基于 S3C2440 的 Uboot 分析与移植①

申　爽

(桂林电子科技大学 计算机科学与工程学院，桂林 541004)

摘　要：首先根据对 Uboot 的结构功能和启动分析，提出了一种基于 s3c2440 大容量 Nand Flash 和 Nor Flash 的移植方案，然后通过多步的移植，完善各个功能模块。最终在 Uboot 下，使用 NFS 方式成功加载内核和文件系统,详细给出了 Uboot 移植方法。

关键词：嵌入式系统；Linux；Uboot；移植

Uboot Analysis and Transplantation Based on S3C2440

SHEN Shuang

(School of Computer Science and Engineering, Guilin University of Electronic Technology, Guilin 541004, China)

Abstract: According to the structure and function of the Uboot, and the analysis of starting codes. This paper firstly proposed a transplant program based on s3c2440 high-capacity Nand Flash and Nor Flash. Then, through a multi-step migration, it improved the various functional modules. Eventually, it loaded the kernel and NFS file systems successfully, by the Uboot, given more Details of the Uboot transplantation.

Key words: embedded system; Linux; Uboot; transplantation

1　移植 Uboot 的软硬件环境

笔者的目标平台，采用 ARM9 系列架构，版本为 ARM920T 的 S3C2440 芯片。支持 32 位 ARM 指令集和 16 位 Thumb 指令集, 带有 5 级流水线。集成了 USB Host 控制器、PCMCIA/SD/MMC 卡控制器、I2C 接口、串口、AC97 控制器、实时时钟、PWM 控制器等外设，内部带有 4KB SRAM,支持 Nor Flash 和 Nand Flash 两种启动方式[1]。核心板资源见表 1。

表 1　核心板资源

CPU	S3C2440A	400~533MHz
SDRAM	H57V2562GTR-75C	16Mx16bit 2 片
Nor Flash	JS28F320-J3D75	4Mx8bit
Nand Flash	K9F2G08	256Mx8bit
Net Card	DM9000A	10/100M

S3C2440 将存储器分成 8 个区域，每个区域大小为 128MB。当从 Nor Flash 启动时，内部 SRAM 被映射到 0x40000000~0x40001000 处。当从 Nand Flash 启动时，内部 SRAM 被映射到 0x00000000~0x00001000 处，并且会自动映射到 Nand Flash 前 4KB 的数据。

本文使用虚拟机搭建软件环境，方便后期开发。安装了 Linux(RHEL5.1),内核版本为 2.6.18xen，选用工具链 arm—linux-gcc-3.4.5 作为编译工具，其中包括了编译器、链接器、汇编器等开发工具。使用交叉网线连接开发板与目标机，选择 JLINK 仿真器，方便笔记本下载和调试。

2　Uboot结构功能与启动分析

2.1 Uboot 结构功能

Uboot 是德国 DENX 小组开发的嵌入式 bootloader。本文在 u-boot-2008.10 的基础上进行分析和移植。从官方的 ftp 服务器下载后，解压即得到全部源代码。Uboot 根目录下共有 30 个子目录，可以分为 4 类：

① 收稿时间:2011-08-11;收到修改稿时间:2011-09-06

（1）平台相关的；

（2）通用的函数和命令；

（3）通用的设备驱动文件；

（4）Uboot 工具、示例程序、文档。

Uboot 提供了系统引导、上电自检、CRC32 校验、设备驱动、支持 NFS 挂载、支持目标板多种方式存储等功能。各个目录之间的层次结构，见图 1。这种结构封装和功能模块化的特点，使开发者可以很容易找到与自己开发板相似的配置，有针对性地对特定模块进行修改，简化移植工作。

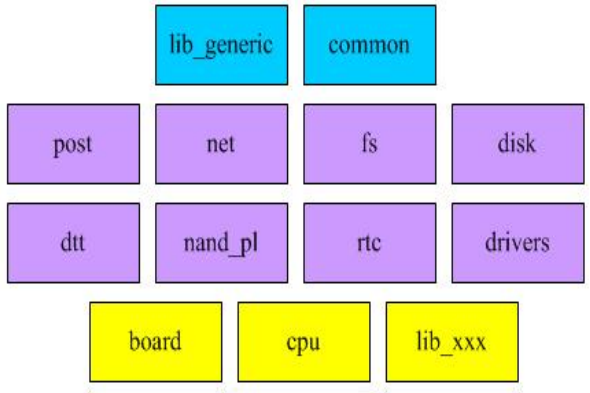

图 1 Uboot 层次结构[2]

