



微控技术论坛翻译义工文章

MSP430 单片机实现远程升级

翻译: 李北山 湖南科大 整理: DC 微控技术论坛版主

原文: Texas Instruments SLAA341.pdf 此翻译只供参考,一切以原文为准. 最后更新日期 2008.6.10.

前言

有什么方法对 MSP430 flash 存储器更新而不用 JTAG 和 BSL

现在这个问题已经得到解决:一段检测监控程序通过 MSP430XXX 单片机通用的内部的 USART 接口可实现对 MSP430F1XX 和 MSP430F4XX 单片机内部的 flash 存储器进行监控和更新.PC 和仿真器是被动式接受指令来对存储器控制访问。最新的这种控制方式可以不用 JTAG 及 BSL 从而实现对 flash 存储器数据更新。

目录

- 1: 简介
- 2: 操作步骤
- 3: 新建监控应用
- 4: 编写程序
- 5: 实例程序
- 6: 参考书目

图表

- 1: 存储器擦写命令例子
- 2: 例 TargetDefs.h
- 3: 与例相关连命令文件

例表

- 1: 控制命令
- 2: Flash 存储器数据更新命令
- 3: 监控软件的源文件
- 4: 应用文档

1 简介

监控程序通过 MSP430 USART 同/异步的接收/发送方式,为存储器的检测和修改提供一个简单可靠的方法.PC 机可用终端仿真软件,如:TeraTerm, 超级中端, 掌上电脑和 MSP430 进行通信。监控程序是基于占用最小系统资源和便于终端应用的操作而设计, 不需要 MSP430 中断处理, 占用 MSP430 FLASH 主要存储器两个段并且程序少于 512 比特位 (64 字节)。

编辑头文件时要注意如下七个选项:

1. 监控程序在何种 MSP430 上运行
2. 可用的中断个数
3. USART 端口选择
4. 系统复位后监控程序 and 用户程序的优先启动顺序
5. 是否需要 Flash 存储器的擦除口令

6. 监控程序是否可清除 MSP430 存储器信息
7. 通过 USART 接收的命令和数据是否需要回送确认

监控程序的操作说明:

监控程序运行于 32.768KHZ 的低频晶振的低频模式下, ACLK 做 USART 的波特率时钟源。

监控程序及示例程序的建立是基于 MSP430 IAR 工作平台 (KkckStart edition)。

2 操作步骤

2.1 配置终端

终端仿真必配制为 9600 波特率, 8 个数据位, 1 个停止位, 无奇偶校验位, 无流控。对终端的编程必须在发送完一个数据或命令后插入 1ms 的延时。通常监控程序会对每个接收到的来自于终端的字符回送一个信息, 这可以在建立选项时添加或删除。

2.2 运行监控

监控程序可以在器件重启或被应用程序唤醒后运行, 在建立程序时是可选项。

监控程序运行时, 关闭看门狗(WDT), 配制 FLASH 存储器的时钟, USART 的 9600 波特率, 8 个数据位, 1 个停止位, 无奇偶校验位, 下一步的操作是监控程序还是应用程序先运行。

如监控程序在系统复位后运行, 定时器 Timer_A 大约在系统启动后的 2 秒钟产生第一次溢出, 然后监控程序迅速进入循环程序检查 USART 接收标志和定时器 A 的溢出标志。

在定时器 A 产生溢出前接 USART 收到任何字符, 监控程序的主程序运行, 等待终端传送字符, 如果在 USART 接收到任何字符前定时器产生溢出, 监控程序停止定时器 A 并且向应用程序传送控制命令。检查应用程序复位向量是否被修改。如果应用程序复位向量被修改, 传送一个说明此复位向量被编址的指令信息。如果复位向量为空, 返回到监控主程序继续执行。

如应用程序配制在系统复位后运行, 监控程序只有在应用程序向监控程序发送指令时才会启动。在这种情况下, 定时器没有被配制或被启动运行, 监控程序进入主循环程序运行。

2.3 使用指令

监控程序处理 5 条指令, 如表 1。每个指令是一个特定字符。传送到任何非控制指令到终端, 并对终端编程的指令都将被屏蔽。所有的指令和 16 进制都以一个特定的值进行传送。一个十六进制的值在传送时, 任何其他正确的十六进制值 (0-9 或 A-F) 被当做是定界符。

表 1 监控指令

指令	参数	功能
C		计算存储器已被使用地址数据的校验和
D	起始地址并计数	显示存储器特定地址的空容
E	口令-起始地址-停止地址	擦除 Flash 存储器
G		运行加载的应用程序
I	段	擦除存储器的信息段 A-B
U	文本文件	用 MSP430-TXT 格式文件对 flash 区进行编程

