MSP430C延时程序设计(为了阅读方便,贴在下面)

MSP430 是超低功耗 16 位单片机，越来越受到电子工程师亲睐并得到广泛应用。C 程序直观，可读性好，易于移植和维护，已被很多单片机编程人员所采用。MSP430 集成开发环境(如 IAR Embedded Workbench 和 AQ430)都集成了 C 编译器和 C 语言级调试器 C—SPY。但是 C 语言难以实现精确延时，这一直困扰着很多 MSP430 单片机程序员。笔者在实际项目开发过程中，遇到很多需要严格时序控制的接口器件，如单总线数字温度传感器 DS18820、实时时钟芯片 PCF8563(需要用普通]/o 模拟 12C 总线时序)、三线制数字电位器 AD8402、CF 卡(Compact Flash Card)等都需要μs 级甚至纳 ns 级精确延时；而一些慢速设备只需要 ms 到 s 级的延时。为此，笔者提出了适合于不同延时级别需要的软件或硬件精确延时方法，并已实际应用，效果良好，大大缩短了开发周期。

1 硬件延时

MSP430 单片机系统程序多采用事件驱动机制，即在没有外部事件触发的情况下 CPU 休眠于低功耗模式中。当外部事件到来时，产生中断激活 CPU，进入相应的中断服务程序(ISR)中。中断响应程序只完成两个任务，一是置位相应事件的标志，二是使 MCU 退出低功耗模式。主程序负责使 MCU 在低功耗模式和事件处理程序之间切换，即在主程序中设一个无限循环，系统初始化以后直接进入低功耗模式。MCU 被唤醒后，判断各标志是否置位。如果是单一标志置位,那么 MCU 执行相应的事件处理程序，完成后转入低功耗模式；若是有多个标志同时置位，主程序按照事先排好的消息队列对它们依次判别并进行处理，所有事件处理完毕以后 MCU 休眠，系统进入低功耗状态(该消息队列的顺序是按照任务的重要性设定的优先级)。在这种前后台系统中，由于主程序是无限循环，就必须关闭看门狗，与其闲置，不如用其定时器的功能作硬件延时。使用 MSP430 单片机看门狗定时器实现任意时长精确延时,既满足了系统实时低功耗的要求，也弥补了使用无限循环延时的时间难确定和占用 CPU 时间长的缺点。通过下例，讲解在同一 WDT ISR 中完成不同时长延时的技巧。

```
#pragma vector=WD_r_VECTOR

interrupt void WDT_Delay(void){

/ /看门狗中断服务程序

if((DelayTime&Delay500ms)==Delay500ms){

/ /判断需要 500 ms 延时的标志是否置位
```

```
static unsigned int n250MS=0;

n250MS++;

if(n250MS==2){ / / 延时 250ms×2=500ms

n250MS=0;  / / 清零计数器

DelayTime&=~Delay500ms; / / 复位标志位

WDTCTL=WDTHOLD+WDTPW;

1El&=～WDTlE; / / 关闭看门狗定时器并禁止其中断

}

}

if((DelayTime&Delay30s)==Delay30s){

/ / 判断需要的 30 s 延时标志是否置位

static unsigned int nS=0;

nS++;

if(nS==30){ / / 延时 1 s×30=30 s

nS=0;  / / 清零计数器

DelayTime&=～Delay30s; / / 复位标志位

WDTCTL=WDTHOLD+WDTPW;

IEl&=～WDTlE;  / / 关闭看门狗定时器并禁止其中断
```

```
}
}
}
```

如果任务 1 需要 500 ms 的延时，只需在需要延时处执行如下语句：

```
WDTCTL=WDT_ADLY_250;
IE┃ =WDTIE;  / /①
DelayTime┃=Delay500ms  / /②
while((DelayTime&Delay500ms)==Delay500ms);  / /③
```

