PET-R58 核心板/核心模组
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：

```
cat   PET_R58_Source.tar.xz* | tar xvJf -
```

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 sun8iw6p1_android
```

编译完成后正确提示如下：

```
make: Leaving directory `/root/Work/A83T_SKD_BASE/lichee/linux-3.4/modules/gpu'
regenerate rootfs cpio
11282 blocks
12031 blocks
build_ramfs
Copy boot.img to output directory ...
Copy modules to target ...

sun8iw6p1 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: ----------------------------------------
```

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 sun8iw6p1

编译完成后正确提示如下

```
/root/Work/A83T_SKD_BASE/lichee/brandy/u-boot-2011.09/../gcc-linaro/bin/arm-linux-gnueabi-ld /root/Work/A83T_SKD
_BASE/lichee/brandy/u-boot-2011.09/arch/arm/cpu/armv7/sun8iw6/dram/libdram.o /root/Work/A83T_SKD_BASE/lichee/bra
ndy/u-boot-2011.09/arch/arm/cpu/armv7/sun8iw6/mmc/libmmc.o /root/Work/A83T_SKD_BASE/lichee/brandy/u-boot-2011.09
/arch/arm/cpu/armv7/sun8iw6/nand/libnand.o /root/Work/A83T_SKD_BASE/lichee/brandy/u-boot-2011.09/lib/openssl/lib
openssl.o /root/Work/A83T_SKD_BASE/lichee/brandy/u-boot-2011.09/sunxi_spl/sbrom/flash/libflash.o /root/Work/A83T
_SKD_BASE/lichee/brandy/u-boot-2011.09/sunxi_spl/sbrom/libs/libgeneric.o /root/Work/A83T_SKD_BASE/lichee/brandy/
u-boot-2011.09/sunxi_spl/sbrom/load/libload.o /root/Work/A83T_SKD_BASE/lichee/brandy/u-boot-2011.09/sunxi_spl/sb
rom/main/libmain.o /root/Work/A83T_SKD_BASE/lichee/brandy/u-boot-2011.09/sunxi_spl/sbrom/spl/libsource_spl.o /ro
ot/Work/A83T_SKD_BASE/lichee/brandy/u-boot-2011.09/sunxi_spl/spl/lib/libgeneric.o -L /root/Work/A83T_SKD_BASE/li
chee/brandy/gcc-linaro/bin/../lib/gcc/arm-linux-gnueabi/4.6.3 -lgcc  -T/root/Work/A83T_SKD_BASE/lichee/brandy/u-
boot-2011.09/sunxi_spl/sbrom/sboot.lds -o sboot.axf -Map sboot.map
/root/Work/A83T_SKD_BASE/lichee/brandy/u-boot-2011.09/../gcc-linaro/bin/arm-linux-gnueabi-objcopy --gap-fill=0xf
f -O binary /root/Work/A83T_SKD_BASE/lichee/brandy/u-boot-2011.09/sunxi_spl/sbrom/sboot.axf /root/Work/A83T_SKD_
BASE/lichee/brandy/u-boot-2011.09/sunxi_spl/sbrom/sboot.bin
make[1]: Leaving directory `/root/Work/A83T_SKD_BASE/lichee/brandy/u-boot-2011.09/sunxi_spl/sbrom'
sboot_sun8iw6p1.bin -> /root/work/A83T_SKD_BASE/lichee/brandy/u-boot-2011.09/../../tools/pack/chips/sun8iw6p1/bi
n/sboot_sun8iw6p1.bin
```

3、编译 android

cd android

source build/envsetup.sh

lunch octopus_f1-eng

extract-bsp

make -j4

pack

编译完成后正确提示如下

```
config.fex Len: 0xd1d4
split_xxxx.fex Len: 0x200
sys_partition.fex Len: 0xf18
boot0_nand.fex Len: 0x8000
boot0_sdcard.fex Len: 0x8000
u-boot.fex Len: 0xd4000
toc1.fex Len: 0x8
toc0.fex Len: 0x8
fes1.fex Len: 0x3040
usbtool.fex Len: 0x23000
aultools.fex Len: 0x26ead
aultls32.fex Len: 0x238dd
cardtool.fex Len: 0x14000
cardscript.fex Len: 0x699
sunxi_mbr.fex Len: 0x10000
dlinfo.fex Len: 0x4000
arisc.fex Len: 0x367a9
boot-resource.fex Len: 0x4e1c00
Vboot-resource.fex Len: 0x4
env.fex Len: 0x20000
Venv.fex Len: 0x4
boot.fex Len: 0xd32800
Vboot.fex Len: 0x4
system.fex Len: 0x141e384c
Vsystem.fex Len: 0x4
recovery.fex Len: 0xdc8800
Vrecovery.fex Len: 0x4
BuildImg 0
Dragon execute image.cfg SUCCESS !
----------image is at----------

/root/Work/A83T_SKD_BASE/lichee/tools/pack/sun8iw6p1_android_f1_uart0.img

pack finish
```

编译完成后会在 lichee/tools/pack 目录下生成 sun8iw6p1_android_f1_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 sun8iw6p1_dragonboard
编译完成后正确提示如下

```
regenerate rootfs cpio
11282 blocks
12031 blocks
build_ramfs
Copy boot.img to output directory ...

sun8iw6p1 compile Kernel successful


INFO: build kernel OK.
INFO: build rootfs ...
Regenerating dragonboard Rootfs...
event num 4
/root/Work/A83T_SKD_BASE/lichee/buildroot/target/dragonboard/rootfs/etc/profile
generating rootfs...
blocks: 491M -> 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 13606/49152 inodes and 132212/196608 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: Sun Mar 19 06:20:18 CST 2017
INFO: build rootfs OK.
INFO: ----------------------------------------
INFO: build lichee OK.