2.2 Uboot 启动分析

Uboot 属于两阶段 bootloader。在第一阶段中，由于没有建立堆栈，全部是汇编级代码，主要涉及两个文件：

cpu/arm920t/start.S

board/{开发板目录}/lowlevel_init.S

前者与平台相关，后者与开发板相关。依次完成如下工作：

（1）设置 CPU 为管理模式（svc），关闭中断、看门狗，设置时钟比，关闭 MMU、CACHE；

（2）初始化 SDRAM；

（3）重定位，复制第二阶段代码到 SDRAM；

（4）建立堆栈；

（5）跳转到第二阶段入口地址处。

在第二阶段中，进入第一个 C 程序。从 lib_arm/board.c 中的 start_armboot 函数开始，进行一系列设置后，最后进入 main_loop 函数死循环，等待输入命令。见图 2。

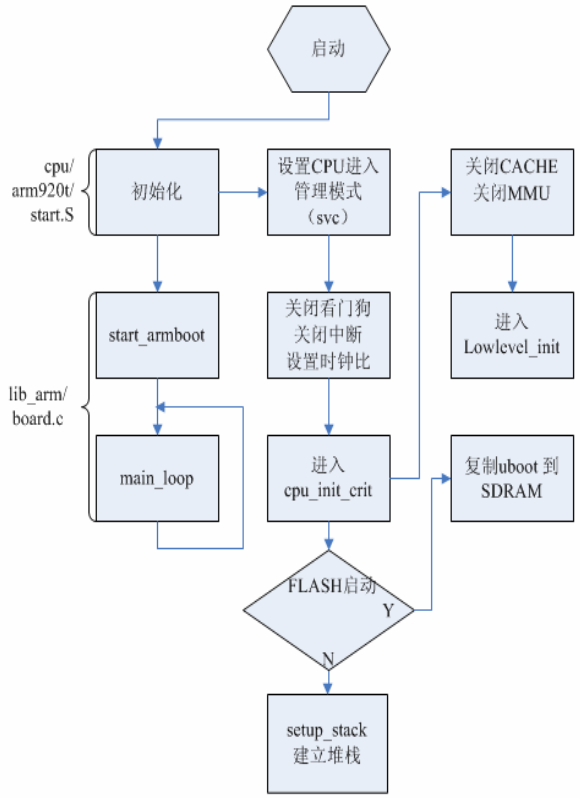

图 2 Uboot 启动分析

3 Uboot移植流程

Uboot 的工作过程与硬件关系密切，所以在移植时必须对 S3C2440 等芯片手册和开发板的硬件有一定的了解。Uboot 中 smdk2410 的配置适用于大多数 s3c2410 开发板，但是目前 Uboot 不支持 s3c2440，需要自己移植。其移植流程主要分为四个步骤:

（1）建立目标平台文件，修改硬件配置，在内存中调试 Uboot；

（2）添加 Uboot 的功能，如网络等；

（3）支持 Nor Flash 方式启动；

（4）支持 Nand Flash 方式启动。

3.1 在内存中调试 Uboot

（1）修改 Makefile

在 Linux 下，使用 make 工具管理工程，在每个目录下都有相应的 Makefile 工程文件。所以在 board 目录下，新建开发板文件夹 fl2440，拷贝 smdk2410 文件夹下的所有文件到该目录中，重命名文件 smdk2410.c 为 fl2440.c 后，需要修改其目录下的 Makefile 文件，替换 smdk2410.o 为 fl2440.o。同样，在根目录下的 Makefile 中，需要添加 fl2440 开发板的编译规则:

fl2440_config : unconfig

@$(MKCONFIG)$(@:_config=) arm arm920t fl2440 samsung s3c24x0

（2）修改平台相关代码

根据前面的分析，修改工作集中在 Uboot 启动第一阶段的 cpu/arm920t/start.S 文件中。由于 s3c2440 的主频提高到了 400MHZ，而 s3c2410 只有 200MHz，首先需要修改原有代码中的锁相环时钟部分。查看芯片手册[3]，s3c2440 的锁相环输出的 MPLL（系统频率）与 UPLL（USB 控制器频率）计算公式如下：

Mpll = (2 * m * Fin)/(p * 2^S);

