

u-boot 的启动流程分析及移植步骤

张 进, 姜 威

(山东大学 信息科学与工程学院, 山东 济南 250100)

摘要:介绍了嵌入式系统中的一个开发源码的软件 u-boot。u-boot 的实现依赖于 CPU 体系结构的代码, 分为 stage 1 和 stage 2 两部分, 分别用汇编语言和 C 语言实现。详细介绍了 u-boot 在 S3C44B0 开发上的移植步骤, 包括交叉编译环境的建立、移植前的预编译等。实例证明该软件功能强大, 适用范围广。

关键词: bootloader; u-boot; 嵌入式系统; 移植; S3C44B0

中图分类号: TP 31

文献标识码: A

文章编号: 1006-6047(2005)07-0068-04

1 bootloader 简介

bootloader 代码(即启动代码)是芯片复位后进入操作系统之前执行的一段代码, 完成由硬件启动到操作系统启动的过渡, 主要是为运行操作系统提供基本的运行环境, 如初始化 CPU、堆栈、初始化存储器系统等。bootloader 代码与 CPU 芯片的内核结构、具体芯片、应用系统的配置及使用的操作系统等因素有关, 其功能类似于 PC 机的 BIOS 程序。由于 bootloader 和 CPU 及电路板的配置情况有关, 这就决定了不可能有通用的 bootloader, 开发时需要根据具体情况进行移植。在嵌入式 Linux 系统中常用的 bootloader 有 armboot, redboot, blob, u-boot 等, 这其中 u-boot 的功能最强大, 但相对也比较复杂。bootloader 的实现依赖于 CPU 的体系结构, 大多数 bootloader 都分为 stage 1 和 stage 2 两大部分。bootloader 的基本原理见文献[1], 这里不详细介绍。

2 u-boot 简介

u-boot 是 sourceforge 上的一个开源的固件程序。提供了对 PowerPC (MPC5xx, MPC8xx, MPC82xx, MPC7xx, MPC74xx), ARM (ARM7, ARM9, StrongARM, Xscale), MIPS (4kc, 5kc), X86 等处理器的支持, 支持的嵌入式操作系统有 Linux, VxWorks, NetBSD, QNX, RTEMS, ARTOS, LynxOS 等。主要用来开发嵌入式系统初始化代码 bootloader。这个软件的主站点是 <http://sourceforge.net/projects/u-boot>, u-boot 最初是由 denx (www.denx.de) 的 PPCboot 发展而来, 对 PowerPC 系列的处理器支持最完善, 对 Linux 的支持最好。这个开源软件项目经常更新, 是学习硬件底层代码开发很好的样例。

3 u-boot 系统启动流程介绍

bootloader 的实现依赖于 CPU 的体系结构, 大多

数的 bootloader 都分为 stage 1 和 stage 2 两大部分, u-boot 也不例外。依赖于 CPU 体系结构的代码, 如设备初始化代码等, 通常都放在 stage 1, 而且用汇编语言实现, 而 stage 2 则通常用 C 语言实现, 这样可以实现复杂功能, 而且有更好的可读性和移植性^[2,3]。

3.1 stage 1 (start.s 代码结构)

u-boot 的 stage 1 代码在 start.s 文件中, 用汇编语言写成, 其主要代码部分是:

a. 定义入口, 一个可执行的 Image 必须有一个入口点并且只能有唯一的全局入口, 通常这个入口放在 ROM (flash) 的 0x0 地址, 值得注意的是必须告诉编译器知道这个入口, 这个工作主要是修改连接器脚本;

b. 设置异常向量(exception vector);

c. 设置 CPU 的速度和时钟频率及中断控制寄存期;

d. 初始化内存控制器;

e. 将 ROM 中的程序复制到 RAM 中;

f. 初始化堆栈;

g. 转到 RAM 中执行。

使用指令 `ldr pc, RAM 中 C 函数地址` 就可以转到 RAM 中去执行。

3.2 stage 2 (C 语言代码部分)

lib_arm/board.c 中的 start_armboot 是 C 语言开始的函数, 也是整个启动代码中 C 语言的主函数, 还是整个 u-boot (armboot) 的主函数, 主要完成如下工作:

a. 调用一系列的初始化函数;

b. 初始化 flash 设备;

c. 初始化系统内存分配函数;

d. 如果目标系统拥有 NAND 设备, 初始化该设备;

e. 如果目标系统有显示设备, 初始化该类设备;

f. 初始化网络设备相关, 填写 IP, MAC 地址等;

g. 进入命令循环, 就是整个 boot 的工作循环, 接受用户从串口输入的命令, 然后做相应的工作。

4 移植实例

4.1 开发板介绍

开发板有如下几部分组成: S3C44B0X, 2 MB 的 Flash(SST39VF160), 8MB 的 SDRAM(HY57V641620), 4 个 LED; ARM JTAG 接口。