INFO: ----------------------------------------
```

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 sun8iw6p1
编译完成后正确提示如下

```
/root/Work/A83T_SKD_BASE/lichee/brandy/u-boot-2011.09/../gcc-linaro/bin/arm-linux-gnueabi-ld /root/Work/A83T_SKD
_BASE/lichee/brandy/u-boot-2011.09/arch/arm/cpu/armv7/sun8iw6/dram/libdram.o /root/Work/A83T_SKD_BASE/lichee/bra
ndy/u-boot-2011.09/arch/arm/cpu/armv7/sun8iw6/mmc/libmmc.o /root/Work/A83T_SKD_BASE/lichee/brandy/u-boot-2011.09
/arch/arm/cpu/armv7/sun8iw6/nand/libnand.o /root/Work/A83T_SKD_BASE/lichee/brandy/u-boot-2011.09/lib/openssl/lib
openssl.o /root/Work/A83T_SKD_BASE/lichee/brandy/u-boot-2011.09/sunxi_spl/sbrom/flash/libflash.o /root/Work/A83T
_SKD_BASE/lichee/brandy/u-boot-2011.09/sunxi_spl/sbrom/libs/libgeneric.o /root/Work/A83T_SKD_BASE/lichee/brandy/
u-boot-2011.09/sunxi_spl/sbrom/load/libload.o /root/Work/A83T_SKD_BASE/lichee/brandy/u-boot-2011.09/sunxi_spl/sb
