

基于串口通信的 DSP 应用程序在线升级方法

汪晶晶, 苏建徽, 孙佩石

(教育部光伏系统工程研究中心 合肥工业大学能源研究所, 安徽 合肥 230009)

摘要:为解决特殊场合 DSP 程序升级困难的问题,以 TMS320F28035 为例,介绍了一种基于串口通信的适合于 TMS320C2000 系列 DSP 实现程序更新的在线升级方法。描述了该在线升级方法的基本思想和实现步骤,给出了关键部分的程序代码。实验证明,该方法简单可靠,可用于嵌入式设备软件程序的升级更新中。

关键词: 在线升级; DSP; 串口通信; Flash

中图分类号: TN409

文献标识码: A

文章编号: 1674-7720(2013)14-0015-03

Method of DSP application upgrading online based on serial communication

Wang Jingjing, Su Jianhui, Sun Peishi

(Research Center for Photovoltaic System Engineering (Ministry of Education), Institute of Energy Research in Hefei University of Technology, Hefei 230009, China)

Abstract: This paper introduces a suitable method based on serial communication for TMS320C2000 series DSP upgrading programs on line, taking TMS320F28035 as an example, to resolve the difficulty of upgrading DSP programs in special occasions. The basic idea and the implementation process steps of the method are described. At the same time, a part of key codes are given. Experiment results show that this method is simple and reliable, and can be used for embedded equipment software upgrading programs.

Key words: upgrading online; DSP; serial communication; Flash

TMS320C2000 系列 DSP 是美国德州仪器公司 (简称 TI) 推出的集微控制器和高性能 DSP 特点于一身的 DSP 系列。该系列的 DSP 具有强大的控制信号处理能力^[1],能够实现复杂的控制算法。随着电子技术的不断发展以及用户需求的不断提升,可能需要经常对已经投入使用的嵌入式设备程序进行更新,而目前一般的程序升级方法是实地取下设备,露出 JTAG 端口后通过仿真器来更新程序^[2-4]。这种方法虽然简单有效,但对于某些特殊场合,会给程序升级带来了极大的不便^[2]。本文以 TMS320F28035 为例,描述了一种可以脱离 JTAG 仿真器,不改变 DSP 上电启动方式,实现 TMS320C2000 系列 DSP 应用程序在线更新的方法。

1 在线升级的基本思想

一般的基于 DSP 的软件程序更新是在 CCS 环境下通过 JTAG 接口操作来实现的。基于 JTAG 接口的方法虽然易于操作,而且调试方便,但经常受空间以及传输距离的限制。例如一台 DSP 系统安装在复杂、封闭的环境下,当程序需要更新或升级时,利用 JTAG 接口难以实

现程序的在线升级^[3]。而基于串口通信的在线升级技术是通过用底层程序烧写应用程序的方法来达到程序升级的目的,该方法则不受复杂系统和复杂环境的限制。另外,在线升级方法不需要改变 DSP 的启动方式,直接采用 DSP 默认的内部 Flash 方式启动^[5],从而省去了要对 DSP 的一些引脚进行硬件设置的麻烦。底层程序指已经固化在 DSP 指定 Flash 空间中的程序,不允许用户修改和擦除,主要用于实现在线升级的时机判断、数据接收及代码烧写等功能,该程序中使用了 Flash2803x_API 库函数(详见 2.2 节);应用程序即为用户的升级程序^[3]。

F28035 DSP 每次上电复位,先运行底层程序,与 PC 机建立联系,然后根据 PC 机的指令来判断是否需要升级应用程序。若需要,则将通过串口发送来的应用程序代码烧写至 F28035 片内 Flash 指定扇区;否则将继续执行原有的应用程序。当应用程序很大或 DSP 的 RAM 空间比较小时,可采用将应用程序代码分批发给 DSP, DSP 接收并烧写完一批代码后,再进行下一批代码的接收和烧写工作,直到所有的应用程序代码都烧写完毕。

