

iTOP-4418&6818-QtE5.7 移植

本文档介绍如何移植高版本的 Qt5.7 到 iTOP-4418&iTOP-6818 开发板，注意：使用本节需要先学习前面编译烧写等基础知识。

对于 QtE5.7 的移植，本节中提供了对应的编译环境，用户如果是第一次移植，建议使用我们提供的版本。

本节中需要使用的大部分重要的软件、工具以及源码在网盘“iTOP4418 开发板资料汇总（不含光盘内容）\04_iTOP-4418 开发板 QtE 和 Qtopia 系统源码以及镜像\05_QtE5.7 镜像和源码以及编译器”目录下“iTOP-4418&6818-QtE5.7 移植”压缩包中可以找到。

1.1 基础软件的下载

移植 QtE5.7 需要 Ubuntu16.04 环境，arm 编译器需要使用“arm-2014.05-29”，触摸 tslib 需要使用 1.4 版本，QtE 的源码为 QtE5.7.0。

下面是具体的下载链接和详细的版本信息。

环境：ubuntu16.04（Qt5.7creator 需要在 16.04 版本上运行）

下载链接：<http://cn.ubuntu.com/download/>

编译器：arm-2014.05-29-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2

下载链接：<http://www.veryarm.com/arm-none-linux-gnueabi-gcc>

触摸工具：tslib-1.4.tar.gz

下载链接：<http://vdisk.weibo.com/s/AcUHxOERV2k0?sudaref=www.baidu.com>

Qt 源码：qt-everywhere-opensource-src-5.7.0.tar.gz

下载链接：http://download.qt.io/official_releases/qt/5.7/5.7.0/

2.2 搭建环境

2.2.1 安装基本工具 vim 和 ssh

本节介绍在 Ubuntu16.04 系统上安装基础软件 vim 和 ssh。

使用命令 “apt-get install vim ssh”

使用 ssh 的时候，会遇到 ssh 连接失败的问题。

解决办法：

先使用命令 “/etc/init.d/ssh restart” 启动 ssh 服务，然后使用命令 “vim /etc/ssh/sshd_config” 打开配置文件。

在配置文件中添加：

```
Ciphers aes128-cbc,aes192-cbc,aes256-cbc,aes128-ctr,aes192-ctr,aes256-ctr,3des-cbc,arcfour128,arcfour256,arcfour,blowfish-cbc,cast128-cbc
MACs hmac-md5,hmac-sha1,umac-64@openssh.com,hmac-ripemd160,hmac-sha1-96,hmac-md5-96
KexAlgorithms diffie-hellman-group1-sha1,diffie-hellman-group14-sha1,diffie-hellman-group-exchange-sha1,diffie-hellman-group-exchange-sha256,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group1-sha1,curve25519-sha256@libssh.org
```

找到下面这段代码，修改如下

```
# Authentication:
LoginGraceTime 120
#PermitRootLogin prohibit-password
PermitRootLogin yes
StrictModes yes
```

之后再重启 service ssh restart 。

2.2.2 安装编译器

将编译器拷贝到 “/usr/local/arm” 目录下，并解压。

修改环境变量 “vim /etc/profile ” 并使用 source 命令更新环境变量，如下图。

```
export PATH=$PATH:/usr/local/arm/arm-2014.05/bin
export CROSS_COMPILE=arm-none-linux-gnueabi-
```

30,

arm+tab 键，确定可以显示 arm-none-linux-gnueabi-

输入 arm-none-linux-gnueabi-gcc -v 显示 gcc version 4.8.3 20140320

(prerelease) (Sourcery CodeBench Lite 2014.05-29)，那就没问题了。如果提示 no such 等，要 sudo apt-get install lib32z1 lib32ncurses5。

2.3 编译 tslib

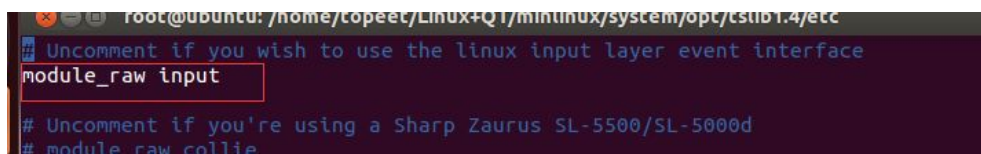
解压源码 tslib-1.4.tar.gz

```
cd tslib
sudo apt-get install autoconf
sudo apt-get install automake
sudo apt-get install libtool
```

