

基于 S3C2410 上 U-Boot 的移植与实现^①

张 伟¹, 刘 斌², 董群锋³

¹(咸阳师范学院 信息工程学院, 咸阳 712000)

²(西安交通大学 计算机教学实验中心, 西安 710049)

³(咸阳师范学院 物理与电子信息工程学院, 咸阳 712000)

摘 要: 操作系统内核移植是嵌入式系统开发的前提和基础, 针对 U-boot 移植的复杂性和多样性, 在分析了 U-boot 的文件结构和启动过程的基础上, 选取了以 SanSung 公司的 S3C2410 为处理器的开发板, 详细介绍了交叉编译环境的搭建、U-boot 的移植、内核的烧写等过程. 移植过程中将 U-boot 的功能与 Linux 的特点相结合, 此方法具有移植速度快、内核修改简单、通用性强的特点. 通过编译测试, 成功实现了 U-boot 在 S3C2410 的移植, 为其他 U-boot 的移植提供了一种参考.

关键词: U-boot; S3C2410; 移植; 嵌入式系统; Bootloader

U-Boot's Transplantation and Implementation Based on S3C2410

ZHANG Wei¹, LIU Bin², Dong Qun-Feng³

¹(School of Information Engineering, Xianyang Normal University, Xianyang 712000, China)

²(Computer Teaching and Experiment Center, Xi'an Jiaotong University, Xi'an 710049, China)

³(School of Physical and Electronic Engineering, Xianyang Normal University, Xianyang 712000, China)

Abstract: The operating system kernel transplantation is the premise and foundation of the embedded system development. In view of the complexity and diversity of the U-boot transplantation, this paper analyzed the file structure and starting process of U-boot, it has chosen S3C2410 of SanSung company for development board, cross-compilation environment construction and U-boot transplantation and the kernel of the burning process were introduced in detail. We combined the function of U-boot with the characteristics of Linux in the transplantation process, this method has the characters of fast transplant, simple modify of kernel and strong commonality. Through compile testing, U-boot transplantation is implemented successful on the S3C2410, provides a reference for other U-boot transplantation.

Key words: U-boot; S3C2410; transplantation; embedded system; bootloader

Bootloader 移植是嵌入式系统开发过程中的一个关键环节, 随着嵌入式系统开发的不断发展和深入, Bootloader 的种类越来越多, 即不同配置的开发板都有与之对应的 Bootloader. Bootloader 的移植依据量体裁衣的原则^[1], 以满足代码的最小化. 在嵌入式开发中, 不管是操作系统的引导, 还是操作系统的配置或运行状态都与 Bootloader 有一定的联系, 所以掌握 Bootloader 的移植是进行嵌入式系统开发的第一步.

1 Bootloader 与 U-Boot

Bootloader 是执行在操作系统运行之前的一段小

程序, 它可以初始化硬件设备、建立内存空间的映像表, 从而建立适当的系统软硬件环境, 为调用操作系统内核做好准备^[2]. 当前流行的可以引导加载 Linux 的 Bootloader 如表 1.

U-Boot(Universal Boot Loader), 是从 FADSROM、8xxROM、PPCBOOT 逐步发展演化而来, 其源码目录、编译形式与 Linux 内核很相似^[3]. 其实, U-Boot 源码就是对应 Linux 内核的精简. 目前, U-Boot 主要支持的操作系统是 Linux, Solaris, VxWorks 等. 随着 U-Boot 的移植在各个不同系列嵌入式处理器上开展, 以支持更多的嵌入式操作系统的装载与引导, 它被认为是功能最

① 基金项目:国家自然科学基金(61102018);陕西省教育厅科研计划(12JK0933);咸阳师范学院专项科研基金(10XSYK308)

收稿时间:2014-01-13; 收到修改稿时间:2014-03-03

多、最具弹性以及开发最积极的开放源码 Bootloader, 其主要特点如下^[4]:

- (1) 源码开放;
- (2) 提供了完备的开发调试文档及网络技术支持;
- (3) 支持如 PowerPC、ARM、x86、MIPS 等多个系列处理器;
- (4) 提供如串口、以太网、SDRAM、FLASH 等丰富的设备驱动源码;
- (5) 提供了灵活的功能设置, 适合 U-Boot 调试, 产品发布等;

表 1 Bootloader 的分类

支持的处理器架构	Bootloader 的名称
X86 体系结构	LIL0,GRUB,ROLO,loadlin, Etherboot, Linux BIOS
ARM 体系结构	Bootldr,Blob,U-Boot, vivi,RedBoot
MIPS 体系结构	PMON
M68k 体系	Sh-boot

2 U-boot相关知识介绍

2.1 U-boot 启动过程

U-boot 启动过程分为两个阶段^[5], 第一阶段主要初始化硬件、加载 U-Boot 第二阶段代码到 RAM、设置好栈跳、转移到第二阶段代码入口等(如图 1); 第二个阶段主要进行配置开发板、初始化环境变量、将内核从 Flash 读取到 RAM 中、配置可用的 RAM 等(如图 2).

