

文章编号:1009-671X(2006)07-0011-04

利用 MSP430 的捕获比较模块实现 UART 功能

韩慧群, 吴简彤, 李金阳

(哈尔滨工程大学 自动化学院, 黑龙江 哈尔滨 150001)

摘要:为了满足单片机系统设计中多个串口的需要,文中使用 TI 公司 MSP430 型单片机的捕获/比较模块实现 UART 功能.文中详细介绍了 UART 的通信方式,并根据 MSP430 型单片机捕获/比较模块的特点,设计实现了用软件来模拟 UART 的方法.通过该方法,用户可以在不增加外部器件的情况下,灵活地为该系列单片机进行多串口扩展.

关键词: MSP430; 捕获比较; UART; 串行通信

中图分类号: TP368.1 **文献标识码:** A

Implementing a UART function with MSP430's capture compare module

HAN Hui-qun, WU Jian-tong, LI Jin-yang

(College of Automation, Harbin Engineering University, Harbin 150001, China)

Abstract: In order to meet the requirement of multi UARTs application, an implementation of UART function for TI's MSP430 family, using the Capture Compare Module, is described in this paper. Communication of the UART is given. This method can be used for expanding serial communication port for MSP430 for the user, without any additional circuit.

Keywords: MSP430; capture compare; UART; serial communication

TI 公司的 MSP430 系列单片机是具有超低功耗特性的功能强大的单片机.它采用 16 位总线“冯-诺依曼结构”,外设和内存进行统一编址,寻址范围可达 64 K.它具有统一的中断管理,丰富的片上外围模块.由于其较全的功能和较高的性价比,应用已日趋广泛.

该系列单片机具有很多型号,它们是由一些基本功能模块按不同的应用目标组合而成.其中的 16 位定时器 Timer_A 和 Timer_B 中每个捕获/比较模块的硬件特性,使得其可以以硬件方式支持实现串行通信,并能够以最少的软件操作和最低的 CPU 负载来实现 UART.此外,利用捕获/比较模块来实现 UART,在配置上还具有很大的灵活性.应用时可以在单工和半双工方式之间进行选择,还可以通过对软件的修改来适应各种不同的串行通信数据格式.

1 捕获/比较模块^[1]

目前 TI 公司的所有 FLASH 型 MSP430 系列单片机内都含有 16 位定时器 Timer_A,在有些系列(如 MSP430F13X, MSP430F14X)中,还含有 Timer_B.该定时器具有 3 个或 7 个相互独立且功能相同的捕获/比较模块.每个模块都可用于捕获事件发生的时间或产生定时间隔,并产生相应中断.

捕获模式常用于确定事件发生时间.如果在选定的输入引脚上发生选定的脉冲触发沿(上升沿、下降沿或任意跳变),则该时刻定时器的计数值将被保存,同时产生中断.因此可以将这一功能特点应用于对串行数据起始位的检测.比较模式一般用于在设定的定时间隔产生中断.当 CCRx 寄存器中的设定值与定时器中的计数值相等时,就会产生中断,

收稿日期:2005-09-08.

作者简介:韩慧群(1981-),男,硕士研究生,主要研究方向:导航、制导与控制,E-mail:gordon.han@hotmail.com.

并通过模块的输出单元产生指定的输出. 这样就可以利用这一特点产生精确的时间间隔来实现串行通讯中各种波特率的发生.

MSP430 的 Timer_A 和 Timer_B 功能基本相同, 除了在计数长度上有所不同之外, 重要的是只有 Timer_A 具有将输入信号锁存到控制寄存器中 SCCI 位的功能. 而且, Timer_A 定时器的捕获/比较模块中, 将比较数据存储到 CCRx 中的动作是自动完成的, 因此 Timer_A 定时器的捕获/比较模块更适合于用来实现 UART 的接收功能, Timer_B 则适合用来实现发送功能. 另外需要注意的是, 在实现 UART 方法中, 要求 Timer_A 和 Timer_B 定时器应一直保持在计数状态, 因此要选取 2 个定时器的工作模式均为连续计数模式.

2 UART 功能实现

2.1 UART 通信方式^[2]

在异步串行通信中, 数据是以帧为单位传送的. 在帧格式中, 一个字符由 4 个部分组成: 起始位、数据位、奇偶检验位和停止位. 起始位(0)信号只占一位, 用来通知接收设备一个待接收的字符开始到达. 线路上在不传送字符时应保持为 1. 接收端不断检测线路的状态, 若连续为 1 以后又检测到一个 0, 就知道发来一个新字符, 应马上准备接收. 图 1 显示了异步通信中一个数据帧的格式.

