# OK6410 linux-3.8.3移植(1)–编译生成

开发环境Ubuntu14.04 arm-linux-gcc 4.4.1 linux-3.8.3

## 1.mkimage工具

把uboot-2013-04-rc2目录下/tools/mkimage文件复制到ubuntu的/usr/bin/目录下

## 2.修改linux-3.8.3目录下的Makeflie

修改架构，添加交叉编译链

ARCH ?= arm

CROSS\_COMPILE ?= /usr/local/arm/4.4.1/bin/arm-linux-

## 3.进入arch/arm/mach-s3c64xx目录，修改Kconfig

Kconfig对应着内核的默认配置菜单。

仿照MINI6410，添加

config MACH\_OK6410

bool “OK6410″

select CPU\_S3C6410

select SAMSUNG\_DEV\_ADC

select S3C\_DEV\_HSMMC

select S3C\_DEV\_HSMMC1

select S3C\_DEV\_I2C1

select SAMSUNG\_DEV\_IDE

select S3C\_DEV\_FB

select S3C\_DEV\_RTC

select SAMSUNG\_DEV\_TS

select S3C\_DEV\_USB\_HOST

#select S3C\_DEV\_USB\_HSOTG

select S3C\_DEV\_WDT

select SAMSUNG\_DEV\_KEYPAD

select SAMSUNG\_DEV\_PWM

select HAVE\_S3C2410\_WATCHDOG if WATCHDOG

select S3C64XX\_SETUP\_SDHCI

select S3C64XX\_SETUP\_I2C1

select S3C64XX\_SETUP\_IDE

select S3C64XX\_SETUP\_FB\_24BPP

select S3C64XX\_SETUP\_KEYPAD

help

Machine support for the foxlinx OK6410

## 4. 添加OK6410支持

在arch/arm/mach-s3c64xx目录下，复制mach-mini6410.c到当前目录并改名为mach-ok6410.c，并把mach-ok6410.c文件中的所有的mini6410修改为ok6410，所有的MINI6410修改为OK6410

修改arch/arm/mach-s3c64xx/mach-ok6410.c中的static struct mtd\_partition ok6410\_nand\_part 分区函数

static struct mtd\_partition ok6410\_nand\_part[] = {

[0] = {

.name = "Bootloader",

.offset = 0,

.size = (1\*SZ\_1M),

.mask\_flags = MTD\_CAP\_NANDFLASH,

},

[1] = {

.name = "Kernel",

.offset = (1\*SZ\_1M),

.size = (5\*SZ\_1M),

.mask\_flags = MTD\_CAP\_NANDFLASH,

},

[2] = {

.name = "File System",

.offset = (6\*SZ\_1M),

.size = (200\*SZ\_1M),

},

[3] = {

.name = "User",

.offset = MTDPART\_OFS\_APPEND,

.size = MTDPART\_SIZ\_FULL,

},

};

在arch/arm/mach-s3c64xx目录下的Makefile中，在obj-$(CONFIG\_MACH\_MINI6410) += mach-mini6410.o下面添加obj-$(CONFIG\_MACH\_OK6410) += mach-ok6410.o

修改drivers/mtd/nand/nand\_ids.c文件，加入对MT的1G SLS nandflash的支持，添加{“NAND 1GiB 1,8V 8-bit”, 0x38, 4096, 1024, 512\*1024, LP\_OPTIONS},//MT

## 5. 进入arch/arm/tools目录，打开mach-types文件

设置与uboot对应的设备号 1626

添加ok6410 MACH\_OK6410 OK6410 1626

## 6.拷贝默认配置文件

内核参考的默认配置文件在 arch/arm/configs/中

确认工作目录是linux-3.8.3顶层目录

执行cp arch/arm/configs/s3c6400\_defconfig ./.config

## 7. uImage的前0X40

一般的u-boot只能识别uImage映像内核，所以我们要制作uImage类型的内核。

uImage是uboot专用的映像文件，它是在zImage之前加上一个长度为64字节的"头"，说明这个内核的版本、加载位置、生成时间、大小等信息；其0x40之后与zImage没区别。

进入scripts/目录，修改Makefile.lib文件，把

UIMAGE\_LOADADDR ?= arch\_must\_set\_this

UIMAGE\_ENTRYADDR ?= $(UIMAGE\_LOADADDR)