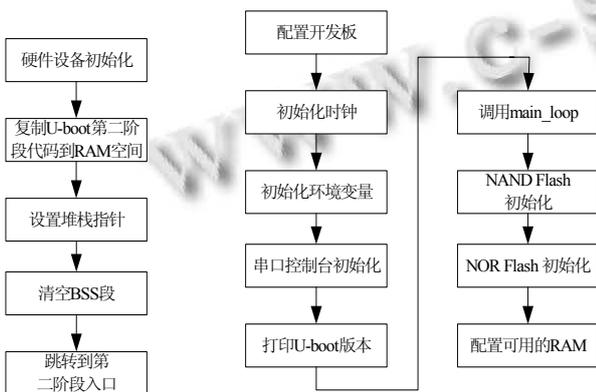


图 1 第一阶段

图 2 第一阶段

2.2 U-boot 的文件结构^[6]

在 U-boot 的移植前, 首先了解一下 U-boot 中的主

要文件结构. 结合移植的需要, 在 ARM 环境下对 U-boot 主要文件结构进行分析, 如图 3 所示, 其中 Board 为依赖于具体的平台目录, 包含 SDRAM、Flash 驱动等; Cpu 为 U-boot 所支持的 CPU 目录, 含串口、网口、LCD 驱动及中断初始化等文件; Lib_arm 为针对 ARM 结构体系的通用文件目录, Include 为 U-Boot 头文件和开发板配置文件目录; Common 为用于实现各种命令的 C 文件目录, 存放独立于处理器体系结构的通用代码, 如内存大小探测与故障检测.

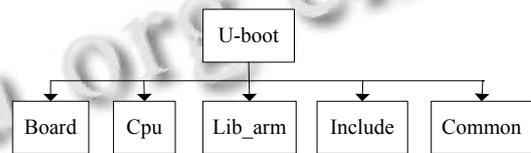


图 3 U-boot 结构

2.3 U-boot 的内存分布

U-boot 在内存空间中设置了一些用于存放环境变量和数据结构的空间, 所以只有熟悉了 U-boot 的内存分布, 才能做好 U-boot 的移植, U-boot 内存主要结构如图 4^[7].

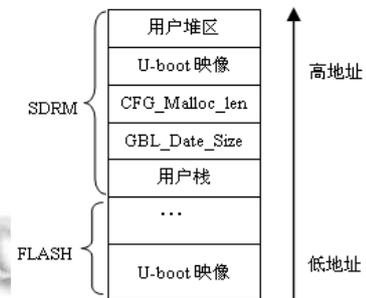


图 4 U-boot 内存分布

(1) U-boot 映像: 是 U-boot 烧写进 flash 的映像, 这部分的大小叫做 monitor_size, 所以在 U-boot 中, 这个二进制文件就叫做 monitor. U-boot 映像会被运送到 RAM 中, 在 RAM 中也有一块 U-boot 映像.

(2) GBL_DATA_SIZE: 存放系统信息.

(3) CFG_MALLOC_LEN: 用于存放堆数据和环境变量.

(4) 环境变量区域: 存储诸如 IP 地址等信息, 在程序被拷贝到 RAM 中时, 环境变量也同时被拷贝到 RAM 中.

2.4 U-boot 的主要功能

U-boot 具有系统引导、上电自检、CRC 校验及基

本辅助等功能,如表 2. 由于 U-boot 具有如此完备、强大的功能,使得其在嵌入式系统开发过程中得到了不断发展, U-boot 的移植支持更多的平台和内核,它必将成为嵌入式系统开发中内核移植的主流工具.