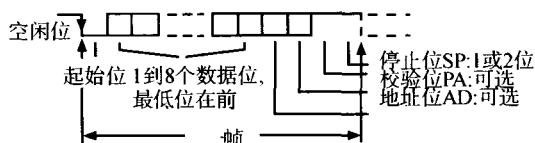


图 1 异步通信一帧的数据格式

2.2 UART 接收功能的实现

如上所述, 接收功能由 Timer_A 来实现. 在接收模式中, 捕获/比较模块首先被设置为捕获下降沿的工作方式, 用于捕获串行通信数据中由空闲位到起始位的下降沿标识, 以确定新的一帧数据开始接收. 采用这种方法, 可以在 UART 没有接收数据的情况下, 不会给 CPU 带来任何额外的负担. 当捕获到接收端的下降沿后, 系统将此时 Timer_A 计数寄存器中的计数值保存在 CCRx 寄存器中, 这一工作是由硬件自动实现的. 而且每个捕获/比较模块的 CCRx 寄存器是相对独立的, 不会产生相互的影响. 捕获的

同时, 系统会产生中断请求, 对于这个中断, 并不一定要求系统立即给予响应. 实际上, 对这一中断的滞后响应, 对 UART 接收功能的实现并不会带来很大的影响. 这样就可以允许 CPU 进行多个事务处理, 或者进行多个 UART 的同时接收或发送.

检测到起始位之后, 将捕获/比较模块设置为比较工作模式, 利用 Timer_A 的计数定时功能作为波特率发生器. 在每个数据位传输的中间时刻对输入端进行检测, 来确定数据位的逻辑值. 首先, 在起始位的捕获中断处理程序中, 一方面要把捕获/比较模块的工作方式改为比较模式进行定时; 另一方面要在 CCRx 中设置定时间隔. 定时间隔是根据 UART 的波特率进行确定的, 即使得第一次比较定时中断发生在第一个数据位信号的中间时刻. 为此, 在捕获中断程序中将 1.5 位的定时间隔加到 CCRx 寄存器中. 这样, 就可以在第一个数据位传输的中间时刻, 因定时器数等于 CCRx 的值而触发中断, 在这个中断处理程序中就可以接收第一个数据位了. 比较中断触发的同时, Timer_A 的 EQU 信号会把输入端的逻辑值同步锁存在透明锁存器中, 这个逻辑值可以由 CCTLx 寄存器中的 SCCI 位读出. UART 功能接收到的数据位就是这样从 SCCI 位获得的.

在接收完第一个数据位以后, 将 1 位的定时间隔加到 CCRx 寄存器中, 使得下一次定时中断发生在下一位数据传输的中间时刻. 以同样的方法系统就可以将串行数据帧的各个数据位依次从 SCCI 读出, 完成一帧数据的接收. 图 2 显示了接收一个数据帧的过程.

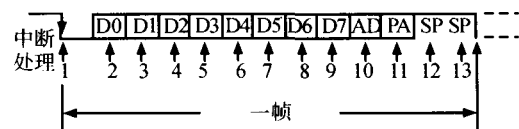


图 2 接收一个数据帧的过程

2.3 UART 发送功能的实现

发送功能由 Timer_B 实现. 与接收功能相比, 发送功能的实现要简单一些. MSP430 利用 Timer_B 的定时器模式进行定时来决定何时发送下一位数据. 在发送之前, 捕获/比较模块被设置为比较模式. 每需要发送一帧数据时, 先在输出端输出逻辑 0 表示数据帧的起始位, 并将 1 位的定时间隔加到 CCRx 寄存器中, 为起始位设置定时. 在起始位的定时中断处理程序中, 再将 1 位的定时间隔加到 CCRx 寄存

器中,并根据第一个数据位的逻辑值预先设定好输出模式.如此反复,就可以将一帧中所有的数据位发送完.可以看出,UART 发送的数据逻辑值,是在比较定时中断产生的同时,由捕获/比较模块自动在输出端根据预定的输出模式输出的.图3显示了发送一个数据帧的过程.

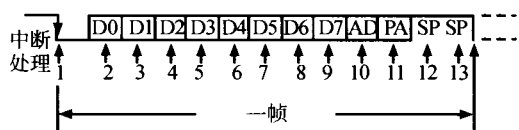


图3 发送一个数据帧的过程

3 波特率计算

在UART的实现中,只需要对加到CCR_x中的数据位定时间隔做相应的改变就可以满足对不同波特率的需要.一个数据位的定时间隔指的是,在UART接收或发送相邻2位数据的时间内Timer_A或Timer_B的累加计数值.如图4所示.一个数据位定时间隔的计算方法是用Timer_A或Timer_B时钟源的频率除以波特率.例如,对于采用8MHz的SMCLK的Timer_A,实现波特率为9600的UART,其一个数据位的定时间隔为

$$\text{定时间隔} = 8\,000\,000 / 9\,600 = 833.333 \approx 833$$

$$\text{实际波特率} = 8\,000\,000 / 833 = 9\,603.8$$

由于定时间隔只能是整数,所以将833.333近似为833,这对波特率发生带来的误差基本上是可以忽略的.

