

文章编号: 1008-9225(2008)06-0114-03

利用 TIMER 的捕获比较功能 模拟 MSP430 UART 的研究

付明

(安徽财经大学 信息工程学院, 安徽 蚌埠 233041)

摘 要: 针对 MSP43014X 系列单片机, 利用片内的定时器 TIMER-A 的捕获比较功能, 通过软件模拟的方式来实现 UART 的功能, 扩展了串行通信的接口, 为工程实际应用提供了更丰富的资源。

关键词: 捕获比较寄存器; 串行通信; MSP430

中图分类号: TP 273+.5 **文献标识码:** A

德州仪器德 MSP430 单片机具有丰富的外围模块, 如 MSP430F149 就包含: 12 位 A/D, 精密模拟比较器, 硬件乘法器, 2 组频率可达 8 MHz 的时钟模块, 2 个可以实现异步、同步及多址访问的 UART。由于在实际应用中, UART 接口具有更大的通用性, 而 MSP430 本身的 UART 有可能不够用, 故往往需要用 TIMER 定时器来模拟 UART, 以获得更多的接口^[1]。本文根据在实际工程中的应用, 以 MSP430F149 为例介绍利用 MSP430 中的定时器模拟串口通信的方法。

TI 所出的全部 FLASH 单片机都含有 Timer-A, 而在 MSP430F13X 系列和 MSP43014X 系列中既含有 Timer-A 也含有 Timer-B, 在 F13X 中有一个带有 3 个捕获比较模块的 Timer-B, 在 F14X 中有一个带有 7 个捕获比较模块的 Timer-B, 它们均是扩展 UART 口的核心。由于在 MSP430 的中断级别中 TIMER-B 的中断级别较 TIMER-A 的中断级别要高, 工程中一般是使用 TIMER-B 来实现定时等功能, 而倾向于用 TIMER-A 来模拟 UART。

1 TIMER 捕获比较功能的介绍

MSP430 系列单片机中都集成了捕获比较的功能模块。捕获比较功能的引入主要是为了提高 I/O 端口处理事务的能力和速度^[2]。以下结合实现串行通信的需要, 简要介绍有关捕获比较的有关概念。

捕获比较模块用于捕获应用事件的发生时

间, 或者用于计时。如果相应的中断允许, 那么完成一个时间捕获或一次定时间隔, 捕获比较模块都将产生中断。每一个捕获比较模块都可以对应一组硬件引脚。

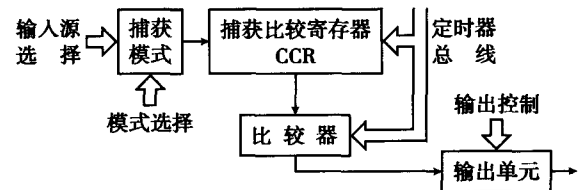


图1 捕获比较模块的结构框图

捕获功能可以捕捉选定输入引脚的状态的变化, 它可以选择捕捉上升沿、下降沿, 或者电平的跳变。如果捕捉到了相应的变化, 则定时器计数值将被复制到捕获比较寄存器 CCRx 中, 并会产生相应的中断。在串行通信中, 正是利用捕获功能的特点来捕捉起始位的信息^[3]。比较功能就是借助比较器不断地把 CCRx 总的设定值与定时器总的计数值相比较, 当两者相等时就产生中断, 并产生设定的中断, 利用比较功能可以获得精确的时间间隔, 利用这个特性可以构造一个精确的波特率发生器为串口通信提供时间基准。

2 UART 的字符格式

在单片机系统中, 串口通信是一个非常重要的部分, UART(Universal Asynchronous Receiver/Transmitter)即异步串行通信, 每个数据帧一般由 1 位起始位、8 位数据位、1 位奇偶校位、1 位停止位组成。用户可以通过软件设置数据位, 停止位的

收稿日期: 2008-03-29

作者简介: 付明(1981-), 男, 湖北仙桃人, 安徽财经大学讲师, 硕士。

位数,还可以设置奇偶位的有无^[4].在实际发送数据的过程中,串口是以字符为基本单位来发送的,奇偶校验位省略.首先把输出电平置高,然后是一个下降沿,输出一个系统时钟的低电平作为起始位,然后再输出数据位,分别占用了八个时钟周期,最后输出一个系统时钟周期的高电平作为结束位.