表 2 U-boot 的主要功能

功能名称	主要含义
系统引导	支持 NFS 挂载、RAMDISK(压缩或非压缩)形式的根文件系统;从 FLASH 中引导压缩或非压缩系统内核;
基本辅助功能	可灵活设置、传递多个关键参数给操作系统,适合系统在不同开发阶段的调试要求与产品发布,尤对 Linux 支持最为强劲;
上电自检	SDRAM、FLASH 大小自动检测;SDRAM 故障检测;CPU 型号;
CRC32 校验	可校验 FLASH 中内核、RAMDISK 镜像文件是否完好;

3 U-boot的移植

3.1 U-boot 移植的基本方法

目前 U-Boot 的移植方法主要有两种^[8].

(1) 用 BDI2000 创建目标板运行环境,先将 U-Boot 镜像文件(u-boot.bin)拷贝到目标板 RAM 中的指定位置,然后再用 BDI2000 调试.优点是不需要将 U-Boot 镜像文件烧写到 FLASH 中;缺点是由于被调试的文件是从 RAM 中开始运行的,与 U-Boot 从 Flash 中启动的实际地址运行不一致, BDI2000 的配置较为复杂.

(2) 先用 BDI2000 先将 U-Boot 镜像文件写入 FLASH,再用 BDI2000 和 GDB 进行调试.该方法 BDI2000 的配置简单,调试过程与 U-Boot 移植后运行过程相一致,即 U-Boot 先从 FLASH 启动,再加载到 RAM 中,并从 RAM 正式运行.优点是调试器配置简单,调试过程与 U-Boot 移植后运行过程相一致.缺点是需要不断烧写 FLASH,会降低 Flash 的寿命.

3.2 硬件平台介绍

嵌入式硬件平台处理器选用的是 Samsung 公司 S3C2410,该处理器带有 32 位微控制器,独立的 16KB 指令 Cache 和 16KB 数据 Cache,支持 TFT 的 LCD 控制器,3 路 UART,4 路 DMA,4 路带 PWM 的 Timer,2 个 USB 主机,1 个 USB 设备. S3C2410 处理器最高时钟频率达 203MHz,为手持设备和一般类型的应用提供了低价格、低功耗和高性能微控制器的解决方案^[9].

3.3 U-boot 移植在 S3C2410 上的实现

Step1. 把 arm-linux-gcc-3.4.1.tar.bz2, u-boot-1.2.0.tar.bz2 两个安装包复制到共享目录下.

Step2. 安装交叉编译器

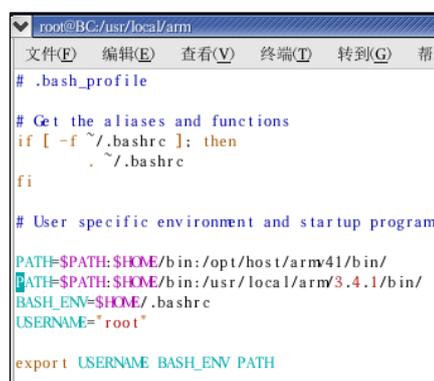
(1) 从共享目录下把 arm-linux-gcc-3.4.1.tar.bz2 复制到/usr/local 下;

(2) 解压 arm-linux-gcc-3.4.1.tar.bz2 (命令: tar -jxvf arm-linux-gcc-3.4.1.tar.bz2);

(3) 解压后会在当前目录下生成一个 usr 目录,将 usr/local 目录下的 arm 目录复制到上一级/usr/local 下;

(4) 修改/root/.bash_profile 配置文件;

(5) 加载配置文件: source /root/.bash_profile;



```

root@BC:/usr/local/arm
文件(E) 编辑(E) 查看(V) 终端(T) 转到(G) 帮助
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin:/opt/host/armv4l/bin/
PATH=$PATH:$HOME/bin:/usr/local/arm3.4.1/bin/
BASH_ENV=$HOME/.bashrc
USERNAME="root"

export USERNAME BASH_ENV PATH

```

图 5 加载配置文件

Step 3.U-boot 移植

(1) 将共享目录下的 u-boot-1.3.3.tar.bz2 复制到/opt 目录下;

(2) 解压 u-boot-1.3.3.tar.bz2(命令: tar -xjcf u-boot-1.3.3.tar.bz2),在当前目录下生成一个 u-boot-1.2.0.目录;

(3) 在当前目录下修改 Makefile,为 mybod2410 添加编译项;

@\$(MKCONFIG) \$(@:_config=) arm arm920t mybod2410 NULL s3c2410

(4) 在 board 下建 mybod/mybod2410 目录;

① cd /uboot-1.3.3/board;

② mkdir my/mybod2410;

(5) 将 board 目录下 SMDK2410 下的所有文件复制到 mybod2410 下;

