

文章编号: 1001-893X(2011)06-0121-04

基于串口通信的 DSP 程序动态加载技术*

李声飞, 代华山

(中国西南电子技术研究所, 成都 610036)

摘要:为解决特殊场合嵌入式设备程序升级困难,提出了一种基于串口的 TMS320F2812 程序动态加载技术。该方法充分利用 F2812 丰富的片内存储器资源和支持在线编程特点,采用二级引导方式,通过调用 API 编程函数对片内 FLASH 擦除和烧写,实现设备程序的在线加载。实验证明,该方法具有烧写稳定、效率高、操作方便等优点,能够应用于嵌入式设备程序的远程升级中。

关键词:嵌入式设备;数字信号处理;串口通信;动态加载;二级引导

中图分类号: TN409 **文献标识码:** A doi: 10.3969/j.issn.1001-893x.2011.06.027

Dynamic Loading Technology for DSP Program Based on Serial Communication

LI Sheng-fei, DAI Hua-shan

(Southwest China Institute of Electronic Technology, Chengdu 610036, China)

Abstract: A new dynamic loading technology is proposed for TMS320F2812 programs based on serial communication, to resolve the difficulty of upgrading embedded equipment software in special occasions. This method takes full advantages of F2812's features, such as rich on-chip memory and supporting on-line programming, adopts secondary booting algorithm, and calls APIs to erase and program on-chip FLASH to realize on-line loading of embedded devices. Experiment results show that this method is featured by stable programming, high efficiency and easy operation, and can be applied to remote upgrade of embedded equipment software.

Key words: embedded equipment; digital signal processing; serial communication; dynamic loading; secondary boot

1 引言

TMS320F2812 是 TI 公司推出的一款高性能的定点 DSP^[1],具有 32 位内核处理器,主频可达 150 MHz,片内集成 128 kbyte 的 FLASH、16 kbyte 的 SARAM,以及丰富多样的外部接口,广泛应用于工业控制、移动通信、军事安全等领域。随着电子技术的发展和用户需求的提升,对已投入使用嵌入式设备程序的升级也越来越频繁,而目前传统的程序升级方法是实地取下设备,通过 JTAG 仿真器来烧写程序^[2]。这种方法简单有效,但对于某些特殊场合,如设备在密闭的箱体内部以及其它不便触及的地方,

这种方式给升级带来了极大的不便,因此开发高效、便捷的升级技术成为一种迫切的需求。笔者考虑到多数设备都有远程通信端口,如串口、SPI 等,因此可采用在线升级方式,在不打开设备箱体的情况下,实现软件更新。

2 系统结构介绍

如图 1 所示,系统硬件结构采用 DSP + FPGA 架构。DSP(TMS320F2812)作为处理中心,通过数据总线和地址总线完成与 FPGA 的数据交换,实现复杂数据处理算法和外部控制。FPGA 利用其大规模硬件资源,实现接口扩展。上电时,FPGA 对 DSP 的模式选择

* 收稿日期:2011-03-01;修回日期:2011-05-17

端进行控制,使 DSP 选择从片内 FLASH 启动或 SCI 启动,并且通过 FPGA,将 PC 机的串口与 DSP 的串口相连,实现 PC 机和 DSP 的数据透明交换。

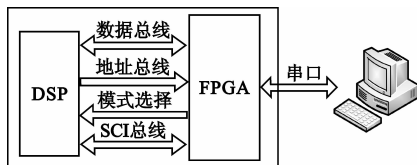


图 1 嵌入式设备系统框图

Fig.1 Block diagram of embedded equipment system

传统的编程方式采用 JTAG 口加载程序,利用 TI 公司编译软件 CCS 中的 On-chip Flash Programmer 插件,将编译生成的 .out 文件直接烧写到片内 FLASH 区^[3]。该方法简单高效,能够实现对模块软件的升级。但对于特殊场合,如模块在密闭箱体内部,该升级方式无法快速有效的实现软件升级,所以本文采用基于串口的动态加载的方式,实现设备软件更新。

3 动态加载的工作机理

TMS320F2812 的启动方式可以分为内部 Flash 引导启动、SPI 引导启动、SCI - A 引导启动、H0 - SARAM 引导启动、OTP 引导启动、GPIO 端口 B 引导启动等几种模式,由其模式选择端实现不同的启动方式。其中,TMS320F2812 脱机运行默认为内部 Flash 启动,本文通过串口实现软件升级,将模式选择端置为 SCI 启动模式。

