KF8F2320——TMR 定时器样例程序

引言

本应用笔记提供了 KF8F2320—TMR0/1/2 定时器相关的配置信息以及如何能够快速的理解并上手使用该模块的一些配置方式。

本应用笔记须与 KF8F2320 数据手册结合使用。

寄存器

寄存器使用说明：

OSCCTL（系统控制寄存器）

寄存器OSCCTL：系统频率控制寄存器(地址:2FH)

	bit7							bit0
复位值 0011 ----	CLKOE	IRCS2	IRCS1	IRCS0	-	-	-	-
	R/W	R/W	R/W	R/W	U	U	U	U

图注：R＝可读　W＝可写　P＝可编程　U＝未使用
-＝读为0　x＝状态未知

P0（P0 口状态寄存器）

寄存器P0：P0口状态寄存器(地址：05H)

	bit7							bit0
复位值 --xx xxxx	-	-	P05	P04	P03	P02	P01	P00
	U	U	R/W	R/W	R/W	R/W	R/W	R/W

P1（P1 口状态寄存器）

寄存器P1：P1口状态寄存器(地址：07H)

	bit7							bit0
复位值 --xx xxxx	-	-	P15	P14	P13	P12	P11	P10
	U	U	R/W	R/W	R/W	R/W	R/W	R/W

TR0（P0 方向控制寄存器）

寄存器TR0：P0口方向控制寄存器(地址：25H)

	bit7							bit0
复位值 --11 1111	-	-	TR05	TR04	TR03	TR02	TR01	TR00
	U	U	R/W	R/W	R/W	R/W	R/W	R/W

TR1（P1 口方向控制寄存器）

TR1：P1口方向控制寄存器(地址：27H)

	bit7							bit0
复位值 --11 1111	-	-	TR15	TR14	TR13	TR12	TR11	TR10
	U	U	R/W	R/W	R/W	R/W	R/W	R/W

P0LR（P0 口输出锁存控制寄存器）

寄存器P0LR：P0口输出锁存寄存器(地址：45H)

	bit7							bit0
复位值 --xx xxxx	-	-	P0LR5	P0LR4	P0LR3	P0LR2	P0LR1	P0LR0
	U	U	R/W	R/W	R/W	R/W	R/W	R/W

P1LR（P1 口输出锁存控制寄存器）

寄存器P1LR: P1口输出锁存寄存器(地址: 47H)

	bit7							bit0
复位值 --xx xxxx	-	-	P1LR5	P1LR4	P1LR3	P1LR2	P1LR1	P1LR0
	U	U	R/W	R/W	R/W	R/W	R/W	R/W

OPTR（选择寄存器）

寄存器OPTR: 选择寄存器(地址: 21H)

	bit7							bit0
复位值 1111 1111	$\overline{PUPH}$	INT0SE	T0CS	T0SE	PSA	PS2	PS1	PS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

T1CTL（T1 控制寄存器）

寄存器T1CTL: T1控制寄存器(地址: 10H)

	bit7							bit0
复位值 -000 0000	-	T1GC	T1CKS1	T1CKS0	-	$\overline{T1SY}$	T1CS	T1ON
	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W

T2CTL（T2 控制寄存器）

寄存器T2CTL: T2控制寄存器(地址: 12H)

	bit7							bit0
复位值 -000 0000	-	T2CKBS3	T2CKBS2	T2CKBS1	T2CKBS0	T2ON	T2CKPS1	T2CKPS0
	U	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位使用说明：

