直接选择 **Next** 即可

4、启动 qt creator 设置交叉编译器和 QT 库文件路径。

`/opt/qtcreator-4.2.1/bin/qtcreator`

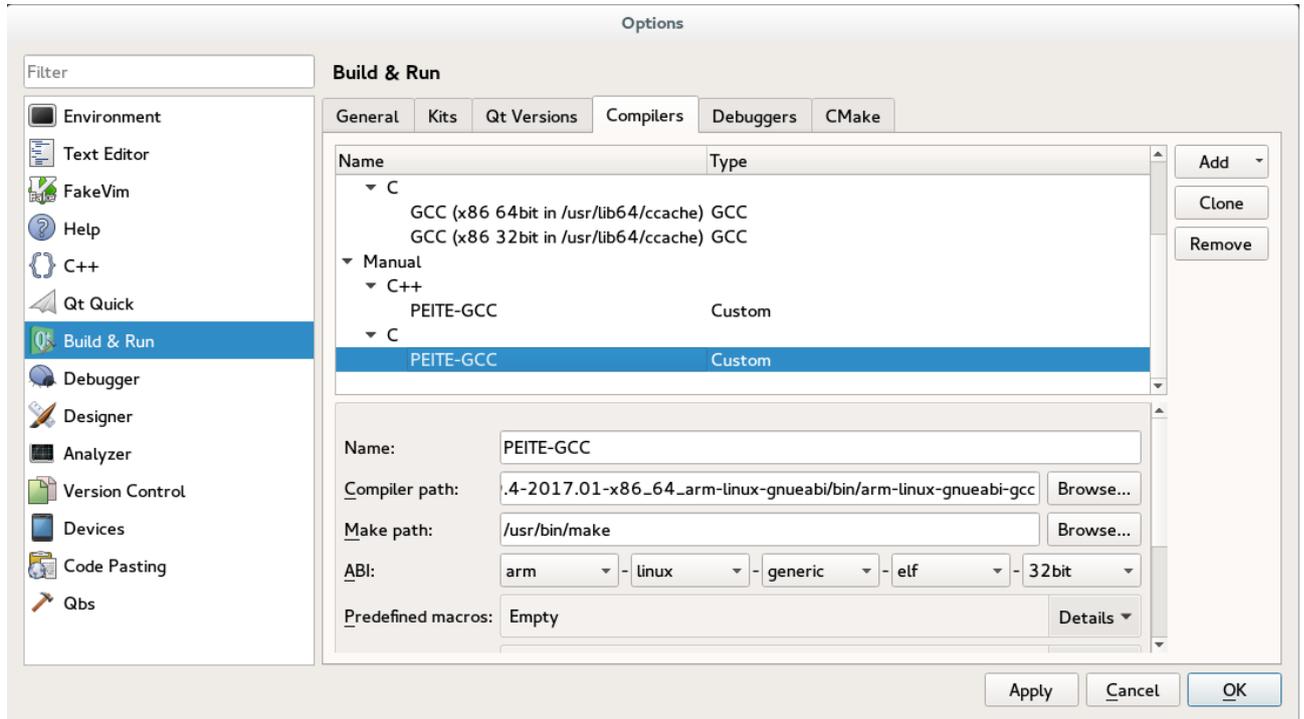


选择菜单 `Tools->Options->Build & Run->Compilers`，点击 `Add ->Custom->C++` 按钮，添加 C++编译器，Compiler path: `/usr/local/gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi/bin/arm-linux-gnueabi-g++`

