

利用 LPC2214 的 IAP 功能实现程序远程更新

赵会宾, 田庆春

(中国电子科技集团公司第 54 研究所, 河北 石家庄 050081)

摘要 介绍了利用芯片 LPC2214 的 IAP 功能实现片内程序的远程下载更新并给出了更新程序的流程图。由于该方法采用的是芯片的内在功能, 因此不需要采用商业化的嵌入式操作系统内核并能够使系统本身达到最小化, 大大降低了实现的硬件成本, 提高了程序下载的可靠性。该技术通过简单修改亦可用于其他具有 IAP 功能的芯片。

关键词 IAP; ARM; LPC2214; 程序下载; ADS1.2

中图分类号 TP311.52 **文献标识码** A

Realization of Program Remote Updating through IAP Function of LPC2214

ZHAO Hui-bin, TIAN Qing-chun

(The 54th Research Institute of CETC, Shijiazhuang Hebei 050081, China)

Abstract Realization of program remote updating through IAP function of chip LPC2214 is introduced and the flow diagram of updating program is given. Due to utilization of the chip's inherent function, the embedded operating system needn't be adopted, and the system can be minimized. The cost is greatly reduced and the reliability is improved. Simply modified, the technology can also be used in other chips with the same IAP function.

Key words IAP; ARM; LPC2214; program download; ADS1.2

0 引言

电力负荷管理系统已成为电力需求侧管理、电力营销与客户服务工作的重要组成部分。为适应系统发展的需求变化, 保证系统建设具有良好的规范性、兼容性、开放性和扩展性, 2004 年底国家电力公司制定了《电力负荷管理系统数据传输规约 - 2004》。

在新的 2004 版传输规约中, 对电力负荷管理终端提出了新的要求, 其中一项重要的指标就是终端必须可通过程序下载实现终端程序的远程更新。因此在研制的新一代电力负荷管理终端中采用 Philip 公司生产的 32 位微处理器 LPC2214, 并利用该芯片的在应用编程技术即 IAP 功能实现了终端程序的远程更新。

1 程序更新的实现方法

LPC2214 是基于一个支持实时仿真和跟踪的 16/32 位 ARM7TDMI - S 的微控制器, 并带有 256 kbytes 嵌入的高速 Flash 存储器和 16 kbytes 的 RAM 存储器。对 Flash 存储器的编程可通过几种方法来实现: 通过内置的串行 JTAG 接口, 通过在系统编程 (ISP) 和 UART0, 或通过在应用编程 (IAP)。用在应用编程 IAP 可以在应用程序运行时对片内

Flash 进行擦除和编程, 这样就为数据存储和现场固件的升级都带来了极大的灵活性。

一般程序的下载可通过 3 种方式来进行实现:

① 通过采用嵌入式实时操作系统的方法来实现程序下载, 但采用操作系统将极大的增加系统对存储容量的要求增加硬件和软件成本; ② 通过采用增加外控制烧录系统的方法来实现, 但这种方法同样增加了系统的复杂度和成本, 同时降低了系统本身的可靠性; ③ 通过应用编程技术即 IAP 技术来实现程序的远程下载, 该技术充分利用芯片本身的功能而不增加外围器件是一种最佳的实现方案。

2 LPC2214 的 IAP 功能

(1) IAP 技术简介

对于 IAP 在应用编程来说, 应当通过寄存器 r0 中的字指针指向存储器包含的命令代码和参数来调用 IAP 程序。IAP 程序是 Thumb 代码, 位于地址 0x7fffff1。

IAP 功能可用下面的 C 代码来调用。

- 定义 IAP 程序的入口地址。
- # define IAP_LOCATION 0x7fffff1
- 定义函数类型指针。

收稿日期: 2006-12-26

```
typedef void (* IAP) (uint32[], uint32[]);
```

```
IAP iap_entry;
```

- 设置函数指针。

```
iap_entry = (IAP) IAP_LOCATION;
```

- 使用下面的语句来调用 IAP。

```
iap_entry(command, result);
```

(2) LPC2214 中的 Flash 扇区

LPC2214 有 256 kBytes 嵌入的高速 Flash 存储器,分为 18 个扇区。扇区 0~7 的大小为 8 kbytes,扇区 8,9 的大小为 64 kbytes,扇区 10~17 的大小为 8 kbytes,具体的扇区分配及地址如表 1 所示。

表 1 LPC2214 中的 Flash 扇区

扇区号	存储器地址	扇区规格
0,1,2,3 4,5,6,7	0x0000 0000—FFFF	8 k
8	0x0001 0000—FFFF	64 k
9	0x0002 0000—FFFF	64 k
10,11,12 13,14,15	0x0003 0000—BFFF	8 k
16	0x0003 C000—DFFF	8 k
17	0x0003 E000—FFFF	8 k

