

君正®

Newton Linux 开发指南

Date: Aug. 2014



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

君正®

Newton Linux 开发指南

Copyright © Ingenic Semiconductor Co. Ltd 201. All rights reserved.

版本发布历史

Date	Revision	Change
八月 2014	1.02	修改了下载源代码的命令 修改了编译工具链的说明 修改了 uboot、kernel 目录结构与说明 修改了制作文件系统的执行命令 修改了烧录的说明
四月 2014	1.01	First release

声明

本文档是提供给使用君正产品，在没有君正授权的情况下，君正不负任何责任，不提供任何明确或暗示的保证供其使用，或者知识产权的侵权除非拥有君正开发团队或者销售部门的授权。

君正产品的不是为医疗或相关设备而设计，并且也不能应用于医疗和相关的设备的应用。

所有这篇文档的内容都是初步的。君正有权在不提供提示的情况下作出修改。任何人员依赖于这篇文档的人员应该及时的联系君正获取最新的文档和勘误表。

北京君正集成电路股份有限公司

地址: 北京市海淀区西北旺东路 10 号院东区 14 号楼君正大厦

电话:(86-10)56345000

传真:(86-10)56345001

Http: //www.ingenic.cn

目录

1	概述	1
2	准备开发环境	1
2.1	准备 Linux 开发主机环境	1
2.2	安装步骤	1
2.3	准备烧录主机环境	2
3	获取 Newton Linux SDK	2
4	Newton Linux SDK 使用方法	2
4.1	Newton Linux SDK 的结构简介	2
4.1.1	u-boot 目录结构	2
4.1.2	kernel 目录结构	3
4.2	设置交叉编译环境	5
4.3	U-Boot 的配置与编译	5
4.4	Linux Kernel 的配置与编译	5
4.5	制作 EXT4 格式的根文件系统	6
4.6	烧录的分区信息	6
4.7	烧录二进制镜像文件	7
4.8	启动 Newton 板子	7
4.9	帮助文档	8
5	关注君正	8
5.1	新浪微博号：北京君正官微	8
5.2	微信号：北京君正	9
5.3	北京君正官网	9

1 概述

本开发指南将向读者介绍如何为君正 Newton 平台搭建 Linux 系统的开发环境，以及如何获取 Linux SDK，如何编译 U-Boot 和 Linux 内核等方法。本指南适合系统软件开发人员使用。