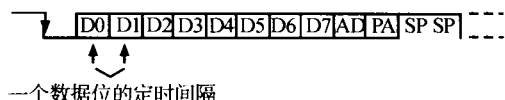


图4 一位定时间隔

4 实现程序代码

以MSP430F169为例,实现波特率为9600的UART.利用Timer_{A3}的第一个捕获/比较模块TA0实现UART的数据接收,利用Timer_{B7}的第一个捕获/比较模块TB0实现UART的数据发送.即P1.1作为UART接收引脚,P4.0作为UART发送引脚.Timer_{A3}和Timer_{B7}均采用系统的8MHz SMCLK作为时钟源.异步通信数据格式为8N1.下面给出C语言实现代码.(以下程序均在AQ430编译环境下测试通过.)

接收准备子程序:

```
unsigned char RXBitCnt0; //接收位数计数
void RXReady (void) {
    RXBitCnt0 = 0x08; //初始化接收数据计数器
    CCTLO = CCIS_0 + OUTMOD0 + CM_2 + CAP +
    CCIE; //初始化捕获比较控制字
}
```

发送子程序:

```
unsigned short TXByte0; //移位发送缓冲,
unsigned char TXBitCnt0; //发送位数计数
void TXByte(unsigned short byte0) {
    //将 byte0 由软串口 0 发送
    TXByte0 = byte0;
    TXBitCnt0 = 0x0a; //1 个起始位,
    //8 个数据位,1 个停止位
    TXByte0 |= 0x100; //添加停止位
    TXByte0 = TXByte0 << 1; //添加起始位
    TBCCR0 = TBR; //从当前计数值开始发送
    TBCCR0 += UARTBitime; //增加 1 数据位定时间隔
    //UARTBitime = 833
    TBCCTL0 = OUTMOD0 + CCIE; //设置输出模式
    //打开中断,开始发送
    while ( TBCCTL0 & CCIE ); //等待一帧发送完毕
}
接收中断服务子程序:
void _INTERRUPT [ TIMERA0 _ VECTOR ] RX0_in-
thandler(void)
{
    TACCRO += UARTBitime; //增加一位定时间隔
    //UARTBitime = 833
    if( CCTLO & CAP ) { //捕获模式,起始位
        CCTLO &= ~ CAP; //改为比较模式
        TACCRO += UARTBitime_5; //增加半位定时
        间隔
        //UARTBitime_5 = 417
    }
    else {
        RXData0 = RXData0 >> 1; //当前接收的数据位
    }
}
```

```

if ( CCTLO & SCC1)
    RXData0 |= 0x80;
RXBitCnt0 --;          //下一位数据
}
}

发送中断服务子程序:
void _INTERRUPT [ TIMERB0 _ VECTOR ] TX0 _ in-
handler( void)
{
    TBCCR0 + = UARTBitime; //增加一位定时间
    隔
    //UARTBitime = 833
    if ( TXBitCnt0 == 0)
        TBCCTL0 & = ~ CCIE; // 所有数据位发送完
        毕
    else {
        TBCCTL0 |= OUTMOD2; //设置输出模式
        if ( TXByte0 & 0x01)
            TBCCTL0 & = ~ OUTMOD2;
            //发送逻辑 1
        TXByte0 = TXByte0 >> 1; //下一位
    }
}

```

```

TXBitCnt0 --;
}
}

```

5 结束语

上述方法已经在实际系统中得到很好的应用. 实践证明,使用这种方法实现的 UART 串口可以支持多种串行数据通信协议,波特率可以达到 115 200 bps 以上. 因此,该实现原理为 MSP430 系列单片机的串行端口扩展提供了一种更加简单、方便、有效的办法. 此外,还为其他类型的单片机进行串口扩展提供了新的思路.

参考文献:

- [1] 胡大可. MSP430 系列 FLASH 型超低功耗 16 位单片机 [M]. 北京:北京航空航天大学出版社,2001.
- [2] 李朝青. 单片机原理及接口技术 [M]. 北京:北京航空航天大学出版社,1999.

[责任编辑:张晓京]