基于 mini2440 平台的 U-Boot 移植方法①

王冰瑶

(哈尔滨工程大学 信息与通信工程学院, 哈尔滨 150001)

摘 要: 研究了以 ARM9 系列的 S3C2440 处理器为核心的 mini2440 平台, 来移植 U-Boot 的方法. 首先根据 mini2440 平台的硬件资源, 对 U-Boot 源代码进行修改, 然后对修改后的源代码进行编译, 将生成 u-boot.bin 文件下载到 mini2440 开发板上运行, 并对 U-Boot 进行功能测试. 测试结果表明, U-Boot 成功地在开发板上运行, 并能实现它的功能. 此移植方法对使用 S3C2440 处理器进行嵌入式系统设计及 U-Boot 在其他处理器上的移植具有参考价值.

关键词: U-Boot; ARM; S3C2440; 嵌入式; BootLoader

U-Boot Transplantation Method Based on Mini2440 Platform

WANG Bing-Yao

(Information and Communication College, Harbin Engineering University, Harbin 150001, China)

Abstract: Research on mini2440 platform with ARM9 series S3C2440 processor core, methods to transplant U-Boot. Based on the mini2440 platform hardware resources, to modify the U-Boot source code, and then to modify the source code to be compiled, will generate the u-boot.bin file to download to the mini2440 development board to run, and the U-Boot functional testing, test results show that U-Boot successfully in the development board to run, and can realize its function. The transplant method to use S3C2440 processor for embedded system design and U-Boot in other processor transplantation has reference value.

Key words: U-Boot; ARM; S3C2440; embedded system; BootLoader

Linux 操作系统因其具有开放源代码、易于移植、资源丰富等优点, 使得它在嵌入式领域中越来越流行. 对于嵌入式 Linux 系统而言, 其软件系统的主要组成部分有 BootLoader、内核、根文件系统和系统应用程序[1]. 而 BootLoader 作为系统加电后运行的第一个程序, 对嵌入式系统的后继开发工作十分重要, 所以它是嵌入式系统开发的重要环节. 在嵌入式领域, BootLoader 的实现严重依赖于具体的硬件平台, 因此想要创建一个通用的 BootLoader 几乎是不可能的. 但是, 仍然可以归纳出一些通用的方法, 用来设计我们所需要的 BootLoader. 同时 ARM 微处理器因为具有高性能、低功耗、价格低等优点, 使其在 32 位嵌入式处理器的市场份额已经超过 70%[2]. 目前 U-Boot 是嵌入式行业主流的 BootLoader, 业内对 U-Boot 在 S3C2410、S3C44B0 等 ARM 处理器上的移植技术已经非常成熟, 可是在 S3C2440 处理器上的移植研究做的很少, 所以在使用 S3C2440 处理器进行嵌入式系统设计时, 工作变得非常困难, 于是此次研究选用使用率非常高的 mini2440 开发板来进行 U-Boot 的移植工作.

1 mini2440开发板简介

mini2440 开发板是由友善之臂公司开发研制, 它采用三星 S3C2440 为微处理器, 并采用专业稳定的 CPU 内核电源芯片和复位芯片来保证系统运行时的稳定性, mini2440 的 PCB 采用沉金工艺的四层板设计, 专业等长布线, 保证关键信号的信号完整性, 是目前国内性价比最高的开发板[3].

由于 U-Boot 的移植依赖于具体的硬件资源, 所以

① 收稿时间:2013-01-24;收到修改稿时间:2013-03-04

必须掌握开发板上的硬件配置，mini2440 开发板的硬件配置如表 1 所示[3].

表 1　mini2440 开发板硬件配置

硬件名称	资源特性
系统时钟源	12M 无源晶振
接口和资源	1 个 100M 以太网接口(采用 DM9000 网络芯片) 3 个串行口，1 个 USB Host，1 个 USB Slave B 接口，1 个 2.0mm 间距 10 针 JTAG 接口
Flash 存储	在板 64M Nand Flash，在板 2M Nor Flash
内存	在板 64M SDRAM，时钟频率 100MHz
CPU 处理器	Samsung S3C2440A，主频 400MHz

2　U-Boot简介

U-Boot 的全称为 Universal Boot Loader，它是完全免费开源的，支持 PowerPC、ARM、MIPS、XScale 等多种系列的嵌入式处理器；同时支持 VxWorks、NetBSD、QNX、RTEMS、Linux 等多种嵌入式操作系统的引导. 就目前来看，U-Boot 对 PowerPC 系列的处理器支持最为丰富，对 Linux 的支持最为完善.