2.3.1 计算校验和

C 指令让监控程序对主存储器有效的应用程序内容的数值产生一个模上 65536 的数, 校验和的数被分成四个十六进制数, 传送到终端来判断程序的正确。

2.3.2 显示 MSP430 存储器

D 指令返回终端存储器的内容, 语法格式: D<起始地址><字数>

起始地址和字节数都是十六进制值, 首先返回的是一个十六进制地址, 后面紧跟 8 个连续的四位十六进制值, 如图 1

D 指令后面紧跟的是起始地址。如果 16 位有效的地址没被终端接收到，这个错误的地址转向存储器有效地址的起始地址。如果没接收到字数，监控程序将会显示一行 8 个字并自动停止直到接收到另外的控制字符。如果接收到字符 Q（退出）监控程序迅速返回这个指令，如果接收到的是其他的字符，监控程序再一次传送 8 个字的数据到终端。如果存储器地址到有效地址的终端时监控器同样迅速返回这个指令。

图 1 显示在地址 0X1500 处的 8 个字数据

```
? D 1500 8
1500 E0F2 0003 0021 4130 0020 4031 0A00 12B0
```

图 1 存储器指令示例

在 0X0000~0X0200 的地址空间包括特殊功能寄存器和外围设备的地址，不会被显示。

2. 3. 3 擦除存储器的内容

E 指令可以擦除除监控程序所在的两个存储段以外的其他所有存储段，或者只擦除特殊范围内的存储区，擦除指令可通过擦除口令减少被非正常擦除的几率。

2. 3. 3. 1 保护口令禁止

如果口令保护被禁止，只要接收到 E 指令，监控器查找存储器需要擦除的起始地址和停止的地址。如果没有接收到一个有效的起始地址，监控器擦除所有可用的 Flash 存储区。如果接收到有效的起始地址而无有效的停止的地址，监控器会擦除从起始地址到存储区最后地址的这一整个地址。

2. 3. 3. 2 保护口令允许

在监测 E 指令的同时查找口令，口令是存储在应用程序的中断向量表的第一个字里的十六位二进制。如果接下来收到的四个字符不是口令的 ASCII 表示方式，监控器不擦除存储区并迅速返回这个指令。整个 Flash 存储区都被擦除后，擦写保护口令的值一直是 0XFFFF。

如果接收到的口令与计算出的口令相匹配，监控器寻找擦除存储区的起始地址和停止地址。如果这个地址不是主存储器的有效地址范围，监控器迅速返回这个指令。起始地址必须要大于或是等于主存储器的有效应用地址。停止地址必须少于或等于存储器的最后有效地址并且要大于或等于起始地址。

Flash 存储器是整段整段地擦除，接收到有效地址后，任何在起始地址和停止范围的段地址的内容都将被擦除。

2. 3. 4 运行加载的应用程序

G 指令不需要判断，只要通过加载应用的复位地址向量 0XFBFE 来间接对整个应用程序段控制。如果向量是空（0XFFFF），监控程序返回主循环程序并执行。

2. 3. 5 擦除 Flash 存储器的信息段

I 指令擦除一个 256 字节的存储区信息段，可以是段 A 或 B。唯一的区别是段名。擦除段后向终端回送一个信号。

2. 3. 6 更新 Flash 存储器的内容

U 指令通过 USART 接收指令/数据的对存储器进行初始化和内容更新。数据更新功能只接收通过 IAR 编译链接产生的 MSP430-TXT 格式的数据。对于监控器编程的应用及介绍在第三节有详细说明。执行更新的功能时，监控器只处理两种指令，一种在表 2 中介绍，这些指令是 MSP430-txt 文件的一部分，并且自动被 IAR 链接器产生。

表 2 Flash 存储器更新指令

指令	功能	说明
@	接收地址	设置 Flash 存储器的起始地址
q	退出	终止 Flash 存储器编程

2. 3. 6. 2 设置 Flash 存储器的起始地址

识别 U 指令后,监控器通过接收@后的地址进入对 Flash 存储器的编程。

如果接收到的不是@,监控器返回到主循环程序为另一个事件的做准备.如果接收到@字符,监控器扫描 USART 输入的 16 位的十六进制 Flash 存储器地址.如果接收到的是规定的(0-9,A-F)以外的四个字符,对 Flash 编程终止,监控器返回主循环程序.