为了构建基于 JZ4775 处理器的 Newton Linux 开发平台，需要准备以下资源：

一台 Linux 开发主机，我们推荐使用 64 位的 Ubuntu-12.04 系统；

一套 Newton 开发板，包括串口线、锂电池、以及 USB Device 线等；

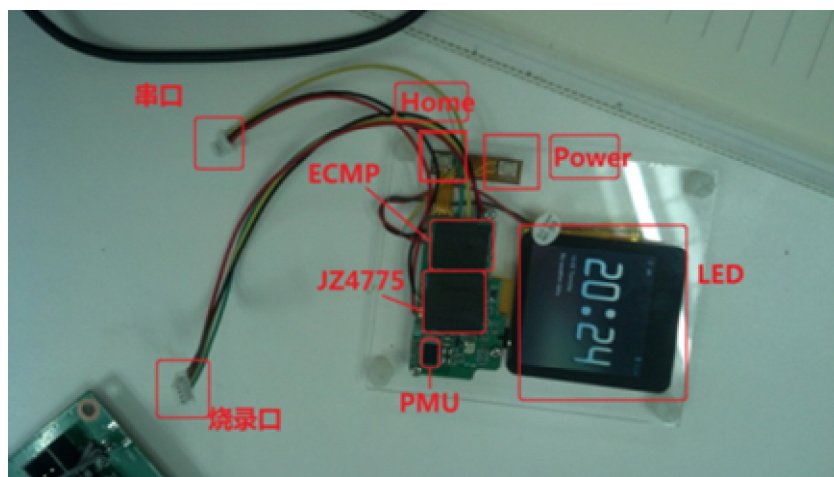


图1-1 Newton 概览

阅读本指南前，需要具有以下知识：

熟悉 Windows XP/7 操作系统，熟悉 Windows 上安装设备驱动的过程；

熟悉 Ubuntu 操作系统，熟悉 Linux 环境变量配置，熟悉 shell 命令和 Linux 命令行操作；

2 准备开发环境

本节介绍搭建 Linux 开发主机环境以及 Windows 烧录主机环境的方法。

2.1 准备 Linux 开发主机环境

需要准备一台安装 Linux 系统的电脑作为开发主机。

Linux 主机系统要求：

- 1) 硬盘容量不少于 128GB。
- 2) RAM 不小于 4GB。
- 3) 安装 Ubuntu-12.04 32 或 64 位系统。

2.2 安装步骤

- 1) 在开发主机上安装好 Ubuntu-12.04 系统。
- 2) 安装必需的软件包：

```
$ sudo apt-get install git build-essentials u-boot-tools
```
- 3) 在 64 位 Ubuntu 上运行 32 位程序需要安装 ia32-libs：

```
$ sudo apt-get install ia32-libs
```

2.3 准备烧录主机环境

君正目前仅提供运行在 Windows XP/7 上的 USB 烧录工具。具体如何安装烧录工具驱动和运行烧录工具进行烧录，请参考另一个文档《Howto_Burn_Newton_Demo_CN.pdf》。

3 获取 Newton Linux SDK

Newton Linux SDK 可从君正 GIT 仓库下载，并使用 repo 对多个 GIT 仓库进行管理。

按照下面步骤进行操作：

1) 下载 repo 脚本

```
$ mkdir newton-linux
$ cd newton-linux
$ wget http://git.ingenic.cn:8082/bj/repo
$ chmod +x repo
```

2) 下载 Newton Linux SDK

```
$ ./repo init -u http://git.ingenic.cn:8082/gerrit/linux/manifest.git -b newton-master
$ ./repo sync
```

以上操作开始下载代码，需要等待一段时间，下载速度依赖于实际网络条件。

4 Newton Linux SDK 使用方法

4.1 Newton Linux SDK 的结构简介

Newton 平台的 Linux SDK 包括 binaries, documents, sources 3 个目录，Linux SDK 包中涵盖了 uboot, kernel, rootfs 及其制作工具，MIPS 交叉编译工具链，第三方开源代码和测试代码等。代码结构如下：

binaries/:

- host-tool/: PC 主机端需要的一些工具，如 mkimage。
- rootfs/rootfs-newton/mkrootfs: 包含制作 rootfs 的脚本和工具。
- rootfs/rootfs-newton/rootfs.ext2: 可以直接烧录并启动的 rootfs。
- toolchain/mips-gcc472-glibc216/: 君正提供的 MIPS 交叉编译工具链。

documents/: 说明文档。

sources/: 源代码。

- bootloader/u-boot/: 君正移植的 u-boot。
- kernel/linux-3.0.8/: 君正基于 Linux3.0.8 移植的内核。
- application/: 第三方开源代码和测试代码。
- buildroot/: buildroot 的代码。

4.1.1 u-boot 目录结构

```
[~]# ls
api      config.mk  drivers   lib        mkconfig  rules.mk   tools     u-boot.srec
arch     COPYING   dts       MAINTAINERS nand_spl  snapshot.commit u-boot    u-boot-with-spl.bin
board    CREDITS   examples  MAKEALL    net       spl        u-boot.bin u-boot-with-spl-mbr.bin
boards.cfg disk      fs        makecp.sh  post      System.map  u-boot.lds
common   doc       include   Makefile   README   test       u-boot.map
```