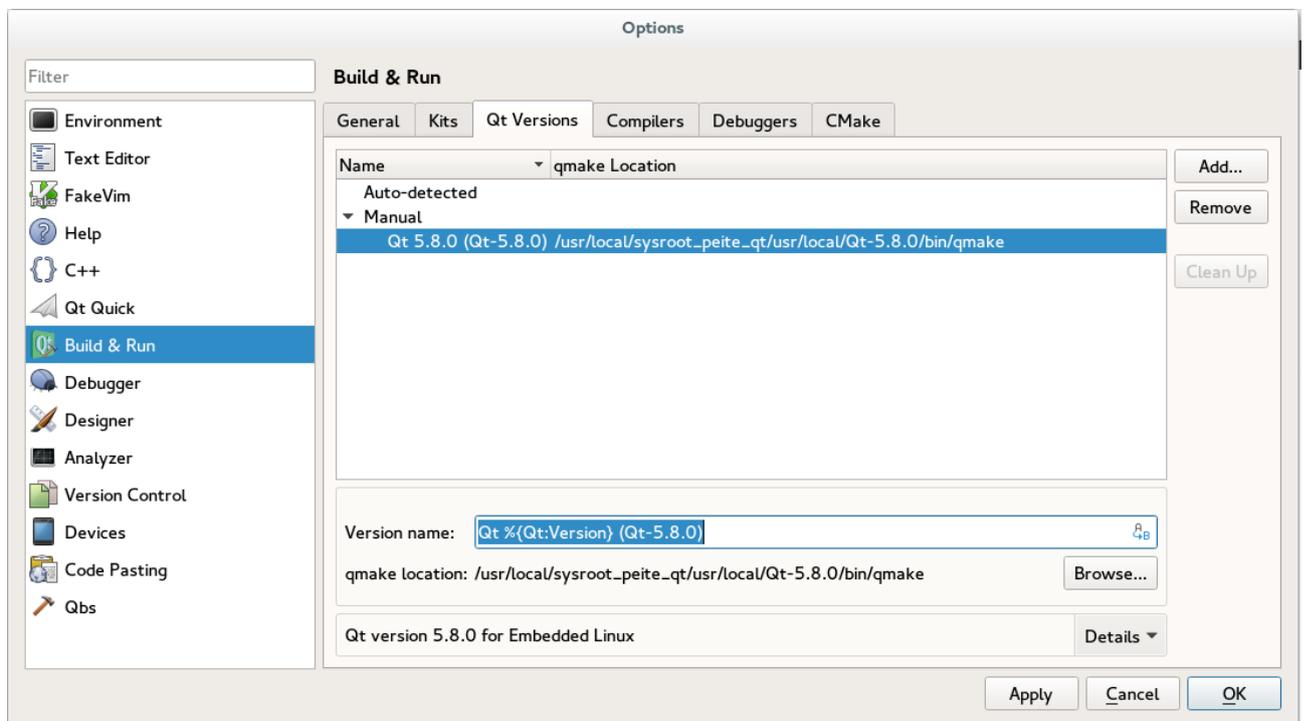


Add ->Custom->C 添加 C 编译器

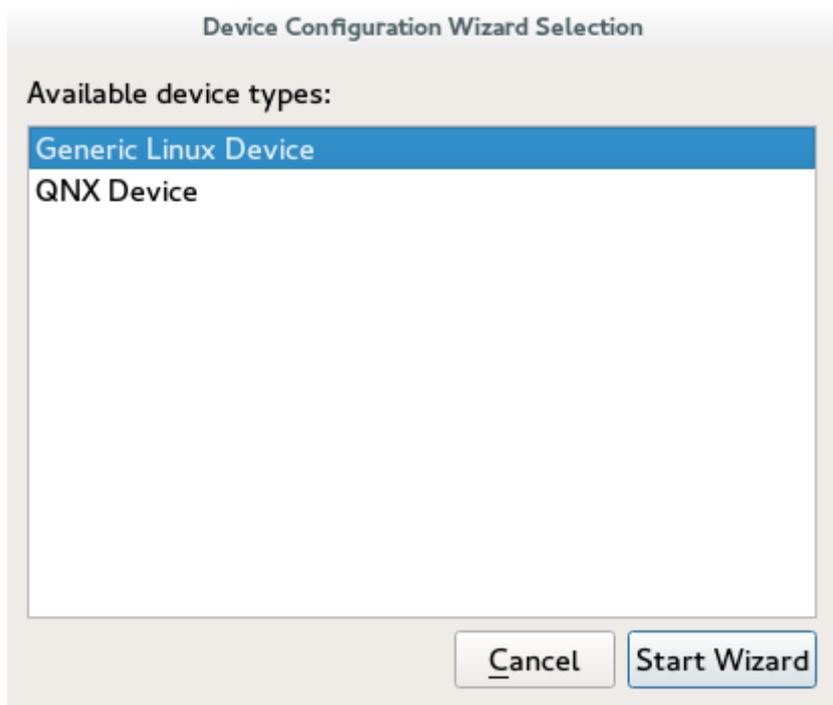
Compiler path: /usr/local/gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi/bin/arm-linux-gnueabi-gcc:



选择菜单 Tools->Options->Build & Run->Qt Versions, 点击 Add 按钮, 配置如下:



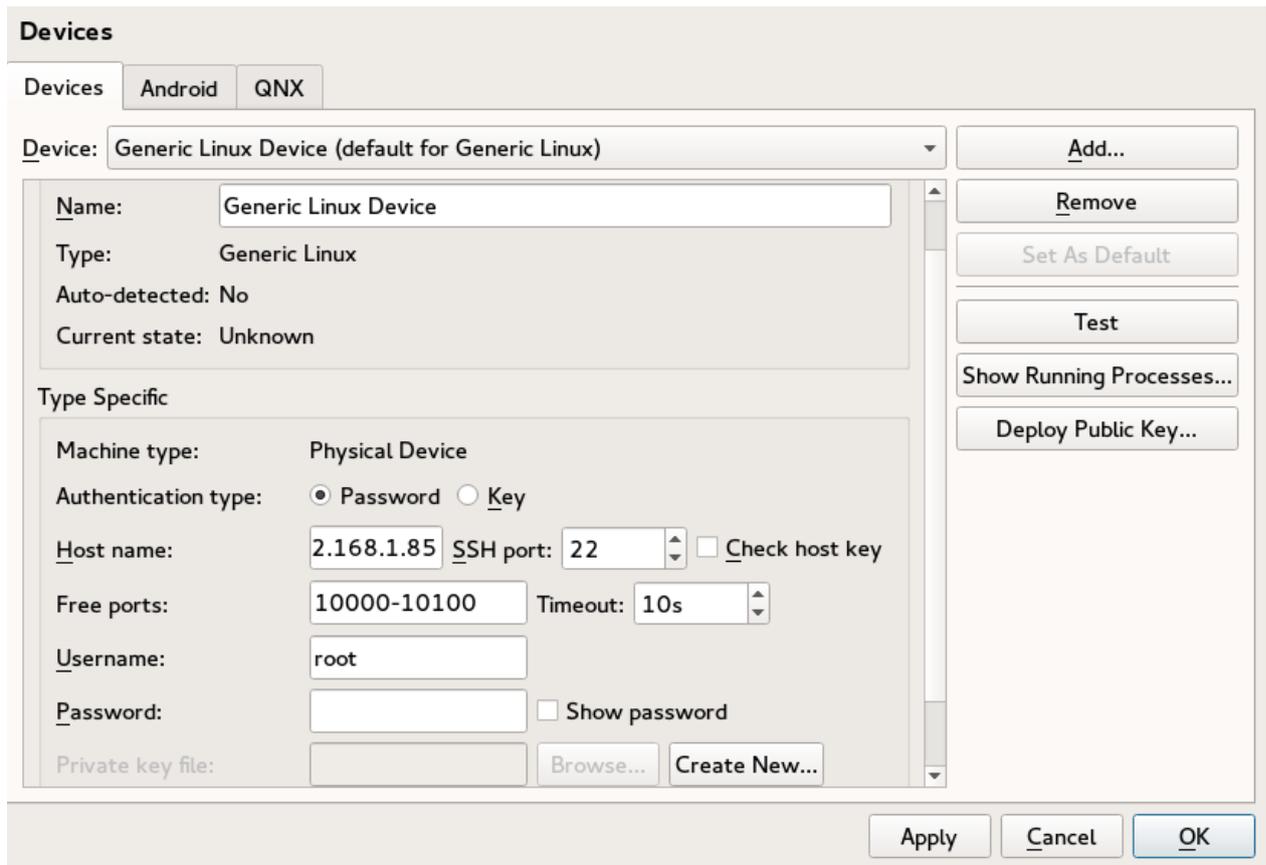
选择菜单 Tools->Options->Devices, 点击 Add 按钮, 配置如下:



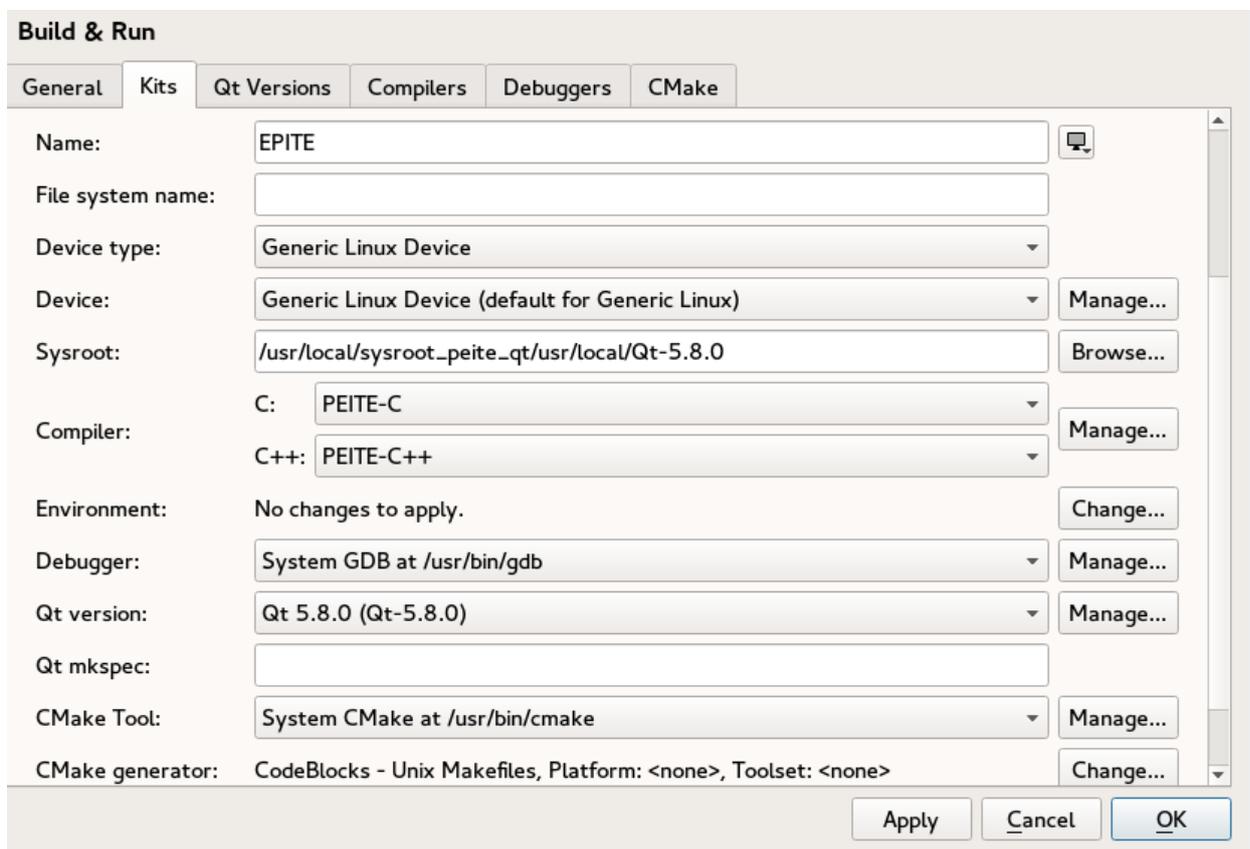
首先需要将开发板与主机在同一局域网内连接好, 