

U-Boot 的启动及移植分析

The Booting and Porting Analyze of U-Boot

(电子科技大学) 黄荐渠 秦东兴 赵曦 贾曦
HUANG Jian-qu QIN Dong-xing ZHAO Xi JIA Xi

摘要:bootloader 开发是嵌入式系统必不可少而且十分重要的部分,U-Boot 为功能强大的 bootloader 开发软件。本文详细分析了 U-Boot 的启动流程,并结合其源码,阐述了 U-Boot 在 S3C2410 目标板上的移植过程,其实现方法和步骤具有一定的通用性。

关键词: bootloader; U-Boot; 嵌入式系统; 移植

中图分类号: TP368.1 文献标识码: B

Abstract: Development of bootloader is the indispensable and very important part of the embedded system,U-Boot is a powerful software to develop a bootloader.In this paper,the booting process of U-Boot is analyzed in detail,the porting process of U-Boot in an Embedded System board based on the S3C2410 CPU are expatiated in terms of the source code of U-Boot, and the realizing methods and step is universal in a sense.

Key words: bootloader; U-Boot; Embedded system; Porting

引言

Bootloader(引导装载器)是用于初始化目标板硬件,给嵌入式操作系统提供板上硬件资源信息,并进一步装载、引导嵌入式操作系统运行的程序,一般固化在 ROM,FLASH 等非易失性存储器上。Bootloader 的设计依赖于 CPU 的体系结构和具体嵌入式板级配置,其设计须依据量体裁衣、定身制作的原则,以满足最小化代码的要求。bootloader 的这种强硬件依赖性决定了开发通用的 Bootloader 相当困难,但 Bootloader 的开发为整个嵌入式系统开发的第一步,十分关键。从可利用资源和实际产品开发周期角度考虑,移植已有的 Bootloader 源码来解决这一问题符合大多数项目的开发要求。

本文详细分析了 Bootloader 的主要任务和结构特点,并结合 U-Boot 源码及 S3C2410 嵌入式开发板,讨论了 U-Boot 移植的方法和步骤。

1 U-Boot 简介

嵌入式 linux 系统中常用的 bootloader 有 armboot, redboot, blob, U-Boot 等,其中 U-Boot(Universal Boot Loader)为当前比较流行,功能比较强大的 bootloader,是遵循 GPL 条款的开放源码项目,其源码目录、编译形式与 Linux 内核相似。支持多种嵌入式操作系统,如 Linux, VxWorks, LynxOS, NetBSD 等,支持多种处理器系列,如 PowerPC, ARM, x86, MIPS, XScale 等,具有较高的可靠性和稳定性,功能设置灵活,调试方便,有丰富的设备驱动源码和开发调试文档。目前 U-Boot 对 PowerPC 系列处理器和 linux 操作系统的支持最为完善。

2 U-Boot 启动流程

由于 Bootloader 的实现依赖于 CPU 的体系结构,大多数

Bootloader 分为 Stage1 和 Stage2 两个部分,分别在 ROM 和 RAM 中运行,U-Boot 也不例外。其中 Stage1 存放与 CPU 体系结构相关的代码,其代码短小精悍,易升级,通常用汇编语言编写。Stage2 通常用 C 语言编写,可以实现更复杂的功能,具有更好的可读性和可移植性。

2.1 Stage1

(1)硬件设备初始化:屏蔽所有中断,设置 CPU 速度和时钟频率, RAM 初始化, LED 初始化, 关闭 CPU 内部指令/数据 cache。

(2)为加载 Bootloader 的 stage2 准备 RAM 空间:空间大小最好为 memory map(通常 4KB)的整数倍,一般而言,1M 的 RAM 足够。

(3)拷贝 Bootloader 的 stage2 到 RAM 空间中:拷贝过程中要确定 stage2 的可执行映像,在固态存储设备的存放地址和终止地址, RAM 空间的起始地址。

(4)设置好堆栈:设置堆栈指针为执行 C 语言代码做好准备。

(5)跳转到 stage2 的 C 入口点:通过修改 PC 寄存器地址实现。

2.2 Stage2

(1)初始化本阶段要使用到的硬件设备:初始化至少一个串口,以便和中断用户进行 I/O 输出信息,初始化计时器等。

(2)检测系统内存映射(memory map):此阶段主要检测 CPU 预留的全部 RAM 地址空间中的哪些被真正映射到 RAM 地址单元,哪些处于未使用状态。

(3)将 kernel 映像和根文件系统映像从 FLASH 上读到 RAM 空间中:首先规划内核映像和根文件系统所占内存范围,明确基地址和映像大小后从 FLASH 上拷贝。