开发板上与 S3C44B0X 相关部分功能框图如图 1 所示。

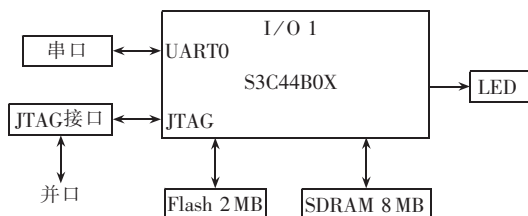


图 1 开发板上与 S3C44B0X 相关部分功能框图

Fig.1 The function blocks relative to S3C44B0X on developing board

4.2 u-boot 文件下载

u-boot 文件下载有两种方法^[4]。

a. 在 Linux 环境下通过 CVS 下载最新的文件。

```
$ cvs -d:pserver:anonymous@cvs.sourceforge.net: /
cvsroot /u-boot login
```

当要求输入匿名登录的密码时,直接按回车键。

```
$ cvs-z6-d:pserver:anonymous@cvs.sourceforge.
net:/cvsroot /u-boot\coP modulename
```

b. 通过 ftp://ftp.denx.de/pub/u-boot/ 下载正式发布的压缩文件。

4.3 u-boot 文件结构

初次下载的文件,解压后 u-boot 文件目录下的文件夹很多,具体的了解可以看其附带的 readme 文件,下面作简单介绍。

a. board: 和一些已有开发板有关的文件,比如 Makefile 和 u-boot.lds 等都和具体开发板的硬件和地址分配有关。

b. common: 与体系结构无关的文件,实现各种命令的 C 文件。

c. CPU: CPU 相关文件,其中的子目录都是以 u-boot 所支持的 CPU 为名,比如有子目录 arm926ejs, mips, mpc8260 和 nios 等,每个特定子目录中都包括 cpu.c 和 interrupt.c, start.s。其中 cpu.c 初始化 CPU、设置指令 Cache 和数据 Cache 等; 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 是 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。

d. disk: disk 驱动的分区分处理代码。

e. doc: 文档。

f. drivers: 通用设备驱动程序,比如各种网卡、支持 CFI 的 Flash、串口和 USB 总线等。

g. fs: 支持文件系统的文件, u-boot 现在支持 cra-

mfs, fat, fdos, jffs2 和 registerfs。

h. include: 头文件,还有对各种硬件平台支持的汇编语言,系统的配置元件和对文件系统支持的文件。

i. net: 与网络有关的代码, BOOTP 协议、TFTP 协议、RARP 协议和 NFS 文件系统的实现。

j. lib_arm: 与 ARM 体系结构相关的代码。

k. tools: 创建 S-Record 格式文件和 u-boot images 的工具。

和移植相关的主要目录有 CPU, BOARD, lib_arm。

4.4 移植步骤

4.4.1 交叉编译环境的建立

要得到下载到目标板的 u-boot 二进制启动代码,需要对下载的 u-boot1.1.1 进行编译。u-boot 的编译一般在 Linux 系统下进行,用 ARM-Linux-GCC 进行编译。自己动手一步一步建立交叉编译环境通常比较复杂,最简单的方法是使用别人编译好的交叉编译工具,下面介绍其方法。

a. 首先在 <http://handhelds.org/download/toolchain> 下载 arm-gcc 3.3.2.tar.bz2。

b. 以 root 用户名登录,将 arm-linux-gcc-3.3.2.tar.bz2 解压到 /root 目录下:

```
# tar jxvf arm-linux-gcc-3.3.2.tar.bz2
```

c. 在 <http://handhelds.org/download/toolchain> 下载 arm-linux-toolchain-post-2.2.13.tar.gz (只是用了它的头文件,主要来自内核 /linux-x.x/include 下)。

d. 将 arm-linux-toolchain-post-2.2.13.tar.gz 解压到 /skiff/local/下:

```
# tar zxvf arm-linux-toolchain-post-2.2.13.tar.gz
```

e. 拷贝头文件到 /root/usr/3.3.2/arm-linux/下,然后删除 /skiff:

```
# cp-dR /skiff/local/arm-linux/include/root/
usr/3.3.2/arm-linux
```

```
# rm-fr /skiff
```

```
# rm-fr /skiff
```