rom/main/libmain.o /root/Work/A83T_SKD_BASE/lichee/brandy/u-boot-2011.09/sunxi_spl/sbrom/spl/libsource_spl.o /ro
ot/Work/A83T_SKD_BASE/lichee/brandy/u-boot-2011.09/sunxi_spl/spl/lib/libgeneric.o -L /root/Work/A83T_SKD_BASE/li
chee/brandy/gcc-linaro/bin/../lib/gcc/arm-linux-gnueabi/4.6.3 -lgcc  -T/root/Work/A83T_SKD_BASE/lichee/brandy/u-
boot-2011.09/sunxi_spl/sbrom/sboot.lds -o sboot.axf -Map sboot.map
/root/Work/A83T_SKD_BASE/lichee/brandy/u-boot-2011.09/../gcc-linaro/bin/arm-linux-gnueabi-objcopy --gap-fill=0xf
f -O binary /root/Work/A83T_SKD_BASE/lichee/brandy/u-boot-2011.09/sunxi_spl/sbrom/sboot.axf /root/Work/A83T_SKD_
BASE/lichee/brandy/u-boot-2011.09/sunxi_spl/sbrom/sboot.bin
make[1]: Leaving directory `/root/Work/A83T_SKD_BASE/lichee/brandy/u-boot-2011.09/sunxi_spl/sbrom'
sboot_sun8iw6p1.bin -> /root/Work/A83T_SKD_BASE/lichee/brandy/u-boot-2011.09/../../tools/pack/chips/sun8iw6p1/bi