(4)为内核设置启动参数:linux 2.4.x 以后内核以标记列表(Tagged list)的形式来传递启动参数。

(5)调用内核:直接跳转到内核的第一条指令处,在跳转时必须满足 CPU 寄存器, CPU 模式以及 Cache 和 MMU 的相关设置。

在整个启动过程中 U-Boot 的主要函数调用关系如下图:

黄荐渠:在读研究生

基金项目:四川省科技厅基金资助项目(01SG003-11)

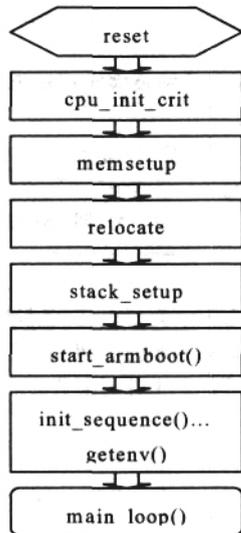


图1 U-Boot 启动代码流程图

3 源码结构

可以从 <http://sourceforge.net/projects/U-Boot> 获得最新发布的 U-Boot 版本, 大多为 tar 或者 bzip2 的压缩包格式, 下载后进行源代码的解压缩:

```
#tar -xjvf u-boot-1.1.2.tar.bz2 或
```

```
#tar -xzvf u-boot-1.1.2.tar.gz
```

解压缩后得到的文件组织形式如下:

- board 目标板相关文件, 主要包含 SDRAM, FLASH 驱动;
- common 独立于处理器体系结构的通用代码;
- cpu 与处理器相关的文件;
- drivers 通用设备驱动程序, 主要有以太网接口的驱动;
- doc U-Boot 说明文档;
- example 在 U-Boot 下运行的示例程序;
- include 头文件和开发板配置文件, 所有开发板和配置文件都在 configs 目录下;
- lib_XXX 处理器体系相关文件;
- net 网络功能相关的文件目录;
- post 上电自检文件目录;
- rtc RTC(Real Timer Clock drivers)驱动程序;
- tools 创建 U-Boot S-RECORD 和 BIN 镜像文件的工具。

4 移植实例

4.1 硬件资源

本次移植硬件资源主要包括 S3C2410 处理器, Intel EP28F128J3A FLASH (16M), MICRON 48LC16M162-75 SDRAM (64M), USB Host/Device, 全功能 JTAG 调口, DM900 AVI COM (10/100M)等。

4.2 建立开发环境

在宿主机上建立 arm-linux-gcc 交叉编译环境。u-boot-1.1.2 目录下的 Makefile 文件中有这一项:

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

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

如果处理器为 arm 则 u-boot 默认交叉编译器为 arm-linux-gcc, 可以从网上下载 arm-linux-gcc-2.95.3, 在 RedHat Linux

系统以 root 用户登陆, 在 /usr/local/arm/2.95.3 目录下安装交叉编译程序, 在 PATH 变量中添加一项: /usr/local/arm/2.95.3/bin,

```
#export ATH=/usr/local/arm/2.95.3/bin:$PATH
```

```
#arm-linux-gcc -v
```

交叉编译环境建好后试着编译 u-boot, 成功的话应该会生成 u-boot.bin 文件。

4.3 移植过程及主要修改文件

在 board 下创建新目录存放开发板相关代码, 并添加文件。

1) 创建 my2410(要移植的目标板名称)目录, 添加 config.mk, flash.c, my2410.c, Makefile, memsetup.S, u-boot.lds 文件。由于大部分支持 S3C2410 的板子都和 smdk2410 相似, 也可以直接复制 smdk2410 的文件到此目录中。smdk2410 平台相关目录为: board/smdk2410/, cpu/arm920t/, cpu/arm920t/s3c2x0/, lib_arm/, include/arm/, include/configs/smdk2410.h。