2.1 U-Boot 源代码的目录结构

要想对 U-Boot 进行移植，首先需要对其源代码的结构以及功能进行详细的分析. 此次研究与移植工作基于 u-boot-1.1.6 进行，对其源码解压后会在 u-boot-1.1.6 目录下生成它的源代码. 其目录结构主要分为与体系结构有关的目录以及与体系结构无关的代码目录，前者主要包含 board 目录和 CPU 目录，移植 U-Boot 的工作主要就集中在对这些目录里面特定文件的修改[4]. 移植 U-Boot 过程中比较重要的目录如表 2 所示.

表 2　移植 U-Boot 所需的重要目录介绍

目录名称	功能说明
driver	通用设备驱动
lib_arm	arm 体系结构相关的文件
tools	存放制作 U-Boot 格式映像等工具
doc	U-Boot 的说明文档
include	头文件和目标板的配置文件，配置文件在其子目录 configs 下
cpu	与处理器相关的文件
board	目标板的相关文件

2.2 U-Boot 的启动流程

大部分 BootLoader 的启动过程可以分为 stage1 和 stage2，U-Boot 也不例外. 在第一阶段用到的文件通常都是依赖于 cpu 体系结构的，而且用汇编语言来实现，从而达到短小精悍的目的. 而第二阶段的代码通常用 C 语言编写，以实现更复杂的功能，并且使代码具有更好的可读性和移植性. U-Boot 的启动流程图如图 1 所示.

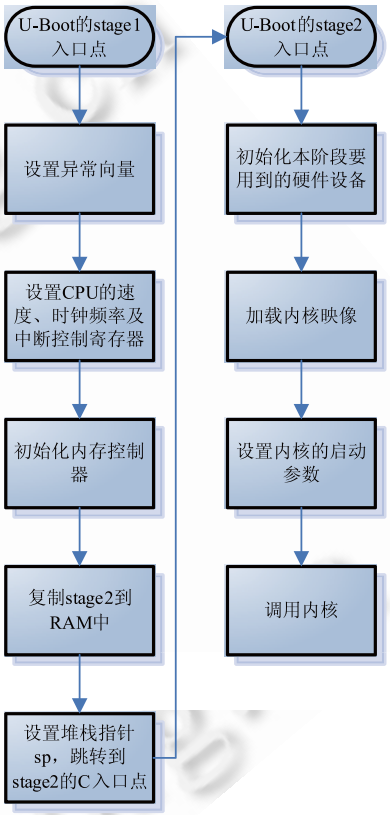

图 1　U-Boot 的启动流程图

3　U-Boot的移植

3.1 建立交叉编译环境

交叉编译工具链是嵌入式开发必备的基础工具. 本文使用的交叉编译工具链版本是 arm-linux-gcc-3.4.5. 首先将交叉编译工具链解压到/usr/local/arm 目录下；然后修改 PATH 环境变量，使其包含交叉编译器的路径；最后输入 arm-linux-gcc-v 以检查交叉编译工具链是否安装成功.

3.2 U-Boot 移植的具体操作方法

在对 U-boot 进行移植之前，首先要对 U-boot 已经支持的开发板进行分析，找出硬件资源最为接近的开发板代码，然后以其为基础进行移植[5]. board 目录下每一个子目录都包含一个 u-boot 支持的硬件开发板的支持代码，其中有 smdk2410 子目录，但没有

smdk2440 目录, 这表明 u-boot 支持 S3C2410,但不支持 S3C2440. S3C2440 与 S3C2410 非常相似, 因此可以用 S3C2410 的支持代码作为基础为 S3C2440 移植 u-boot.

需要说明的是, 在移植过程中需要修改的文件较多, 并且有的文件还需要重写, 但是因为篇幅有限, 所以本文并未给出全部的修改代码. 大体移植步骤如下.

3.2.1 建立 mini2440 开发板目录以及配置文件

复制 board/smdk2410 到 board/mini2440, 并将 mini2440 目录下的 smdk2410.c 重命名为 mini2440.c, 在 include/configs/目录下复制 smdk2410.h 文件, 并重命名为 mini2440.h.