n/sboot_sun8iw6p1.bin
```

3、编译 QT Rootfs

首先切换到 pack 目录
cd lichee/tools/pack/
./pack -c sun8iw6p1 -p dragonboard -b f1 -d uart0 -s none -v none
编译完成后正确提示如下

```
split_xxxx.fex Len: 0x200
sys_partition.fex Len: 0xaa0
boot0_nand.fex Len: 0x8000
boot0_sdcard.fex Len: 0x8000
u-boot.fex Len: 0xd4000
toc1.fex Len: 0x8
toc0.fex Len: 0x8
fes1.fex Len: 0x3040
usbtool.fex Len: 0x23000
aultools.fex Len: 0x26ead
aultls32.fex Len: 0x238dd
cardtool.fex Len: 0x14000
cardscript.fex Len: 0x699
sunxi_mbr.fex Len: 0x10000
dlinfo.fex Len: 0x4000
arisc.fex Len: 0x367a9
boot-resource.fex Len: 0x4e4c00
Vboot-resource.fex Len: 0x4
env.fex Len: 0x20000
Venv.fex Len: 0x4
boot.fex Len: 0xe5f000
Vboot.fex Len: 0x4
rootfs.fex Len: 0x1fc09c00
Vrootfs.fex Len: 0x4
BuildImg 0
Dragon execute image.cfg SUCCESS !
----------image is at----------

/root/work/A83T_SKD_BASE/lichee/tools/pack/sun8iw6p1_dragonboard_f1_uart0.img
```