这样 arm linux 交叉编译环境就建立好。

f. 增加 /root/usr/local/arm/3.3.2/bin 到路径环境变量, path = \$path:/root/usr/local/arm/3.3.2/bin, 可以检查路径变量是不是设置正确, #echo \$path。

4.4.2 移植前的预编译

移植 u-boot 到新的开发板上仅需要修改和硬件相关的部分。移植主要包括两个层面的移植,第一层是针对 CPU 的移植,第二层是针对 board 的移植。由于 u-boot1.1.1 里已经包含 s3c44b0 的移植,所以笔者对自己的板子 myboard 的移植主要是针对 board 的移植。移植之前需要详细阅读 u-boot 目录下的 readme 文件,其中对如何移植作了精简的介绍。为了减少移植的工作量,可以在 include/config 目录下选一个和要移植的硬件相似的开发板,笔者选的是 b2 开发板。下面介绍具体步骤。

a. u-boot 1.1.1 下 CPU 文件夹里已经包括了 S3C44B0 的目录,其下已经有了 start.s, interrupts.c 以及 cpu.c, serial.c 几个文件,不需要建立与 CPU 相关的目录了。

b. board 目录下创建 myboard 目录,创建 myboard.c 以及 flash.c, memsetup.S, u-boot.lds 等。不需要从零开始,可选择一个相似的目录,直接复制过来,修改文件名以及内容。在移植 u-boot 过程中,选择的是 u-boot-1.1.1\board\dave\B2 目录。

c. include/configs 目录下添加 myboard.h,在这里放上全局的宏定义等,也不需要从头写,可以在 include/configs 目录下找相似的 CPU 的头文件复制,这里用的是 B2.h,然后进行相关的修改。

d. u-boot 根目录下的 Makefile 文件修改加入:

```
myboard_config:unconfig
```

```
@./mkconfig $(@:_config=) arm s3c44b0 myboard
```

e. 修改 u-boot 根目录下 MAKEALL 文件,加入对板子的申明。在 MAKEALL 中添加入 myboard LIST_ARM7="B2 ep7312 impa7 myboard"。

f. 运行 make clobber,删除错误的 depend 文件。

g. 运行 make myboard_config。

h. 执行完毕,就建立好了整个软件的 makefile 了,这时需要修改生成的 makefile 中的交叉编译选项,打开 makefile 文件,找到其中的语句:

```
ifeq ($(ARCH),arm)
```

```
CROSS_COMPILE=arm-linux-
```

```
end if
```

将其改成

```
ifeq ($(ARCH),arm)
```

```
CROSS_COMPILE=/root/usr/local/3.3.2
```

```
/bin/arm-linux-
```

```
end if
```

这一步和上面的设置环境变量只要有一个即可。

执行 make,报告有一个错误,修改 myboard/flash.c 中的 #include "../common/flash.c" 为 "u-boot/board/dave/common/flash.c",重新编译就可通过。

4.4.3 具体修改要点

预先编译没有错误就可以开始硬件相关代码移植^[3-5],首先对要移植的硬件有清楚的了解,如 CPU, CPU 的控制寄存器及启动各阶段程序在 flash, sdrum 中的布局有清楚的了解。

移植过程中是按照先修改 /include/config/myboard.h 头文件中的大部分参数(大部分的宏定义都在这里设置),然后按照 u-boot 的启动流程逐步修改。需要熟悉 ARM 汇编语言,C 语言,及对 u-boot 启动流程代码有深入的了解。B2 板 CPU 频率为 75 MHz, flash 为 4 Mbit, sdrum 为 16 Mbit, 串口波特率为 115 200 bit/s, 环境变量放在 EEPROM 中,对比两个开发板的不同,需要修改的有 CPU 的频率、flash 和 sdrum 容量的大小、环境变量的位置等。由于参考板已经有了大部分的代码,只需要针对 myboard 进行相应的修改即可。与之相关的文件有 /include/config/myboard.h(大部分的宏定义都在这里设置)、/board/myboard/flash.c(flash 的驱动程序)、/board/myboard/myboard.c(SDRAM 的驱动程序)、/cpu/s3c44b0/serial.c(串口的驱动使能的部分)等。