3 模拟功能的实现

3.1 接收过程

图2所示为一个3位数据帧的时序.在接收这种格式的数据帧时,首先要确定起始位,用来进行帧同步.在MSP430中是利用捕获功能来捕捉起始位的下降沿.定义TIMER-A的CAP位为1,则MSP430进入捕获模式,如图2,在A点捕获到起始位,系统将此刻的定时器值T存入CCR_x中,并由于捕获而产生中断.在TIMER-A的中断处理程序中,对A点所产生中断的处理非常重要,一进入中断就将捕获功能转换为比较功能.另外设置TACTL中的MC0位为1,使430进入增计数模式,并将1.5位的时间间隔(1.5T)加到CCR_x中,即CCR_x=1.5T.同时将MSP430内部的计数器TAR的值清零.这样,当MSP430内部的计数器TAR到达1.5位时间间隔时(B点),即定时器TAR的值等于CCR_x中的值,就会由此比较功能触发中断,这样就实现了1.5位时间间隔的精确定时.由于1.5T的时间跳过了起始位,刚好是在第一位数据位的中间,这时的数据位是最稳定的,没有电平的突变,在中断处理程序中可以读取输入引脚的状态,从而接收到Bit1的信息.同时,重新设定CCR_x中的值,将其设定为一位数据位T的时间,TAR清零重新开始计数,当TAR计数达到CCR_x中的值时,又产生一次中断,这时刚好是在第二位的数据位的中间部分,中断服务程序中可以读取Bit2的信息.如此重复8

次,就可以完成一个字节数据的接收.收完一个字符后,再重复上述过程,可以连续接收多个字符.

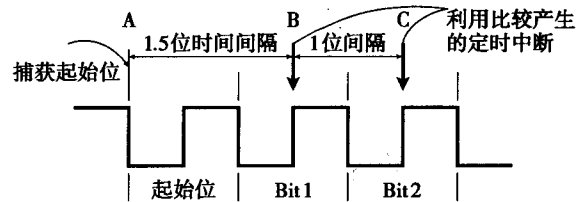


图2 在串行通信接收过程中捕获比较功能时序分析示意图

3.2 发送过程

发送过程比较简单.利用比较功能产生一个间隔为1位时间(T)的时序,相当于一个波特率发生器.首先,设定TIMER为比较模式,即TACTL中的CAP设为1,设定CCR0的值为T,清除TAR,设为增计数模式,则TAR从0计数到T,到达T是触发中断,进入中断处理程序,然后TAR再清零(硬件自动完成),重新开始计数,计到T时又触发中断,如此反复.在每一次比较功能触发的中断服务程序中发送一位数据位,如此循环执行,这样就可以完成一个数据帧的发送.异步串行通信的一个数据帧往往是10位(省略了奇偶校验位)或11位.可以利用将数据帧的所有位安排在一个待发送字中,然后移位发送,而不需要专门编程产生起始位和停止位.

3.3 波特率的确定以及中断的安排

从以上的分析可以看出,串行通信的波特率主要是与1位时间间隔T有关,T的有关值和MSP430使用的晶振的频率有关,也和TIMER-A选择的MSP430时钟源有关,在这里以实际工程出发,详细地阐述一下时间T的计算算法.

为MSP430选用8MHz的晶振,在主程序里对系统的时钟进行定义,定义了子系统时钟SMCLK,选用8MHz的晶振,再把SMCLK的时钟八分频,这样SMCLK的时钟频率就是1MHz,选用SMCLK为TIMER-A的时钟.

(1) 时钟初始化的程序如下:

```
BCSCTL1 &= ~XT2OFF;           //XT2CLK 模式
BCSCTL1 &= ~XTS;              //aclk 32k
Do                               //循环是为了保证晶振稳定
{
    IFG1 &= ~OFIFG;           //清除震荡器故障中断标志
    for(i=0xff;i>0;i--);      //时钟至少要稳定的时间.
}while(OFIFG&IFG1);           //如果存在震荡器故障则循环
BCSCTL2|= SELM1;              //MCLK 时钟源为 XT2CLK 8 MHz
BCSCTL2 &= ~DIVM0;           //MCLK 不分频
BCSCTL2 &= ~DIVM1;           //MCLK 不分频
BCSCTL2|= SELS;              //选择 SMCLK 的时钟是 xt2clk 8 MHz
BCSCTL2|= DIVS1 + DIVS0;     //设置 SMCLK 为 8 分频后是 1 MHz
```

然后对TIMER-A进行初始化,使其进入到捕获模式,来捕捉起始位.P1.1是MSP430的捕

获的输入脚,初始化必须在进入中断前执行.