图4-1 u-boot 代码

api:	该目录主要存放一些系统调用, 包含有显示用的 api , 网络部分的 api , 和一些跟平台相关但独立出来的 api 。是一个扩展应用的独立的 api 库。
arch:	主要以体系结构进行统一划分, 比如 avr32 、 arm 、 mips 、 sh 等。每种体系结构下统一有三个目录: cpu 、 lib 、 include 。
- cpu:	君正系列的 CPU 相关的代码都位于 arch/mips/cpu/xburst/ 目录下, 主要文件包括: <ul style="list-style-type: none"> - cpu.c 初始化 cpu、设置指令 cache 和数据 cache 等。 - jz4775/jz4775.c JZ4775 相关代码, 如系统 timer、PLL 的初始化等。 - jz_serial.c 串行口 UART 驱动程序。 - interrupt.c 设置系统的各种中断和异常, 比如快速中断, 开关中断、时钟中断、软件中断、预取中止和未定义指令等。 - start.S u-boot 启动时执行的第一个文件, 它主要是设置系统堆栈和工作方式, 为进入 C 程序奠定基础。
- lib:	该目录存放相应的体系结构 CPU 通用的文件。
- Include:	该目录存放与相应体系结构对应的头文件。
board:	该目录下存放目标板相关目录文件, 比如 Makefile 和 u-boot.lids 等都和具体开发板的硬件和地址分配有关。Newton 的文件在 ingenic/newton 中。
Boards.cfg:	该文件是目标板配置参数文件, 里边有很多种目标板的配置参数。
common:	该目录主要实现 uboot 命令行下支持的命令, 每一条命令都对应一个文件。例如 bootm 命令对应就是 cmd_bootm.c 。除此之外, 还存放一些与体系结构无关的通用文件。
Config.mk:	这个文件里面主要定义了交叉编译器及选项和编译规则。
disk:	该目录主要存放驱动的分区处理代码、硬盘接口程序。
doc:	该目录下主要是 U-Boot 的说明文档。
drivers:	通用设备驱动程序, 如各种网卡驱动、 CFI 标准 Flash 驱动、 USB Device 驱动等。
fs:	各种文件系统的驱动, 如 EXT4 、 FAT 、 JFFS2 、 CRAMFS 、 yaffs2 等。
include:	该目录主要 U-Boot 头文件, 以通用的头文件为主。其中 configs/ 下包含板级头文件 include/configs/newton.h 基于 JZ4775 Newton 开发平台的配置文件。
lib:	该目录存放通用的多功能函数。
nand_spl:	NAND 存储器相关代码。
tools:	创建 S-Record 和 U-Boot 映像的工具, 如 mkimage 。
Test:	测试命令的实现, 测试系统是否运行正常时使用。
net:	网络功能相关的文件目录, 如 bootp 、 nfs 、 tftp 等网络功能。

4.1.2 kernel 目录结构

```

-----[ PWD = ~/work/dev_newton_master/sources/kernel/linux-3.0.8 ]
[-yyhuang:linux-3.0.8]:$ ls
arch    crypto    fs        Kbuild    MAINTAINERS  mm        REPORTING-BUGS  sound    virt
block   Documentation  include  Kconfig   makecp.sh    Module.symvers  samples        System.map  vmlinux
COPYING drivers    init      kernel    Makefile     net          scripts         tools      vmlinux.o
CREDITS firmware  ipc       lib        mklinux.sh   README       security        usr

```

图4-2 kernel 代码

arch/mips/: MIPS 体系相关目录和文件, 目录结构如下

```

-----[ PWD = ~/work/dev_newton_master/sources/kernel/linux-3.0.8/arch/mips ]
[~yyhuang:mips]:$ ls
alchemy boot      dec      jz4740    kernel  Makefile  mti-sead3  pnx833x  sgi-ip22  txx9
ar7      built-in.o  emma     Kbuild    lantiq  math-emu  netlogic  pnx8550  sgi-ip27  vr4lxx
ath79    cavium-octeon fw       Kbuild.platforms lasat  mipssim  oprofile  power    sgi-ip32  wrppmc
bcm47xx  cobalt     include  Kconfig    lib     mm        pci       powertv  sibyte   xburst
bcm63xx  configs    jazz     Kconfig.debug loongson mti-malta pmc-sierra rb532    sni