2) 修改目标板中的主要文件。

* flash.c. 移植过程中 FLASH 驱动一般都需要重写, Intel EP28F128J3A150 在 cmi 板中有支持, 用 board/cmi/flash.c 替换原来的 flash.c, 删除 write_short() 和 write_buff 函数, 把 board/ep7312/flash.c 文件中的 write_word() 和 write_buff() 复制过来。

```
修改 #define FLASH_BLOCK_SIZE 0x00010000 为
```

```
#define FLASH_BLOCK_SIZE 0x00020000 (flash 块大小为 128k)
```

设置 flash 时序为最慢, 把 s3c2410 设成 202.8MHZ, 并且工作在异步模式, 在 cpu/arm920t/start.S 文件中添加如下代码:

```
Mrc p15,0,r1,c1,c0,0
```

```
Orr r1,r1,#0x00000000
```

```
Mcr p15,0,r1,c1,c0,0
```

* memsetup.S. 为了以后能用 u-boot 的 GO 命令执行修改过的用 load 或 tftp 下载的 u-boot, 在标记符 '0:' 添加代码:

```
Mov r3,pc
```

```
Ldr r4, =0x3FFF0000
```

```
and r3, r3, r4 //以上三句得到实际启动的内存地址
```

```
add r0, r0, r3 //用 GO 命令调试 uboot 时, 启动地址在 RAM
```

```
add r2, r2, r3 //把初始化内存信息的地址, 加上实际启动地址
```

* config.mk. 此文件用于设置程序链接的起始地址, 预留 1M 空间, 修改 0x33F80000 为 0x33F00000

* my2410.c. 主要为 SDRAM 的驱动程序, 可不作修改。

* u-boot.lds. 作如下修改:

```
.text
```

```
{
```

```
cpu/arm920t/start.o(.text)
```

```
*(.text)
```

```
}
```

* Makefile. 修改 COBJS := smdk2410.o flash.o 为 COBJS := my2410.o flash.o

3) 在 include/configs 目录下复制 smdk2410.h 并改名为 my2410.h, 在这里放上全局的宏定义, 很多参数都要在这里修改, 包括 CPU, 系统时钟, RAM, FLASH 配置信息, 内存映射参数, u-boot 环境变量, 内核启动参数等。主要参考目标资源的硬件手册来修改。

4) 实现网卡功能

DM9000 网卡在 u-boot 中没有相应驱动, 必须自己添加驱

动,具体步骤如下:

把驱动复制到 drivers/dm9000.c

在 drivers/Makefile 中加入 dm9000.o

在 lib_arm/board.c 中修改 CS8900 为 DM9000

在 include/configs/my2410.h 中添加如下代码:

```
#define CONFIG_DRIVER_DM9000
```

```
#define DM9000_BASE 0x08000000
```

```
#define DM9000_bus16 1
```

```
#define CONFIG_ETHADDR 08:00:3e:26:0a:5b
```

```
#define CONFIG_NETMASK 255.255.255.0
```

```
#define CONFIG_IPADDR 192.168.1.45
```

```
#define CONFIG_SERVERIP 192.168.1.23
```

5)回到 u-boot 根目录下修改 Makefile,加入如下代码:

```
my2410_config: unconfig
```

```
@/mkconfig $(@_config=) arm arm920t my2410
```

其中 ARM 是 CPU 的种类,arm920t 是 ARM CPU 对应的代码目录,my2410 是自己主板对应的目录。

6)编译。

```
#make my2410_config
```

```
#make
```

编译成功生成 u-boot.bin。

7)烧写。把编译生成的 u-boot.bin 烧写到目标板 flash 中。

移植过程结束。

5 结束语

在整个 U-Boot 的移植过程中,往往会遇到许多事先无法预料的问题,特别是对 U-Boot 源码目录中相关文件不熟悉的情况下,移植会更加困难。读者必须耐心仔细地阅读 U-Boot 的使用说明文档和目标板的硬件手册。目前,笔者移植的 U-Boot 已经在目标板上稳定的运行,希望本文在移植过程中采用的方法和步骤能为相关工作的开发人员提供参考。

本文作者创新点:本文根据具体开发板的硬件配置,在 U-Boot 中寻找已有良好支持,且与目标板硬件配置相似的标准版作为移植的参考板,并修改相关源代码以适应具体应用需要。这种方法避免了在嵌入式系统开发过程中遇到新的应用需求时 bootloader 需要重新编写的局面,通过移植无疑能缩短整个系统的开发周期。虽然移植的具体方法和步骤在不同的应用环境中存在差异,但善于利用已有资源的思想是每个开发人员应该具备的。