如果接收到的十六进制的地址有效,检查应用程序可访问 Flash 存储器的地址范围,Flash 存储器的地址范围从顶部的第 2 段到 N 段。对于存储器地址范围访问是由处理器头文件: TargetDefs.h 的取值决定。接收到的地址超出 Flash 存储器的有效地址范围,对 Flash 编程终止。监控器返回主循环程序。

2. 3. 6. 2 接收对 Flash 编程的数据

一旦接收到有效地址,监控器扫描 USRAT 接收到的十六进制编码数据,连续接收到两个(0-9,0-F)有效的十六进制才被认为是有效的数据。除@和 Q 以外的任何非十六进制字符或从终端返回的字符都被忽略,字符的发送可没有定界符。

每接收到一个字节,存储在 RAM 的缓冲区。监控器监测到一个换行字符后将 ARM 缓冲区的内容复制到 Flash 存储器。

在监控器每写一个新的值入 Flash 存储器时会对相关字节单元迅速检查。如果字节单元不为空,监控器产生一个 NB 的错误信息并终止编程。其他的接收到的内容被忽略。

监控器在写完一个字节数据后同样进行检查,如果回读的值与刚写入的值不一致,监控器产生一个 ERR 错误信息并终止编程。其他的接收到的内容被忽略。

如果接收到@字符,监控器等待接收 Flash 地址。如果在@之后收到有效的地址,监控器设置新的目的地址,继续在新目的地址写入数据。如果没有接收到有效的地址,对 Flash 编程终止,返回到主循环程序执行。

2. 3. 6. 3 终止对 Flash 存储器数据更新

如果在对 Flash 存储器编程中接收到 q (0X61) 指令。对 Flash 编程终止,返回到主循环程序执行。

2. 3.6.3 上位机下载程序

下面用支持多种版本 Windows 操作系统的上位机软件经过以下步骤给 Flash 下载程序。

1. 启动上位机程序
2. 从程序菜单中选择,文件-**a** 目录。在弹出的对话框窗口中,从下拉列表选择正确的串口,单击配制按钮。
3. 在弹出的对话框,选串口波特率 9600 b/S, 8 个数据位,无奇偶校验,一个停止位和 Xon/Xoff 流控制。单击 OK 按钮。
4. 确认用适当的连接线将 PC 串口和 MSP430 USART 电路连接起来。上电重启 MSP430 控制器。如果 Flash 监控器确认与终端正确连接并配制好,终端显示屏上就会有相关显示。
5. 确认文件传输的格式与监控编程软件要求的是否一样。在设置页面对话框中单击 ASCII 栏,检查发送与接收标签,在输入数据前延时 50ms。
6. 确认所加载应用程序的存储器范围内所有的存储单元被擦除。如果有必要,擦除整个 Flash 存储器。
7. 上位机输入 U 指令。从上位机菜单选择,传输-**a** 发送 Text 文件。
8. 在弹出对话框中,选择传送文件的 MSP430-TXT 格式,单击 OK 按钮。
所有数据向终端传送完后,回送一个信号。监控器返回到主循环程序,进行新一轮数据处理。
- 3 建立监控应用程序
监控程序是非常紧凑并且除占用 2 个 Flash 段外不占用任何处理器资源。
建立监控程序及应用程序要注意以下几个方面:
 1. 常用 MSP430F1XX 或 MSP430F4XX 系列任何芯片只要有一个 USART 就可以建立监控。

2. 对于目标设备可有两个 USART，分别是 USART0 和 USART1。
3. 监控程序占用 Flash 存储器的第 1, 2 段或 Flash 存储器的日子最初的 512 个字节。
4. 监控程序和应用程序分开建立，调试，和仿真。

3.1 监控程序选项

建立监控程序的原文件如表 3 所示

表 3 监控程序源文件

文件名	说明
Flash_monitor.S43	监控程序的汇编源代码
TargetDefs.h	用于对部分特殊目标板设置的头文件，有以下几个可选。 MSP430 目标设置 同主机通信的串口 是否在终端显示存储区内容 监控程序或是应用程序先启动 擦除 Flash 时是否要求口令
Flash_monitor.h	选择串口及目标设备内存区边界值检测