```

图4-3 arch/mips 目录

- kernel/: 内核相关文件。
- mm/: 内存管理相关文件。
- lib/: 公用库函数。
- xburst/soc-4775/: JZ4775 处理器相关目录和文件
 - board/s2523b_15m/: JZ4775 处理器 Newton 平台通用处理文件
 - common/: JZ4775 处理器通用处理文件
 - include /: 各种头文件, 包含 JZ4775 相关的定义和开发板的配置文件等
- boot/compressed/: ulmage 生成目录
- Kconfig: Kconfig
- Makefile: Makefile
- configs/: 板级配置文件
- include/asm-generic/: 通用文件
- Sound:
 - oss/jzsound/: JZ4775/JZ4780 处理器 OSS 音频驱动
 - devices/codecs: 编解码器驱动
 - interface/: 常用接口驱动
- kernel: Linux 通用内核文件
- mm/: Linux 通用内存管理文件
- lib/: Linux 通用库函数
- init/: Linux 初始化函数
- ipc/: Linux 进程间通信函数
- net/: 网络相关文件
- fs/: 文件系统相关文件
 - jffs2/: JFFS/JFFS2 文件系统
 - ubifs/: UBIFS 文件系统
- drivers/: 设备驱动目录, 目录结构如下

```

-----[ PWD = ~/work/dev_newton_master/sources/kernel/linux-3.0.8/drivers ]
[~yyhuang:drivers]:$ ls
accessibility bluetooth crypto  gpu      input  media  nfc    pnp    sbus  switch
acpi          built-in.o dca     hid      isdn    memstick nubus  power  scsi  target
amba         cdrom      dio     hwmon    Kconfig message of      pps    sfi    tc
ata          char       dma     hwspinlock leds    mfd     oprofile ps3    sh     telephony
atm          clk        edac    i2c      lquest  misc    parisc  ptp    slpt  thermal
auxdisplay   clocksource eisa    ide      macintosh mmc      parport rapidio sn     tty
base         connector  firewire idle     Makefile mtd      pci     regulator spi    uio
bcma         cpufreq   firmware ieee802154 mca     nand    pcmcia  rtc    ssb    usb
block        cpuidle   gpio    infiniband md       net     platform s390   staging uwb

```

图4-4 drivers 目录

- block/: 块设备驱动
- char/: 字符设备驱动
- cpufreq: cpufreq 驱动
- input/: 输入设备驱动 (keyboard, mouse, touchscreen 等各种输入设备驱动)
- mmc/: MMC/SD 卡驱动
- mtd/: MTD 设备驱动
- mtd/ubi/: UBI 驱动
- net/: 网络设备驱动
- tty/serial/: UART 驱动
- spi/: 同步串行接口驱动
- usb/host: USB host 驱动
- usb/otg: USB otg 驱动
- usb/dwc2: USB dwc2 驱动
- usb/gadget: USB device gadget 驱动
- video/jz4780_fb: LCD framebuffer 驱动
- misc/jz_cim: Camera 驱动

4.2 设置交叉编译环境

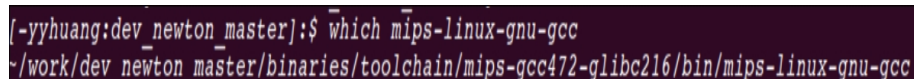
在 Newton 平台上进行 Linux 3.0.8 内核开发之前，首先需要安装好 MIPS 的交叉编译工具链。针对 Linux 3.0.8 内核的开发，君正提供基于 GNU gcc-4.7.2 和 glibc-2.16 的 MIPS 交叉编译工具链，位于 Linux SDK 包 binaries/toolchain/mips-gcc472-glibc216 目录下。