/include/config/myboard.h 是全局宏定义的地方,主要的修改有

a. 将 #define CONFIG_S3C44B0_CLOCK_SPEED 75 改为 #define CONFIG_S3C44B0_CLOCK_SPEED 64;

b. 将 #define PHYS_SDRAM_1_SIZE 0x01000000 /* 16 MB */ 改为 #define PHYS_SDRAM_1_SIZE 0x00800000 /* 8 MB */;

c. 将 #define PHYS_FLASH_SIZE 0x00400000 /* 4 MB */ 改为 #define PHYS_FLASH_SIZE 0x00200000 /* 2 MB */;

d. 将 #define CFG_MAX_FLASH_SECT 256 /* max number of sectors on one chip */ 改为 #define CFG_MAX_FLASH_SECT 35;

e. 将 #define CFG_ENV_IS_IN_EEPROM 1 /* use EEPROM for environment vars */ 改为 #define CFG_ENV_IS_IN_FLASH 1。

其他如堆栈的大小根据需求修改。

由于 flash, sdrum 的容量发生变化,需要对启动阶段程序在 flash, sdrum 中的位置重新作出安排。将 flash 中的 u-boot 代码放在 0x0 开始的地方,将复制到 sdrum 中的 u-boot 代码安排在 0xc700000 开始的地方。

flash 的修改不仅和容量有关,还和具体型号有关,Flash 存储器的烧写和擦除一般不具有通用性,需要针对不同型号的存储器作相应的修改,这需要查看厂家的使用说明书。修改过程中需要了解 flash 擦写特定寄存器的写入地址和数据命令以及扇区的大小、位置,并进行正确设置。

sdrum 要修改的地方主要是初始化内存控制器部分,由 start.s 文件中的 cpu_init_crit 完成 CPU cache 的设置,由 board/myboard/memsetup.s 中 memsetup 完成初始化 sdrum。S3C44B0 提供了 sdrum 控制器,与一些 CPU 需要 UPM 表编程相比,只需进行相关寄存器的设置修改,降低了开发的难度。

串口波特率不要修改,都是 115 200 bit/s,用 B2 板的串口驱动即可。串口的设置主要包括初始化串口部分,值得注意的串口的 Baudrate 与时钟 MCLK 有很大关系,这可以在 CPU 用户手册中查到。

配置好以后,重新编译 u-boot 代码,将得到的 u-boot.bin 通过 JTAG 口下载到目标板后,如果能从串口输出正确的启动信息,基本上移植就成功了。实际过程中因考虑不周需要多次修改才能成功。移植成功后可以在此基础上添加一些其他功能,如 LCD 驱动等,在此基础上添加功能还是比较容易的。

5 结语

u-boot 是一个功能强大的 bootloader 开发软件,适用的 CPU 平台、支持的嵌入式操作系统很多。本文是笔者在实际的开发过程中根据相关资料进行摸索,成功地移植了 u-boot 的基础上总结出来的。对于不同的 CPU 和开发板其基本方法和步骤是相同的,希望能对有关嵌入式设计人员有所帮助。

参考文献:

- [1] 詹荣开. 嵌入式 BootLoader 技术内幕[EB/OL]. <http://linuxsir.zahui.net/html/34/107039.htm>, 2003-12.
- [2] 吴明辉. 基于 ARM 的嵌入式系统的开发与应用[M]. 北京:人民邮电出版社, 2004.
- [3] 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[J]. 单片机与嵌入式系统, 2004, (10): 41-42.
- SONG Guo-jun, ZHANG Kan-yu, LIN Xue-long. The based character of u-boot in embedded system and porting method [J]. **Singlechip and Embedded System**, 2004, (10): 41-42.
- [4] DENK W. The DENX u-boot and linux guide (DULG) for TQM8xxL[EB/OL]. <http://www.denx.de/twiki/bin/view>

/DULG/Manual, 2005-02.