监控程序的选项在 TargetDefs.h 头文件中通过#define 声明，如下所示。下面这个段讨论这个控制选项。

```
// TargetDefs.h 建立监控设置选项
#define CHIP F169           //MSP430F169 芯片
#define MEMDUMP 1         //允许建立内存区
#define SERIAL_PORT_1     //用USART1
#define ECHO 1            //对于所有的 USART 输入的数据都返回一个值
#define DELAY_START 1     //定义监控程序优先运行
#define PASSWORD 1       //允许口令保护
#define DIRECT_INTERRUPTS //未定义，监控程序则位于Flash的低位存储区
// 如未定义，监控程序则位于Flash的高位存储区
```

主要的选项是目标设备，USART 的使用，监控程序放置在 Flash 存储区的那个段。其次是否要显示 MSP430 的内存区，监控程序或是应用程序那个在启动是先运行，是否要 Flash 存储区擦除口令保护，是否需要终端回显至 USART，是否需要显示启动进度条。

3.1.1 主要选项

3.1.1.1 选择目标设置

TargetDefs.h 头文件通过#define CHIP 语法定义来选择目标设备。TargetDefs.h 对 CHIP 的定义后面通常接一个具体的型号。例如：监控程序运行于 MSP430F169,则芯片的定义行必须如下定义: #define CHIP F169。

3.1.1.2 选择 USART

USART 接口的选择是通过#define SERIAL_PORT_1 语句选择 USART1。

3.1.1.3 应用程序的内容放置在存储器位置

监控程序占用 MSP430 主存储器的两个 512 字节的段.位于 MSP430 所有系列芯片的 0XFFC0~0XFFFF,可选择为存储器的 0,1 段.或可位于最低的两个 512 字节的段中,一般位于 N 和 N-1 段中,N 的值取决于 Flash 总的存储器的容量。

每种方法都有其优点与不足.当监控程序位于存储器的高地址,中断向量表和监控向量程序位于同一个段.因为 Flash 存储器只能一段段的擦除.,意思就是中断向量表中必须有固定的值.为达到一致,只能用特定的间接方式处理中断请求.间接跳转的中断向量点位于 1 段的末端.ISRs 中断表位于 PC 加载的 2 段(0XFBE0~FBFF)的中断向量表.增加三个钟周期来处理理中断.这个方法的好处就是监控程序不会擦除中断向量表.因为主

的向量表与监控程序位于 Flash 存储器 0 段。监控程序不会擦本身的程序段。不会被错误的程序段所加载，假设应用程序不能擦除 Flash 存储器。用这种方法，监控程序的访问只有通过重启来实现。

当监控程序位于存储器的低地址，应用程序 ISRs 的中断向量表位于 0XFFE0~0XFFFF，可省略三个时钟周期处理中断，但是要求在加载应用程序时对中断向量表重新写。如查程序在加载时带无效的向量表，或应用程序不能通过监控程序进行编程，只有通能 ROMbootlader 或 JTAG 接口。

监控程序位于存储器的低地址，TargetDefs.h 头文件要包括#define DIRECT_INTERRUPTS 这行。如果这行被删除或被注释掉。监控程序则位于 Flash 存储器的 0,1 段。给 Flash 存储器的应用程序设置一个边界值。

3.1.2 附加选项

3.1.2.1 允许显示 MSP430 存储器内容

如果 MEMDUMP 是非零值，可显示 MSP430 的存储器内容。如果 MEMDUMP 没定义或者这义的值是零，存储器内容不显示。

3.1.2.2 终端返回值

如果 ECHO 定义的值是非零值，监控器返回 USART 每次传输过来的值。禁止这个功能只要将 ECHO 不定义或定义为 0。

3.1.2.3 Flash 擦除口令保护

口令保护功能是防止 Flash 存储器的不经意的擦除。不保证 Flash 内容的安全或被非法读取。这样做的目的只是防止操作人员对 Flash 存储器的失手操作。

口令保护允许时，监控程序在接收到 E 指令时准备接收 16 位的口令。将接收到的 16 位的口令在接收到应用程序的中断时与存储的口令进行对比。如果口令不匹配。监控程序返回到命令状态。

3.2 建立应用程序

运行在监控器里的应用程序可以用汇编或 C 语言编写。监控程序设计得非常紧凑小巧。其中一个方面考虑的是能否将监控程序植入应用程序中。如果监控程序在复位后没运行，应用程序要能够启动监控程序，通过监控程序的中断向量。定义为公共量 (Reset)，在建立链接器的位图文件里可找到其绝对地址。

Flash 监控器在建立监控应用软件时有如下要求：

1. 不能擦除监控程序占用的 Flash 段
2. 输出的文件格式是 MSP430-TXT
3. 复位后监控程序不能首先运行，应用程序能提供一种主监控程序运行的方法。