8 位单片机支持对寄存器的位进行直接的操作，因此在使用的过程中不仅可以通过给寄存器赋值来达到想要的配置，同时还可以直接对位进行操作来达到需要的配置。

以下是对程序中使用到的位进行说明：

T0IE：T0 中断使能位（T1IE、T2IE）

PUIE：外设中断使能位

AIE：总中断使能位

T0IF：T0 中断标志位（T1IF、T2IF）

T1H：T1 寄存器的高 8 位

T1L：T1 寄存器的低 8 位

PP3：PWM3 周期寄存器

T2ON：T2 使能位

T0：T0 寄存器（T1、T2）

T2CCR：T2 触发 AD 启动寄存器

TMR 定时器样例程序框图

注：定时器使用时一定要赋初值。

TMR0 定时器样例简述：

开发环境：ChipON IDE

功能简述：使用 T0 定时器，使得 LED2 每 1 秒亮灭状态变化一次。

硬件连接：连接 JP2

TMR1 定时器样例简述：

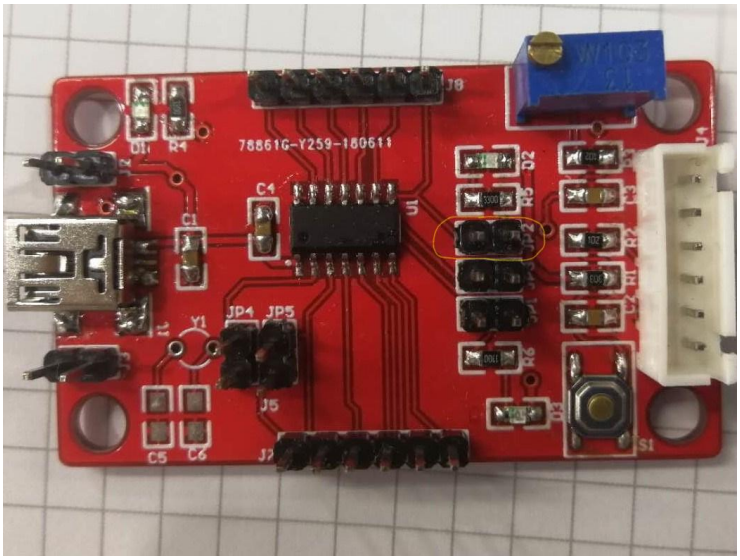
功能简述：使用 T1 定时器，使得 LED2 每 1 秒亮灭状态变化一次。

硬件连接：连接 JP2

TMR2 定时器样例简述：

功能简述：使用 T2 定时器，使得 LED2 每 1 秒亮灭状态变化一次。

硬件连接：连接 JP2（将黄色框中的插针用跳线帽连接）



TMR0 定时器样例程序：

MCU 初始化：

```
void Init_fun()
{
    OSCCTL = 0x70;                              //设置为8M
    /*********端口初始化***********/

     //P07 NC  P06 NC     P05 PP05 P04 PP04
     //P03 S1    P02 NC   P01 NC     P00 NC
     P0 = 0X08;
     P0LR = 0X08;
     TR0 = 0X08;                                //P03输入 S1按键

     //P17 NC  P16 NC     P15 NC       P14 NC
     //P13 NC    P12 LED   P11 LED     P10 NC
     P1 = 0X06;
     P1LR = 0X06;                         //默认P12口指示灯亮，采样通道为AN11
     TR1 = 0;

     /****初始化定时器1*****/
     OPTR=0x43;                       // 使能总弱上拉，脚需要前面单独配置
16分频的计时。16M可记1024us  500us C5  250us C4 125us C3
     T0=0X00;
     T0IF=0;
     T0IE=1;

     AIE = 1;
     PUIE = 1;
}
```

中断函数：

```
void INT_FUN0() __interrupt (0)
{
    if(T0IF)
    {
        time_1s++;
        T0IF=0;
        T0=0X00;
        if(time_1s>=250)
        {
            time_1s = 0;
            LED=!LED;
        }
    }
}
```

主函数：

```
void main()
{
     time_1s = 0;
     Init_fun();
     P1LR2 = 1;
     TR12 = 0;
     while(1)
     {
          _NOP();
     }
}
```