3.2.2 修改 CPU 目录下的相关部分

通过阅读分析 U-Boot 的源代码, 了解到程序的第一条指令所在文件为 cpu/arm920t/start.S, 所以先从此文件进行修改. 在 114 行#if defined(config_ S3C2400) ||defined(config_S3C2410)后添加 CPU S3C2440 的定义 defined(config_ S3C2440).

在 cpu/arm920t/s3c24x0 目录下 speed.c 文件中的 get_HCLK、get_PCLK、get_ PLLCLK 函数的功能是为了获得 HCLK 和 PCLK, 但因为 S3C2410 与 S3C2440 计算 HCLK 和 PCLK 的方法并不一样, 所以这 3 个函数的代码都需要修改. 此外, U-Boot 开发者并不知道我们的开发板所配置的 Nand Flash 类型, 所以 Nand Flash 的驱动程序需要我们自己编写 nand_flash.c.

3.2.3 修改 board 目录下的相关部分

S3C2410 需要 4 个频率, 分别是 CPU 核 arm920t 使用的 FCLK、快速设备控制器使用的 HCLK、慢速设备控制器使用的 PCLK 以及 USB 控制器使用的 UPLL. 这 4 个频率都是频率控制器芯片把晶振频率(12MHz)倍频或者分频所得, 它们分别为 200MHz、100MHz、50MHz、48MHz. 但是 S3C2440 相应的频率分别是 400MHz、100MHz、50MHz、48MHz, 同时 S3C2440 与 S3C2410 的频率控制器芯片的设置方法也稍有差异, 所以需要修改相关代码以匹配 S3C2440.

将/board/mini2440 目录下 mini2440.c 文件的第 77 行改为:

```
#define S3C2440_MPLL_400MHz ((0x5c<<12)|(0x01<<4)|(0x01))
#define S3C2440_UPLL_48MHz ((0x38<<12)|(0x02<<4)|(0x02))
#define S3C2440_CLKDIV 0x05
```

```
clk_power->CLKDIVN = S3C2440_CLKDIV;
__asm__ (“mrc p15,0,r1,c1,c0,0\n”
    “orr r1,r1,#0xc0000000\n”
    “mcr p15,0,r1,c1,c0,0\n”
    ::: “r1”);
clk_power->MPLLCON= S3C2440_MPLL_400MHz;
```

将 83 行改为:

```
clk_power->UPLLCON=S3C2440_UPLL_48MHz;
```

114 行为设定机器类型 ID, 将在调用 kernel 时传给 kernel, 需将 MACH_TYPE_ SMDK2410 改为 MACH _ TYPE_S3C2440, 从而匹配 S3C2440.

mini2440 使用的 SDRAM 为 HY57V561620FTP, 根据它的芯片手册可知: 64ms 内最少需要刷新 8192 次, 再根据 S3C2440 的硬件手册关于内存控制器的说明可以计算出 REFCNT 应设为 1268. 所以修改/board /mini2440 下的 lowlevel_init.S 文件, 将 126 行的#define REFCNT 1113 改为#define REFCNT 1268.

3.2.4 修改头文件的相关部分

S3C2440 与 S3C2410 相比, 频率控制器多了一个寄存器 CAMDIVN, 该寄存器在上述程序中需要使用, 所以需要在 include/s3c24x0.h 中的结构体 S3C24X0 _CLOCK_POWER 的最后增加一个字段 CAMDIVN, 在 129 行增加:S3C24X0_ REG32 CAMDIVN;

由于初始化 Nand Flash 的代码是条件编译的, 故需要修改 include/configs 目录下的 mini2440.h 文件, 将第 81 行的注释去掉, 并在倒数第二行增加以下宏定义:

```
#define CFG_MAX_NAND_DEVICE 1
#define NAND_MAX_CHIPS 1
#define CFG_NAND_BASE 0
```

在 include/s3c24x0.h 文件中, 依照 S3C2410_ NAND 定义 2440 的 Nand Flash 控制器寄存器数据结构, 以供 board_nand_init 函数使用. 同时依照 S3C2410_GetBase_NAND 函数定义 S3C2440_GetBase _NAND 函数, 以供 board_nand_init 函数使用.

3.2.5 修改 Makefile 文件的相关部分

U-Boot 源文件中有多个 Makefile 文件, 其中主要需要修改的如下所示:

将 board/mini2440/Makefile 的第 28 行 COBJS := smdk2410.o flash.o 改为 COBJS := mini2440.o flash.o.

