



# PET-R18 核心板/核心模组 开发手册

## 一、编译环境搭建指南

- 安装 Ubuntu 16.04 64 位。
- 安装依赖软件

```

sudo apt clean
sudo apt update
sudo apt -y upgrade
sudo apt -y dist-upgrade
sudo apt -y install openssh-server
sudo apt -y install git flex bison gperf build-essential libncurses5-dev:i386
sudo apt -y install libx11-dev:i386 libreadline6-dev:i386 libgl1-mesa-dev g++-multilib
sudo apt -y install tofrodos python-markdown libxml2-utils xsltproc zlib1g-dev:i386
sudo apt -y install dpkg-dev libsdl1.2-dev libesd0-dev
sudo apt -y install git-core gnupg flex bison gperf build-essential
sudo apt -y install zip curl zlib1g-dev gcc-multilib g++-multilib
sudo apt -y install libc6-dev-i386
sudo apt -y install lib32ncurses5-dev x11proto-core-dev libx11-dev
sudo apt -y install lib32z-dev ccache
sudo apt -y install libgl1-mesa-dev libxml2-utils xsltproc unzip m4
sudo apt -y install gawk fakeroot g++-multilib gcc-multilib
sudo apt -y install u-boot-tools make texinfo clang cmake dos2unix unix2dos
    
```

- 安装 openjdk-8-jdk。

```
sudo apt -y install openjdk-8-jdk
```

- 输入命令 `java -version` 检查 java 的主版本号是否为 1.8。
- 在开发工具目录下有安装好的虚拟机磁盘镜像文件，使用时内存最少需要 8G，编译安卓如果遇到内存不足错误可以加大内存容量或减少编译线程数量。

## 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_R18_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 sun50iw1p1_android -k linux-3.10
编译完成后正确提示如下：
    
```

```

make: Leaving directory `/root/work/A64_7.1.1_SKD_BASE/lichee/linux-3.10/modules/gpu'
[GPU]: mali400 device driver has been built.
regenerate rootfs cpio
15756 blocks
15756 blocks
build_ramfs
Copy boot.img to output directory ...
Copy modules to target ...

sun50iw1p1 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:~/work/A64_7.1.1_SKD_BASE/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 sun50iw1p1
编译完成后正确提示如下
/root/work/A64_7.1.1_SKD_BASE/lichee/brandy/u-boot-2014.07/./gcc-linaro
opy -O binary /root/work/A64_7.1.1_SKD_BASE/lichee/brandy/u-boot-2014.07
14_sunxi_spl/sunxi_spl/sbrom/sboot.axf /root/work/A64_7.1.1_SKD_BASE/lic
././bootloader/uboot_2014_sunxi_spl/sunxi_spl/sbrom/sboot.bin
/root/work/A64_7.1.1_SKD_BASE/lichee/brandy/u-boot-2014.07/././bootload
nxi_spl/././tools/add_hash.sh -f /root/work/A64_7.1.1_SKD_BASE/lichee/l
/bootloader/uboot_2014_sunxi_spl/sunxi_spl/sbrom/sboot.bin -m sboot
build_sboot
'sboot_sun50iw1p1.bin' -> '/root/work/A64_7.1.1_SKD_BASE/lichee/brandy/u-
ack/chips/sun50iw1p1/bin/sboot_sun50iw1p1.bin'
root@SeKeDe:~/work/A64_7.1.1_SKD_BASE/lichee/brandy#
    
```

## 3、编译 android

```

cd android
source build/envsetup.sh
lunch tulip_p3-eng
extract-bsp
make -j4
pack
编译完成后正确提示如下
    
```

```
config.fex Len: 0xc400
split_xxx.fex Len: 0x200
sys_partition.fex Len: 0x12c8
sunxi.fex Len: 0x14000
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: 0x4100
boot_package.fex Len: 0x130000
usbtool.fex Len: 0x23600
aultools.fex Len: 0x2847b
aultls32.fex Len: 0x24d23
cardtool.fex Len: 0x41600
cardscript.fex Len: 0x6de
sunxi_mbr.fex Len: 0x10000
dlinfo.fex Len: 0x4000
arisc.fex Len: 0x6
boot-resource.fex Len: 0x4ef400
Vboot-resource.fex Len: 0x4
env.fex Len: 0x20000
Venv.fex Len: 0x4
boot.fex Len: 0xf30800
Vboot.fex Len: 0x4
system.fex Len: 0x2e768144
Vsystem.fex Len: 0x4
recovery.fex Len: 0x110c800
Vrecovery.fex Len: 0x4
diskfs.fex Len: 0x200
Vdiskfs.fex Len: 0x4
BuildImg 0
Dragon execute image.cfg SUCCESS !
-----image is at-----

/root/work/A64_7.1.1_SKD_BASE/lichee/tools/pack/sun50iw1p1_android_p3_uart0.img
pack finish
```

编译完成后会在 `lichee/tools/pack` 目录下生成 `sun50iw1p1_android_p3_uart0.img` 系统烧写镜像文件。

## 四、编译 Linux + QT 5.9

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 sun50iw1p1_dragonboard -k linux-3.10
编译完成后正确提示如下
Copy boot.img to output directory ...
./scripts/build.sh
sun50iw1p1 compile kernel successful
INFO: build kernel OK.
INFO: build rootfs ...
Regenerating dragonboard rootfs...
extract dragonboard rootfs.tar.xz
INFO: Prepare dragonboard toolchain ...
cp: cannot stat 'extra/*': No such file or directory
generating rootfs...
blocks: 695M -> 1024M
Creating filesystem with parameters:
  Size: 1073741824
  Block size: 4096
  Blocks per group: 32768
  Inodes per group: 8192
  Inode size: 256
  Journal blocks: 4096
  Label:
  Blocks: 262144
  Block groups: 8
  Reserved block group size: 63
Created filesystem with 14921/65536 inodes and 186636/262144 blocks
e2fsck 1.44.1 (24-Mar-2018)
fsck.ext4: Bad magic number in super-block while trying to open rootfs.ext4
success in generating rootfs
Build at: sun Apr  7 20:29:02 CST 2019
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 sun50iw1p1
编译完成后正确提示如下