①处是配置看门狗工作在定时器模式，WDT 每隔 250 ms 产生一次中断请求。可以根据需要改变时钟节拍，在使用 32768 Hz 晶振作为时钟源时，可以产生 1. 9ms、16 ms、250 ms 和 1000 ms 的延时基数。在头文件 msp430xl4x. h 中，将这 4 种翻转时间的 WDT 配置宏定义为：WDT_ADLY_1_9、WDT_ADLY_16、WDT_ADLY_250 和 WDT_ADLY_1000。如果用 DCOCLK 作为 SMCLK 的时钟源，WDT 选择 SMCLK=1 MHz 为时钟源，这样可以有 0. 064 ms、0. 5 ms、8 ms 和 32 ms 延时基数可供使用。

②处设置一个标志位，方便 WDT ISR 判别并进入相应的延时分支。

③处一直判别 DelayTime 标志组中的 Delay500ms 位，如果处于置位状态，说明所需的延时未到，执行空操作，直到延时时间到，在 WDTISR 中将 Delay500ms 复位，跳出 while() 循环，执行下一条指令。

同理，如果任务 2 需要 30 s 延时，通过 WDTCTL=WDT_ADLY_1000 激活 WDT 中断，每隔 1 s 进中断一次，在 WDT ISR 中判别标志发现是 Delay30s 置位而不是 Delay500ms 执行 30 s 延时程序分支。每中断一次，计数器 nS 加 1，直到计到 30，说明 30 s 延时完成，清零计数器，停止看门狗(WETCTL=WE)THOLD+WDTPW;)可停止产生中断，并复位该延时标志，以通知任务延时时间到，可以执行下面的指令了。

在 WDT ISR 中可以根据延时基数和计数器的搭配实现任意长度的时间延时。在系统程序设计时，先确定所需的不同延时时间，然后在 WDT。ISR 中添加相应的延时分支即可。嵌入式实时操作系统μC／OS—II 移植于 MSP430 单片机就是使用看门狗定时器产生时钟节拍的。

对于系统比较简单，只需要单一时长的延时．而又要考虑系统功耗时，介绍另一种使用看门狗定时器中断完成延时的方法。若要延时 1 s，则设定 WDT 每 250 ms 中断一次。在需要延时处，启动看门狗定时器并允许其中断，系统进入低功耗模式 3(共有 5 种．模式)休眠。在中断服务程序中对延时时间累加，当达到 1 s 时唤醒 CPU，并停止看门狗定时器中断。实例代码如下：

```
vold main(vold){

WDTCTL=WDT_ADT_ADLY_250)

／／启动 WDT,每 250 ms 中断一次

IEII=WDTIE)／／使能看门狗定时器中断

_BIS_SR(LPM3_bitS+GIE);

／／系统休眠于低功耗模式 3，开总中断

}

#pragrna vector=WDT_VECTOR

—interrupt void WDT_Delay(void){ ／／看门狗中断服务程序

statlc unsigned charn=4;

if(一一 n==0){ ／／延时 4×250 ms=1 s

—BlC_SR_IRQ(LPM3_blts);
```

/ /将 CPU 从低功耗模式 3 唤醒

WDTCTL=WDTHOLD+WDTPW：

IEl＆=～WDTIE；）

/ /关闭看门狗定时器并禁止其中断

}

这种方法充分发挥了 MSP430 系列的超低功耗特性，在等待延时的过程中，CPU 不需要一直判断标志位以得知延时结束，而是进入省电模式。等待过程中，只有极短的时间会在中断服务程序中累计时间并进行判断。可以根据需要设置 CPU 进入不同的低功耗模式 LPMx。如果系统使用了多种外设中断，并在其他中断服务程序中也有

