

谈如何通过网络接口实现 DSP 片外 Flash 在线系统编程

陈洪钢, 黄华伟

(电子科技大学, 四川 成都 610054)

摘要:文章针对传统设计采用 RS232 作为数据传输手段而造成的速度过慢的问题,设计了一种通过网络接口,使用一种特殊的 UDP 包对 BF531 片外 Flash 在线系统编程的方案,并给出了在系统编程的具体步骤和实现结果。实践证明该方法无须实现 TCP/IP 协议栈,速度快,体积很小,很好地解决了以 DSP 为核心的应用系统软件的维护和升级问题。

关键词:BF531; Flash; 网络接口; ISP; UDP

中图分类号:TP368

文献标识码:A

文章编号:1009-2374(2009)07-0129-02

在以 DSP 为核心的现代电子设备中,通常将可执行代码存放在 Flash 中,设备启动的时候,DSP 通过引导加载机制将 Flash 中的代码转移到高速存储器中运行。Flash 在系统编程 (ISP) 是指以在不改变硬件环境的条件下,在线将程序代码 (固件)下载到 Flash 中,从而方便地实现系统软件的调试和升级。BF531 是 ADI 公司生产的一款高性能 DSP。ADI 公司在其软件 VDSP 下提供了通用的在系统编程工具“Flash Programmer”,但是该工具需要使用 JTAG 接口,并且需要专用仿真器。另外一种方法是采用 RS232 作为通信接口,通过一段引导程序来进行 Flash 的在系统编程,缺点是速度很慢。由于目前大多数电子仪器都已经具有网络接口,本文设计了一种通过网络接口进行 Flash 在系统编程的方案。采用的网络接口芯片为常见的 DM9000a,由于网络通信中常见的 TCP/IP 协议栈过于庞大,本文中未使用协议栈,而是设计了一种简单的实现方法:直接构建含特殊标记头的 UDP 广播包来代替。

一、BF531 的 Flash 启动加载过程

LDR 文件是最终烧写到 FLASH 上面的程序文件,可以有多种格式。这里,我们生成的 LDR 文件是二进制格式,LDR 文件分为多个块,每个块都分为一个“BLOCK”和一个 10 字节的“HEAD”。在“HEAD”中,头四个字节表示块应加载在 BF531 地址空间的地址,后四个字节表示“BLOCK”的大小,最后两个字节表示块的属性。

BF531 芯片本身已经提供了 4 种启动代码加载方式,通过启动模式引脚 BMODE 来选择,本设计中将该引脚设置为最常见的“01”模式,即“从 8/16 位的 FLASH/PROM 启动”模式。

BF531 内部有一个 BOOTROM 区,用以存放“引导核”,BF531 启动的时候会先运行该引导核,自动将从 Flash 起始地址 0x2000000 开始的 LDR 文件分块按照用户的要求分别加载到片内 RAM 和 SDRAM 中。加载完毕后,引导核会修改程序指针 PC,使之成为片内 RAM 的启动地址 0xFFA08000,启动用户自己的应用程序。

二、总体设计

需要设计一个 Bootloader 程序和一个与之配合的上位机程序,并将 Bootloader 烧在地址 0x20000000 处。DSP 启动后,“引导核”会自动启动该程序,Bootloader 程序延时判断上位机

是否发出了连接命令。如果有,进入更新固件模式,可以进行 Flash 的烧写;否则,则进入正常启动模式,用以加载并执行用户程序。

(一) Bootloader 与用户程序

根据上面所述,启动的顺序是:“引导核”自动启动 Bootloader,运行完成后再加载运行用户程序。所以 Bootloader 程序应该独立于用户程序,因为 Bootloader 一般不需要更新,而且,在系统编程出现意外的时候,不会导致 Bootloader 被破坏。由于 Bootloader 需要在启动的时候就运行,所以应该事先通过 JTAG 使用仿真器将其烧写到地址 0x20000000 处。而对于常用的 Nor Flash,编程的时候会先把相应的整个扇区擦除,所以为了方便编程,应该使用用户程序的 LDR 文件的起始烧写地址固定,并且为一个扇区的起始地址,本设计中固定为 0x20004000。