```

```

/root/work/A64_7.1.1_SKD_BASE/lichee/brandy/u-boot-2014.07/../../gcc-linaro
opy -O binary /root/work/A64_7.1.1_SKD_BASE/lichee/brandy/u-boot-2014.07
14_sunxi_spl/sunxi_spl/sbrom/sboot.axf /root/work/A64_7.1.1_SKD_BASE/lic
../../bootloader/uboot_2014_sunxi_spl/sunxi_spl/sbrom/sboot.bin
/root/work/A64_7.1.1_SKD_BASE/lichee/brandy/u-boot-2014.07/../../bootload
nxi_spl/../../tools/add_hash.sh -f /root/work/A64_7.1.1_SKD_BASE/lichee/
/bootloader/uboot_2014_sunxi_spl/sunxi_spl/sbrom/sboot.bin -m sboot
build_sboot
'sboot_sun50iw1p1.bin' -> '/root/work/A64_7.1.1_SKD_BASE/lichee/brandy/u-
ack/chips/sun50iw1p1/bin/sboot_sun50iw1p1.bin'
root@SeKeDe:~/work/A64_7.1.1_SKD_BASE/lichee/brandy#

```

### 3、编译 QT Rootfs

首先切换到 pack 目录

```
cd lichee/tools/pack/
```

```
./pack -c sun50iw1p1 -p dragonboard -b p3
```

编译完成后正确提示如下

```

config.fex Len: 0xc400
split_xxxx.fex Len: 0x200
sys_partition.fex Len: 0xaa0
sunxi.fex Len: 0x14000
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: 0x4100
boot_package.fex Len: 0x130000
usbtool.fex Len: 0x23600
aultools.fex Len: 0x2847b
aultls32.fex Len: 0x24d23
cardtool.fex Len: 0x41600
cardscript.fex Len: 0x6de
sunxi_mbr.fex Len: 0x10000
dlinfno.fex Len: 0x4000
arisc.fex Len: 0x6
boot-resource.fex Len: 0x4f4c00
Vboot-resource.fex Len: 0x4
env.fex Len: 0x20000
Venv.fex Len: 0x4
boot.fex Len: 0xfa7800
Vboot.fex Len: 0x4
rootfs.fex Len: 0x2cc5b6dc
Vrootfs.fex Len: 0x4
BuildImg 0
Dragon execute image.cfg SUCCESS !
-----image is at-----
/root/work/A64_7.1.1_SKD_BASE/lichee/tools/pack/sun50iw1p1_dragonboard_p3_uart0.img
pack finish

```

编译完成后会在 lichee/tools/pack 目录下生成 sun50iw1p1\_dragonboard\_p3\_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位于开发资料《开发工具/交叉编译器》目录下：

```
gcc-linaro-7.2.1-2017.11-x86_64_aarch64-linux-gnu.tar.xz
```

编译所需的其他库文件是 sysroot\_peite.tar.xz(交叉编译器目录内)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

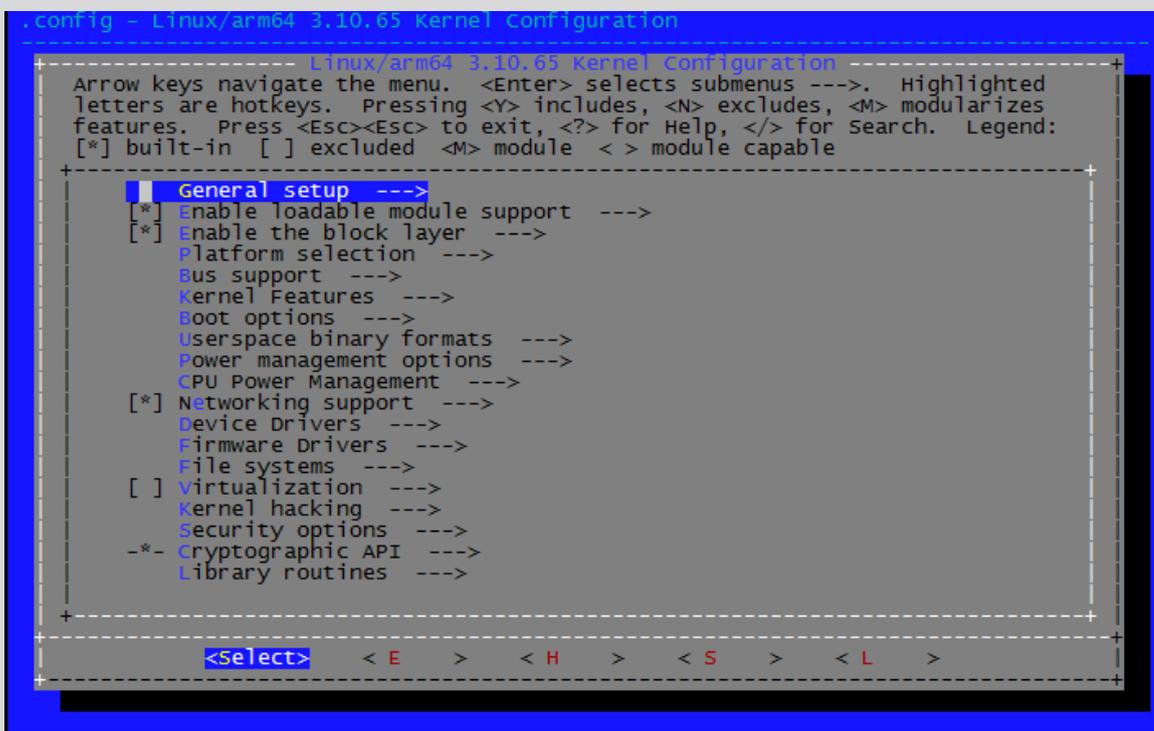
```
cd lichee/linux-3.10/
```

加载默认配置

```
make sun50iw1p1smp_android_defconfig
```

启动内核配置

```
make menuconfig
```



```
.config - Linux/arm64 3.10.65 Kernel Configuration
----- Linux/arm64 3.10.65 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
+-----+
| General setup ---> |
| [*] Enable loadable module support ---> |
| [*] Enable the block layer ---> |
| Platform selection ---> |
| Bus support ---> |
| Kernel Features ---> |
| Boot options ---> |
| Userspace binary formats ---> |
| Power management options ---> |
| CPU Power Management ---> |
| [*] Networking support ---> |
| Device Drivers ---> |
| Firmware Drivers ---> |
| File systems ---> |
| [ ] Virtualization ---> |
| Kernel hacking ---> |
| Security options ---> |
| *- Cryptographic API ---> |
| Library routines ---> |
+-----+
| <Select> < E > < H > < S > < L > |
+-----+
```