主机可以正常 ping 通开发板



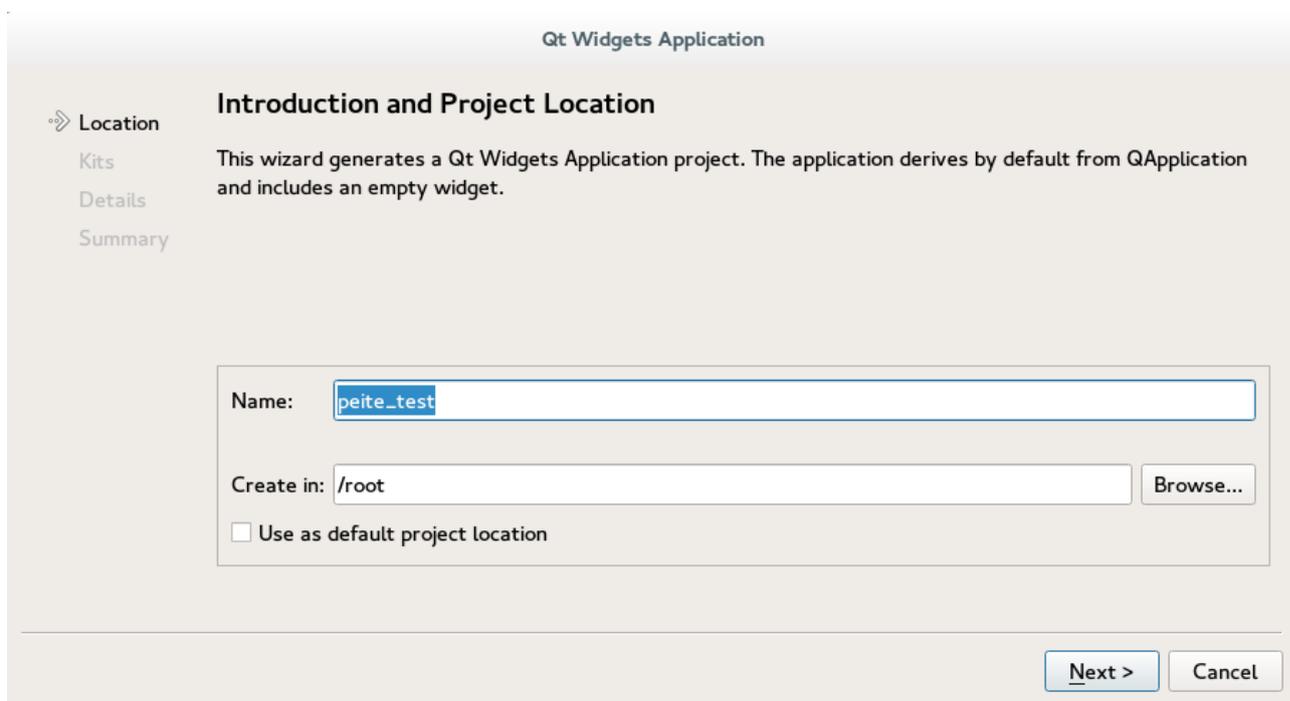