Boot Block 一般位于片内 Flash 存储器顶端。在 LPC2214 中 Boot Block 占有的 17 号 Flash 存储器扇区不能用来存放代码。

(3) IAP 命令简介

IAP 功能的使用主要通过调用 iap_entry() 函数来实现。IAP 命令主要包括以下几项内容:

- 准备编程扇区命令,命令代码为 50。该命令使 Flash 写/擦除操作分两步执行,即先准备写操作的扇区,然后进行写/擦除操作。该命令必须在执行“将 RAM 内容复制到 Flash”或“擦除扇区”命令之前执行;

- 将 RAM 内容复制到 Flash 命令,命令代码为 51。其参数分别为要写入数据的目标地址、读出数据字节地址、写入字节的数目和系统时钟频率。该命令用于编程片内 Flash 存储器;

- 擦除扇区,命令代码为 52。其参数分别为起始扇区号、结束扇区号和系统是种频率。该命令用于擦除片内 Flash 存储器;

- IAP 扇区查空命令,命令代码为 53。其参数为起始扇区和结束扇区,该命令用于对片内 Flash 存储器的一个或多个扇区进行查空操作;

- 读器件 ID 命令,命令代码为 54。该命令用于读取器件的 ID 号;

- 读 Boot 代码命令版本,命令代码为 55。该命令用于读取 boot 代码版本号;

- 比较命令,命令代码为 56。该命令用来比较两个地址单元存储器的内容,主要用来验证烧写程

序数据的正确性。

3 程序远程更新的实现

3.1 编译链接

软件开发采用 ADS1.2, ARM ADS 全称为 ARM Developer Suite。是 ARM 公司推出的新一代 ARM 集成开发工具。终端软件主体采用了简单可靠的前、后台工作方式。利用 IAP 功能实现程序的远程更新,将终端的程序分为主程序和更新程序 2 部分。

主程序用来实现终端的功能,是需要进行更新的程序部分。更新程序仅负责主程序的更新工作。编译链接时主程序占据 0~15 扇区,更新程序占据 16 扇区。更新程序用到的数据单独定义到片内 RAM 中 0x40003800~0x40003FFF 区域。

在进行编译和链接前首先应对 ARM Linker 进行设置,这些设置对最终生成的目标文件有着直接的影响。Linktype 中提供了 3 种链接方式。Partial 方式表示链接器只进行部分链接,经过部分链接生成的目标文件,可以作为以后进一步链接时的输入文件。Simple 方式是默认的链接方式,也是最为频繁使用的链接方式,它链接生成简单的 ELF 格式的目标文件,使用的是链接器选项中指定的地址映射方式。Scattered 方式使得链接器要根据 Scatter 格式文件中指定的地址映射,生成复杂的 ELF 格式的映像文件。因此为了实现主程序和更新程序的精确定位,采用 Scattered 方式。

链接用 Scatter 格式文件定义如下:

```
ROM_LOAD1 0x00000000 ←主模块起始地址
```

```
ROM_EXEC1 0x00000000
```

```
{ main.o (+ RO) } ←主程序代码区域
```

```
SRAM1 0x40000000
```

```
{ main.o (+ RW, + ZI) } ←主程序数据区域
```

```
ROM_LOAD2 0x0003C000 ←更新模块起始地址
```

```
ROM_EXEC2 0x0003C000 FIXED
```

```
{ iap.o (+ RO) } ←更新程序数据区域
```

```
SRAM2 0x40003800
```

```
{ iap.o (+ RW, + ZI) } ←更新程序数据区域
```

.....

在 scatter 链接文件中,RO 表示程序代码数据内容,RW 和 ZI 表示该程序部分中定义的通用数据变量和需进行零初始化的数据变量。这里仅列出了与程序远程更新相关的部分链接文件内容。其他部分例如需保存历史数据区和需掉电保存的临时数据区

等也需要单独进行定义,这里就不再列出。

从该链接文件中可以看出,ROM_LOAD1 部分主要包含主程序和其中定义的数据。ROM_EXEC1 中是主程序代码数据,编译链接后的执行起始地址为 0x00000000 即片内 Flash 区。SRAM1 中是主程序中定义的数据变量,编译链接后的起始地址为 0x40000000 即片内 RAM 区域。ROM_LOAD2 部分主要包含更新程序和其中定义的数据。ROM_EXEC2 中是主程序代码数据,编译链接后的执行地址为 0x0003C000。SRAM2 中是更新程序中定义的数据变量,编译链接后的起始地址为 0x40003800 即片内 RAM 区域的顶端区域。