假设 Linux SDK 的路径为 SDK_ROOT。

安装交叉编译工具链只需设置 PATH 环境变量到交叉编译工具链路径即可，比如：

```
$ export PATH=SDK_ROOT/binaries/toolchain/mips-gcc472-glibc216/bin:$PATH
$ export CROSS_COMPILE=mips-linux-gnu-
```

按照上面建立好交叉编译环境后，通过“which mips-linux-gnu-gcc”命令验证交叉编译工具链是否已经设置成功：



```
[~yyhuang:dev newton master]:$ which mips-linux-gnu-gcc
~/work/dev_newton_master/binaries/toolchain/mips-gcc472-glibc216/bin/mips-linux-gnu-gcc
```

图4-5 查看环境变量结果

4.3 U-Boot 的配置与编译

配置和编译的操作如下：

```
$ cd sources/bootloader/u-boot
$ make distclean
$ make newton_ulmage_msc0
```

编译后生成 u-boot-with-spl-mbr.bin

4.4 Linux Kernel 的配置与编译

首先，选择 Newton 板的配置：

```
$ cd sources/kernel/linux-3.0.8
$ make newton_msc_defconfig
```

然后编译内核：

```
$ make ulmage
```

命令“make ulmage”编译生成 U-Boot 可以引导的二进制映像 ulmage，位于 linux-3.0.8/arch/mips/boot/compressed/目录下。

4.5 制作 EXT4 格式的根文件系统

代码中上传了可以启动的 rootfs: binaries/rootfs/rootfs-newton/rootfs.ext2，但更多的时候 rootfs 还是需要定制的。如果简单地添加以下文件到 rootfs 中，请按照一下步骤操作。如果需要细致地选择组件，请参考文档《BuildRoot 使用手册》。

执行以下命令需要 root 权限：

```
$ sudo -s
```

进入 SDK_ROOT/binaries/rootfs/rootfs-newton/mkrootfs 目录，执行以下命令：

```
# mkdir rootfs
# cd rootfs/
# tar xvf ../rootfs.tar.bz2
# cd ../
# ./mk_rootfsimg.sh rootfs
```

该命令会在当前目录下生成 EXT4 格式的 rootfs.img。

4.6 烧录的分区信息

JZ4775 Newton 开发板烧录文件分区表

Board	Hardware	File	Offset(B)	Option	Configuration
Newton	Storage: EMMC	u-boot-with-spl-mbr.bin	0	MMC0	Newton_mmc_lpddr.cfg
	4G, 512B/Sector	ulmage	0x300000	MMC0	
	Mem: Mobile DDR	rootfs.img/rootfs.ext2	0x3800000	MMC0	

烧录的文件及其对应的位置如下图所示：



图4-6 镜像文件及其对应的烧录位置

4.7 烧录二进制镜像文件

将编译生成的 u-boot-with-spl-mbr.bin, ulmage, rootfs.img/rootfs.ext2 烧录到 Newton 开发板。

4.8 启动 Newton 板子

烧录完成后, 连接好 Newton 板串口线, 设置好 UART 参数 (波特率: 57600, 数据位: 8bit, 奇偶校验位: 无, 停止位: 1, 数据流控制: 无), 然后按照正常方式启动 Newton 板。