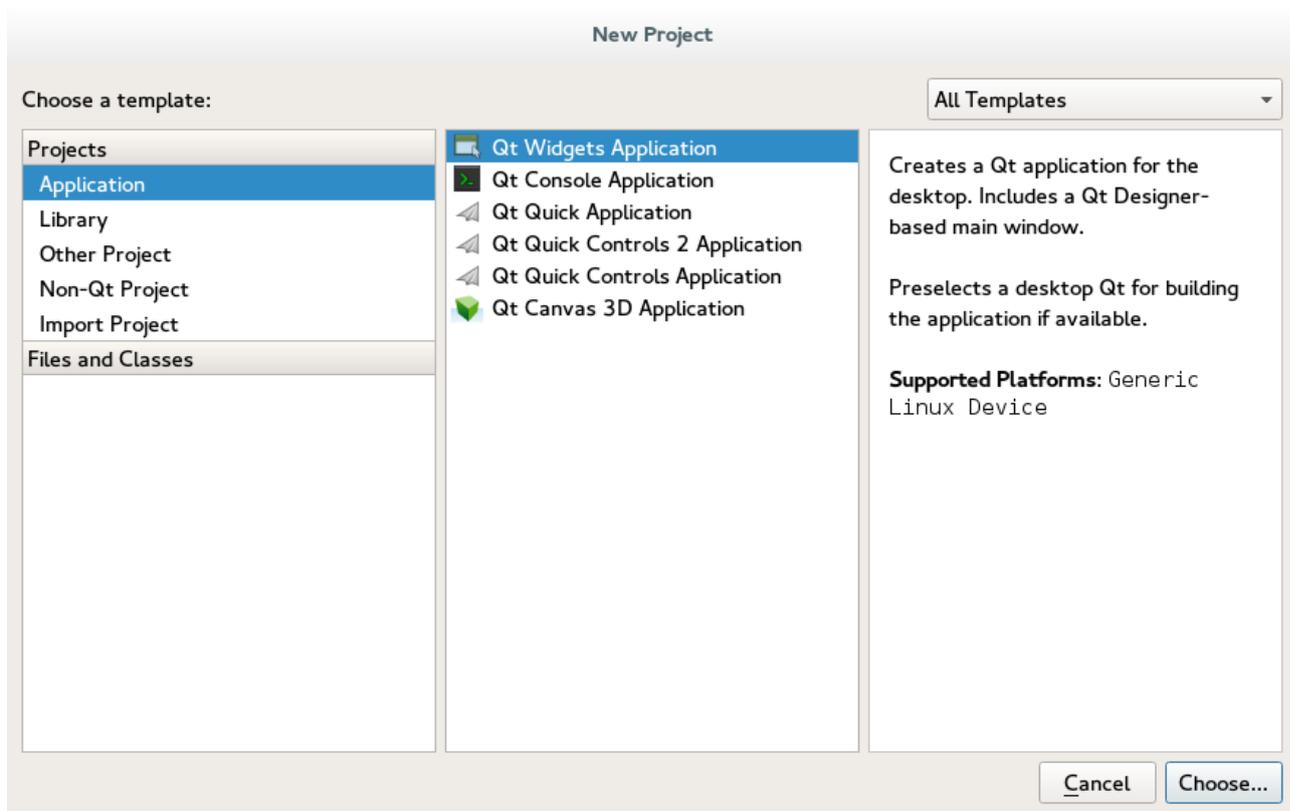
测试通过后的配置如下：