《微型机与应用》2013 年 第 32 卷 第 14 期

2 在线升级的具体实现

2.1 应用程序

用户的应用程序经过 CCS 编译连接生成具有模块化格式的目标文件 (.out), 该文件中的代码和数据分别存放在不同的段中, 因而不能直接用来烧写 Flash, 需将其转换为 Flash 能识别的数据格式——二进制文件 (.bin)。本文采用 hex2000.exe 和 FileOshell.exe 工具来实现文件转换。首先, 应用程序经过编译连接生成 .out 文件, 然后通过 hex2000.exe 把 .out 文件转换成 .hex 文件, 再通过 FileOshell.exe 将文件转换成 .bin 文件。先做一个批处理文件, 内容如下:

```
Example_2803xAdcSoc.out
-map Example_2803xAdcSoc.map
-o Example_2803xAdcSoc.hex
-m
-memwidth 16
-image
ROMS
{
Flash28035: origin = 0x3e8000, len= 0x1000, romwidth=
16, fill=0xFFFF
}
```

其中, Example_2803xAdcSoc.out 是应用程序经过 CCS 生成的文件; -map 是生成 map 文件; -o 是生成 hex 文件; -m 是 Motorola-S 格式; -memwidth 16 指存储器位数为 16 bit; -image 指选择映像文件; ROMS 是所需要转换的起始地址、长度、位数及填充。本文选择从 0x3e8000 开始, 长度是 4KB, 即 FlashH, FlashH 中未用的部分用 0xFFFF 填充, 本文把这个批处理文件命名为: Example_2803xAdcSoc.cmd。接下来要生成 .bin 文件, 先做一个 MS-DOS 型批处理文件, 其内容如下:

```
IFileIOshell.exe -i Example_2803xAdcSoc.hex -o Example_2803xAdcSoc.bin
```

注意要把 Example_2803xAdcSoc.out、hex2000.exe、FileIOShell.exe、Example_2803xAdcSoc.cmd 和 MS-DOS 型批处理文件放在同一目录下, 然后双击 MS-DOS 型批处理文件, 即生成所需要的 Example_2803xAdcSoc.bin 文件。

2.2 底层程序

底层程序用于实现将串口发送的数据烧写至 Flash 的指定部分, 涉及到应用程序的正确定位和复位后的启动过程, 是实现整个在线升级的重点。底层程序流程图如图 1 所示。底层程序主要实现以下功能^[3]:

(1) 上电复位查询功能。上电复位后通过接收上位机发送的命令判断是否升级。若上位机发送的是升级命令, 则跳转到底层程序中升级部分执行; 否则, 跳转到原有的应用程序处执行。

(2) 搬移烧写程序的功能。由于 F28035 片上 Flash 不支持在其中一个扇区运行程序去擦除或烧写其他扇区, 故完成接收数据和烧写 Flash 工作的这部分程序(即底层