本文描述的程序加载采用两级引导方式^[4],分为引导程序加载和应用程序 FLASH 编程。程序动态加载原理如下:首先 DSP 上电并扫描模式选择口,选择 SCI - A 启动模式,然后等待接收来自 SCI 口数据;PC 机通过串口将引导程序发送给 DSP 的 SCI 通信口,将引导程序缓存至 RAM 区;然后再发送应用程序的烧写文件,DSP 在引导程序的控制下,将应用程序烧写到 DSP 的 FLASH 区。由于 TMS320F2812 有 128 kbyte 的 FLASH 空间,能够满足一般的应用需求。

3.1 引导程序加载

DSP 的 RAM 空间分为安全区和非安全区,为了代码的安全性,一般将代码放在安全区域内运行。所以要把加载引导程序加载至 RAM 的安全区内。

如图 2 所示,DSP 在 SCI-BOOT 模式下,接收来自串口的数据,先将引导程序存储到非安全 RAM 区(安全 RAM 处于锁定状态),该区域在 DSP 存储器的低地址位。然后加载引导程序接管 DSP 的控制权,对 DSP

进行解锁操作,并将自身拷贝到 RAM 的安全区。

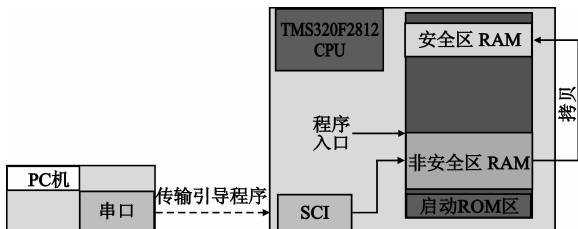


图 2 引导程序加载

Fig.2 Download boot program

3.2 应用程序烧写

在引导程序的控制下,将应用程序烧写到 DSP 的 FLASH 区,如图 3 所示。

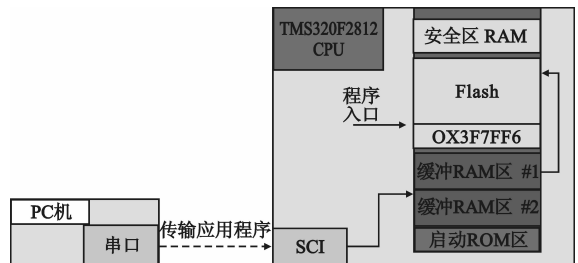


图 3 应用程序烧写

Fig.3 Programming APP code

首先 PC 机通过串口将编译好的应用程序发到 DSP,DSP 接收并缓存程序到 RAM 区。鉴于应用程序占用空间较大,一般将其存储在较大的 RAM 空间中。笔者根据需求,将其定位到 L0L1 区域(地址空间 0x008000 ~ 0x00A000)。预定义两个数组 Uint16BlockBuffer1[4096]、Uint16BlockBuffer2[4096]用于存储应用程序,在引导程序中采用存储器定位语句,将以上两个缓冲数组定位到相应存储空间:

```
# pragma DATA_SECTION (BlockBuffer1, "BlockTransfer-Buffer1");
# pragma DATA_SECTION (BlockBuffer2, "BlockTransfer-Buffer2");
```

在 CMD 文件中,采用定位语句,将 Buffer1/2 定位到 DSP 较大的 RAM 空间:

```
BlockTransferBuffer1: > RAMH0_1 PAGE = 2//地址空间:0x3F8000 ~ 0x3F9000
BlockTransferBuffer2: > RAMH0_2 PAGE = 2//地址空间:0x3F9000 ~ 0x3FA000
```

通过以上引导程序的设置,可将应用程序缓存至 RAM 区中。

当缓冲区 Buffer1 存满后,引导程序调用 FLASH 编程 API 函数,将 Buffer1 中应用程序烧写到 FLASH 中。在烧写前要对 FLASH 进行擦除,采用 TI 公司

提供的 API 函数对 FLASH 区域擦除:

Flash_Erase(SECTOR_F2812, &EraseStatus)

其中,第一个参数 SECTOR_F2812 表示要擦除的扇区,第二个参数返回状态值;通过 EraseStatus 来判断擦写是否成功,若成功,则调用 FLASH 编程 API 函数如下:

Flash_Program (Flash_ptr, BlockBuffer1, Length, &ProgStatus)

其中,参数 Flash_ptr 表示编程首地址,BlockBuffer1 为应用程序暂存 RAM 的首地址,Length 为烧写字节数,ProgStatus 表示返回状态。通过调用该 API 函数,将暂存在 RAM 区的应用程序烧写至指定的 FLASH 区域,并可以通过指定烧写首地址方式来自行分配 FLASH 空间。