选择菜单 Tools->Options->Build & Run->Kits，点击 Add 按钮，配置如下：



八、创建并编译 QT 程序



Qt Widgets Application

Location

→ Kits

Details

Summary

Kit Selection

Qt Creator can use the following kits for project **peite_test**:

Select all kits

<input checked="" type="checkbox"/>	EPITE		Details ▲
<input checked="" type="checkbox"/>	Debug	/root/build-peite_test-EPITE-Debug	Browse...
<input checked="" type="checkbox"/>	Release	/root/build-peite_test-EPITE-Release	Browse...
<input checked="" type="checkbox"/>	Profile	/root/build-peite_test-EPITE-Profile	Browse...

< Back Next > Cancel

Qt Widgets Application

Location

Kits

→ Details

Summary

Class Information

Specify basic information about the classes for which you want to generate skeleton source code files.

Class name:

Base class:

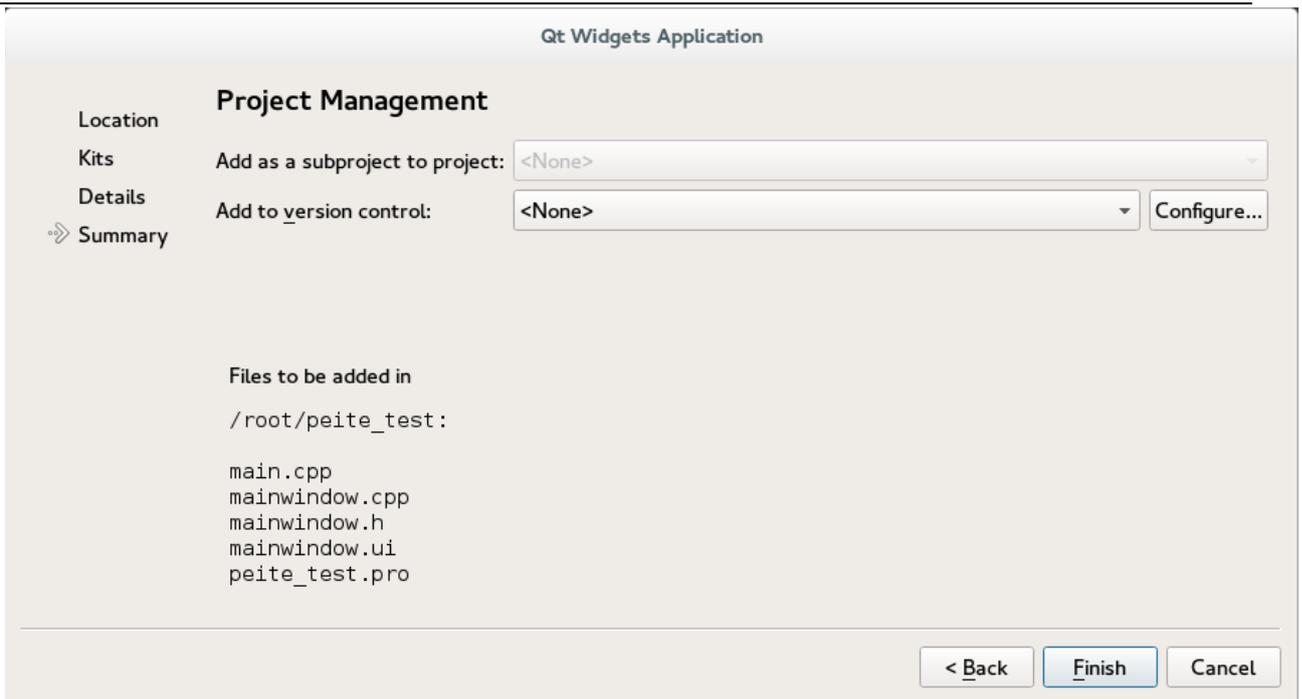
Header file:

Source file:

Generate form:

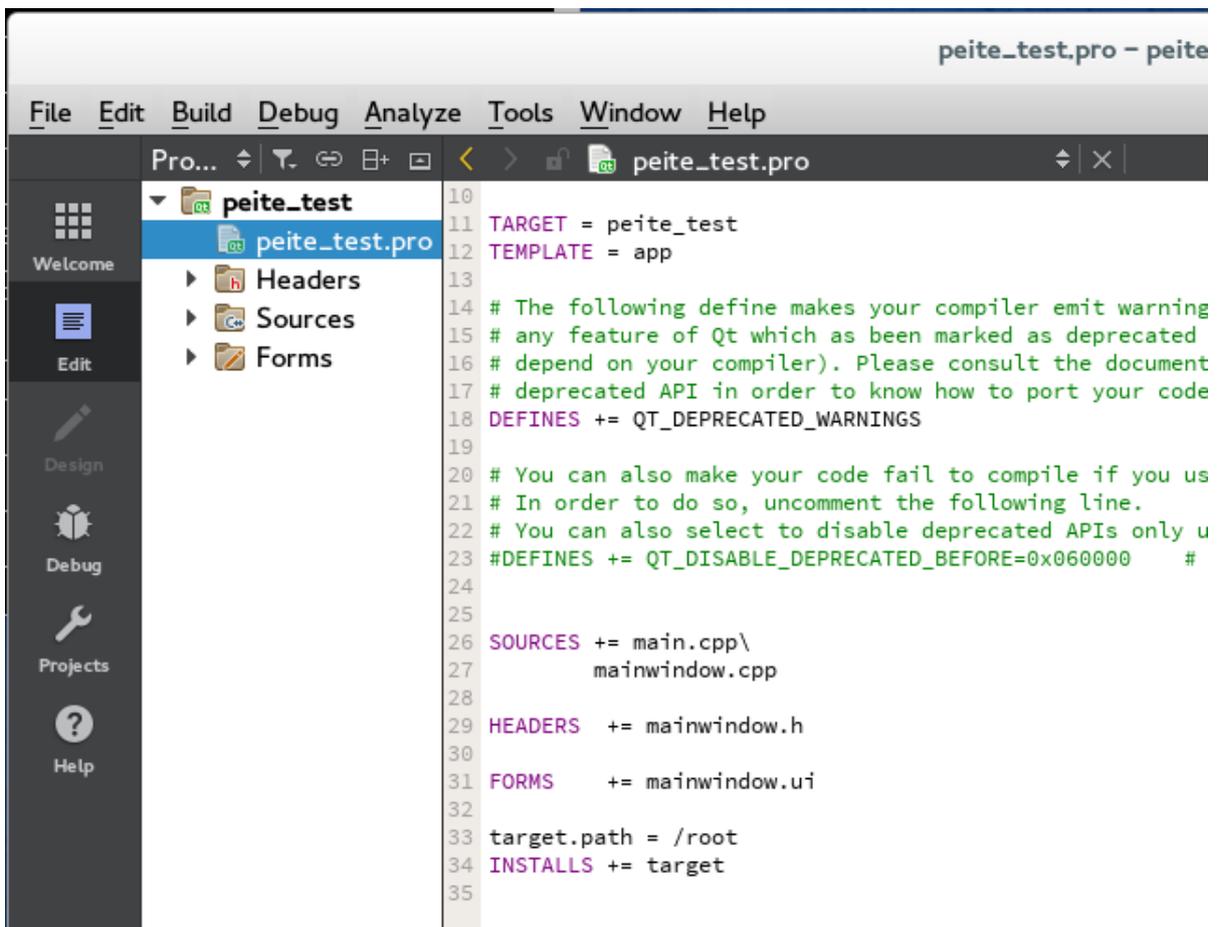
Form file:

< Back Next > Cancel



完成创建后，需要修改工程目录下的 peite_test.pro 文件，在文件最后添加下面两行代码

```
target.path = /root
INSTALLS += target
```



然后在 QT Creator 中重新打开工程，编译、运行后，可以在开发板上查看运行效果。

九、定制编译 QT 源码

客户可以自行编译 QT 的源码，可以对 QT 源码进行修改及定制，以下执行步骤需要 ROOT 权限。

- 1、复制 QT_Source 目录及所有文件到编译主机。
- 2、进入 QT_Source 目录，运行 config.sh 进行编译环境准备及选项配置。
- 3、运行 build.sh 编译
- 4、编译完成后的 QT 安装目录为 /usr/local/sysroot_peite_qt/usr/local/Qt-5.8.0

十、联系方式

地址 : 广州市天河区大观中路新塘大街鑫盛工业园 A1 栋 201

电话 : 020-85625526

传真 : 020-85625526-606

主页 : <http://www.gzpeite.net>

淘宝店 : <https://shop149045251.taobao.com>

核心板 : 王先生

移动电话: 18926288206

电子信箱: 18926288206@gzpeite.net

业务 QQ: 594190286

定制研发: 杨先生

移动电话: 18902281981

电子信箱: 18902281981@gzpeite.net

业务 QQ: 151988801

广州佩特电子科技有限公司

2017 年 12 月