编译链接完成后,在生成的 MAP 文件中找到更新函数 refresh() 的起始地址,在主程序中声明该外部函数即可在主程序中进行调用。例如在 MAP 文件中找到更新函数的地址为 0x0003c2ac。则定义更新函数为:

```
# define refresh() ((void (*)))(0x0003c2ac)()
```

在主程序文件中通过调用更新函数 refresh(), 即可完成主程序的更新工作。这里需要注意的是当编译模式改变时由于代码的优化等级等条件发生了变化代码的大小也会发生相应的变化。因此,要重新检查设定更新函数的起始地址。

3.2 程序远程更新的实现

refresh() 更新函数的主要工作流程如图 1 所示。程序远程更新过程可大体上分为 2 个阶段即程序更新的准备阶段和程序更新阶段。

准备阶段有可以分 3 步来完成。首先主站通过无线信道将编译过的新的终端的主程序(不包含更新模块)分成小的数据单元下传给终端,终端将收到的经校验正确的数据存储在片外 Flash 存储器中;程序下传完成后主站查询终端代码的接收情况,并对传输错误的部分重新下传进行更正;最后主站发送启动更新指

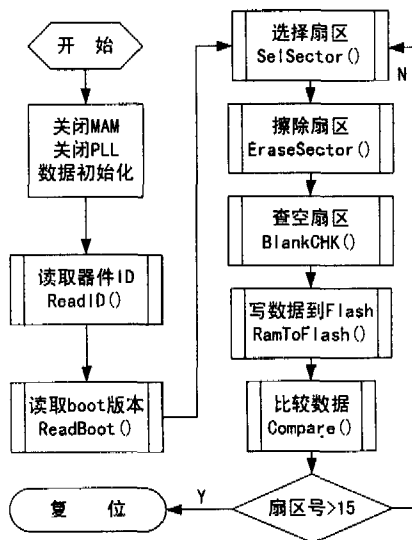


图 1 程序更新函数流程图

令,终端检验程序数据正确性并置上更新标志,停止刷新看门狗,从而使终端复位。

程序更新阶段也分为 2 步来完成。首先终端重启时检测到更新标志有效即可调用更新程序 refresh() 进行程序更新;程序更新完成后再次使终端自复位,即可运行更新后的程序代码,完成程序更新全过程。

由于 IAP 服务代码是 Thumb 指令,所以调用时要进行处理器状态切换或使用 BX 指令调用,若使用 C 程序直接调用,须设置编译参数时,ATPCS 中应选中 Arm/Thumb Interworking 项。

Flash 存储器在写或擦除操作过程中不可被访问。执行 IAP 命令使用片内 RAM 顶端的 32 个 bytes 空间。因此用户程序不应该使用该空间。

调用 IAP 功能前,要关闭 PLL、MAM 部件及所有中断。在调用擦除扇区命令和将 RAM 内容复制到 Flash 命令时需正确设置系统时钟频率。

将 RAM 内容复制到 Flash 命令的参数目的地址的边界应当为 512 bytes,源地址应以字为边界,程序复制时写入字节的数目应当为 512, 1 024, 4 096 或 8 192。本程序中采用长度为 1 024 bytes。

4 结束语

本文简要介绍了利用芯片 LPC2214 的 IAP 功能实现程序远程更新的实现方法,更详细的内容请参考文献和器件的数据手册中查找。以上程序基于 ADS1.2 开发平台并调试通过。该技术已成功应用于开发的新一代电力负荷管理终端,取得了并在实践中取得了良好的运行效果和经济效益。在实际运用时为了保证下传的可靠性,在程序数据传输的过程中采用了 CRC 校验的方法提高数据可靠性。将该技术略加改动亦可应用于其它具有 IAP 功能的处理器芯片。

参考文献

- [1] LABROSSE J J. 嵌入式实时操作系统 uC/OS-II[M]. 邵贝贝译. 北京:北京航空航天大学出版社, 2003.
- [2] 周立功. ARM 嵌入式系统基础教程[M]. 北京:北京航空航天大学出版社, 2005.
- [3] 周立功. ARM 微控制器基础与实战(第二版)[M]. 北京:北京航空航天大学出版社, 2005.
- [4] 姜大光. 一种嵌入式系统集成调试器的实现[J]. 无线电工程, 2005, 35(11): 62-64.

作者简介

赵会宾 男, (1976-), 工程师, 工学硕士。主要研究方向: 信号处理和电路与系统。

田庆春 男, (1975-), 工程师, 主要研究方向: 嵌入式操作系统的研究。