在 BlockBuffer1 程序 FLASH 烧写同时,引导程序继续接收剩下的应用程序代码至 BlockBuffer2 空间,然后重复以上步骤,将 BlockBuffer2 中代码烧写至 FLASH 指定位置,如此反复,待 FLASH 全部烧写完毕,即完成了程序的升级。

4 动态加载的实现

4.1 硬件环境搭建

首先要搭建设备升级的硬件环境,将设备串口与 PC 机串口相连,保证数据传输的硬件通路正常;然后通过 FPGA 控制 DSP 的启动模式^[5],配置 DSP 上电 SCI-A 启动。如表 1 所示,FPGA 控制 4 条模式选择线 GPIOF4、GPIOF12、GPIOF3、GPIOF2 输出为 0011。

表 1 DSP 启动模式控制
Table 1 DSP boot mode control

模式选择	GPIO 初始状态	Jump to Flash/Rom address 0x3F7FF6	Call SCI_Boot to load SCI-A
GPIOF4 (SCITXDA)	上拉	1	0
GPIOF12 (MDXA)	未上拉	×	0
GPIOF3 (SPISTEA)	未上拉	×	1
GPIOF2 (SPICLK)	未上拉	×	1

4.2 DSP 资源分配

TMS320F2812 片内^[6]有两块 1 k × 16 bit 的 SARAM MO/M1,两块 4 k × 16 bit 的 SARAM L0/L1,1 块 8 k × 16 bit 的 SARAM H0、128 k × 16 bit 的 FLASH。由于 SARAM 具有读写速度快的优点,所以优先考虑程序在 SARAM 中运行,同时大容量的 SARAM 可以

作为数据暂存。FLASH 具有掉电数据不丢失的特性,用来保存代码和重要数据信息。基于以上考虑,引导程序代码量较小,笔者选择将其代码段存放到 MO/M1 空间中,运行程序放在较大的 L0/L1 段,CMD 映射关系如下所示:

```
SECTIONS
{
    .text_unsecured: > RAMMOM1PAGE = 0
    .text                :LOAD = RAMH0_1,
    RUN = RAML0L1,
    LOAD_START(_textLoadStart),
    LOAD_END(_textLoadEnd),
    RUN_START(_textRunStart),
    PAGE = 1}

```

应用程序的 CMD 文件控制其取址和运行的空间,如下所示:

```
SECTIONS
{
    .econst: > FLASH_E PAGE = 0
    /* Allocate program areas: */
    .text: > FLASHD PAGE = 0
    .cinit      : > FLASHD PAGE = 0
    codestart   : > BEGIN PAGE = 0
    ramfuncs    : LOAD = FLASHD,
    RUN = RAML0,
    LOAD_START(_RamfuncsLoadStart),
    LOAD_END(_RamfuncsLoadEnd),
    RUN_START(_RamfuncsRunStart),
    PAGE = 0}

```

econst 段为存放外部数据和常量的空间,分配一个较大的 8 k FLASH 的区域来保存外部数据;text 段为代码段,将程序映射到 FLASHD(地址空间: 0x3EC000 ~ 0x3F0000)中;cinit 用于保存已初始化全局数据,存放到 FLASHD 区域。Ramfuncs 控制程序的执行方式,上电后程序从 FLASHD 中拷贝代码到 RAML0(地址空间: 0x008000 ~ 0x009000)中运行,实现了程序从 FLASH 区到 RAM 区的映射。

4.3 转换待传输的文件

由于 CCS 编译生成的目标文件(.out)是模块化格式,即程序中代码和数据分别存放在不同段中,该文件不能直接用来烧写 FLASH,需将其转换为 FLASH 能识别的数据格式——二进制文件(.bin)。笔者采用 HEX2000 和 Fileoshell.exe 工具来实现文件转换。编写 AppCode_hex_2812.cmd 文件,将 CCS 编译后生成文件 AppCode.out 转换成 HEX 文件 AppCode.hex,并填充未用的 FLASH 区域为全 1。

```
AppCode.out
- map AppCode_hex.map

```

```

- o AppCode.hex
- m
- memwidth 16
- image
ROMS
{
FLASH2812; origin = 0x3e8000, len = 0x10000, romwidth
= 16, fill = 0xFFFF
}

```

利用 AppCode_ COFF2BIN_ 2812. bat 批处理文件,控制 FillOShell 工具生成 bin 文件,可以直接传输该文件来实现在线升级。

4.4 烧写步骤

首先配置串口调试工具,配置参数为:波特率 9 600 bit/s,8 bit数据位,1 bit截止位。然后,选择发送文本文件方式,发送引导程序的 bin 文件到 DSP,如图 4 所示。