编译完成后会在 lichee/tools/pack 目录下生成 sun8iw6p1_dragonboard_f1_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

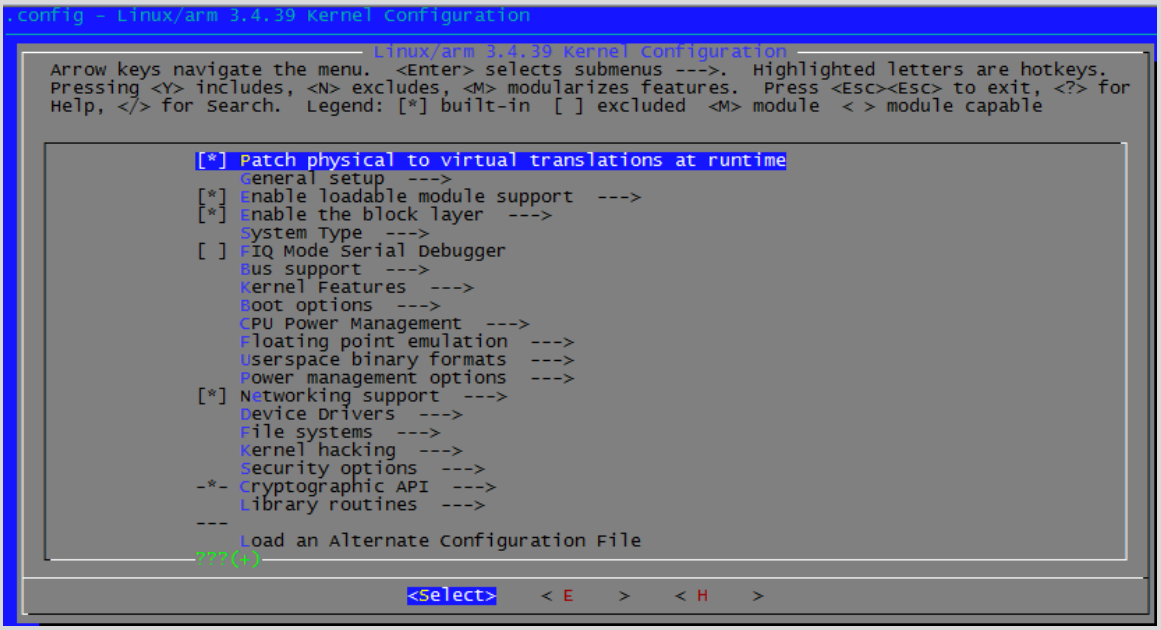
cd lichee/linux-3.4/

加载默认配置

make sun8iw6p1smp_android_defconfig

启动内核配置

make menuconfig



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/sun8iw6p1smp_android_defconfig ，

此步骤非常重要，如果不执行的话会自动恢复为默认配置。

cp .config arch/arm/config/sun8iw6p1smp_android_defconfig

完成内核配置修改后，从新编译 android 或 linux 即可。

注意 Android 和 Linux 共用同一个默认配置，修改内核选项对 android 和 Linux+QT 同时生效。

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文档目录下的《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，系统检测到的设备信息如下：

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。

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

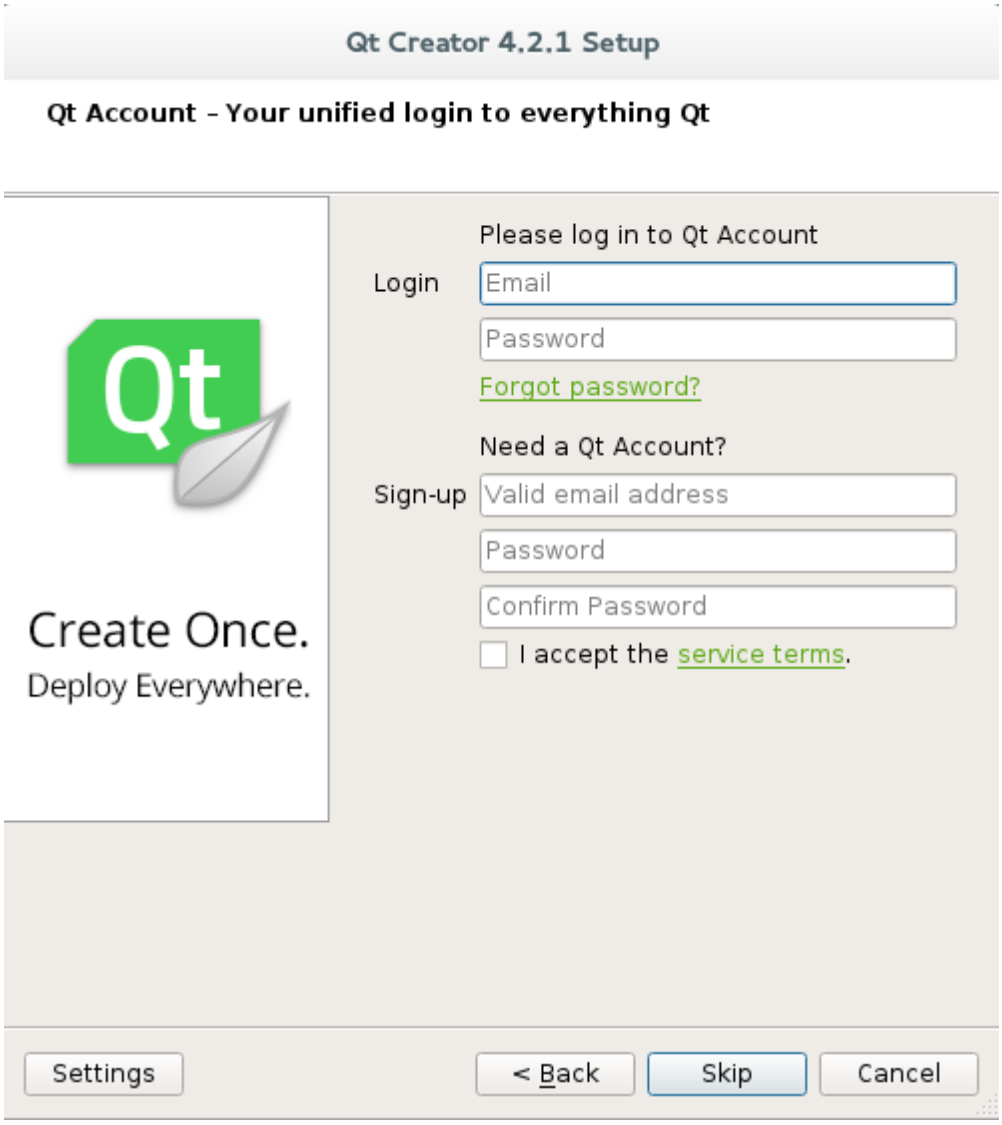
```
tar -xJf gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi.tar.xz -C /usr/local
```

