

基于 ARM 嵌入式系统的 Bootloader 的设计与实现

刘 娅

(连云港师范高等专科学校 江苏 连云港 222000)

摘 要:在计算机、互联网和通信技术高速发展的同时,嵌入式系统开发技术也取得迅速发展,嵌入式技术应用范围的急剧扩大。但是在嵌入式系统设计中,针对嵌入式处理器和操作系统的 Bootloader 代码的设计是一个难点。在基于 ARM720T 核的嵌入式处理器芯片的 EP7312 上进行设计,总结出嵌入式处理器芯片的 Bootloader 设计的一般模式及其中的一些关键技术,并给出了 Bootloader 的详细设计流程。

关键词:嵌入式系统;Bootloader;ARM;操作系统

中图分类号:TP316

文献标识码:B

文章编号:1004-373X(2006)07-142-03

Design and Realization of Bootloader Based on ARM Embedded System

LIU Ya

(Lianyungang Teachers College, Lianyungang, 222000, China)

Abstract: When computer, Internet and telecom technology developing quickly, embedded system is also developing fast. Application of embedded system is more and more. But in embedded system design, it is difficult to design a Bootloader for embedded CPU and operation system. In this paper, a design and realization of Bootloader of embedded processor chip called EP7312 based on ARM720T. It gives the main framework of Bootloader and explains the key technology. Furthermore, it shows flow chart in detail.

Keywords: embedded system; Bootloader; ARM; operation system

1 引 言

随着嵌入式产品中高端微处理器 ARM 的加入以及软件上操作系统的支持,使得整个嵌入式系统拥有了完整的构架。现在,专门为嵌入式产品开发的各个操作系统层出不穷,Window CE, Pocket PC, Linux 等等。然而,如何进行加载操作系统这个问题却很少有人提出。这就产生了另一个相关主题 Bootloader。Bootloader 本身的功能就是引导与加载内核镜像。如何实现 Bootloader 的基本功能就是本课题的一个论题。除了基本功能,Bootloader 还能有什么更加具体的扩展功能来方便各个系统开发者,都属于本课题讨论的范畴。

简单地说,Bootloader 就是在操作系统内核运行前运行的一段小程序,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固件^[1]。通过这段小程序,可以初始化必要的硬件设备,完成处理器和周边电路和设备正常运行所要的初始化工作;可以屏蔽底层硬件的差异,使上层应用程序的编写和移植更加方便;调用操作系统内核,真正起到引导和加载内核的作用。Bootloader 是底层硬件和上层应用软件之间的一个中间件软件。他创建内核需要的一些信息并将

这些信息通过相关机制传递给内核,从而将系统的软硬件环境带到一个合适的状态。同时还提供基本输入、输出系统监控程序功能,还可具有一定的调试功能。

2 Bootloader 的实现流程

Bootloader 的启动过程可以是单阶段的,也可以是多阶段的。通常多阶段的 Bootloader 能提供更为复杂的功能,以及更好的可移植性。从固态存储设备上启动的 Bootloader 大多数是二阶段的启动过程:stage 1 和 stage 2 两部分^[2]。

整个 Bootloader 的实现流程如图 1 所示。

具体思路为:在 Bootloader 做完一些硬件初始化工作后,而在加载内核镜像之前,先在一定的时间内等待用户键盘输入,如果没有,则进入启动加载模式,直接加载内核镜像进行启动;如果有,则进入下载模式,这时就可以根据自己的需要以及 Bootloader 的支持情况,做一些其他的工作。

Bootloader 是依赖于硬件而实现的,特别是在嵌入式系统中。不同的体系结构需求的 Bootloader 是不同的;除了体系结构,Bootloader 还依赖于具体的嵌入式板级设备的配置。也就是说,对于两块不同的嵌入式板而言,即使他们基于相同的 CPU 构建,运行在其中一块电路板上的 Bootloader,未必能够运行在另一块电路开发板上。由于

收稿日期:2005-10-20

Boot Loader 的实现依赖于 CPU 的体系结构,因此大多数 Boot Loader 都分为 stage 1 和 stage 2 两大部分。依赖于 CPU 体系结构的代码,比如设备初始化代码等,通常都放在 stage 1 中,而且通常都用汇编语言来实现,以达到短小精悍的目的。而 stage 2 则通常用 C 语言来实现,这样可以实现较复杂的功能,而且代码会具有更好的可读性和可移植性^[2]。模式的转换设计主要在阶段 2 中实现。stage 1 通常包括以下步骤^[3]:

- (1) 初始化 CPU 各种模式的堆栈和寄存器;
- (2) 初始化系统时钟、系统总线速率以及一些其他要用到的系统常量;
- (3) 初始化存储控制器,如 FLASH 和 SDRAM 大小、类型、数据宽度和地址范围、划分板上内存空间范围;
- (4) 初始化通用 I/O 和各种控制器^[4];
- (5) 为 stage 2 做准备。