(二) 网络通信协议设计

嵌入式仪器一般资源都比较少,所以 Bootloader 体积应该设计的尽量的小。最常用的网络通信协议为 TCP/IP 协议栈,虽然功能强大,但是体积过于庞大。考虑 Bootloader 只需要网络接口完成最简单的通讯功能,而且通信量很小,本文设计了一种简单的方法:直接构建含特殊标记头的 UDP 广播包。

详细的实现方法:上位机端发送数据的时候,在普通的 UDP 包的“payload”区加上一个自定义字符串的“标记头”,并定义为广播包,发送。Bootloader 端接受时采用“杂散模式”接受,直接分析网络芯片 DM9000A 接受到的数据,如果发现是 UDP 包并且具有自定义字符串的“标记头”,则接受,否则,抛弃。相反,Bootloader 端发送数据的时候,按照 UDP 包标准格式直接手动填充一个完整的 UDP 广播包,并加上“标记头”,上位机接受的时候,按正常接收方式,即可根据“标记头”判断是否是 Bootloader 传出来的包。

三、Bootloader 设计

如图 1 所示,Bootloader 功能上可以分为 4 个模块,主控制模块,数据通信模块,Flash 操作模块,用户程序加载启动模块。Bootloader 运行的时候,先向上位机发出握手请求,等待 1 秒,如果有握手回应信号则证明有上位机相连,Bootloader 进入命令等待状态,如果没有,则直接进入用户程序加载启动模块。当一个完整的上位机命令被接收到后,Bootloader 解析

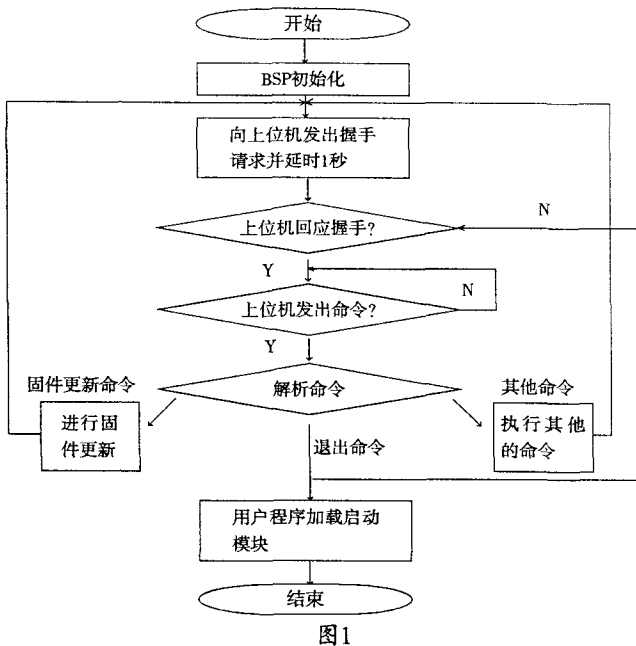


图1

命令,并执行相应的功能。执行完相应的功能后会再发出握手请求(除非命令为退出命令),等待上位机握手,周而复始。

(一)主控制模块

实现功能:实现 Bootloader 与上位机的握手,解析并执行上位机发出的命令,并总体控制各个模块。

(二)数据通信模块

实现功能:控制网络控制芯片 DM9000A 进行数据输入和输出。主要需要实现数据接收功能,数据发送功能,DM9000A 芯片的初始化和复位功能。

本设计中采用了上文中已经叙述的特殊的网络通信方式。设计数据发送函数时,先定义一个 UDP 的标准广播包的模版,在 UDP 包的“payload”区开始的地方添加好自定义的“标记头”,发送的时候直接在该模版的“payload”区后上加入准备传送的数据,通知 DM9000A 发送即可,为了避免计算校验位造成速度减慢,本设计中将 UDP 头的校验位设置为 0,即不校验。设计数据接收函数时,只需要先读取“payload”的头几个字节,判断是否为定义的“标记头”,如果是,则去掉该“标记头”和 UDP 包的头,获得真正需要的数据,如果不是,则抛弃。