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_64-4.2.1.tar.xz

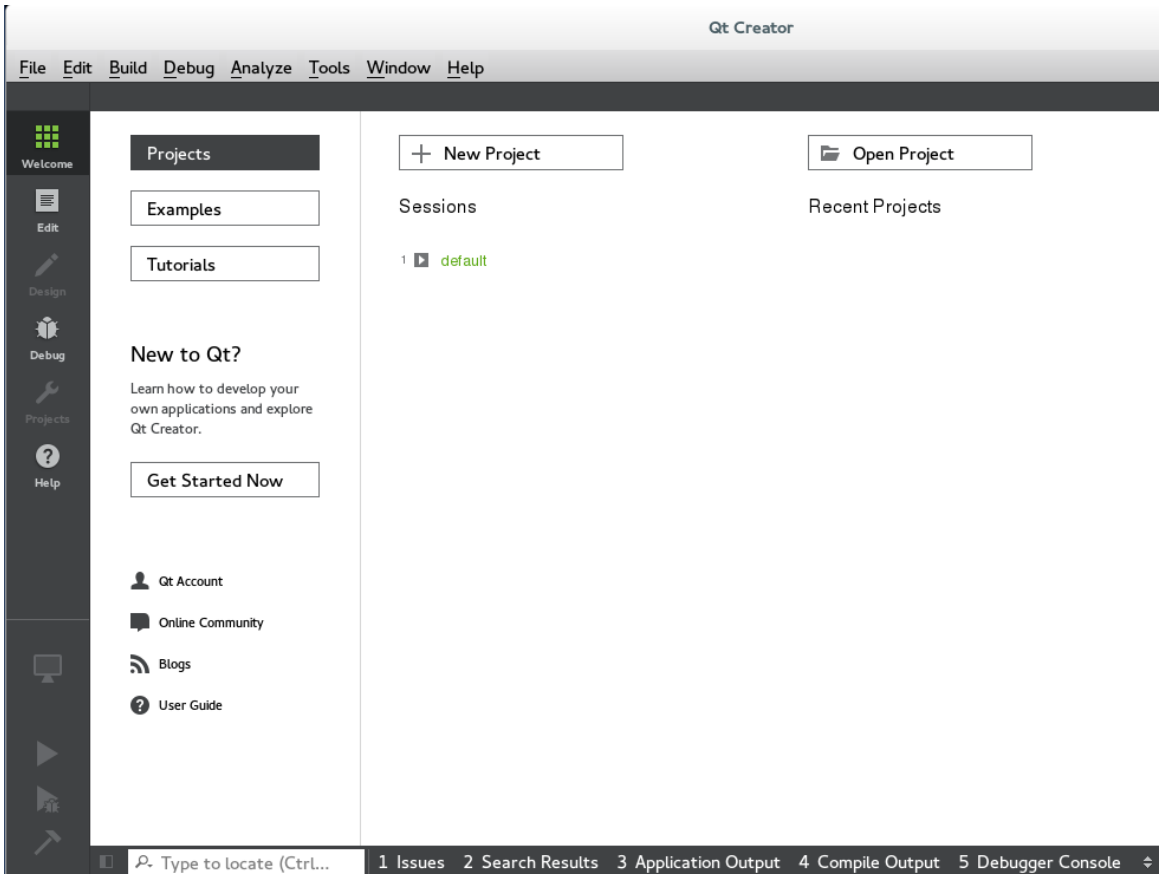
```
tar -xJf qt-creator-opensource-linux-x86_64-4.2.1.tar.xz
./qt-creator-opensource-linux-x86_64-4.2.1.run
```



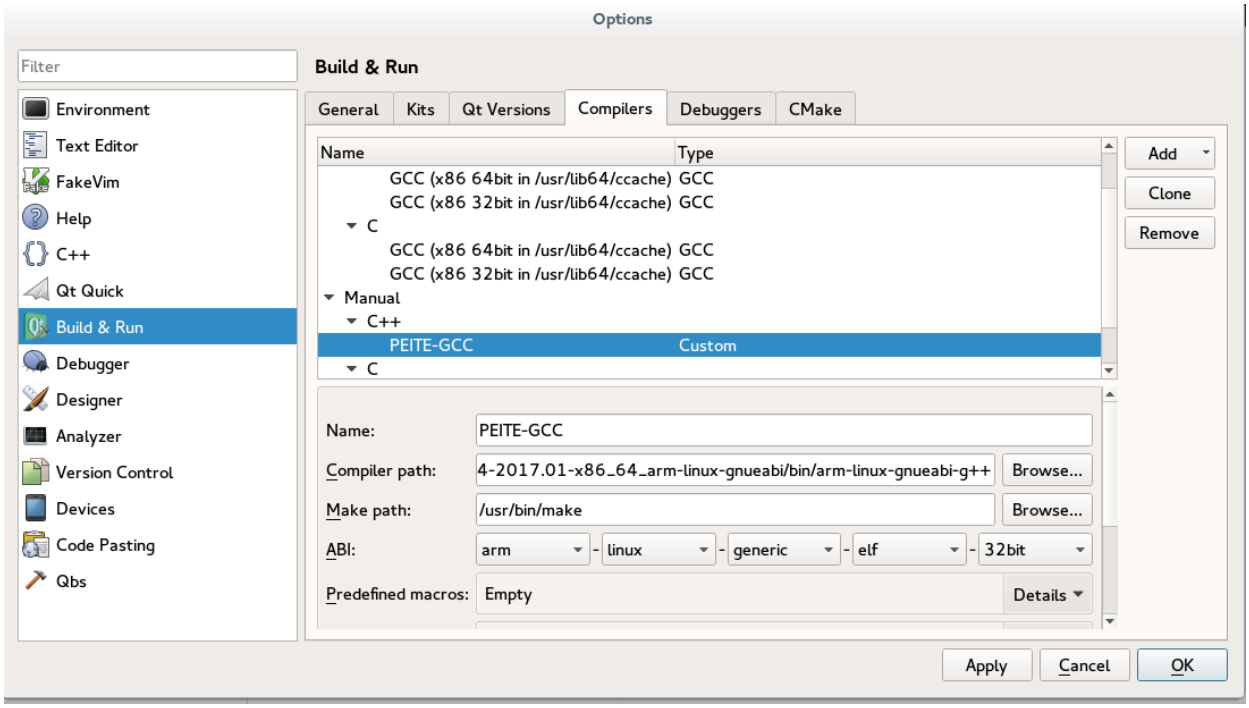
注意在上面这一步选择 Skip，其他直接选择 Next 即可

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

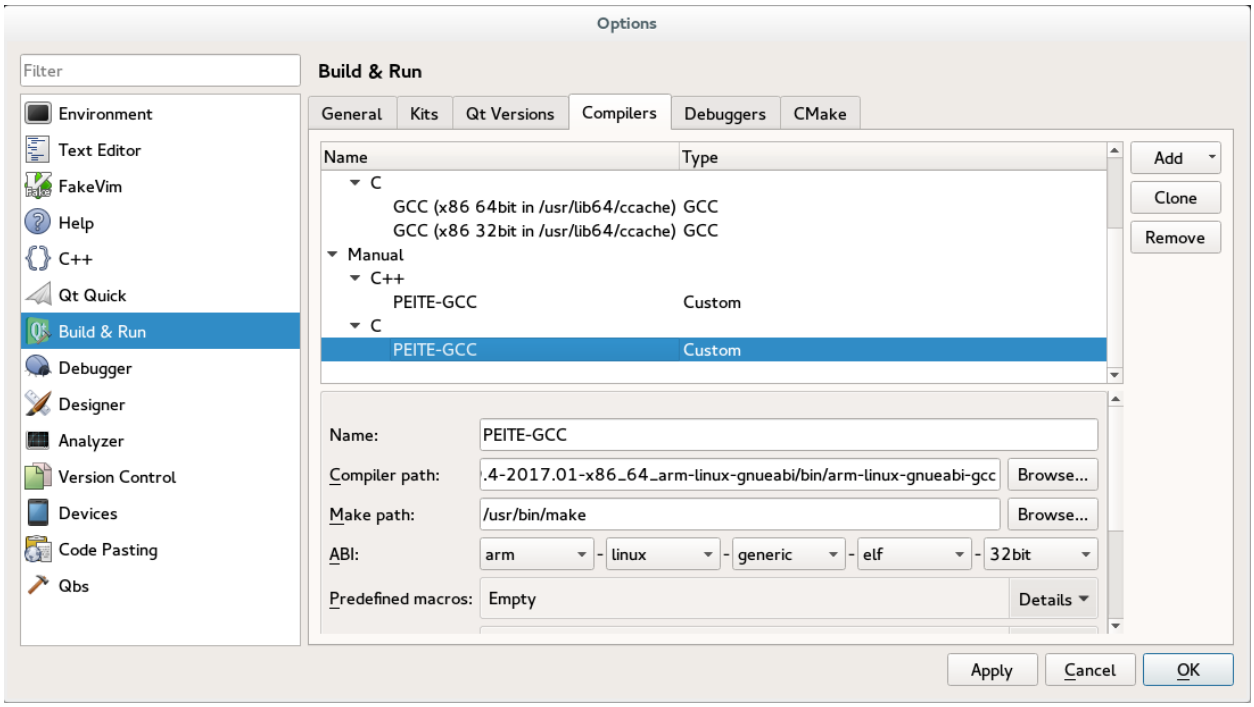


选择菜单 Tools->Options->Build & Run->Compilers，点击 Add ->Custom->C++ 按钮，添加 C++编译器，Complier path：/usr/local/gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi/bin/arm-linux-gnueabi-g++

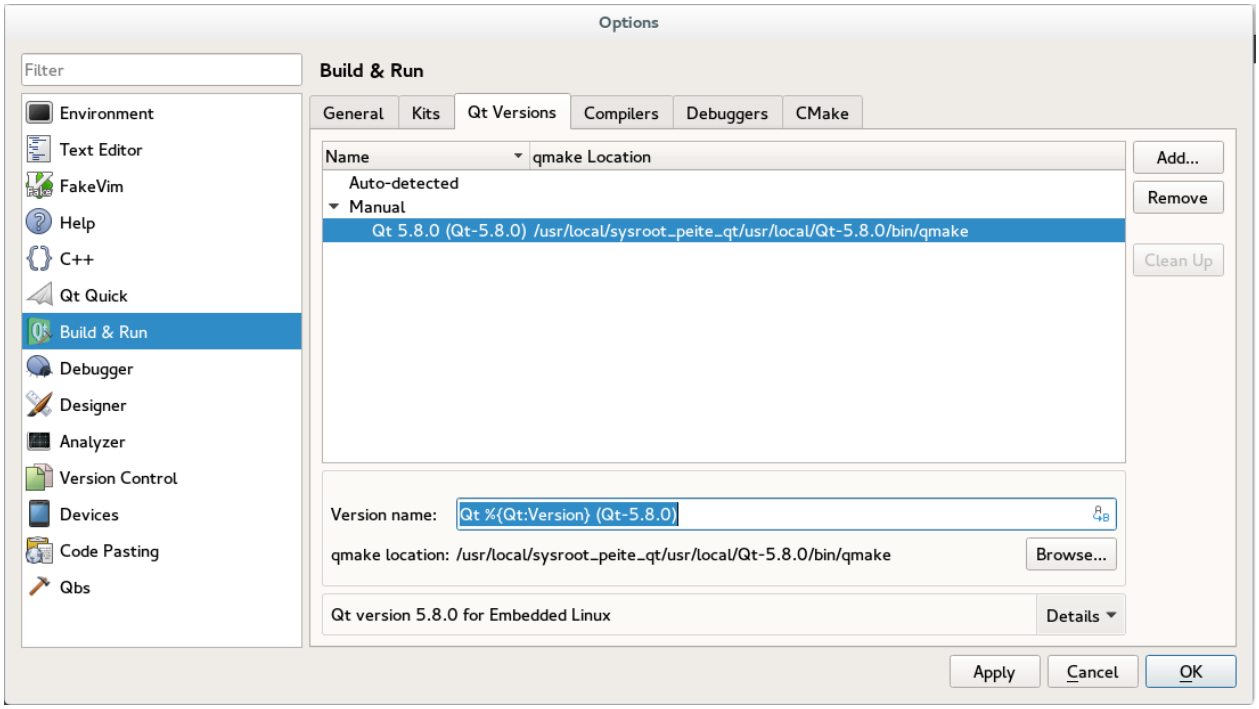


Add ->Custom->C 添加 C 编译器

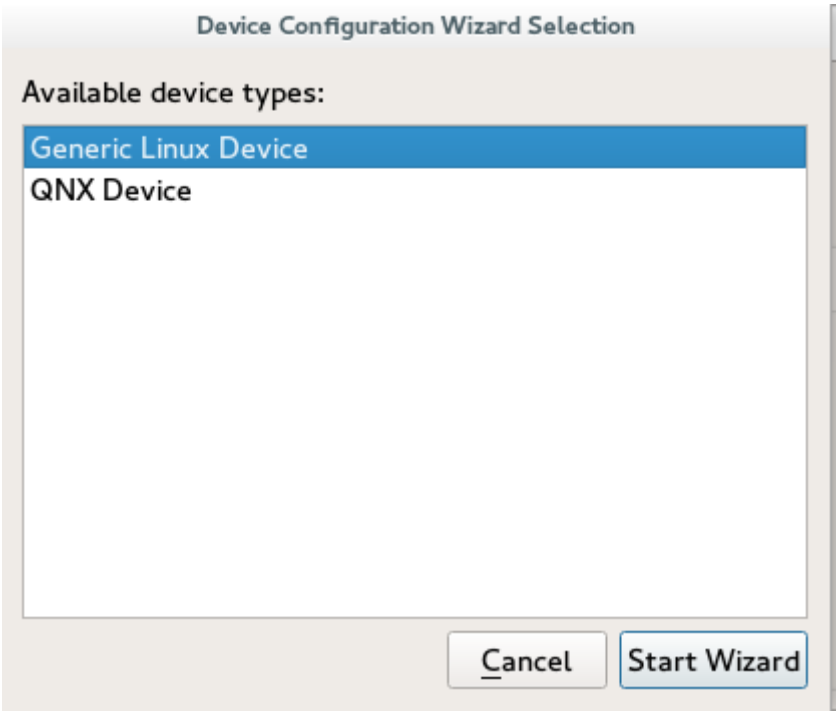
Complier path：/usr/local/gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi/bin/arm-linux-gnueabi-gcc：



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

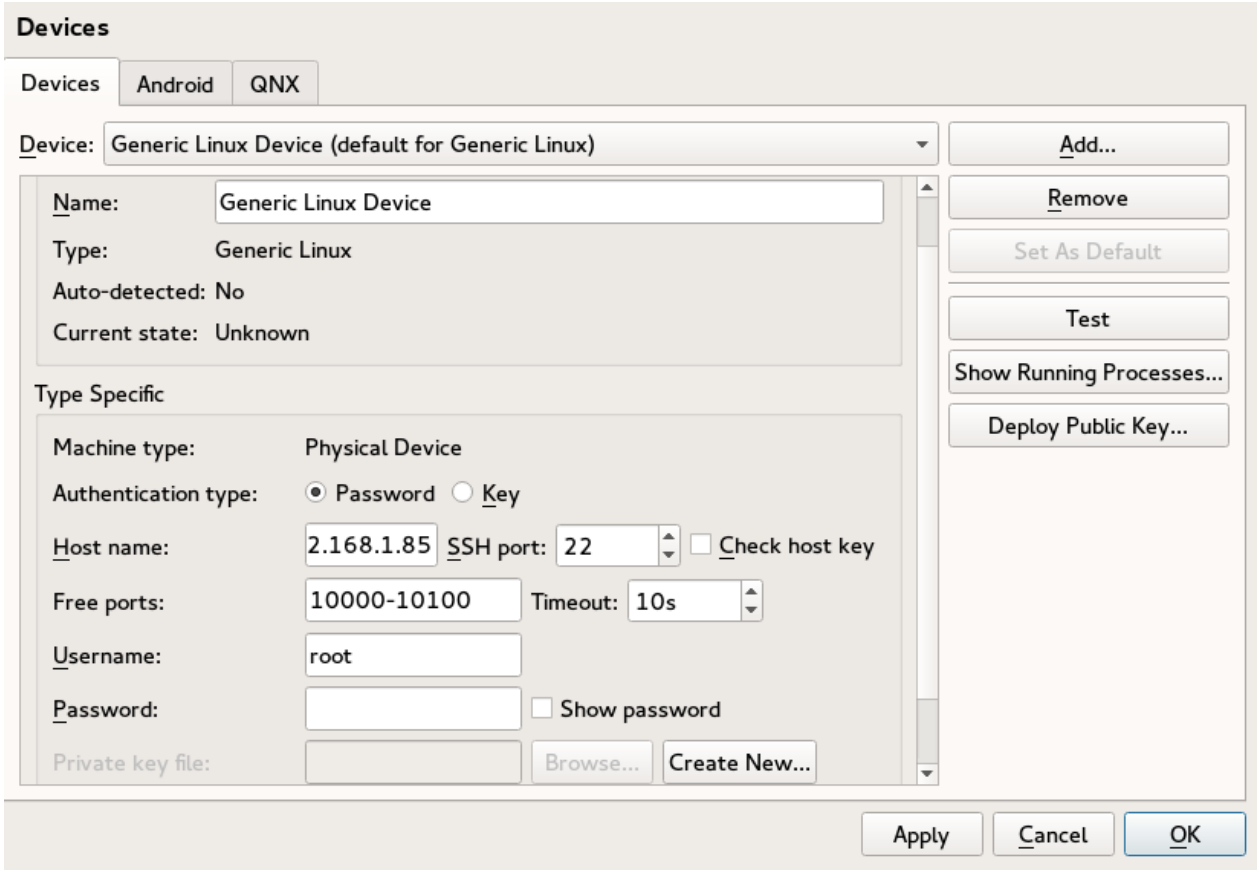