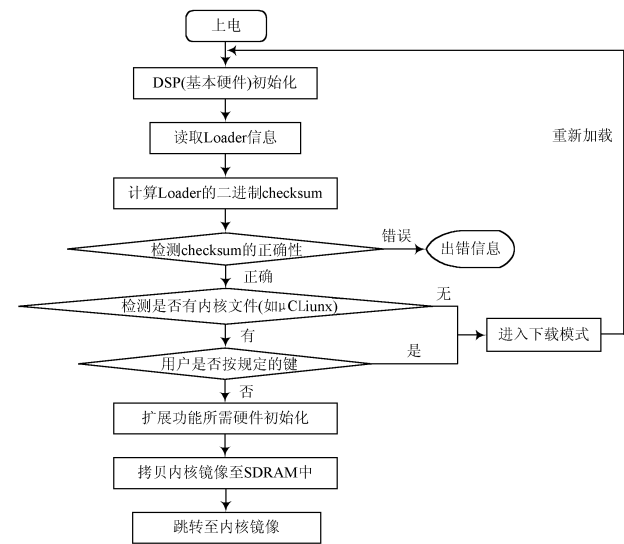


图 1 Bootloader 流程图

stage 2 通常包括以下步骤(以执行的先后顺序):为了增加 Bootloader 对平台的控制,一般 Bootloader 都会设计成支持命令输入的方式,通过串口来接收用户的命令。一般在平台调试完毕后,可以在不用人工干预的情况下自动加载内核;而在调试阶段,基本上是通过 Loader 所支持的命令来进行操作的。提供足够丰富的命令,能极大简化和全面测试开发平台^[5]。

- (1) 初始化一些硬件设备,如串口、网口、LED 等^[4];
- (2) 检测系统内存映射;
- (3) 将内核和根文件系统映像从 FLASH 上读到 RAM 空间中,引导加载内核。

3 模式设计

3.1 模式分类

大多数 Bootloader 都包含两种不同的操作模式:“启动加载”模式和“下载”模式,这两种区别对于开发人员才

有意义。

启动加载模式 也称“自主”模式,即 Bootloader 从目标机上的某个固态存储设备上将操作系统加载到 RAM 中运行,整个过程没有用户的介入。这种模式是 Bootloader 的正常工作模式,因此当以嵌入式产品发布的时候,Bootloader 必须工作在这种模式下。

下载模式 在这种模式下,目标机上的 Bootloader 将通过串口或者网络连接或者其他通信手段从主机下载文件,比如:下载内核镜像和配置文件等。

对于普通用户来说只需要 Bootloader 的启动加载模式,但是对于开发者来说,则需要下载模式,因为他们需要时时刻刻地进行一些镜像的更新。为了在两者之间做到兼顾,所以既支持启动加载模式,也支持下载模式。这里主要介绍下载模式的设计。

3.2 下载模式

由于启动加载模式比较简单,只要 Bootloader 运行正常,内核和文件系统正确无误,一般就可以。这里就不再多说。

在下载模式中,从主机下载的文件通常首先被 Bootloader 保存到目标机的 RAM 中,然后被 Bootloader 写到目标机上的 FLASH 类固态存储设备中。Bootloader 的这种模式通常在第一次安装内核与根文件系统时使用;此外,以后的系统更新也会使用 Bootloader 的这种工作模式。工作于这种模式下的 Bootloader 通常都会向他的中断用户提供一个简单的命令行接口。

本课题允许用户在固态存储设备上有多个内核文件和配置文件等(内核是限定最多只能放两个的,而配置文件则是依赖存储系统的,存储系统的剩余空间能放多少就放多少)。但是被设置成运行文件的各只有一个。所以下载模式提供了以下几个功能:

- (1) 下载文件;
- (2) 设置默认启动文件;
- (3) 删除文件;
- (4) 退出。

3.2.1 下载文件

下载文件的功能将用户需要用的新内核文件,或者其他的配置文件等等放到存储器中。用户输入“load”后,下载模式给出提示信息,要求用户输入合法文件名,然后程序就会下载文件到 RAM 中,如果从 ROM 中申请到足够的空间,就会再把他保存到 ROM 中去。整个下载过程如图 2 所示。

3.2.2 删除文件

存储设备的容量是有限的,但是用户有些时候可能要更新文件,这个时候就需要删除存储设备中的一些文件。用户输入“del”后,下载模式给出提示信息,要求输入被删除文件名。如果文件存在,就会删除文件。如果文件不存在,则提示出错信息。如果被删除文件是默认运行文件,则给出确认提示信息。如图 3 所示。