安装完之后执行

```
./autogen.sh
./configure CC=arm-none-linux-gnueabi-gcc CXX=arm-none-linux-gnueabi-g++
--host=arm-none-linux-gnueabi --prefix=/opt/tslib1.4 ac_cv_func_malloc_0_nonnull=yes
make
make install
```

vim /opt/tslib1.4/etc/ts.conf 修改第二行如下



```
root@ubuntu: /home/topeet/Linux+QT/mintlinux/system/opt/tslib1.4/etc
# Uncomment if you wish to use the linux input layer event interface
module_raw input
# Uncomment if you're using a Sharp Zaurus SL-5500/SL-5000d
# module_raw collie
```

2.4 编译 Qt5.7 的源码

使用 “tar -vxf qt-everywhere-opensource-src-5.7.0” 命令解压 Qt5.7 源码，并进入解压生成的 “qt-everywhere-opensource-src-5.7.0” 目录。

在 “qt-everywhere-opensource-src-5.7.0” 目录下，使用 “vi qtbase/mkspecs/linux-arm-gnueabi-g++/qmake.conf” 命令，打开 qmake.conf 文件并修改为如下所示内容。

```
#
```

```
# qmake configuration for building with arm-linux-gnueabi-g++
#
MAKEFILE_GENERATOR      = UNIX
CONFIG                  += incremental
QMAKE_INCREMENTAL_STYLE = sublib

QT_QPA_DEFAULT_PLATFORM = linux #eglfs

QMAKE_CFLAGS_RELEASE += -O2 -march=armv7-a

QMAKE_CXXFLAGS_RELEASE += -O2 -march=armv7-a

include(../common/linux.conf)
include(../common/gcc-base-unix.conf)include(../common/g++-unix.conf)

# modifications to g++.conf
QMAKE_CC              = arm-none-linux-gnueabi-gcc
QMAKE_CXX              = arm-none-linux-gnueabi-g++
QMAKE_LINK              = arm-none-linux-gnueabi-g++
QMAKE_LINK_SHLIB        = arm-none-linux-gnueabi-g++

# modifications to linux.conf
QMAKE_AR                = arm-none-linux-gnueabi-ar cqs
QMAKE_OBJCOPY            = arm-none-linux-gnueabi-objcopy
QMAKE_NM                = arm-none-linux-gnueabi-nm -P
QMAKE_STRIP              = arm-none-linux-gnueabi-strip
load(qt_config)
```

如下图。

```
root@ubuntu:/home/topeet/neo/qt5.7/qt-everywhere-opensource-src-5.7.0# vi qtbase/mkspecs/linux-arm-gnueabi-g++/qmake.conf
```

使用“vi autoconfigure.sh”命令，新建“autoconfigure.sh”脚本。脚本内容如下所示。

```
#!/bin/sh
./configure \
-v \
-prefix /opt/qt5.7.0 \
-release \
-opensource \
-make libs \
```

```
-xplatform linux-arm-gnueabi-g++ \  
-optimized-qmake \  
-pch \  
-qt-sql-sqlite \  
-qt-libjpeg \  
-qt-zlib \  
-no-opengl \  
-skip qt3d \  
-skip qtcanvas3d \  
-skip qtpurchasing \  
-no-sse2 \  
-no-openssl \  
-no-nis \  
-no-cups \  
-no-glib \  
-no-iconv \  
-nomake examples \  
-nomake tools \  
-skip qtvirtualkeyboard \  
-I/opt/tslib1.4/include \  
-L/opt/tslib1.4/lib  
  
exit
```

保存，退出。

使用 “chmod 777 autoconfigure.sh ” 命令修改 “autoconfigure.sh” 文件权限。

然后使用 “./autoconfigure.sh” 命令运行该脚本。

运行完成后，在当前目录下，继续使用 “make” 命令进行编译。make 运行结束后，执行 “make install” 命令。

完成后，可以在/opt 目录下可以查看到生成的 “qt5.7.0” 文件（接下来要用到）。

2.5 生成 system.img

使用提供好的根文件系统（“4418/6818 最小系统”，以 4418 为例），将前两节生成的“qt5.7.0”“tslib1.4”拷贝到“opt”目录下如下图。

```
root@ubuntu:/home/topeet/4418_6818_qt5.7/4418/minilinux/system/opt# ls
qt5.7.0  tslib1.4
root@ubuntu:/home/topeet/4418_6818_qt5.7/4418/minilinux/system/opt#
```

解压“fonts.zip”，将解压生成的将字库文件夹“fonts”拷贝到“opt/qt5.7.0/lib/”下。

同时将“libstdc.tar.gz”拷贝到“opt/qt5.7.0/lib/”目录下解压，并输入命令：

```
ln -s libstdc++.so.6.0.19 libstdc++.so.6
```