TMR1 定时器样例程序：

MCU 初始化：

```
void Init_fun()
{
    OSCCTL = 0x70;                              //设置为4M
    /*********端口初始化**********/

     //P07 NC  P06 NC     P05 PP05 P04 PP04
     //P03 S1    P02 NC   P01 NC     P00 NC
     P0 = 0X08;
     P0LR = 0X08;
     TR0 = 0X08;                                //P03输入 S1按键

     //P17 NC  P16 NC     P15 NC       P14 NC
     //P13 NC    P12 LED   P11 LED     P10 NC
     P1 = 0X06;
     P1LR = 0X06;                         //默认P12口指示灯亮，采样通道为AN11
     TR1 = 0;


     /****初始化定时器1*****/
     T1CTL = 0X01;            //设置定时器1，预分频为1:1，则计数1为1us
     T1H = 0xec;
     T1L = 0x78;                    //设定定时时间为5ms，  此时的0XEC78=60536
经过5000计数后为65536溢出中断
     PUIE = 1;
     T1IF = 0;                //清定时器1的标志位
     T1IE = 1;                //定时器1使能
     AIE = 1;                 //总中断
}
```

中断函数：

```
void INT_FUN0() __interrupt (0)
{
     if(T1IF)
     {
          T1IF = 0;                                    //清零中断标志
          T1H = 0xec;                                       //T1计数器重新赋值
          T1L = 0x78;
          if(time_1s++ >= 200)                          //判断是否达到1s记时
5ms*200 = 1s
          {
               time_1s = 0;                           //清零计数器
               P1LR2 = !P1LR2;
          }
     }
}
```

主函数：

```
void main()
{
      time_1s = 0;
      Init_fun();
      P1LR2 = 1;
      TR12 = 0;
      while(1)
      {
            _NOP();
      }
}
```

TMR2 定时器样例程序：

MCU 初始化：

```
void Init_fun()
{
    OSCCTL = 0x40;                          //设置为4M
    /*********端口初始化***********/

     //P07 NC  P06 NC     P05 PP05 P04 PP04
     //P03 S1    P02 NC   P01 NC     P00 NC
     P0 = 0X08;
     P0LR = 0X08;
     TR0 = 0X08;                            //P03输入 S1按键

     //P17 NC  P16 NC     P15 NC       P14 NC
     //P13 NC    P12 LED   P11 LED     P10 NC
     P1 = 0X06;
     P1LR = 0X06;                      //默认P12口指示灯亮，采样通道为AN11
     TR1 = 0;


     /****初始化定时器1*****/
     T2=0;
     PP3=0XC4;
     T2CTL=0x38;    // 计数源系统时钟，非4分频，设定8分频 暂不启动 可 32.767ms，定时25ms
     T2IF=0;
     T2IE = 1;
     AIE = 1;
     PUIE = 1;
     T2ON = 1;
}
```

中断函数：

```
void INT_FUN0() __interrupt (0)
{
     if(T2IF)
     {
           T2IF = 0;                                  //清零中断标志
           T2 = 0;                                    //T1计数器重新赋值
           PP3=0XC4;
           if(time_1s++ >= 40)                      //判断是否达到1s记时    5ms*200 = 1s
           {
                time_1s = 0;                            //清零计数器
                P1LR2 = !P1LR2;
           }
     }
}
```

主函数：

```
void main()
{
	time_1s = 0;
	Init_fun();
	P1LR2 = 1;
	TR12 = 0;
	while(1)
	{
		_NOP();
	}
}
```

程序说明：

注意事项：

1、当预分频器用于 T0 模块时，所有写入 T0 寄存器的指令都会将预分频器清 0。当预分频器用于 WDT 时，CWDT 指令会同时将预分频器和看门狗定时器清 0。

2、在计数模式下，当不使用预分频器时，要求 TOCK 的高电平状态和低电平状态分别保持至少 2Tosc 的时间，以实现 TOCK 与内部相位时钟的同步。

3、在任何使用到中断函数的程序中，打开相应模块中断的同时也要打开 AIE（总中断），如需要设置的为外设模块时还需要使能外设中断（PUIE）。

4、T2 只能作为定时器，因此是没有外部计数时钟输入脚。