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/arm64/config/sun50iw1p1smp_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/arm64/config/sun50iw1p1smp_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。

**注意**，在点击烧写镜像后，设备会重启黑屏，如果没有开始烧写进程，此时需要在 PC 端的设备管理区中对黄色感叹号设备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/交叉编译器》目录下:

- 1、解压交叉编译器 gcc-linaro-4.9.4-2017.01-x86\_64\_arm-linux-gnueabi.tar.xz

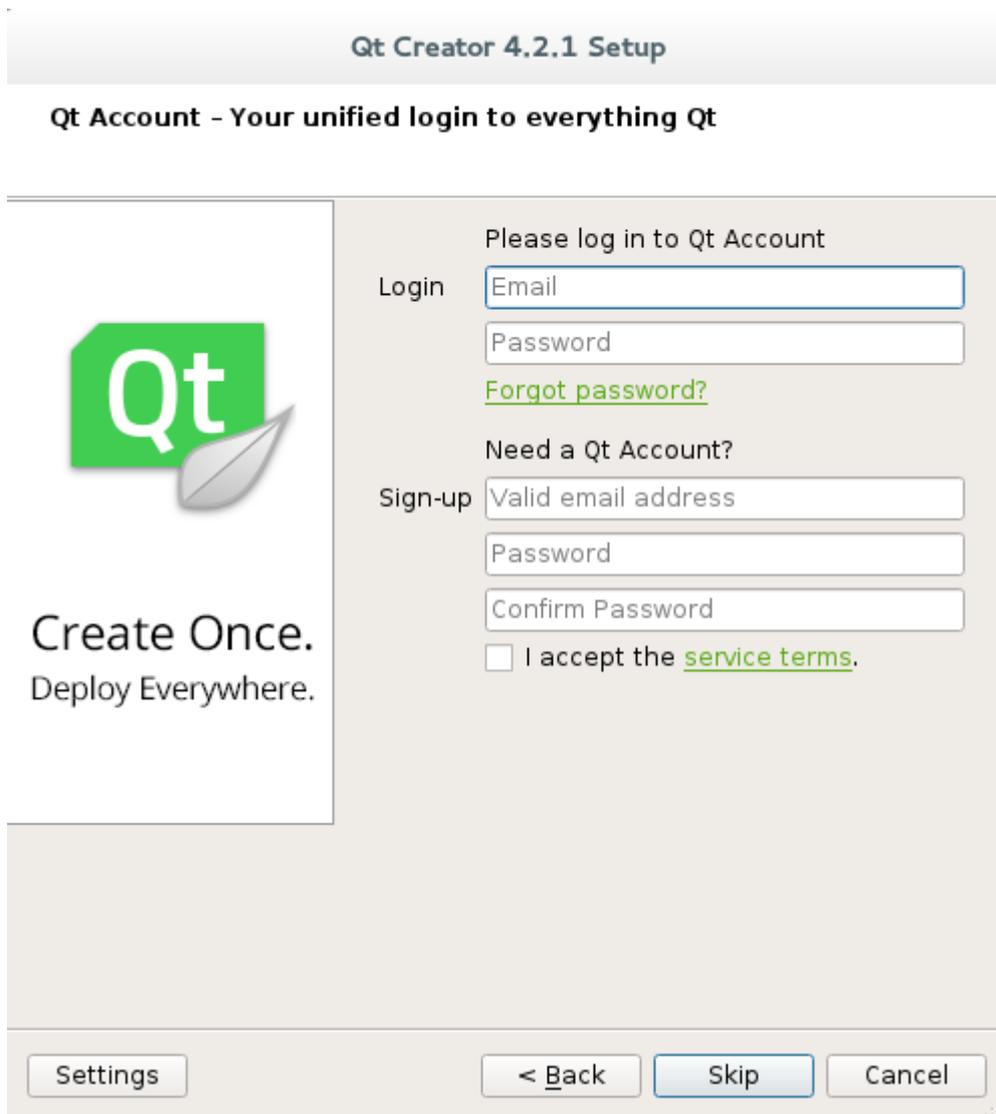
```
sudo mkdir -p /usr/local/arm
sudo tar -xJf gcc-linaro-7.2.1-2017.11-x86_64_aarch64-linux-gnu.tar.xz -C /usr/local/arm
```

- 2、解压库文件 sysroot\_peite\_qt.tar.xz

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

- 3、安装 qt-creator-opensource-linux-x86\_64-4.4.1.run

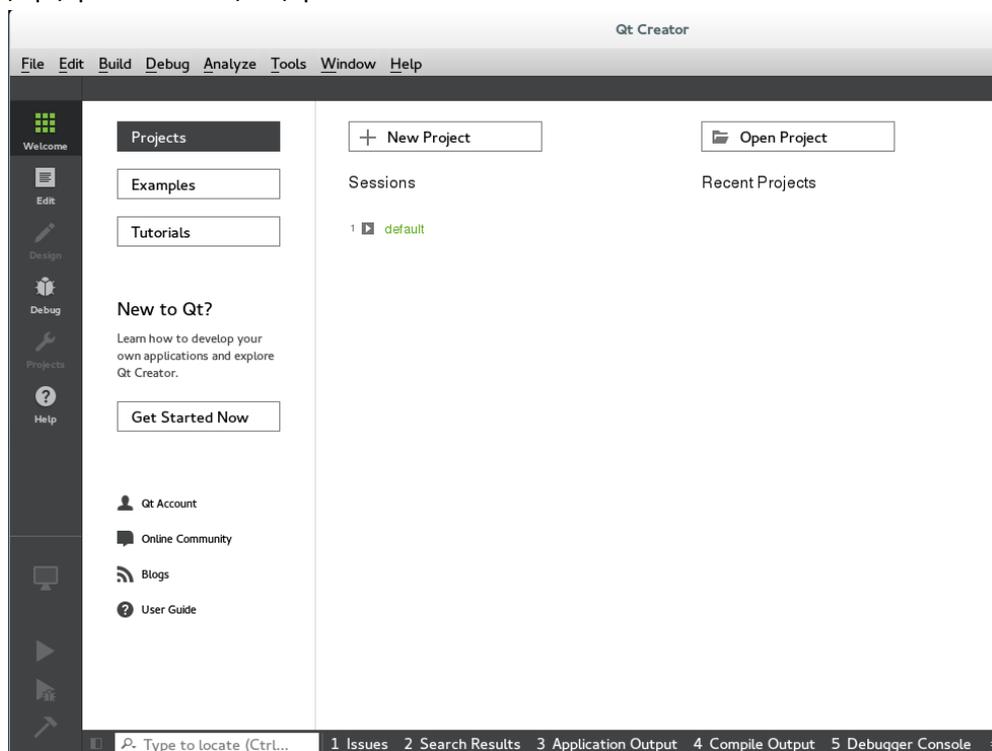
```
chmod +x qt-creator-opensource-linux-x86_64-4.4.1.run
./qt-creator-opensource-linux-x86_64-4.4.1.run
```