选择菜单 Tools->Options->Devices，点击 Add 按钮，配置如下：

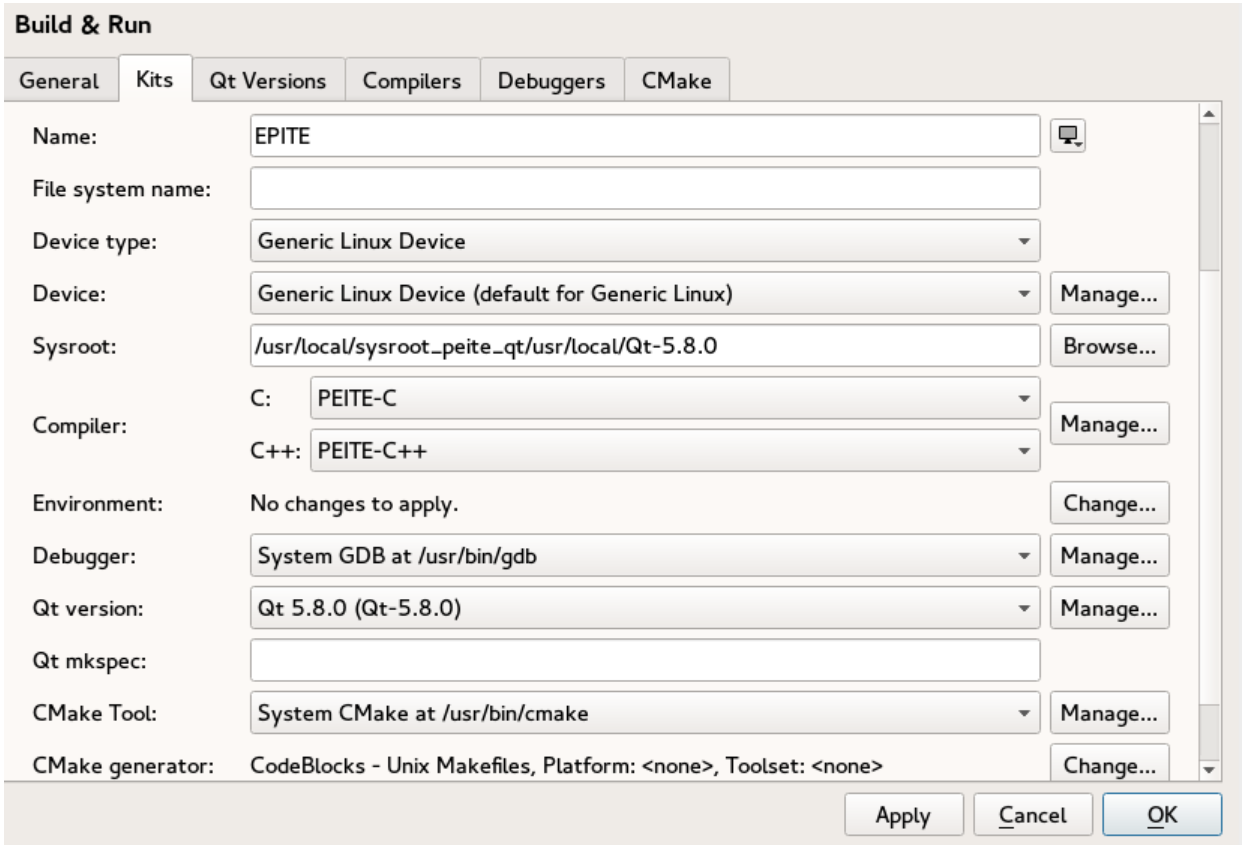


首先需要将开发板与主机在同一局域网内连接好，主机可以正常 ping 通开发板

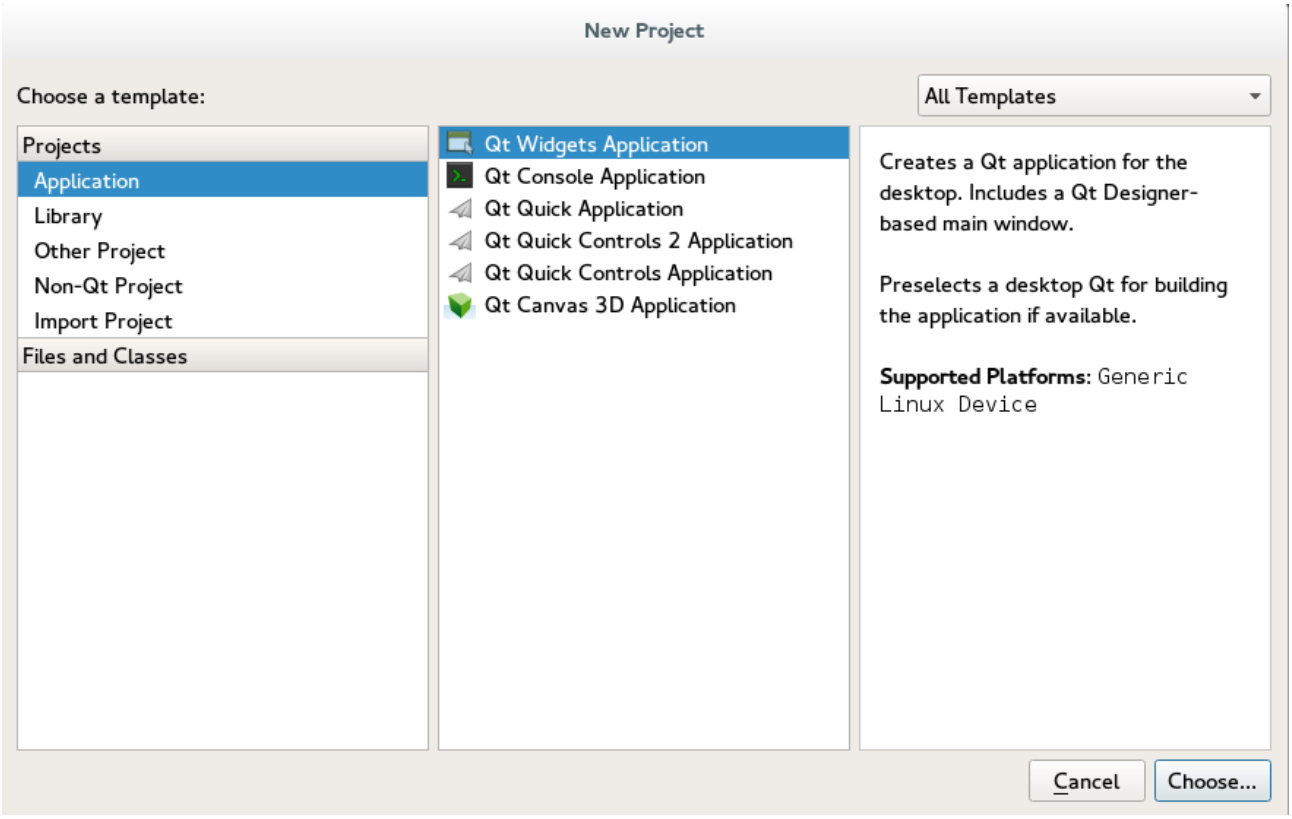
测试通过后的配置如下：

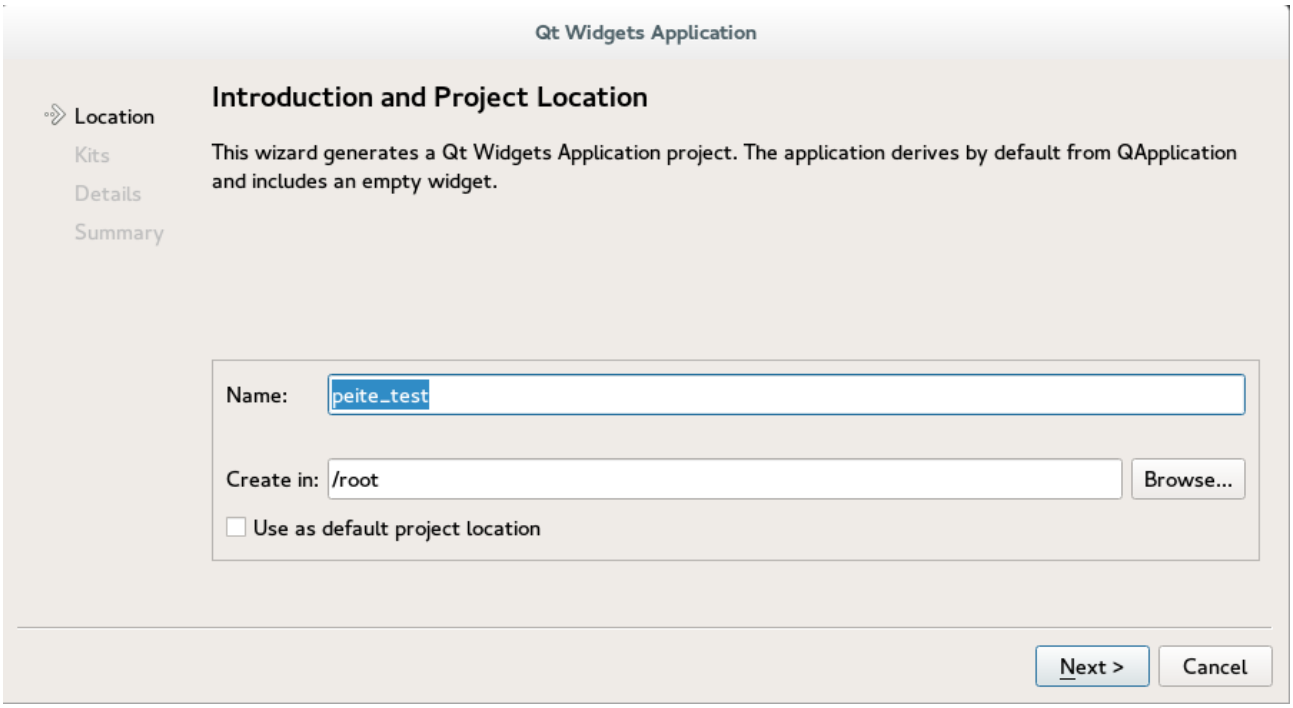


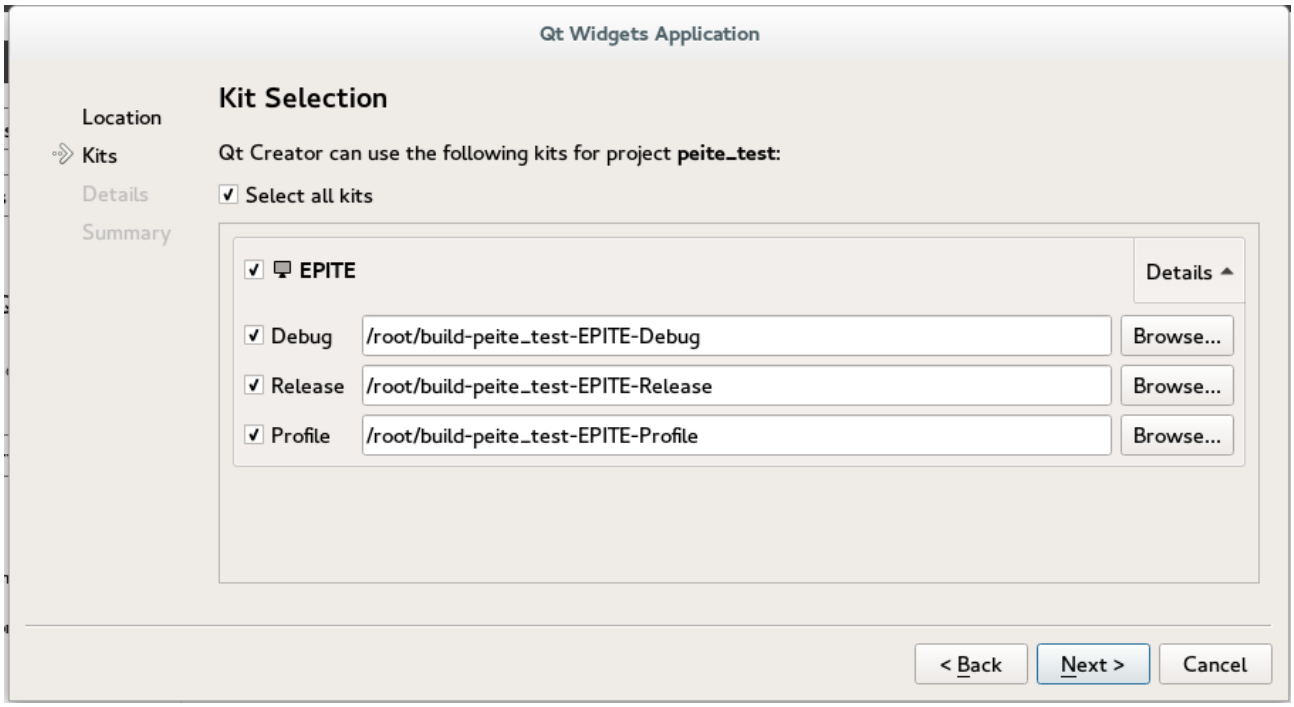
选择菜单 Tools->Options->Build & Run->Kits，点击 Add 按钮，配置如下：

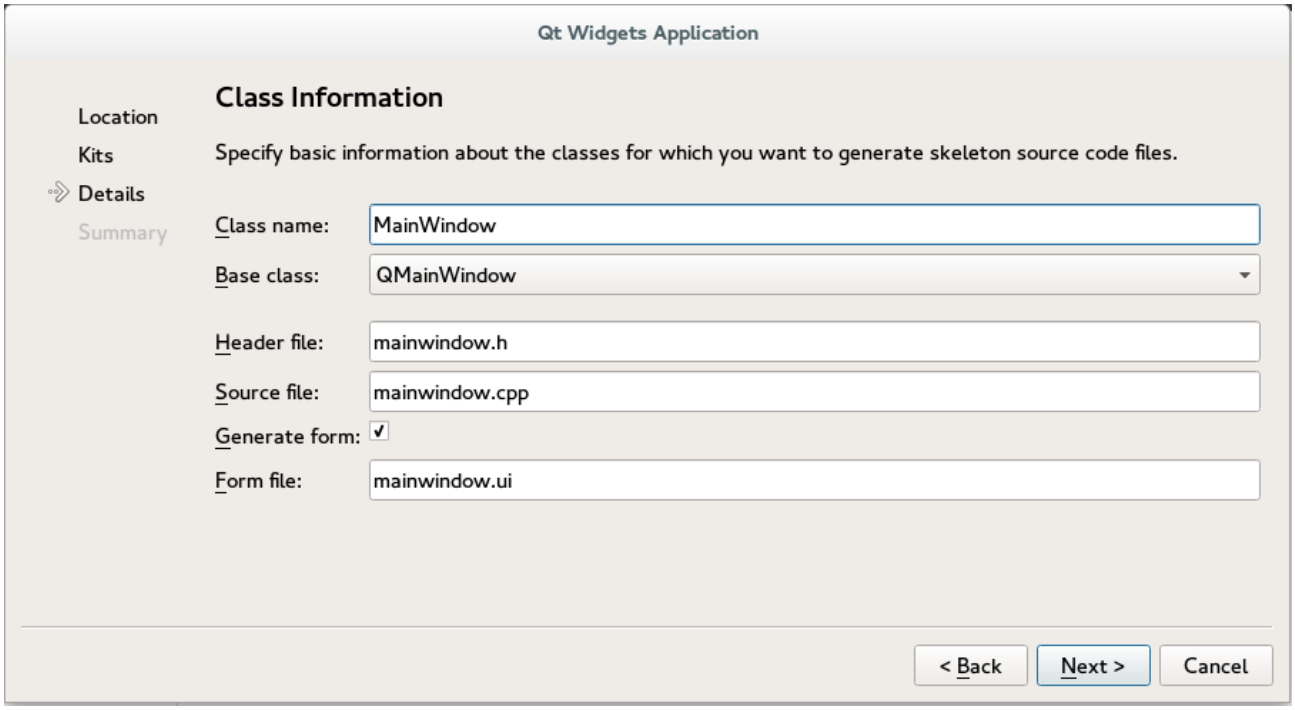


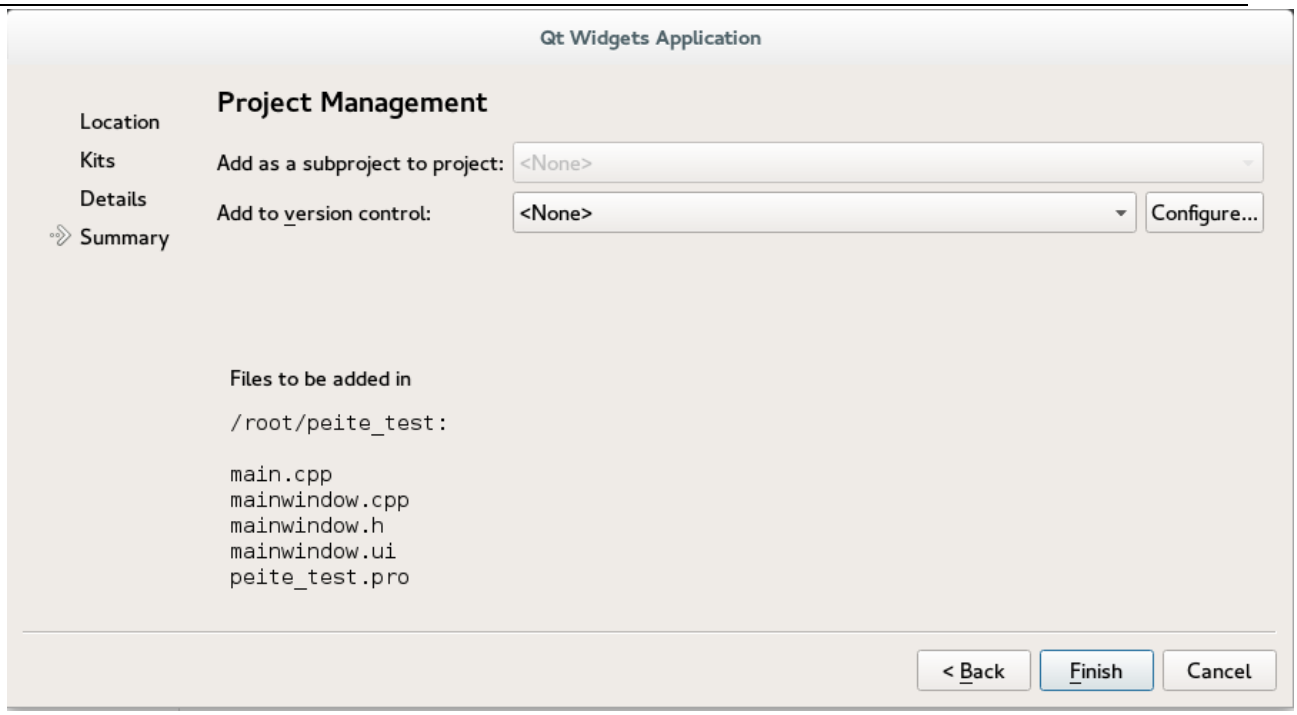
八、创建并编译 QT 程序


New Project
Choose a template:
All Templates
Projects
Application
Library
Other Project
Non-Qt Project
Import Project
Files and Classes
Qt Widgets Application
Qt Console Application
Qt Quick Application
Qt Quick Controls 2 Application
Qt Quick Controls Application
Qt Canvas 3D Application
Creates a Qt application for the desktop. Includes a Qt Designer-based main window.
Preselects a desktop Qt for building the application if available.
Supported Platforms: Generic Linux Device
Cancel
Choose...