修改环境变量“vim etc/profile”如下图。

```
# Ash profile
# vim: syntax=sh

# No core files by default
ulimit -S -c 0 > /dev/null 2>&1

USER=`id -un`
LOGNAME=$USER
PS1='[$USER@$HOSTNAME]# '
PATH=$PATH

HOSTNAME=`/bin/hostname`

export USER LOGNAME PS1 PATH

export TSLIB_ROOT=/opt/tslib1.4
export QT_ROOT=/opt/qt5.7.0
export TSLIB_TSDEVICE=/dev/input/event2
export TSLIB_TSEVENTTYPE=input
export TSLIB_CONFFILE=/opt/tslib1.4/etc/ts.conf
export TSLIB_PLUGININDIR=/opt/tslib1.4/lib/ts
export TSLIB_CONSOLEDEVICE=none
export TSLIB_FBDEVICE=/dev/fb0
export QWS_MOUSE_PROTO=tslib:/dev/input/event2
export LD_LIBRARY_PATH=/lib:/usr/lib:/usr/local/lib:$QT_ROOT/lib:$TSLIB_ROOT/lib:$TSLIB_ROOT/lib/
export QT_QPA_PLATFORM_PLUGIN_PATH=$QT_ROOT/plugins
```

```
export QT_QPA_PLATFORM=linuxfb:tty=/dev/fb0
export QT_QPA_FONTDIR=$QT_ROOT/lib/fonts
export QT_QPA_GENERIC_PLUGINS=tslib
~
```

修改开机启动脚本 “vi etc/init.d/rcS” , 最后加一行 “/bin/qt5.7 &” 。

将迅为提供的 “qt5.7” 文件拷贝到 “bin” 下。

参照 “Linux 系统编程_v2.1.pdf” 编译最小 linux , 打包生成 system.img。若出现

```
make_ext4fs: error while loading shared libraries: libstdc++.so.6: cannot open shared object file: No such file or
directory
```

一般出现该错误是由于应用程序是 32 位导致的 , 可以使用 file 命令来查看 :

```
root@ubuntu:/usr/local/bin# ls
hostapd hostapd_cli make_ext4fs mkimage
root@ubuntu:/usr/local/bin# file make_ext4fs
make_ext4fs: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamically linked (uses shared
libs), for GNU/Linux 2.6.8, stripped
```

从上面可以看到 make_ext4fs 为 32 位程序 , 所以我们需要让 64 位机支持运行 32 位应用。 执行下面两条命令即可解决问题 :

```
apt-get install lib32c-dev
apt-get install lib32stdc++6
```

再运行 make_ext4fs 就不会报错了。输入命令打包系统镜像命令 :

```
make_ext4fs -s -l 314572800 -a root -L linux system.img system
```

内核使用最小 linux 即可。烧写镜像到开发板。

使用迅为提供的 “hello” 测试程序进行测试 , 成功。

2.6 安装 qtcreator 编译测试程序 hello

将 “qt-opensource-linux-x64-5.7.0.run” 拷贝到 ubuntu16.04 任意文件夹下 ,

“chmod 777 qt-opensource-linux-x64-5.7.0.run” , 之后直接 “./qt-opensource-linux-x64-5.7.0.run” 执行 , 安装一直点下一步就 OK !

然后就可以通过 qtcreator 编译 hello 测试程序了 , 编译方法和 QtE4.7.1 版本的类似。

联系方式

北京迅为电子有限公司致力于嵌入式软硬件设计，是高端开发平台以及移动设备方案提供商；基于多年的技术积累，在工控、仪表、教育、医疗、车载等领域通过 OEM/ODM 方式为客户创造价值。

iTOP-6818 开发板是迅为电子基于三星最新四核处理器 Exynos6818 研制的一款实验开发平台，可以通过该产品评估 Exynos 6818 处理器相关性能，并以此为基础开发出用户需要的特定产品。

本手册主要介绍 iTOP-6818 开发板的使用方法，旨在帮助用户快速掌握该产品的应用特点，通过对开发板进行后续软硬件开发，衍生出符合特定需求的应用系统。

如需平板电脑案支持，请访问迅为平板方案网“<http://www.topeet.com>”，我司将有能力为您提供全方位的技术服务，保证您产品设计无忧！

本手册将持续更新，并通过多种方式发布给新老用户，希望迅为电子的努力能给您的学习和开发带来帮助。

迅为电子

2018 年 3 月