3.2.1 C 语言编写应用程序

对于 C 语言编写的应用程序，要求能通过链接命令文件来改变存储区限制及修改中断向量存放的位置。

对于 IAR 链接器在链接器命令文件的相关章节的注释。

```
//-----  
// Rom 存储器 (FLASH)  
//-----
```

占用存储器高地址的监控程序的地址可以被修改。应用程序的中断向量要存放在用户 Flash 的末端。如图 3 所示显示了用于 MSP430F169 的 IAR 链接器文件。用 MSP430F169.XCL 修改并建立一个位于 Flash 存储器 0, 1 段应用程序。

```
// Code

-Z(CODE) CSTART=1100-FBDF
-Z(CODE) CODE=1100-FBDF

// Constant data

-Z(CONST) DATA16_C, DATA16_ID, DIFUNCT, CHECKSUM=1100-FBDF

// Interrupt vectors

-Z(CONST) INTVEC=FBE0-FBFF
-Z(CONST) RESET=FBFE-FBFF
```

图 3 链接指令示例

-Z (CODE) 定义声明了存储器段内容将会被链接器的代码替换。如：MSP430F169，存储空间范围是 0X1100—0XFBDF。

-Z (CONST) 定义声明了存储器段内容将会被链接器的数据替换。

32 字节 INTVEC 用作中断向量，2 字节 RESET 用作复位 向量。

修改链接文件后，存储器使用区的程序与监控软件所要求的相兼容。确认没有用修改后的链接指令文件来重写 IAR 提供的指令文件。改变修改后链接文件的默认选项输出格式为 MSP430-TXT 的文件。如果这个选项被选择后，链接器输出的二进制文件可以通过调试器下载进 FLASH 也可通过 JTAG 仿真器和将 TXT 格式的监控程序加载进来。

1. 在链接目录中，选择输出表，检查产生输出文件及默认的文件，输入将要产生输出的文件名，选择 MSP430-TXT 格式的文件并下载。
2. 选择配制表，在链接文件区，检查默认的文件为修改的链接文件输入文件路径名。

3.2.2 用汇编语言编写应用程序

当用汇编语言编写应用程序，用常用的 ORG 和 DW 来定义中断向量，如建立的应用程序和监控程序相匹配，只需要替换 FLASH 存储器 2 段的中断向量，而不是 FLASH 存储器 0 的段。

4. 建立监控程序

监控程序及其应用可通过分开建立然后再合在一起。例如：Flash 初始化编程。有利于分开建立单个的二进制监控文件和应用文件，他们可用 JTAG 仿真器存储到 FLASH 存储器。监控软件可用 USART 来擦除和下载程序到 FLASH 存储器。

为单独建立监控程序和应用程序，只需要添加监控源文件（flash_monitor.s43）到目标文件并查看包含（TargetDefs.h）文件的路径。

如果监控程序作为单独建立并存储在 FLASH 的程序，在系统复位后总会首先运行。

5 应用文档

应用程序中包含了监控程序的源程序。基于 FET 程序文档的 FET140_UART15_9600.C.用于 MSP430F169 芯片并运行于 SoftBaugh Dlr169 的目标板。

示例应用程序在初始化 USART 端口和定时器 A 后进入 LPM3 直到中断发生。

定时器 A 用于产生 0.5HZ(2S)的中断率。定时器 A 的中断在 P1 的 0 位。TI MSP-FET 的目标板上及 SoftBaugh Dlr169 板上让 LED 灯闪烁。

USART 的 ISR 每个输入缓冲区接收到的字符复制进输出缓冲区，同样检查每个接收到的字符来判断是否为

^C 字符 (0x03 或 ETX)。收到 ETX 字符后, 在 0XFC40 的情况下应用程序对监控器的传输控制通过 PC 在线联机的形式来热启动。

对应用文档建立的要求如表 4

表 4 应用文档

文件名	说明
Monitor_demo_app.c	是 Fet140_uart15_9600.C 文件基础上修改所得文件
Ink430F169_boot.xcl	对 IAR MSP430F169 链接文件修改来兼容监控器在 3.2.1 节

6 参考资料

1. MSP430X1XX 用户手册(TI 文献编号 SLAU049)
2. MSP430IAR 汇编参考手册(IAR EW 使用帮助)
3. IAR 链接及库工具参考手册(IAR EW 使用帮助)
4. SoftBaugh Dlr169 示例板原理图

参考例程

MSP430 远程更新程序 <http://focus.ti.com.cn/cn/lit/an/slaa341/slaa341.zip>