参考文献:

- [1]The DENX U-Boot and Linux Guide (DULG) for TQM8xxL[DB/OL].<http://www.denx.de/wiki/view/DULG/Manual>
- [2]<http://cvs.sourceforge.net/viewcvs.py/u-boot/u-boot>
- [3]Samsung S3C2410A User's Manual
- [4]詹荣开. 嵌入式 Bootloader 技术内幕[DB/OL].http://blog.chin-aunix.net/u/1807/showart_262602.html
- [5]钱峰,刘晔. U-Boot 在 S3C44B0 上的移植[J]微计算机信息,2006,9-z119-121

作者简介:黄荐渠(1983-),男(汉族),在读研究生;主要研究方向:嵌入式系统;秦东兴(1970-),男(汉族),副教授,博士,硕导,主要研究方向:精密仪器与机械,嵌入式系统;赵曦(1984-),男(汉族),在读研究生;主要研究方向:精密仪器与机械;贾曦(1983-),男(汉族),在读研究生;主要研究方向:精密仪器与机械。

Biography:Huang Jianqu, Male (Nationality:Han), Graduate student; Research field:Embedded system.

(610054 四川 成都电子科技大学机械电子工程学院) 黄荐渠 秦东兴 赵曦 贾曦

通讯地址:(610054 四川省 成都市成华区建设北路二段四号,电子科技大学机械电子工程学院)黄荐渠

(收稿日期:2008.2.15)(修稿日期:2008.3.25)

(上接第 138 页)

在本运动控制系统中为了提高配置过程的速度,达到上电后的高速硬件精插补器的实时性,并且由于仅对单个的 FPGA 进行配置。这里就选择了主并行配置。ARM 与 FPGA 连接示意图如图 5 所示。

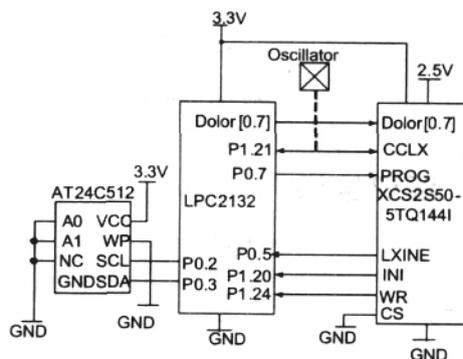


图 5 主并行配置模式

由图 5 可以看出这里使用 ARM 的 P0.2 口和 P0.3 口外部扩展 AT24C512 的 E²PROM 存储芯片。而对于 FPGA 的连接则采用并行连接方式,根据 FPGA 并行配置规则,使用 ARM 的相应口连接 FPGA。

本文作者创新点

在本文中主要就所设计的运动控制系统的硬件进行了详细的介绍,首先提出了本运动控制系统的总体规划。根据规划,对其中的几个重要模块做了详细的设计说明。最后对系统中的硬件精插补器的开发流程做了简单介绍,并在此基础上做了详细的设计说明,接着对精插补器中的几个重要功能模块进行设计,并给出了详细的实现方式说明。

参考文献:

- [1]刘文生,宋昌才.运动控制技术的应用.Vol. 31 No. 6 2005.12: 25- 27
- [2]丛爽,李泽湘编著.实用运动控制技术.电子工业出版社 2006.
- [3]黄益群,张海荣,严彩忠,陈先锋,舒志兵.运动控制新技术.Servo Control 2005.11: 20- 24.
- [4]田家林,陈利学,寇向辉. FPGA 在运动控制系统中的设计.[J]微计算机信息,2007.3,2: 212- 213

作者简介:苏蕊,(1972-),女,山东济宁人,信息工程学院讲师,主要研究方向为多媒体应用技术,智能控制等。

Biography:SU Rui, (1972-), Female, Born in Shandong Jining, Lectuer, Main Research Areas: Muliti-media Application Technology, Intelligent Control.

(276826 山东 日照 济宁医学院信息工程学院)苏蕊

通讯地址:(276826 山东省日照市学苑路 669 号济宁医学院信息工程学院软件教研室)苏蕊

(收稿日期:2008.2.15)(修稿日期:2008.3.25)