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。

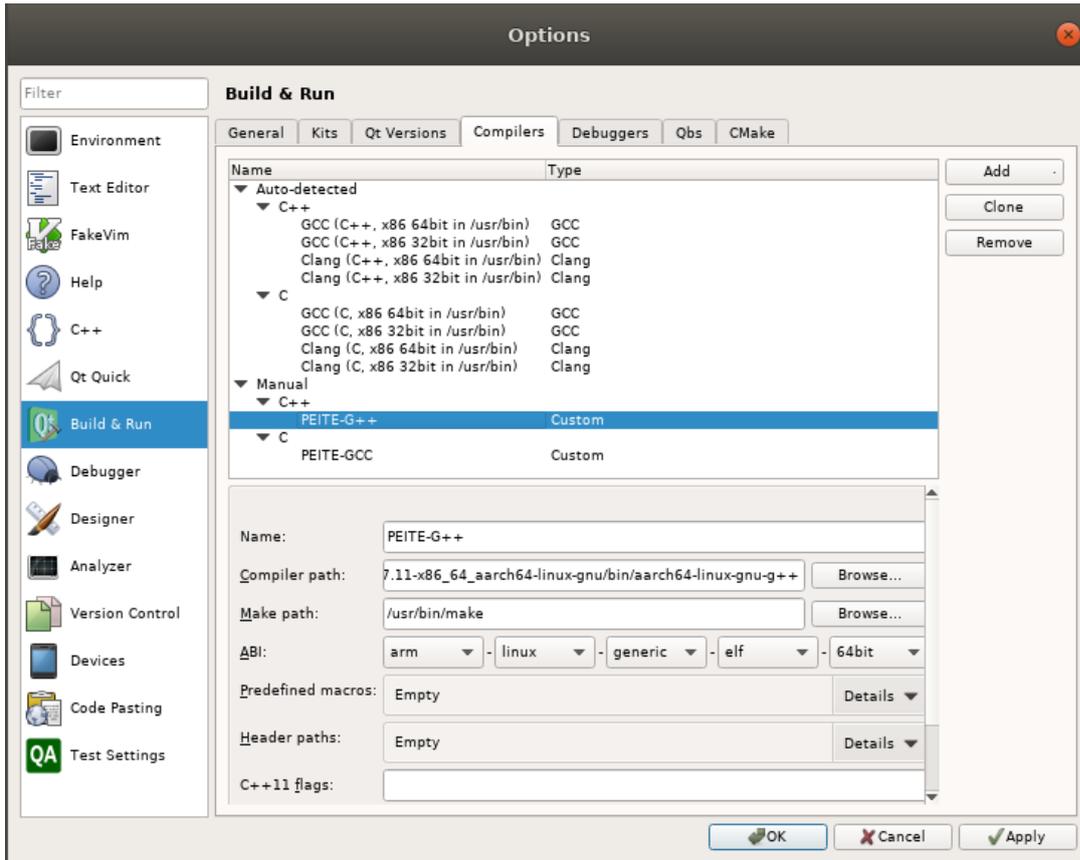
/opt/qtcreator-4.4.1/bin/qtcreator



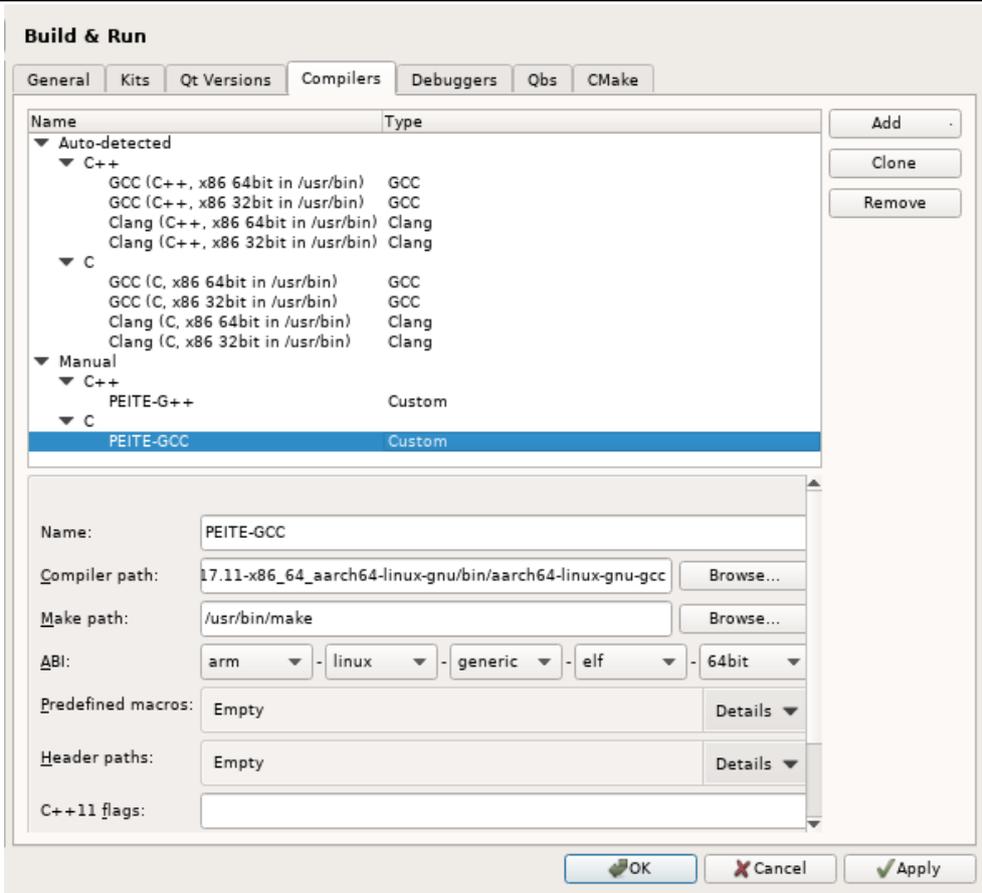
选择菜单 Tools->Options->Build & Run->Compilers, 点击 Add ->Custom->C++ 按钮, 添加 C++编译器,