Upll = (m * Fin)/(p * 2^S)

m = (MDIV + 8), p = (PDIV + 2), s = SDIV

其中 Fin 为晶振频率 12MHz，m、p、s 分别为配置参数。

在添加 s3c2440 的锁相环寄存器的宏定义后，分别给 MDIV、PDIV、SDIV 寄存器赋值 0x7f、0x01、0x01 完成配置：

#define S3C2440_MPLL_405_MHz ((0x7f<<12)|(0x01<<4)|(0x01))

#define S3C2440_UPLL_48_MHz ((0x38<<12)|(0x02<<4)|(0x02))

#define S3C2440_CLKDIV 0x05

/* FCLK:HCLK:PCLK = 1:4:8, UCLK = UPLL */

因为当 Uboot 下载到内存中调试时，不需要进行底层初始化，也不需要进行 Uboot 代码重定位，所以在配置文件 include/configs/fl2440.h 中添加如下宏定义，跳过设置：

#define CONFIG_SKIP_LOWLEVEL_INIT 1

#define CONFIG_SKIP_RELOCATE_UBOOT 1

最后在宿主机 linux 环境的终端下进入到 Uboot 的根目录，执行如下命令编译 Uboot：

#make fl2440_config

#make CROSS_COMPILE={arm-linux-的安装路径}

用 JLINK 将生成的 u-boot.bin 文件，烧写到 SDRAM 的 0x33f8,0000 处，设置 pc 指针从该位置处运行，通过串口终端就可以成功打印出目标板的信息。

3.2 添加网络等功能

Uboot 通过向位于 include/configs/fl2440.h 文件的 CONFIG_COMMANDS 宏中添加新字段的方式支持新功能。只需以逻辑或的形式添加宏 CFG_CMD_DFL、CFG_CMD_NET 和 CFG_CMD_ PING，这样就可以开启 Uboot 的默认功能和网络功能了。

查看开发板原理图，网卡芯片位于 0x20000000 地址处。Uboot 支持网卡驱动，要使用 DM9000A 网卡芯片，需要接着在 include/configs/fl2440.h 文件中，配置如下：

#define CONFIG_DRIVER_DM9000 1

#define CONFIG_DM9000_USE_16BIT 1

#define CONFIG_DM9000_BASE 0x20000000

#define DM9000_IO 0x20000000?

#define DM9000_DATA 0x20000004

重新编译 Uboot 后，通过串口终端，先使用 setenv 命令设置开发板 mac 地址、ip 地址、服务器 ip 地址等环境变量，再使用 ping 命令测试网卡是否可用。

3.3 Nor Flash 方式启动

不同与其他开发板的 Nor Flash 芯片（1、2MB），目标平台的 Nor Flash 采用了更大容量的 Nor Flash 芯片 Intel 的 JS28F3201（4MB）。由前面的启动分析，当 Uboot 在 Flash 中运行时需要进行内存初始化、代码重定位。所谓的代码重定位，即是在上电后，将从 0x00000000 处开始的 NorFlash 中的 Uboot，搬到 0x3000_0000 之后运行。具体的内存分布，见图 3。

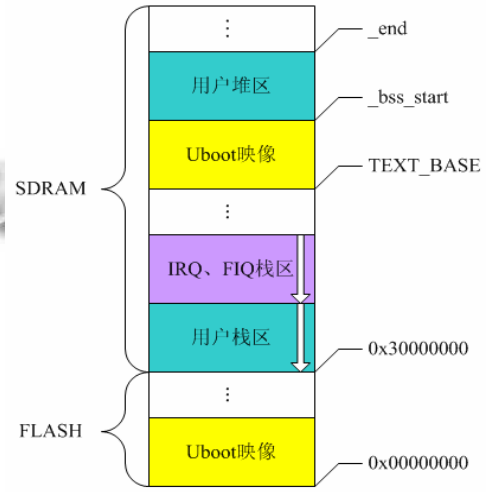

图 3　内存分布

参考芯片手册[4],Flash 芯片的核心功能是读、写、擦除所对应的软件命令操作序列。使用 grep 命令，查找到目录中 drivers/cmi/flash.c 支持该芯片，将该文件复制并重命名为 board/fl2440/flash.c。主要工作集中在重新实现该文件中的读取函数（flash_get_size）、擦除