(三)Flash 操作模块

实现功能:实现一些常用的操作 Flash 功能接口函数,供主控制模块使用。主要需要实现的功能函数有:Flash 写函数,Flash 读函数,Flash 块擦除函数,Flash 解锁函数,Flash 校验函数。

普通的 nor Flash 编程前需要将相应的扇区解锁并擦除,所以如果该扇区有别的数据,会被一并擦除。为了能自由地编程任意地址的数据,本设计中特别定义了一个 1 个块大小的缓冲区,暂存原 Flash 块里面的数据,编程的时候将更新数据和原数据进行合并再后在一起编程进入 Flash,解决了该问题。

(四)用户程序加载启动模块

实现功能:完成类似 BOOTROM 区“引导核”的工作,即加载起始地址为 0x20004000 的用户程序 LDR 文件到片内 RAM 或 SDRAM,然后清空所有相关堆栈和寄存器,最后修改 PC 指针为 0xFFA08000,开始执行用户程序,所以该程序应该使用汇编编写。ADI 在 VDSP 下提供“引导核”的示例程序,可以参考该程序设计本模块。需要注意的是本模块绝对不能被加

载的用户程序覆盖掉,所以不能把这段程序放到片内 RAM,而应该把这段程序放在 SDRAM 的最后,保护起来。

四、上位机设计

本设计中上位机在 Windows XP 操作系统下采用 NI 公司的 Labwindows/CVI 8.0 开发设计,如图 2 所示。需要通过 UDP 包与 Bootloader 的通信和程控面板的显示。由于 CVI8.0 自身没有带 UDP 相关驱动,需要加入 Windows 的 API 库 wsock32.lib 进行开发。调用该库进行 SOCKET 编程,设计好 UDP 的读、写、打开、关闭函数。在设计发送/接收函数的时候应该加上或去掉定义的“标记头”。

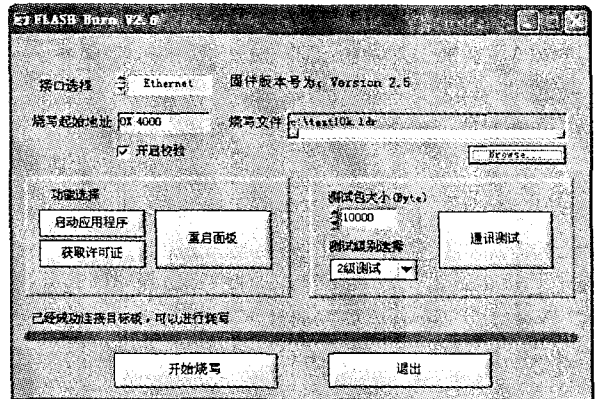


图2

五、实现结果

本设计已经在使用 DSP 芯片 BF531 和网络控制芯片 DM9000A 的目标板上成功实现。完整的实现步骤如下:先将 Bootloader 生成 LDR 文件,再将其通过仿真器烧写到 0x20000000,使用交叉双绞线将目标板与电脑相连接,在电脑上打开上位机,给目标板上电,此时上位机与目标板进行握手,如果成功,面板会显示“已经成功连接目标板”,选择需要烧写的用户程序的 LDR 文件和烧写地址,点击开始烧写即可。

六、结语

本文提出了一种通过特殊的 UDP 广播包进行 Flash 的在系统编程的方法,并成功实现。该方法抛弃了 TCP/IP 协议栈,因而大大的减小了自身的体积,而且采用网络接口,使系统编程的速度得到了极大的提升。很好地解决了目标系统的系统程序升级和调试的问题。

参考文献

- [1]Analog Devices, Inc. ADSP -BF533 Blackfin Processor Hardware Reference, 2007.
- [2]Analog Devices, Inc. VisualDSP ++4.5 Loader and Utilities Manual, 2006.
- [3]Analog Devices, Inc. ADSP-BF533 Blackfin Booting Process, 2006.
- [4]Lvzhongyong. Auto-load and auto-update the DSO's software, 2007.

作者简介:陈洪钢(1981-),男,四川南充人,电子科技大学自动化工程学院硕士研究生,研究方向:测控技术与仪器;黄华伟(1978-),男,四川成都人,电子科技大学自动化工程学院硕士研究生,研究方向:通用测试技术与仪器设计。