Compiler path:

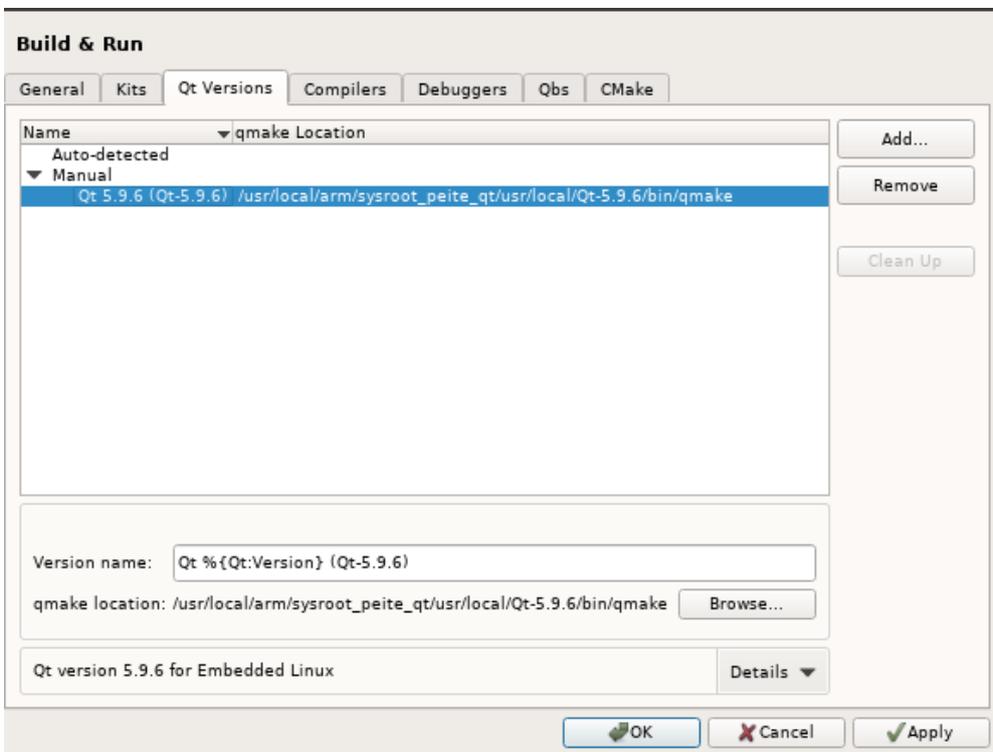
/usr/local/arm/gcc-linaro-7.2.1-2017.11-x86\_64\_aarch64-linux-gnu/bin/ aarch64-linux-gnu-g++



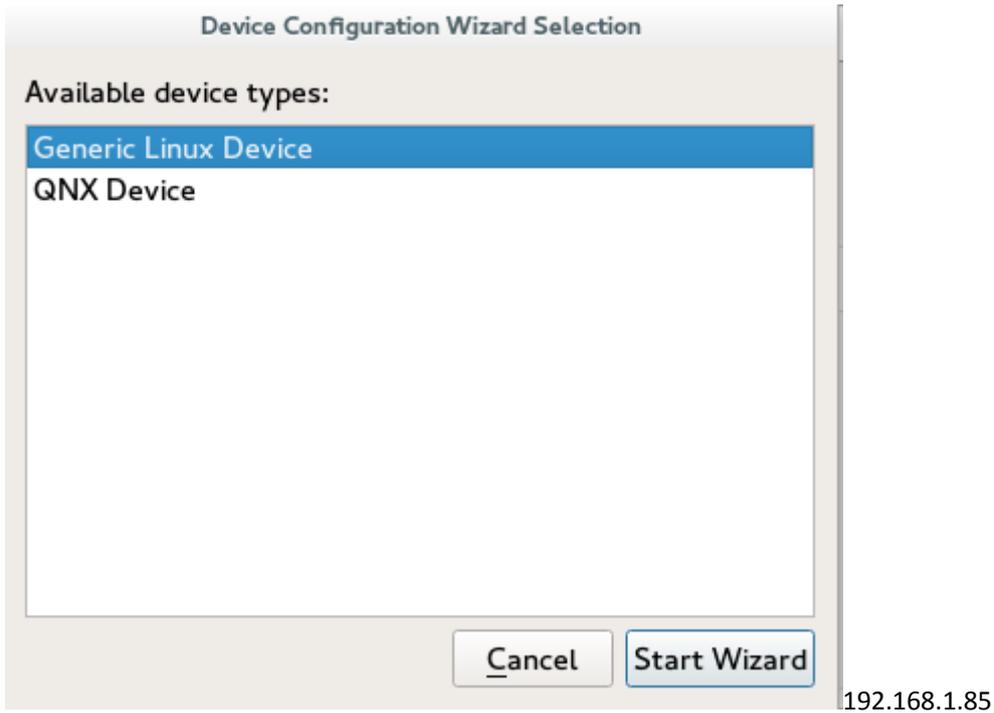
Add ->Custom->C 添加 C 编译器 Compiler path:  
/usr/local/arm/gcc-linaro-7.2.1-2017.11-x86\_64\_aarch64-linux-gnu/bin/ aarch64-linux-gnu-gcc