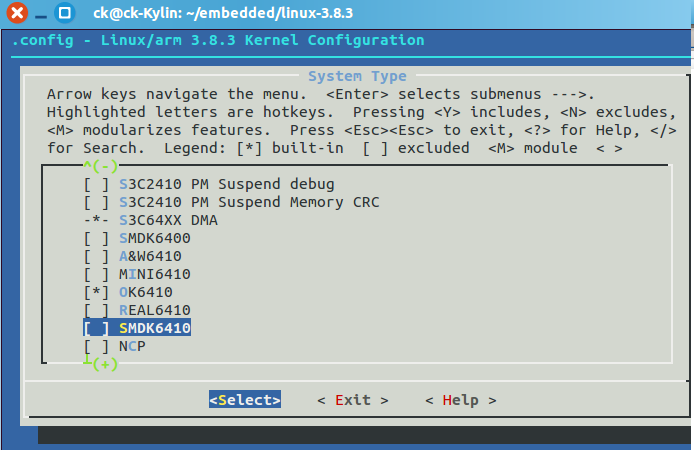
修改成

UIMAGE\_LOADADDR ?= arch\_must\_set\_this

#UIMAGE\_ENTRYADDR ?= $(UIMAGE\_LOADADDR)

UIMAGE\_ENTRYADDR ?= $(shell echo $(UIMAGE\_LOADADDR) | sed -e "s/..$$/40/")

## 8.回到linux-3.8.3目录下，执行make menuconfig

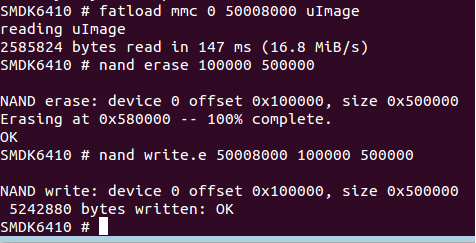


选择System Type,取消SMDK6400 ，A&W6410，SMDK6410等平台，**只选择OK6410**

选择Save an Alternate Configuration File，保存为.config然后退出。  
执行make uImage  
执行结束后可在arch/arm/boot目录下找到uImage文件

## 9.把uImage文件拷贝到SD卡中

开发板从nandflash中启动，执行以下命令将uImage拷贝到nandflash的100000-600000(与第4步，分区kernel对应)

10.NAND启动

修改u-boot源码 /include/configs/smdk6410.h

确保启动后执行如下两条指令，分区地址对应

#define CONFIG\_BOOTCOMMAND “nand read 0x50008000 0x100000 0x500000;” \

“bootm 50008000″

# MMC驱动移植

## 1.mach-ok6410.c文件里面增加以下内容：

#include <plat/sdhci.h>

/\* MMC/SD config \*/

static struct s3c\_sdhci\_platdata ok6410\_hsmmc0\_pdata = {

.max\_width = 4,

.cd\_type = S3C\_SDHCI\_CD\_INTERNAL,

};

static struct s3c\_sdhci\_platdata ok6410\_hsmmc1\_pdata = {

.max\_width = 4,

.cd\_type = S3C\_SDHCI\_CD\_PERMANENT,

};

ok6410\_machine\_init()函数中添加：

s3c\_sdhci0\_set\_platdata(&ok6410\_hsmmc0\_pdata);

s3c\_sdhci1\_set\_platdata(&ok6410\_hsmmc1\_pdata);

## 2.内核配置：

系统默认已经有选上MMC/SD驱动选项，所以需要增加的是File systems 内容如下：

File systems —>

Native language support —>

\* Codepage 437 (United States, Canada)

\* NLS ISO 8859-1 (Latin 1; Western European Language)

更改u-boot启动参数，使内核加载SD卡中ext3格式根文件系统，具体参照 <cramfs,yaffs2,ext3根文件系统构建>

NAND方式启动可以使用mount命令挂载SD卡

# mount -t ext3 /dev/mtdblkp2 /home

至此， 若是使用SD卡存储uboot，uImage，rootfs，那么可以实现可移植性极强的一整套系统。

但是将根文件系统放置SD卡中缺乏安全性，读者可根据个人需求选择不同的根文件系统，然后挂载其他用户文件系统。

# OK6410 linux-3.8.3移植(3)–NAND驱动移植