唤醒 CPU 的语句，这种方法便不再适用了。

μs 级延时不宜使用硬件延时，因为频繁的进出中断会使 CPU 用大量时间来响应中断和执行中断返回等操作。硬件延时的方法适用于 ms 级以上的长时间延时。

2 软件延时

在对数字温度传感器 DS18820 的操作中，用到的延时有：15 μs、90μs、270 μs、540 μs 等。这些延时短暂，占用 CPU 时间不是太多，所以比较适合软件延时的方法。通过汇编语言编写的程序，很容易控制时间，我们知道每条语句的执行时间，每段宏的执行时间及每段子程序加调用的语句所消耗的时间。因此，要用 C 语言编制出较为精确的延时程序，就必须研究该段 C 程序生成的汇编代码。

循环结构延时：延时时间等于指令执行时间与指令循环次数的乘积，举例来讲，对如下延时程序进行实验分析。

void delay(unsigned int time){

while(time 一一){};

在 main()中调用延时函数 delayr(n)；得到的延时时间是多少，需要在 MSP430 单片机的集成编译环境 IAR Em—bedded Wclrkbeneh IDE

3. 10A 中编制测试。

使用 C430 写好一段可执行代码，在其中加入延时函数，并在主函数中调用，以 delay(100)为例。设置工程选项 Options，在 Debugger 栏中将 Drivet 选为 Simulator，进行软件仿真。在仿真环境 C—SPY Debugger 中，从菜单 View 中调出 Disassembly 和 Register 窗口，前者显示编程软件根据 C 语言程序编译生成的汇编程序，在后者窗口中打开 CPU Register 子窗体，观察指令周期计数器 CYCLE—COUNTER。可以看到，delay()编译得到如下代码段：

```
delav:

001112 0F4C mov. w R12, R15

001114 0C4F mov. w R15. R12

001116 3C53 add. w #0xFFFF. R12

001118 0F93 tst. w R15

00111A FB23 jne delay
```

单步执行，观察 CYCI 正 COUNTER，发现每执行一条指令，CYCLECOUNTER 的值加 1，说明这 5 条指令各占用 1 个指令周期，循环体 while()每执行一次需要 5 个指令周期，加上函数调用和函数返回各占用 3 个指令周期，delay(100)延时了 5×100+6—506 个指令周期。只要知道指令周期，就能容易的计算出延时时长了。延时函数因循环语句和编译器的不同，执行时间也有所不同，依照上述方法具体分析，可以达到灵活编程的目的。

MSP430 的指令执行速度即指令所用的周期数，这里的时钟周期指主系统时钟 MCLK 的周期。单片机上电后，如果不对时钟系统进行设置，默认 800 kHz 的 DCOCLK 为 MCLK 和 SMCLK 的时钟源，LFXT1 接 32768 Hz 晶体，工作在低频模式(XTS=0)作为 ACLK 的时钟源。CPU 的

指令周期由 MCLK 决定，所以默认的指令周期就是 1 / 800 kHz=1. 25μs。要得到 1μs 的指令周期需要调整 DCO 频率，即 MCLK=1 MHz，只需进行如下设置：BCSCTL1=XT2OFF+RSEL2；

/ / 关闭 XT2 振荡器，设定 DCO 频率为 1 MHz

DCOCTL=DCO2

/ / 使得单指令周期为 1μs

并不是说 MSP430 单片机软件延时最小的延时基准是 1μs，当开启 XT2=8 MHz 高频振荡器，指令周期可以达到 125 ns。MSP430F4XX 系列的单片机由于采用了增强型锁频环技术 FLL+，可以将 DCO 频率倍增到 40MHz，从而得到最快 25 ns 的指令周期。

调用延时函数的方法适合于 100 μs～1 ms 之间的延时，100μs 以下的短延时最好通过空操作语句_NoP()或其任意个组合来实现。可使用宏定义实现需要的延时，如要延时 3 μs，则：

```
#define DELAY5US{_NOP();_NOP(); _NOP(); }
```

结语

本文提出的基于 MSP430 片内看门狗定时器的硬件延时方案和软件延时方法满足了不同时宽级别的延时需求，尤其软件延时，采用汇编程序分析法得到了延时函数准确的延时时间，大大提高了软件延时精确度和程序调试效率，并在多种芯片接口程序中应用，运行效果良好。