函数(flash_erase)、写入函数(write_buf 、write_word)。

然后修改 include/configs/fl2440.h 文件配置中的物理内存映射和 Flash 的部分，在编译之前，将链接脚本文件 board/fl2440/U-boot.lds 中的程序初始化运行地址改为.=0x0000,0000。跳线选择从 Nor Flash 启动后，输入 flinfo 命令，可以从串口终端打印出 Flash 信息。

3.4 Nand Flash 方式启动

不同与其他开发板的 Nor Flash 芯片（64、128MB），目标平台采用了大容量的 Nand Flash 芯片 K9F2G08（256MB），该芯片是 2k 每页的 flash。当操作 Nand Flash 时，先传输命令，然后传输地址，最后传输读/写数据，期间要查看 Flash 的状态。该芯片的地址需要 5 个周期才能传完，前两个周期用来寻页内地址，后三个周期是页间寻址。

u-boot-2008.10 对于 Nand Flash 的支持有两套代码，旧代码在 drivers/nand_legacy 下，新代码在 drivers/mtd 下。根据文档 doc/README.nand 的说明，新代码移植自 Linux 内核 2.6.12, 支持更多型号的 Nand Flash，所以本文选择使用新代码。

首先需要在配置文件 include/configs/fl2440.h 中，以逻辑或的方式在宏 CONFIG_COMMANDS 中添加字段 CFG_CMD_NAND，然后再添加 Nand Flash 的基地址、片数、是否进行校验等信息，在 Nand Flash 中保存环境变量。

根据 MTD 的驱动框架，使用 nand_chip 数据结构表示一个 Nand Flash 芯片。这个结构体中包含了关于 Nand Flash 的地址信息、读写方法、ECC 模式、硬件控制等一系列低层机制[5]，因此新驱动的编写将变得更加方便，只需修改该结构体的相应成员函数。

主要集中在重新编写 cpu/arm920t/s3c24x0/nand.c 文件，实现相应的初始化函数（board_nand_init）和命令控制函数（s3c2440_hwcontrol），并在文件中加入以下代码，进行条件编译：

#if(CONFIG_COMMANDS&CFG_CMD_NAND) && !defined(CFG_NAND_LEGACY)

由于已经加入了串口通信和网络通信的功能，只要连接好串口线和交叉网线就可以进行内核的移植工作了，见图 4。

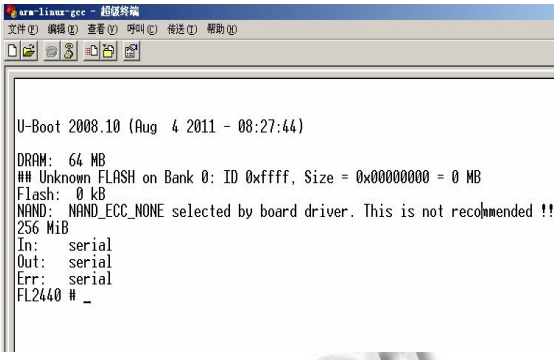

图 4 Nand Flash 方式启动

4 Uboot 加载内核及文件系统

本文采用 NFS 方式起根文件系统，方便后面的驱动开发的调试工作。这就需要对内核进行配置，在配置文件中的 File systems -> Net File Systems 选项下，选中 NFS client support for NFS version 3；Root file system on NFS。然后对配置文件的 CMDLINE 字段进行修改，设置启动参数：

CMDLINE= ” root=/dev/nfs nfsroot=192.168.0.10:/nfsroot/rootfs ip=192.168.0.11 console=ttySAC0 mem=64”

最后使用如下命令进行编译后生成最终的内核镜象文件：

make uImage ARCH=arm CROSS_COMPILE=arm-linux-

在 Linux 下的/etc/xinetd.d/tftp 文件中，设置 disable 字段为 no，设置服务器目录为/tftpboot 后，关闭防火墙，重启 xinetd.d 服务，开放 Tftp 服务。在/etc/exports 文件中，写入 /nfsroot 192.168.0.* (rw) ,并重启 Nfs 服

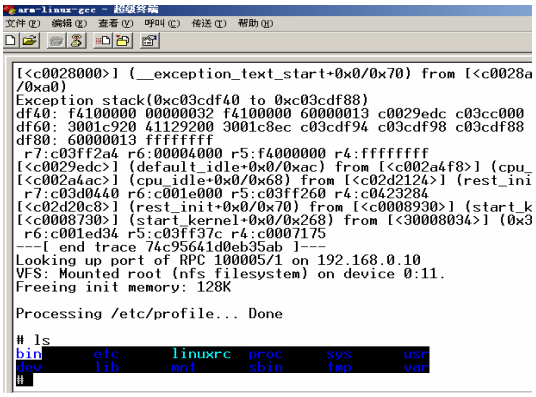

图 5 Uboot 加载内核和 NFS 文件系统

(下转第 229 页)