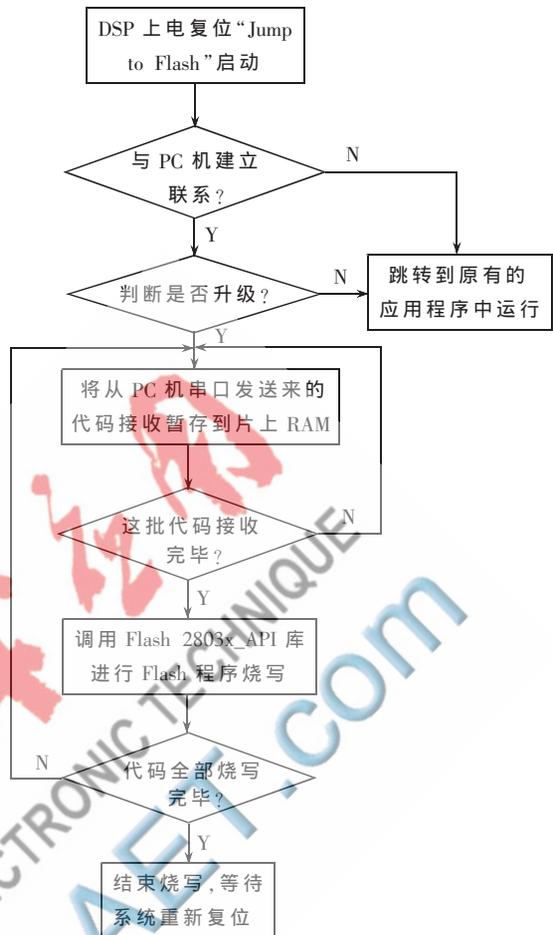


图 1 底层程序流程图

程序中的升级部分程序) 需搬移至片内 RAM 或片外 RAM 上运行。实现程序搬移的函数为:

```
void MemCopy (Uint16 *SourceAddr, Uint16 *Source End
Addr, Uint16 *DestAddr)
{
while (SourceAddr < SourceEndAddr)
{
*DestAddr++ = *SourceAddr++;
}
return ;
}
```

其中, SourceAddr 为 Flash 中升级程序的起始地址, SourceEndAddr 为 Flash 中升级程序的结束地址; DestAddr 为搬移至内存的首地址。

(3) 接收上位机发送的应用程序代码并保存到 DSP 指定的内存中(一般为 RAM 区)。这是通过串口 RS232 来实现的。并确定用于数据保存的这部分内存未被占用。例如, 若需要将应用程序代码暂存到 F28035 的 L0 SARAM 区域(地址空间 0x3F8000-0x3F8800)。定义数组 Uint16 BlockBuffer[2048]用于存储应用程序代码, 在底层程序中采用存储器定位语句, 将上面的缓冲数组定位到相应的存储空间:

欢迎网上投稿 www.pcachina.com 17

```
#pragma DATA_SECTION(BlockBuffer, "BlockTransferbuffer");
```

在底层程序 CMD 文件中,采用定位语句,将 BlockTransferbuffer 定位到 DSP 的 L0 SRAM 空间:

```
BlockTransferBuffer; > L0 SRAM PAGE=2
//地址空间:0x3F8000~0x3F8800
```

通过以上底层程序的设置,可将应用程序缓存到指定的 RAM 区域中。

(4) 代码接收结束后,将内存中的代码烧写至指定 Flash 扇区,该步骤通过调用 Flash2803x_API 库函数完成。底层程序中所用到的 Flash2803x_API 库函数如下^[6]:

① 擦除扇区的函数为 Uint16 Flash28035_Erase (Uint16 SectorMask, &Fstatus), 其中, SectorMask 为即将被擦除的扇区; &Fstatus 为执行擦除操作后返回的状态值,用来判断擦除操作是否成功。② 将程序烧写到 Flash 扇区的函数为 Uint16 Flash28035_Program (&FlashAddr, &BuffAddr, Length, &Fstatus), 其中, &FlashAddr 为即将被烧写的 Flash 扇区的起始地址; &BuffAddr 为即将准备烧写的程序当前存放在内存空间的首地址; Length 为程序长度; &Fstatus 为执行烧写操作后返回的状态值,用来判断烧写操作是否成功。③ 校验烧写到 Flash 中的程序为 Uint16 Flash28035_Verify (&FlashAddr, &BuffAddr, Length, &Fstatus), 其中, &FlashAddr 指定从 Flash 内开始比较的首地址; &BuffAddr 为被比较文件的存储首地址; Length 是需要比较的 16 bit 字的个数, 程序长度; &Fstatus 是执行校验操作后返回的状态值,用来判断校验操作是否成功。

2.3 底层程序和应用程序的定位

DSP F28035 上电复位后, CPU 将从内部 Boot Rom 获得复位向量。复位向量指向 Boot Rom 并执行其内部的 Bootloader 程序, 执行完毕后确定从内部 Flash 启动。程序指针跳转到 Flash 的 0x3F7FF6 处。由于这个地址是固定的, 因此底层程序必须烧写在以这个地址为起始地址的空间内。DSP 进入底层软件程序中运行, 首先通过接收上位机的命令来判断是否进行在线升级, 如果进行在线升级, 则跳转到相应升级程序中执行; 否则, 跳转到原有的应用程序处执行。由底层程序跳转到原有的应用程序处执行时, 采用绝对地址跳转。部分程序如下所示:

```
#define Jumpgxcx (void (*)(void))0x3E8FFE
//定义应用程序的跳转地址
SCI_SendStatus("upgrade program? (y/n):")
//向上位机询问是否升级
temp = SCIA_GetByteData_app();
//接收上位机发送来的是否升级命令
if (temp == 'y')
{
    main2(); //如果升级,则跳转到升级程序中执行
}
Else
{
```

```
(*Jumpgxcx)();
```

```
//如果不升级,则采用绝对地址跳转到应用程序中执行
}
```