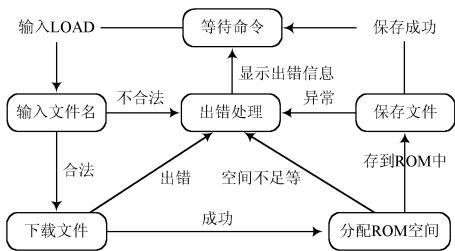


图 2 下载文件

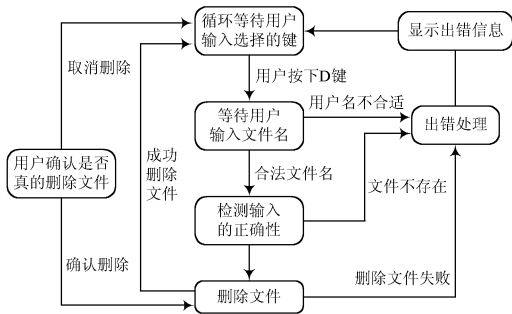


图 3 删除文件

3.2.3 设置默认文件

由于本系统允许存储设备中同时存放多个内核文件和配置文件,所以系统运行以后需要有默认的启动文件部分。用户输入“set”后,下载模式给出提示信息,要求用户输入默认运行文件名。然后程序会去获取文件的相关信息,如果文件存在,为文件设置默认运行文件标志。设置成功后,程序会回到循环状态,等待用户的命令;如果文件不存在,则提示出错信息。

3.2.4 退出下载模式

用户在下载模式中是无法加载内核,让内核运行的,只有退出下载模式,重新启动,进入启动加载模式后,引导内核,运行内核。

4 结 语

Bootloader 是操作系统和硬件的枢纽,相对于操作系

统内核来说他是一个硬件抽象层,他负责初始化硬件,引导操作系统内核,检测各种参数给操作系统内核使用,一个功能完备的大型 Bootloader 工作量相当于一个小型的操作系统。嵌入式领域中操作系统的移植关键在于 Bootloader 的移植和操作系统内核硬件相关部分移植。设计和实现一个好的 Bootloader 将大大提高操作系统移植的稳定性并大大加快操作系统移植的周期^[4]。

本文介绍的 Bootloader 代码已经在基于 EP7312 芯片开发的系统上运行并测试通过。以上所列的设计流程不是一成不变的,在具体应用中要权衡取舍。本系统还需要进一步完善,在文件的控制上还需要加强,比如察看文件属性,以及如何实现 TFTP 模式下载等等。

参 考 文 献

[1] 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应用,2004(10): 78~81.

[2] 詹荣开. 嵌入式 Bootloader 技术内幕[Z]. IBM DW,2003.

[3] 胥静. 嵌入式系统设计与开发实例详解——基于 ARM 的应用[M]. 北京:北京航空航天大学出版社,2005.

[4] 李汉强,邱巍. 基于 IntelPXA26X 处理器的 Bootloader 的设计与实现[J]. 武汉理工大学学报,2003,27(6):770~773.

[5] 白浪,张思东. WinCE 系统下 Bootloader 的开发[J]. 单片机及嵌入式系统应用,2004(2):24~27,42.

[6] 朱颖,黄光明. Motorola 微处理器的 Bootloader 分析与应用[J]. 单片机及嵌入式系统应用,2004(6):27~29.

[7] 杜春雷. ARM 体系结构与编程[M]. 北京:清华大学出版社,2004.

[8] 田立峰,崔健,陈伟,等. 双核 DSP 芯片 TMS320VC5421 的并行引导方案[J]. 国外电子元器件,2004(8):14~16.

[9] 孙天泽,袁文菊,张海峰. 嵌入式设计及 Linux 驱动开发指南——基于 ARM9 处理器[M]. 北京:电子工业出版社,2005.

电子标签普及 RFID 市场规模今年达 60 亿

随着第二代身份证的换发及电子标签的日益普及,中国 RFID(无线射频识别)的市场也急剧扩大,其规模有望在今年达到 60 亿元。

在近日由软件巨头 SAP 牵头举行的 RFID 专家座谈会上,信产部信息化推进司司长陈伟透露,去年中国的 RFID 市场规模已经达到 36.9 亿元,而且这个数字今年还将大幅增长,达到 59.7 亿元。

陈伟表示,目前我国对 RFID 技术的应用主要集中在食品及医药安全等领域,但是第二代身份证的全面发放则是 RFID 市场规模在去年得以迅速扩大的最重要原因。此外,随着第二代身份证换发工作的开展及

结束,中国 RFID 市场将在 2008 年达到高潮,并在之后有所回落。但随着零售业的逐渐成熟,RFID 技术应用还将在 2010 年再度爆发。

SAP 全球执行董事会董事韩迎科表示,目前零售巨头沃尔玛已经广泛采用了 RFID 技术,其使用 RFID 标签的商品进货量是非 RFID 商品的三倍之多,并因此使其公司库存成功下降了超过 10%。

此外,韩迎科也称,上海农产品中心批发市场经营管理有限公司已经成为 SAP 在华首个商务 RFID 客户。其表示,由于中国经济在全球经济体系中的重要地位,RFID 技术在中国的应用将产生全球性的影响。