将 cpu/arm920t/s3c24x0 目录下 Makefile 文件的第 29 行改为 usb_ohci.o nand_flash.o 以将 nand flash 驱动

程序编译进 U-Boot.

修改 U-Boot 目录下的 Makefile 文件, 在第 1882 行处模仿 smdk2410_config 目标增加新目标 mini2440_config.

mini2440_config : unconfig

@$(MKCONFIG) $(@:_config=) arm arm920t mini2440 NULL s3c24x0

3.3 u-boot.bin 文件的生成与测试

从终端进入到 u-boot-1.1.6 目录后, 依次执行以下命令:

make mini2440_config

make

等待几分钟后, 会在 u-boot-1.1.6 目录下生成 u-boot.bin 文件. 使用 H-JTAG 软件将开发板的 Nor Flash 全部擦除, 然后将 u-boot.bin 烧写到 Nor Flash 中. 烧写成功后, 复位开发板, 可以从超级终端中看到如图 2 所示的信息.

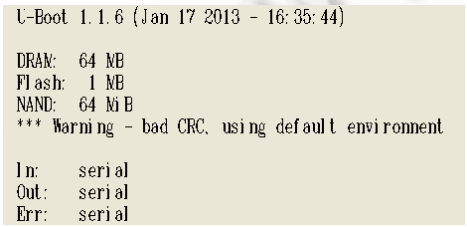

图 2 U-Boot 成功启动后超级终端的输出信息

超级终端输出的信息表明, CPU 和串口能够正常工作, 此时可以按下任意键进入 U-Boot 控制界面如图 3 所示. 在控制界面用 U-Boot 的相关命令测试其功能, 测试结果表明 U-Boot 可以在 mini2440 上稳定地运行, 并能实现其功能. 至此, U-Boot 的移植工作结束.

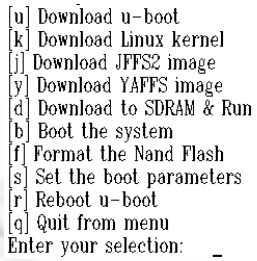

图 3 U-Boot 控制界面

4 结语

利用上述的 U-Boot 移植方法, 可以使 U-Boot 稳定地运行在以 S3C2440 为处理器的 mini2440 开发板上, 并且能够实现利用 U-Boot 下载内核、加载根文件系统等功能, 最终可以正确引导 Linux 内核启动, 为后续嵌入式系统开发打下了坚实的基础.

参考文献

1 韦东山.嵌入式 Linux 应用开发完全手册.北京:人民邮电出版社,2008.22–30.

2 田泽.嵌入式系统开发原理与实践.北京:清华大学出版社,2005.38–62.

3 友善之臂.MINI2440 用户手册.友善之臂,2009.10–12.

4 詹荣开.嵌入式系统 BootLoader 技术内幕. [2012-12-18]. http://www.ibm.com/developerworks/cn/linux/l-btloader/

5 孙弋,等.ARM-Linux 嵌入式系统开发基础.西安:西安电子科技大学出版社,2008.102–115.

(上接第 108 页)

1993(10):57–78.

20 Short RD, Fukunaga K. The optimal distance measure for nearest neighbour classification. IEEE Trans.Inform.Theory, 1981(27),622–627.

21 Blanzieri E, Ricci F. Probability Based Metrics for Nearest Neighbor Classification and Case-Based Reasoning. Proc. of the 3rd International Conference on Case-Based Reasoning Research and Development (ICCBR'99),1999: 14–28.

22 Liu B, Pan J, McKay RI. Entropy-based metrics in swarm clustering. International Journal of Intelligent Systems, 2009(24): 989–1011.

23 Nie Z, Kambhampati S. A Frequency-based Approach for Mining Coverage Statistics in Data Integration.http://www.public.asu.edu/~zaiqingn/freqbased.pdf.

24 Samet S, Miri A. Privacy preserving ID3 using Gini Index over horizontally partitioned data. International Conference on Computer Systems and Applications,2008(1-3):645–651.

25 Li R, Tao X, Tang L, Hu Y. Using Maximum Entropy Model for Chinese Text Categorization. Computer Science, 2004, 3007: 578–587.

26 UCI Repository of Machine Learning Databases. [2012-12-12]. http://repository.seasr.org/Datasets/UCI/arff/

27 郭躬德,黄杰,陈黎飞.基于 KNN 模型的增量学习算法.模式识别与人工智能,2010,5:86–94.