Qt Widgets Application
Location
Kits
Details
Summary
Introduction and Project Location
This wizard generates a Qt Widgets Application project. The application derives by default from QApplication and includes an empty widget.
Name:
peite_test
Create in:
/root
Browse...
Use as default project location
Next >
Cancel

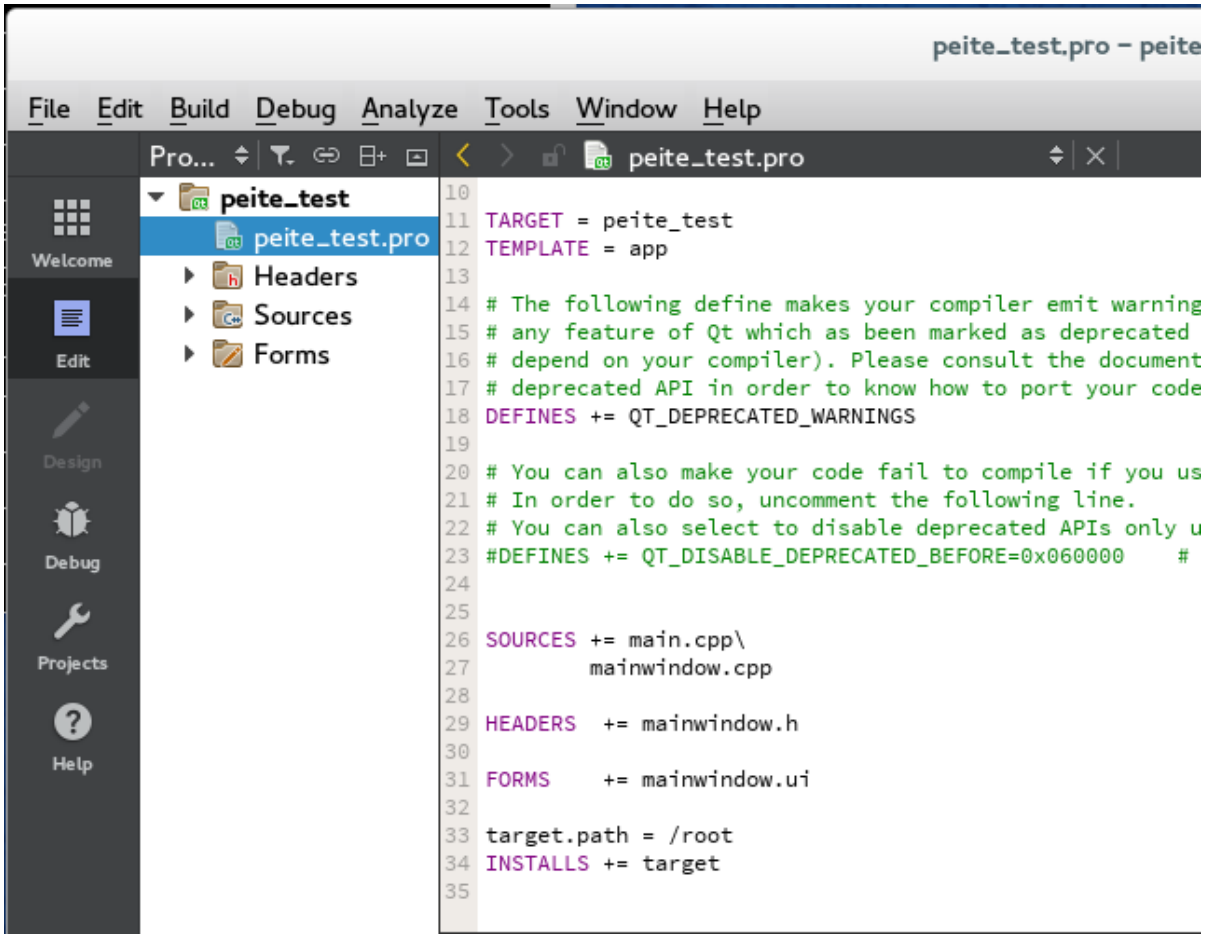

Qt Widgets Application
Location
Kits
Details
Summary
Kit Selection
Qt Creator can use the following kits for project peite_test:
Select all kits
EPITE
Details
Debug
/root/build-peite_test-EPITE-Debug
Browse...
Release
/root/build-peite_test-EPITE-Release
Browse...
Profile
/root/build-peite_test-EPITE-Profile
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:
MainWindow
Base class:
QMainWindow
Header file:
mainwindow.h
Source file:
mainwindow.cpp
Generate form:
Form file:
mainwindow.ui
< 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

2018 年 1 月