4 实验结果分析

（1）测量精度实验：选取 1MΩ、2MΩ、5MΩ、10MΩ、20MΩ高精度电阻进行测试, 测得数据如表 1 所示。

表 1

标准电阻 MΩ	1	2	5	10	20
测试 1	1.03	2.03	5.03	10.09	20.09
测试 2	1.04	2.05	5.03	10.1	20.06
测试 3	1.03	2.01	5.06	10.03	20.04
测试 4	1.02	2.04	5.04	10.06	20.02
测试 5	1.04	2.02	5.02	10.02	20.08
测试 6	1.05	2.03	5.02	10.04	20.12
测试 7	1.03	2.02	5.02	10.08	20.03
测试 8	1.02	2.07	5.08	10.06	20.22
测试 9	1.05	2.06	5.06	10.06	20.03
测试 10	1.01	2.05	5.04	10.11	20.26
平均值	1.032	2.038	5.04	10.065	20.095
相对误差	3.20%	1.90%	0.80%	0.65%	0.47%

（2）测量速度实验（一人操作，包括接线时间在内）：以 32 芯电缆为例，一个人用手摇兆欧表对电缆芯线之间的绝缘电阻和芯线通断进行测量, 所用时间为 105 分钟。使用本测试仪所用时间为 10 分 26 秒。

（3）功耗测试：静态功耗≤120mW；最大功耗≤2W。

实验结果表明，系统具有较高的测量精度，相对误差小于 4%。测量速度快、功耗低，符合设计要求。

5 结语

该测试仪能自动检测 32 芯电缆各芯线对地、芯线之间的绝缘电阻及芯线的通断。通过对高压开关电源适时控制，以及专门设置的看门狗电路使系统具有稳定性好，功耗低，测量精度高，测量速度快等特点。另外，系统采用模块化设计，可以方便扩展为 64 或 128 芯电缆的测试。可广泛用于机车信号、电务通信、邮电、航空等多芯电缆的测量。

参考文献

1 闵海波,王仕成,张金生.导弹绝缘电阻快速自动化测试仪研制.自动化仪表,2006,(10):76–77.
2 孙玉胜,邹玉炜,崔光照.多芯电缆测试仪的研制.传感器与仪器仪表,2007,23(7-1):192–194.
3 徐俊刚,张立材.绝缘电阻在线检测研究.山西建筑,2008,34(4):185–186.
4 冀飞,王顺喜.高压直流电源技术的发展现状及应.农村电气化,2004,8:34–35.
5 刘亚平,邢济收,刘相权.AVR 单片机串行口的软件扩展技术.北京信息科技大学学报(自然科学版),2010:25(4):54–56.
6 罗大成,王仕成,闵海波.一种导弹绝缘电阻测试仪的软件设计.战术导弹控制技术,2007,(1):26–28.

(上接第 225 页)

务，允许 192.168.0.* 网段内的节点访问该主机。将 uImage 放置/tftpboot 目录下，在 Uboot 下输入命令 tftp 31000000 uImage；bootm 31000000 后就可以加载 Linux 内核和文件系统了。见图 5。

5 小结

本文在分析Uboot原理后，通过4个步骤，将Uboot移植到 s3c2440 目标平台，并使用 NFS 方式加载 Linux 系统。 逐步实现了串口通信、网络、Flash 烧写、Flash 启动等功能，提出了一种基于 s3c2440 大容量 Flash 的 Uboot 移植方法，为后续开发打下了坚实的基础。

参考文献

1 杜春雷.ARM 体系结构与编程.北京:清华大学出版社,2003.8–9.
2 韦东山.嵌入式 Linux 应用开发完全手册.北京:人民邮电出版社,2008.246–248.
3 Sumsuang Electronics. S3C2440A 32–BIT CMOS Micro-controller User’s Manual. Republic of Korea: Sumsang, 2003.
4 Intel. Intel StrataFlash Embedded Memory (P30).The United States of America: Intel, 2005.
5 宋宝华.Linux 设备驱动开发详解.北京:人民邮电出版社,2008.541–543.