如果串口输出以下信息, 说明 u-boot 启动成功:

```
sdrn init ok
MMC init ok
Starting U-Boot ...

U-Boot 1.1.6-g5352e480 (Apr 23 2014 - 12:02:44)

Board: Ingenic NEWTON (4775 SOC CPU Speed 1008 MHz)
MEM Clock: 168 MHz
DRAM: 512 MB Ram size > EMC_LOW_SDRAM_SPACE_SIZE, set ram size = EMC_LOW_SDRAM_SPACE_SIZE: 256 MB
Error: Unknown flash ID, force set to 'SST_ID_395F040'
Flash: 512 kB
MMC init ok
*** Warning - MMC/SD first load, using default environment

--==-- 0x8fe88000 ==--
d2041 set_lcd_power_on
Line is 762
DEFAULT_BACKLIGHT_LEVELIn: serial
Out: lcd
Err: lcd
Net: JZ ETHERNET
Hit any key to stop autoboot: 0
MMC init ok
6291456 bytes : OK
## Booting image at 80600000 ...
Image Name: Linux-3.0.8-00132-gde62c9c
Image Type: MIPS Linux Kernel Image (gzip compressed)
Data Size: 2580324 Bytes = 2.5 MB
Load Address: 80010000
Entry Point: 80425700
Verifying Checksum ... OK
Uncompressing Kernel Image ... OK

Starting kernel ...
```

图4-7 U-boot 启动成功

如果串口输出以下信息, 说明 Linux 内核启动成功:

```
[ 4.943623] x2d x2d: Virtual Driver of JZ X2D registered
[ 4.954509] Virtual Driver of JZ X2D registered
[ 4.964442] regulator_init_complete: LDO_AUD: incomplete constraints, leaving on
[ 4.985821] jz-rtc jz-rtc.0: setting system clock to 2013-03-01 07:59:44 UTC (1362124784)
[ 5.007577] EXT4-fs (mmcblk0p1): couldn't mount as ext3 due to feature incompatibilities
[ 5.043737] EXT4-fs (mmcblk0p1): couldn't mount as ext2 due to feature incompatibilities
[ 5.080652] EXT4-fs (mmcblk0p1): mounted filesystem without journal. Opts: (null)
[ 5.096083] VFS: Mounted root (ext4 filesystem) on device 179:1.
[ 5.109367] Freeing unused kernel memory: 212k freed
Starting udevd ...
```

图4-8 Kernel 启动成功

如果串口输出以下信息, 说明 Linux 内核挂载根文件系统成功, 输入 root 进入系统:

```
Starting logging: OK
Populating /dev using udev: [ 7.252142] Battery driver:The voltage is 4188mV
[ 7.298758] Battery driver:The adjust voltage is 4228mV
[ 7.348936] Battery: Not charging USB: offline AC: offline
done
Initializing random number generator... done.
Starting system message bus: done
Starting network...
Starting connman ... done.
Starting ofono ... done.
Starting ncard: OK
Starting domain name daemon: named[ 8.426821] warning: 'named' uses 32-bit capabilities (legacy support in use)
failed

Welcome to Buildroot
buildroot login: root
# cd /
# ls
bin      home    libexec  media    proc     sbin     usr
dev      lib     linuxrc  mnt      root     sys      var
etc      lib32   lost+found opt      run      tmp
#
CTRL-A Z for help 57600 8N1 NOR Minicom 2.5 VT102 Offline
```

图4-9 Linux 根文件系统挂载成功

4.9 帮助文档

SDK_ROOT/documents 下有相关外设模块的帮助文档，请阅读了解详情：

- README_WIFI: WIFI 设备的配置及测试帮助文档。
- README_BLUETOOTH: Bluetooth 设备的配置及测试帮助文档。
- README_SENSOR: 传感器设备的配置及测试帮助文档。
- README_MISC: 其它设备，如 Audio, USB 的配置及测试帮助文档。

5 关注君正

欢迎关注君正的微博和微信，第一时间获得君正最新动态。

5.1 新浪微博号：北京君正官微



图5-1 北京君正新浪微博

5.2 微信号：北京君正



图5-2 北京君正微信

5.3 北京君正官网

官网主页：<http://www.ingenic.cn/>

芯片产品主页：<http://www.ingenic.cn/cn/cn/product.php?fid=782>

开发支持主页：<http://www.ingenic.cn/cn/cn/kfzc.php>

如有技术方面问题和建议，请联系君正技术支持：

邮箱：support@ingenic.cn

电话：4008-900-223