

君正®

君正 JZ4780 Android4.1 开发指南

Date: Mar. 2013



北京君正集成电路股份有限公司
Ingenic Semiconductor Co., Ltd.

君正®

JZ4780 Android4.1 开发指南

Copyright © Ingenic Semiconductor Co. Ltd 2005 - 2013. All rights reserved.

Release history

Date	Revision	Change
Feb. 2013	1.0.1	First release
Mar. 2013	1.0.2	1. Remove the operation detail of Burning. About Burning please reference Burn Tool Guide or quick start. 2. Add Warrior compile command.

Disclaimer

This documentation is provided for use with Ingenic products. No license to Ingenic property rights is granted. Ingenic assumes no liability, provides no warranty either expressed or implied relating to the usage, or intellectual property right infringement except as provided for by Ingenic Terms and Conditions of Sale.

Ingenic products are not designed for and should not be used in any medical or life sustaining or supporting equipment.

All information in this document should be treated as preliminary. Ingenic may make changes to this document without notice. Anyone relying on this documentation should contact Ingenic for the current documentation and errata.

北京君正集成电路股份有限公司

地址:北京市海淀区东北旺西路 8 号中关村软件园信息中心 A 座 108 室

电话: 86-10-82826661

传真: 86-10-82825845

Http: //www.ingenic.cn

目录

1	概述	1
2	建立开发环境.....	3
2.1	准备 PC 上的基础环境	3
2.2	额外的编译环境.....	3
2.3	下载源码.....	3
3	编译	5
3.1	代码目录说明	5
3.2	编译系统文件	5
3.3	编译 X-boot	5
4	烧录	7
5	其他	9

1 概述

本文基于君正 JZ4780（32 位的 RISC）处理器、 Android4.1 系统的 Warrior PAD 或 Grus 开发板进行编写，以帮助读者尽快熟悉基于 JZ4780、Android4.1 系统的开发环境。

该文档主要介绍了以下几个方面内容：

- 1) 基于 64 位 PC 机准备的基础环境；
- 2) 搭建编译环境；
- 3) 下载源码；
- 4) 编译；
- 5) 如何烧录；

2 建立开发环境

开始搭建开发环境，需要准备一台 Linux 机器，我们推荐使用 64 位的 Ubuntu 12.04 系统的 PC 机

2.1 准备 PC 上的基础环境

- 1) 准备一台环境干净的 64 位 Ubuntu（推荐版本是 12.04）系统的 PC 机；
- 2) 可以如下命令来使 64 位的系统能够搭建君正 32 位处理器的开发环境：

```
sudo apt-get install cpp-4.6 g++-4.6 gcc-4.6 gcc-4.6-multilib gcc g++ cpp gcc-multilib
g++-4.6-multilib git-core gnupg flex bison gperf build-essential zip curl libc6-dev
libncurses5-dev:i386 x11proto-core-dev libx11-dev:i386 libreadline6-dev:i386
libgl1-mesa-glx:i386 libgl1-mesa-dev g++-multilib mingw32 tofrodos python-markdown
libxml2-utils xsltproc zlib1g-dev:i386
```

```
sudo apt-get install ia32-libs /* only for 64bit PC OS */
```

```
sudo apt-get install gawk; /*only for 32bit PC OS */
```

- 3) 君正 4780 的 android 开发环境与 Google 官方完全兼容，更详细的编译环境建立方法，请参考 android 的官方网站：<http://source.android.com/source/initializing.html>

2.2 额外的编译环境

当标准的 android 编译环境准备好之后，有几个额外的准备步骤，这些可能有别于官方的标准环境。

- 1) 默认的 Ubuntu 环境中预装的 JAVA 环境是 openjdk，而 Android 编译必须用 oracle(sun)的 JAVA。请将 Ubuntu 上自带的 openjdk 先完全卸载，随后安装 JAVA 官网上下载的最新的对应于 64 位机器的 JDK 安装包，我们推荐安装 `jdk-6u37-linux-x64` 的版本；
- 2) 编译 Android 的 Linux 内核，需要安装我们的交叉编译链工具：**mips-4.3-mxu.tar.bz2** 将 `mips-4.3-mxu.tar.bz2` 解压到 Ubuntu 的 `/opt` 下面，然后将其添加到环境变量 `PATH` 中，以便任意 shell 都能访问到。

2.3 下载源码

君正 4780Android 可以提供完整的代码，请通过正确的渠道获得，下载的方式可通过 GIT（筹建中）、FTP 等。

3 编译

3.1 代码目录说明

将代码的压缩包解开，会得到以下的一个目录结构：

```
user@ubuntu:~/android-41$ ls
abi          cts          external    kernel      ndk         sdk
bionic      dalvik       frameworks  libcore    packages   system
bootable    development  gdk         libnativehelper  pdk
build       device      hardware    Makefile   prebuilts
```

其中，

- 1) Bootloader 的源码，位于：bootable/bootloader/xboot
- 2) Android 的 Linux 内核源码，位于：kernel
- 3) 完整 android 的源码。

3.2 编译系统文件

进入到工程根目录，执行编译命令：

Warrior PAD 的编译命令。

```
user@ubuntu:~/android-41$ ./build/scripts/mbuild_nand allimg warrior
warrior_release_nand
```

Grus 参考开发板的编译命令。

```
user@ubuntu:~/android-41$ ./build/scripts/mbuild_nand allimg grus
grus_nand
```

整个编译过程需要持续几个小时，具体的时间取决于主机的性能情况。

编译完成后，会在 out/target/product/ 的 warrior 或 grus 目录下生成二进制文件 system.img system.img.bin 和 boot.img 文件。

3.3 编译 X-boot

进入到 bootable/bootloader/xboot/ 目录下，执行命令：

Warrior PAD 的编译命令。

```
user@ubuntu:~/android-41$ make clean; make warrior_nand_config; make
```

Grus 参考开发板的编译命令。

```
user@ubuntu:~/android-41$ make clean; make grus_nand_config; make
```

编译完成后，会在当前目录下生成 x-boot-nand.bin 文件。

4 烧录

用于烧录的文件一般为三个

- x-boot-nand.bin
- boot.img
- system.img (或 system.img.bin)

system.img 用于 SDBurnTool 烧录 (推荐使用)

system.img.bin 用于 USBBurnTool 烧录。

具体烧录过程, 请参考 SDBurnTool 或 USBBurnTool 的指导手册或与产品相应的快速操作指南。

5 其他

Android 开发环境的搭建大同小异，基本上与 Android 官方网站介绍的一致，区别主要在于：

- a) 君正的 Android 开发板，同样 64 位的 Linux 系统，需要额外安装兼容 32 位的相应插件；
- b) 需要君正提供的交叉编译工具；
- c) 君正提供的对应的烧录工具；