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



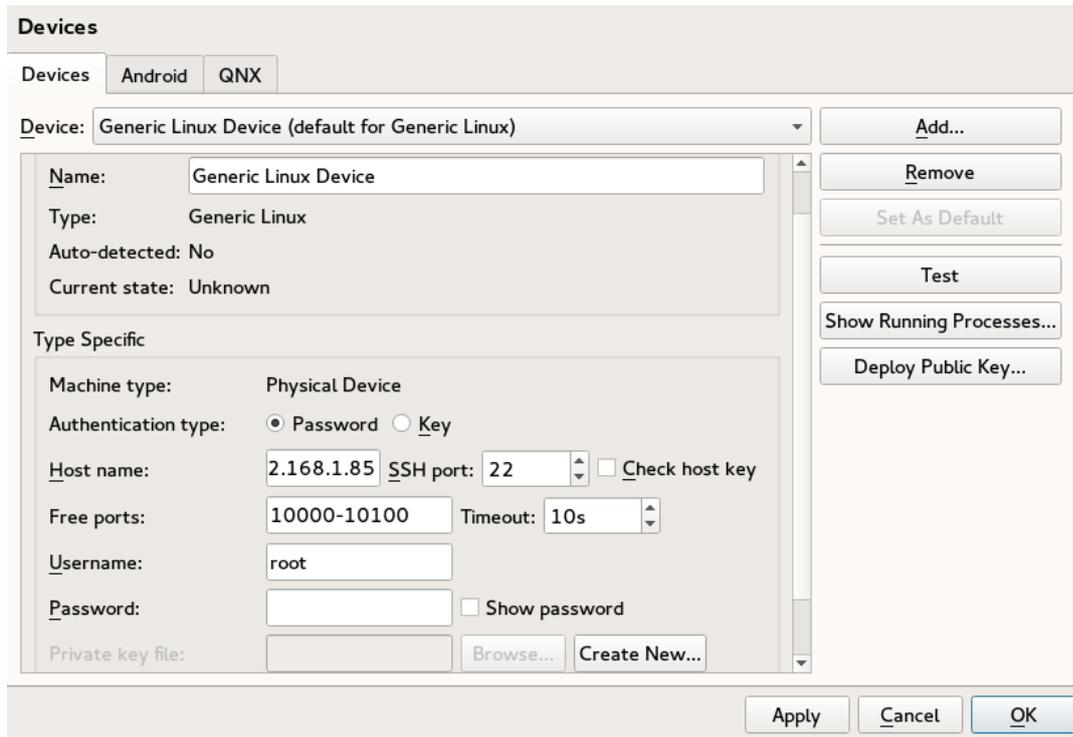
选择菜单 Tools->Options->Devices, 点击 Add 按钮, 配置如下:



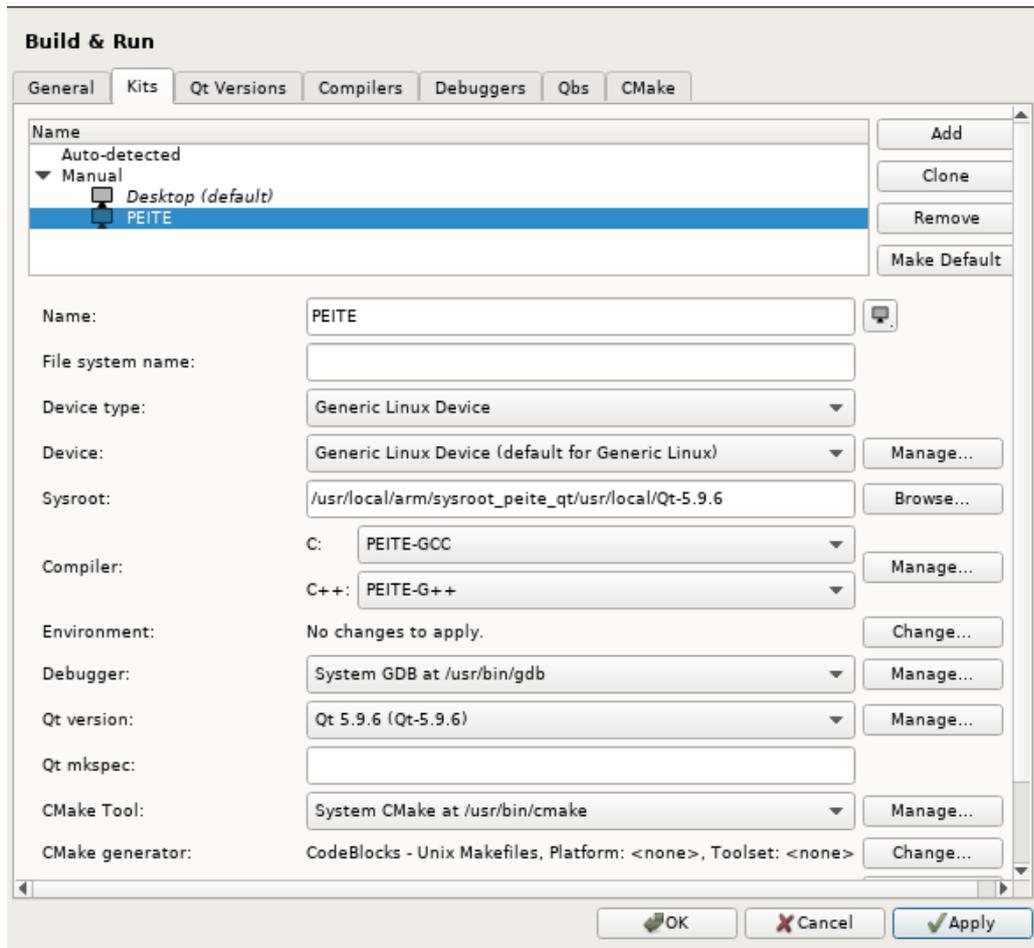
首先需要将开发板与主机在同一局域网内连接好，主机可以正常 ping 通开发板



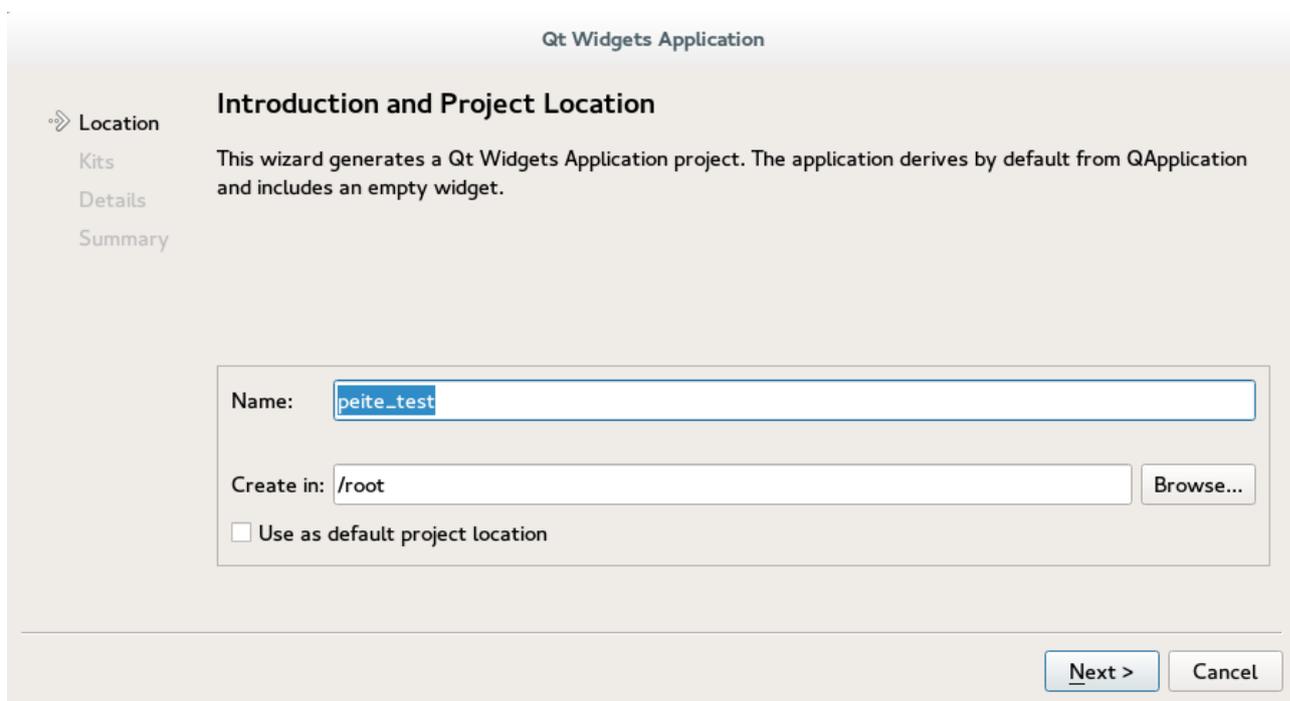
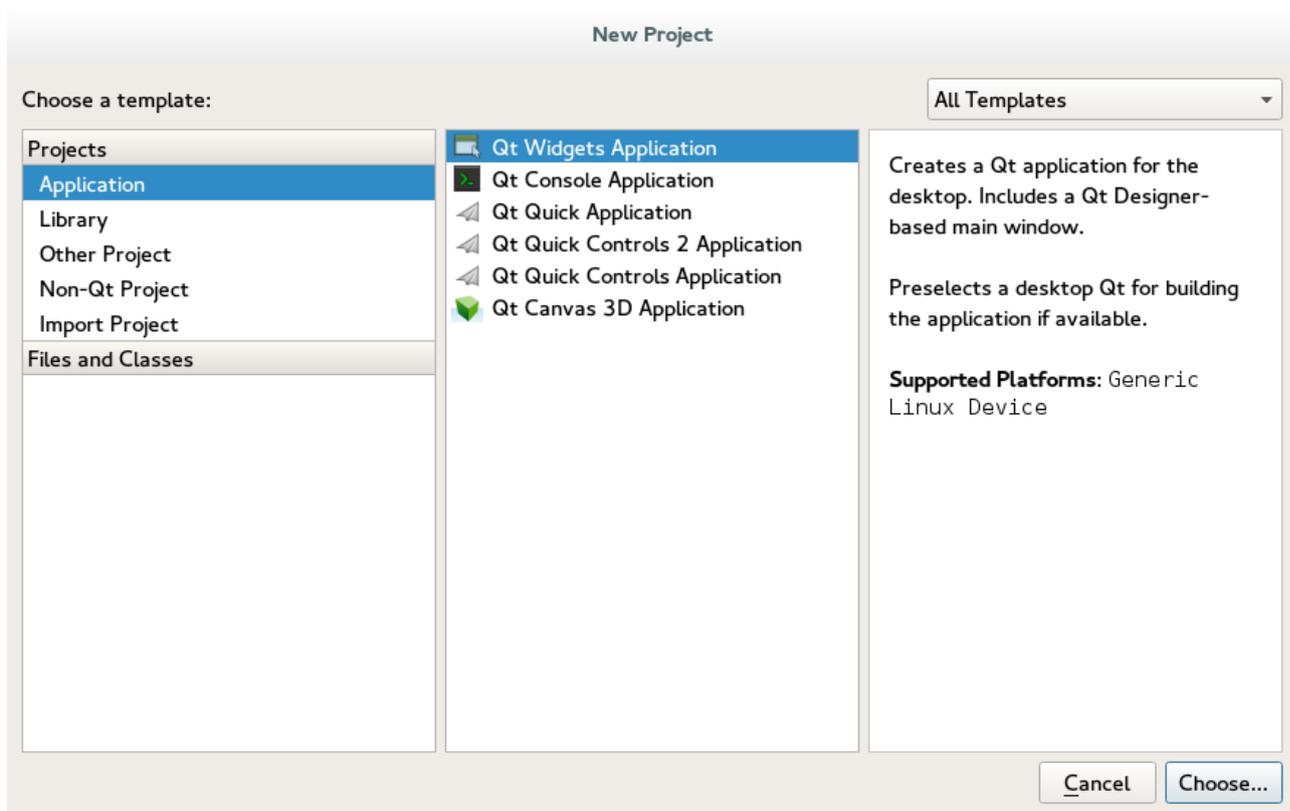
测试通过后的配置如下：



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:

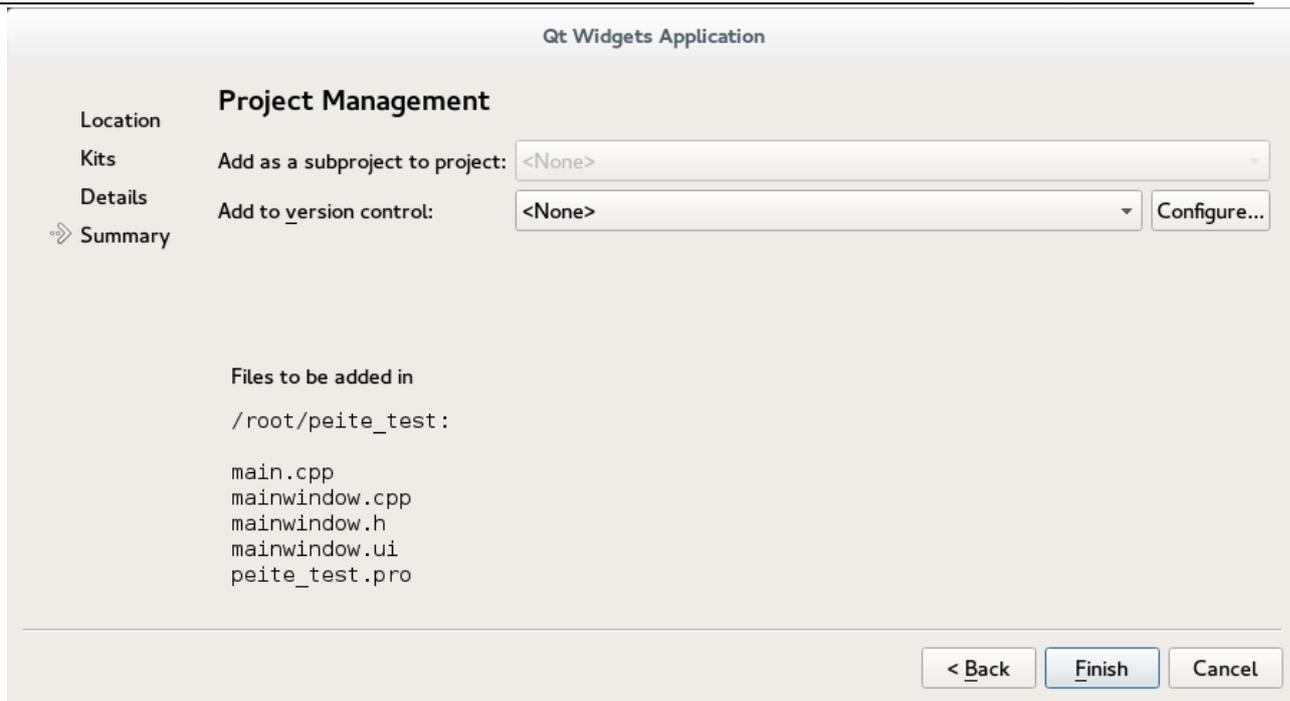
Header file:

Source file:

Generate form:

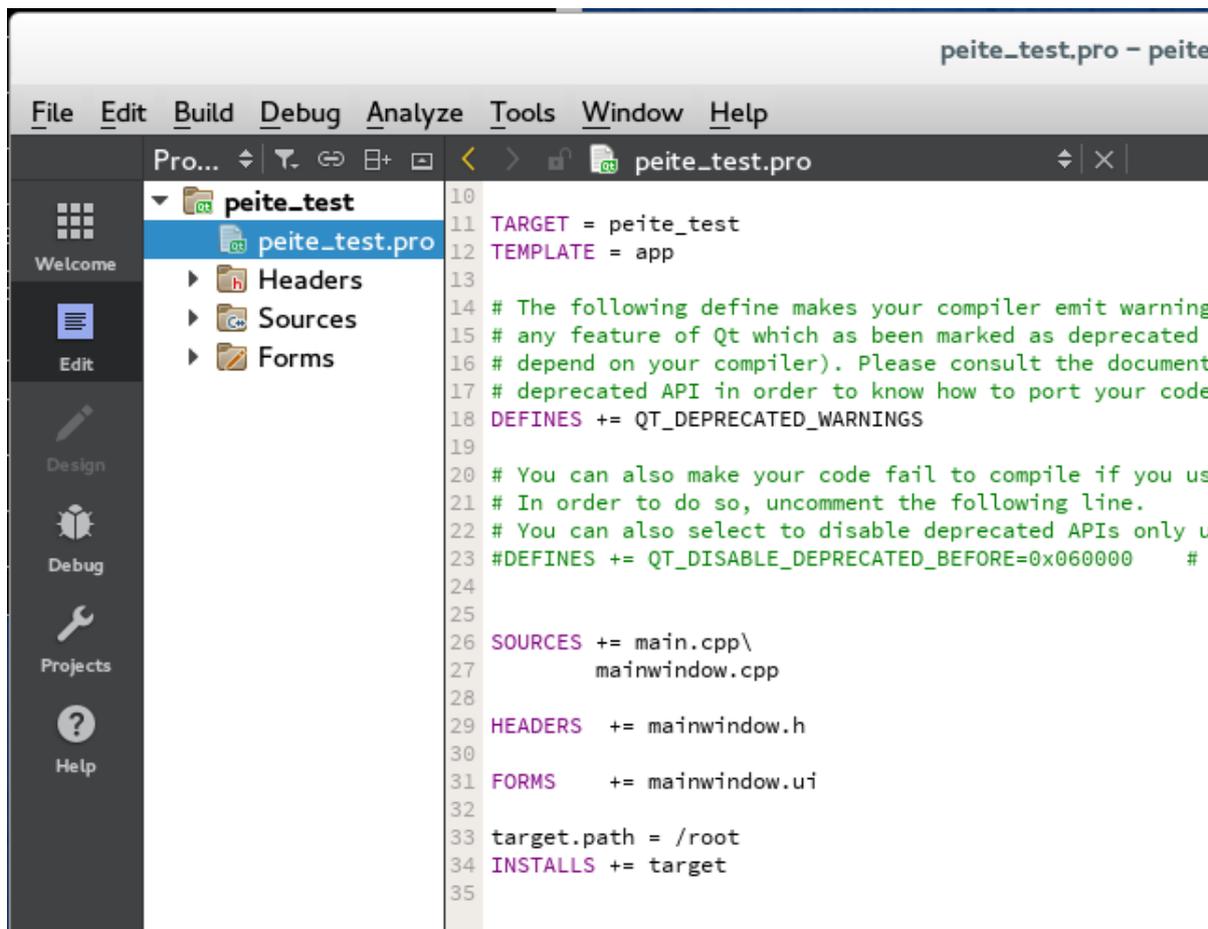
Form file:

< 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/arm/sysroot\_peite\_qt/usr/local/Qt-5.9.6

## 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

2019 年 4 月