- [5] 赵俊良, 李哲英. blob 在 S3C44B0 上的移植[J]. 单片机与嵌入式系统, 2004, (9): 37-39.
- ZHAO Jun-liang, LI Zhe-ying. The porting of blob on S3C44B0 board[J]. **Singlechip and Embedded System**, 2004, (9): 37-39.
- [6] 杜春雷. ARM 体系结构与编程[M]. 北京:清华大学出版社, 2003.

(责任编辑: 李 玲)

作者简介:

张 进(1974-), 男, 江苏盐城人, 讲师, 硕士研究生, 研究方向为嵌入式开发与应用(E-mail: zhang6jin@sdu.edu.cn);

姜 威(1963-), 男, 山东济南人, 教授, 主要研究方向为电路设计和数字图像处理。

Start sequence and porting steps of u-boot

ZHANG Jin, JIANG Wei

(Shandong University, Ji'nan 250100, China)

Abstract: U-boot, a software developing source code in the embedded system, is introduced. The realization of u-boot depends on the structure code of CPU. It consists of stage1 and stage2, which are developed with the assemble language and the C language respectively. The porting steps of u-boot on S3C44B0 board are expounded, including the construction of across compiling environment and the pre-compiling before porting. The example proves its powerful functions and wide applications.

Key words: bootloader; u-boot; embedded system; porting; S3C44B0

中国电力教育大学院(校)长联席会第二届第二次会议纪要

中国电力教育大学院(校)长联席会第二届第二次会议于 2005 年 5 月 11~16 日在华南理工大学和澳门大学召开, 各联席会议成员单位和香港理工大学、澳门大学等单位的代表共 60 余人出席了会议。

12 日上午举行了大会开幕式, 华南理工大学副校长李琳教授、联席会主席张保会教授、中国南方电网公司研究中心赵杰主任、广东电网公司陈荣真副总经理、广东省粤电集团公司田燕军副书记、澳门大学科技学院李怡平院长等领导在开幕式上发表了讲话, 对大会的召开表示热烈祝贺。随后, 大会举行了两场学术报告会, 来自香港理工大学和内地六所高校及科研院所的专家、教授就西南电力东输中的七个重大技术问题与高等电力教育的热点主题作了专题报告(详细内容将刊登于《电力自动化设备》杂志)。报告会内容丰富, 精彩纷呈, 参加报告会的有中国南方电网公司、广东电力工程界技术人员和华南理工大学的师生共计一百余人。

13 日, 与会代表应邀到澳门大学交流访问, 参观了校园和图书馆, 与澳门大学科技学院电机电子工程系的领导和师生们进行了广泛的接触, 并在澳门大学国际图书馆举行了互赠纪念品仪式, 表达了真诚的合作愿望。

14~15 日, 联席会以“电力教育的国际需求与联席会议的

国际化”为主要议题进行了交流和探讨, 并特邀香港理工大学电机系主任黄杰波教授作经验交流报告。与会院(校)长们就电力高等教育的课程设置、学生培养、师资队伍建设和教育理念转变、大学院校工作评价体系等议题展开了深入讨论, 发言踊跃, 各抒己见。是难得的观念碰撞、思想凝练, 更是难得的群英荟萃、相得益彰, 充分体现了院(校)长联席会会议的宗旨。

考虑到联席会工作的实际需要, 会议决定四川大学刘俊勇教授担任联席会执行主席, 协助张保会主席开展工作; 并增补西安交通大学方万良教授为联席会副主席。会议商定, 中国电力教育大学院(校)长联席会第二届第三次会议将于 2006 年 5 月在东北电力学院举行, 第二届第四次会议将由重庆大学、四川大学联合承办。会议还决定, 指定《电力自动化设备》杂志为中国电力教育大学院(校)长联席会议信息发布期刊。

华南理工大学和澳门大学为本次会议的成功召开进行了大量认真而细致的准备工作, 并为此付出了大量辛勤的劳动, 与会代表表示由衷的感谢。中国南方电网公司、广东电网公司、广东省粤电集团公司作为协办单位, 给大会以极大的支持, 此外, 香港理工大学、珠海供电公司、斗门供电公司、中山供电公司等单位也对本次会议给予积极的支持, 一并表示感谢。

中国电力教育大学院(校)长联席会