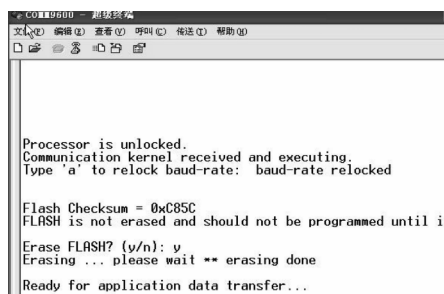


图 4 传输引导程序
Fig.4 Transmission boot program

传完后再次锁定波特率,并选择擦除 FLASH,将未编程的 FLASH 区域全部擦除,再传输应用程序的 bin 文件,如图 5 所示。FLASH 烧写完成,并校验应用程序大小,若校验通过则 FLASH 烧写成功。

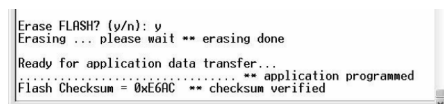


图 5 烧写成功
Fig.5 Programming success

5 结束语

本文提出了一种 DSP 软件动态加载方法,可以在不打开机箱条件下实现模块软件升级。通过实际测试,应用程序代码在 10 kbyte 左右时,升级耗时 2 min 左右,与传统的 JTAG 升级方式相比,虽然时间有所增加,但解决了高空、密闭等特殊情况下程序升级的困难,可满足基于 TMS320F2812 DSP 开发的嵌入式系统升级要求。文中采用二级引导方式,通过

串口传输数据并调用 API 编程函数实现程序动态加载。由于片内 FLASH 容量有限(128 kbyte),扩充片外 FLASH 资源以满足大型应用需求,将成为下一阶段工作的重点,若配合以太网或总线技术,还可实现软件的远程升级。总之,该方法具有操作简单、烧写稳定等优点,可广泛应用于嵌入式设备软件的更新升级中,是智能化设备的必然要求和趋势。

参考文献:

- [1] 王虹,杨甦.基于 TMS320F2812 的步进电机 SVPWM 细分驱动[J]. 电讯技术,2008,48(8):189-193.
WANG Hong, YANG Su. Subdivision Driving with SVPWM of Stepping Motor Based on TMS320F2812[J]. Telecommunication Engineering,2008,48(8):189-193. (in Chinese)
- [2] 李静,张树团. TMS320F2812 片内 Flash 在线烧写技术研究[J]. 计算机应用,2008(10):37-40.
LI Jing, ZHANG Shu-tuan. Research on on-line programming techniques for Flash in TMS320F2812[J]. Journal of Computer Applications,2008(10):37-40. (in Chinese)
- [3] 陈代媛. C6000 外部 FLASH 在线编程引导技术[J]. 电讯技术,2009,49(5):86-88.
CHEN Dai-yuan. External FLASH Memory's Bootloader System for C6000[J]. Telecommunication Engineering,2009,49(5):86-88. (in Chinese)
- [4] 朱望纯,胡汉武,李智. DSP + FLASH 的二次程序加载实现方法[J]. 理论与方法,2008,27(4):14-16.
ZHU Wang-chun, HU Han-wu, LI Zhi. Secondary boot loader method for DSP and FLASH[J]. Journal of Theory and Methods,2008,27(4):14-16. (in Chinese)
- [5] 黄玉梅. 一种基于 DSP + FPGA 的控制系统方案设计[J]. 电讯技术,2004,44(3):136-138.
HUANG Yu-mei. A DSP and FPGA Based Scheme for Control System[J]. Telecommunication Engineering,2004,44(3):136-138. (in Chinese)
- [6] 李晶,钟瑜,郑百衡. 基于 DSP 的 FPGA 动态配置技术[J]. 电讯技术,2005,45(1):156-159.
LI Jing, ZHONG Yu, ZHENG Bai-heng. FPGA Dynamic Reconfiguration Based on DSP[J]. Telecommunication Engineering,2005,45(1):156-159. (in Chinese)

作者简介:

李声飞(1985—),男,河南光山人,2010 年于重庆大学获硕士学位,现为助理工程师,主要研究方向为信号与信息处理;

LI Sheng-fei was born in Guangshan, Henan Province, in 1985. He received the M. S. degree from Chongqing University in 2010. He is now an assistant engineer. His research direction is signal and information processing.

Email: happyrogerli@163.com

代华山(1978—),男,四川三台人,2005 年于电子科技大学获硕士学位,现为工程师,主要研究方向为航空电子。

DAI Hua-shan was born in Santai, Sichuan Province, in 1978. He received the M. S. degree from University of Electronic Science and Technology of China in 2005. He is now an engineer. His research direction is avionics.