(2) 接受数据的初始化程序如下:

```
TACTL|=TASSEL1+TACLR; //选择 SMCLK 为 TIMER 的时钟,清除 TAR
P1SEL|=BIT1; //P1.1 用于捕捉下降沿
P1DIR&=~BIT1; //P1.1 用于输入
TACCTL0=CM1+CCIE+CAP+OUT; //下降沿捕获,中断允许
```

由于在串口通信中,最常用的波特率是 9 600 bps 或者 4 800 bps,绝大多数的设备都支持该速率,在这里就用 9 600 bps 作为串口的波特率,TIMER_A 的时钟是 1 MHz,一个周期就是 $T=$

SMCLK/9 600 bps,在 9 600 bps 的速率下,一个 T 就是 $1\text{ MHz}/9\ 600\ \text{bps}=104\ \text{bit}$,那么一个半周期就是 $1.5T=156\ \text{bit}$,由于在工程实际应用中都采用 16 进制,即为 $1.5T=0X9C$, $T=0X68$.

(3) 初始化 TIMER_A 的程序如下:

```
TACTL|=TASSEL1+TACLR; //MSCLK
TACTL|=MC0; //当波特率为 4 800 bps 时,CCR0=0x400;
CCR0=0x9c; //0X9c=156;
CCTL0=CCIE;
P1SEL&=0x7D; //P1.1 改为 IO 端口,设为输入
P1DIR&=0XFD;
BitCount=11; //记录发送或接受的位数
RXTXData=0;
```

这样就设置好了 TIMER_A 定时器,打开中断,那么,当 P1.1 上出现了下降沿,就会硬件触发中断,进入中断。

(4) 中断程序设计:

定义 IO 端口的 P1.5 发数据,P1.1 收数据,这样就在中断函数中实现了 UART 的发送和接受功能.当下降沿触发,进入中断后,把 CCRx 的值改为一个周期 T,跳过了起始位。

非常小.例如,当定时器的计数时钟频率为 1 MHz,波特率为 9 600 bps 时,要求定时器计数 104.166 6,然而,定时器只能计整数,因此选择每计数 104 为一个比特.这样产生的误差是非常小的。

4 结 论

本文研究的设计方法在工程实践中经过了调试,在 IAR Embedded Workbench IDE 开发环境中成功地实现了模拟 UART 的功能,而且发送和接受都安排在一个中断函数中,方便了程序的开发.MAP430 单片机的定时器功能强大,中断源较多,可以任意嵌套,这为工程开发人员提供了较多的选择余地.由于以前大部分的程序代码都用汇编给出,而现在工程实际中一般都是采用 C 编写程序,所以在这里给出了部分 C 源代码。

3.4 TIMER_A 模拟 UART 时间误差分析

对于 UART 口来说,由于字节间存在累积误差,因此,对波特率的要求较高,需要时钟频率非常接近波特率的整数倍.而实际波特率往往是通过时钟分频得到,无法产生精确的波特率,不可避免地会扩大累积误差.对于利用定时器实现的 UART 口来说,在发送或接收下一个字节时,定时器重新开始捕获下一个字节的起始位的下降沿,重新开始计数,因此,字节之间不存在累积误差,仅存在字节内的累积误差,因此,这大大降低了之前需要时钟接近波特率的整数倍的要求.只要 TIMER 的时钟频率足够高,就完全可以实现精确的接受和发送。

参考文献:

在定时器实现的 UART 口中,当定时器的计数时钟频率比较高时,字节内产生的累积误差也

- [1] 胡大可. MSP430 系列超低功耗 16 位单片机原理与应用 [M]. 北京:北京航空航天大学出版社, 2000:28-31.
- [2] 刘立群. 基于 MSP430 单片机的超低功耗数据采集器设计 [J]. 自动化仪表, 2005,26(4):30-31.
- [3] 刘玉宏. MSP430 单片机 C 语言和汇编语言混合编程 [J]. 微计算机信息, 2003,19(10):56-57.
- [4] 赵战克. 单片机在移频信号频率检测中的应用 [J]. 微计算机信息, 2004,20(2):76-77.

On Simulation of MSP430 UART Using Capture Comparison Function of TIMER

FU Ming

(College of Information and Technology, Anhui University of Finance and Economics, Bengbu 233041, China)

Abstract: The simulation problem of universal asynchronous receiver/transmitter (UART) interface is discussed. To MSP43014X chips, a method of extending I/O interface through software stimulation is presented to realize the function of UART by using the capture comparison function of the inner chip TIMER_A, which can provide more resources for practical project.

Key words: Capture/compare register; I/O Function; MSP430

【责任编辑 张耀华】