(6) 将 mybod 下的 SMDK2410.c 重命名为 mybod2410.c;

(7) 在 mybod2410 目录下修改 Makefile;

```
SRCS :=$(SOBJS:.o=.S)$(COBJS:.o=.c)
OBJS :=$(addprefix $(obj),$(COBJS))
SOBJS :=$(addprefix $(obj),$(COBJS))
```

(8) 在 include/configs 中建立配置头文件, 将 smdk2410.h 改为 mybod2410.h;

① cd /u-boot-1.2.0

② cd /include/configs

③ cp include/configs/smdk2410.h include/ configs/
mybod2410.h

Step4. 测试编译

(1) make clean; //清除上次的 make

命令所产生的 object 文件

(2) make mybod2410_configs //定义目标板
make

Step 5. 下载并烧写内核

(1) 用网线将开发板和 PC 机连接起来;

(2) 启动 U-boot 并设置环境变量:

```
setenv ipaddr 192.168.2.100 //设置开发板的  
IP 地址
```

```
setenv serverip 192.22.60.99 //设置 tftp 服务  
器的 IP 地址
```

```
set ethaddr 11.22.33.55.66 //设置 MAC 地  
址
```

saveenv

(3) 在打开服务器, 把编译生产的 U-boot.bin 文件
拷贝到 tftp 服务器程序所在的目录下;

(4) 下载并烧写;

在 U-boot 下用以下命令^[10]

```
tftp 0x30008000 zImage //获取 zImage, 存  
储在内存的 0x30008000 处;
```

```
nand erase 0x40000 0x1c0000 //从 Nand  
flash 的 0x40000 开始擦除 0x1c0000 字节;
```

```
nand write 0x30008000 0x40000 0x1c0000
```

//将内存 0x30008000 为起始地址的 0x1c0000 字节
(zImage)烧写到 Nand flash 的 0x40000 处; 如图 6 所示,
成功完成了 U-boot 的烧写;

通过以上移植过程, 成功地完成了共享目录的建
立、安装交叉编译器、U-boot 的移植、测试编译、内
核的下载及烧写等步骤. 最终, 在 S3C2410 上实现了
U-boot 的移植, 为嵌入式系统开发过程中 U-boot 的移

植提供了一种方法.

```
U-Boot 1.3.3(Oct 12 2013 - 17:29:41)
DARM: 64MB
Flash: 2MB
***Warning-bad CRC, using default environment
In: serial
Out: serial
Err: serial
Mini2410 nand info
```

图 6 U-boot 的烧写

4 结论

U-boot 是功能最强、开放源码、灵活性高的
Bootloader, Linux 操作系统具有易于开发、开源代码、
功能强大及成本低等特点. 本文将两者的特点相结合,
成功地实现了 U-boot 在 S3C2410 上的移植, 该方法克
服了以往嵌入式内核移植过程中的移植速度慢、内核
修改复杂、通用性差的缺点. 通过实验, 移植后的系
统在 ARM 平台上运行稳定, 此方法为嵌入式系统开发
奠定了基础, 并为其他 U-boot 移植提供了一种参考.

参考文献

- 冯林琳, 耿恒山. 基于 S3C6410 的 Uboot 分析与移植. 计算机与现代化, 2013, (1): 119-120.
- 张徽, 张华春. U-Boot 在 S3C2440 上的移植方法. 电子器件, 2007, 8(30): 1423-1424.
- U-boot: the Universal Boot Loader. <http://www.denx.de/wiki/U-boot/WebHome>.
- U-Boot 移植体会. <http://blog.csdn.net/qiaohuic/article/details/4020319>, 2013-06-15/2014-1-10
- 宋国军, 张侃谕, 林学龙. 嵌入式系统中 U-Boot 基本特点及其移植方法. 单片机与嵌入式系统应用, 2004, (10): 79-80.
- 缙新科, 滕永. 基于 ARM9 的嵌入式 Linux 系统分析与移植. 甘肃科学学报, 2011, 23(4): 107-108.
- DENX. The Universal Boot Loader ("Das U-Boot"). <http://www.denx.de/wiki/U-Bootdoc/Presentation>, 2013-10-13/2014-1-10.
- 王亚刚. 嵌入式 Bootloader 机制的分析与移植. 计算机工程, 2010, 36(6): 267-268.
- 刘洪涛. ARM 嵌入式体系结构与接口技术. 北京: 人民邮电出版社, 2012.
- 罗怡桂. 嵌入式 Linux 实践教程. 北京: 清华大学出版社, 2011.