底层程序的 cmd 配置与应用程序的 cmd 配置要保持一致, 不能产生地址冲突。同时, 要注意底层程序和应用程序的跳转地址配置。

底层程序 cmd 文件的部分配置如下:

```
BEGIN : origin = 0x3F7FF6, length = 0x000002
RESET : origin = 0x3FFFC0, length = 0x000002 /*
codestart : > BEGIN PAGE = 0
```

应用程序 cmd 文件的部分配置如下:

```
BEGIN : origin = 0x3E8FFE, length = 0x000002
codestart : > BEGIN PAGE = 0
```

3 烧写步骤

首先把底层程序通过 JTAG 接口烧写到 F28035 中, 然后再进行应用程序的烧写。应用程序的烧写步骤为: 先把串口调试工具的参数配置为波特率 9 600 bit/s, 8 bit 数据位, 1 bit 停止位, 没有奇偶校验位; 选择发送文本文件方式, 发送应用程序的 .bin 文件到 DSP。由于 F28035 的 RAM 区比较小, 可以采取把应用程序代码分为多次发送的方式。烧写过程如图 2 所示。

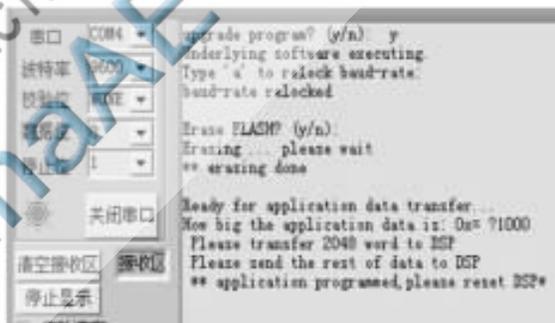


图 2 烧写过程

本文介绍了一种基于串口通信的 DSP 应用程序在线升级技术, 可以在不打开机箱的条件下实现模块软件的更新升级。经过实验发现, 采用在线升级技术来更新程序所耗费的时间比采用 JTAG 口烧写程序所耗费的时间要长一些, 但解决了复杂情况下程序升级困难的问题。总之, 该方法简单可靠, 可应用于嵌入式设备的软件程序更新升级中。

参考文献

- [1] 苏奎峰, 吕强, 耿庆锋, 等. TMS320F2812 原理与开发[M]. 北京: 电子工业出版社, 2005.
- [2] 李声飞, 代华山. 基于串口通信的 DSP 程序动态加载技术[J]. 电讯技术, 2011, 51(6): 121-124.
- [3] 李静, 张树团. TMS320F2812 片内 Flash 在线烧写技术研究[J]. 国外电子元器件, 2008(10): 37-40.
- [4] 孙轶, 许少尉. TMS320F2812 芯片的 Flash 烧写技术[J].

航空计算技术, 2006, 36(5): 46-49.

- [5] TI 公司. TMS320F281x boot ROM serial Flash programming[Z]. 2007.
- [6] TI 公司. TMS320F2803x piccolo Flash API[Z]. 2010.

(收稿日期: 2013-03-19)

作者简介:

汪晶晶, 女, 1988 年生, 硕士研究生, 主要研究方向: 电力电子与电力传动。

苏建徽, 男, 1963 年生, 博士, 教授, 博士研究生导师, 主要研究方向: 光伏发电技术、分布式发电技术等。

孙佩石, 男, 1954 年生, 教授, 硕士生导师, 主要研究方向: 自动控制、电力电子与电力传动、光伏系统的研制。

