

KF8F2320——CMP2 样例程序

引言

本应用笔记提供了 KF8F2320—CMP2 比较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)

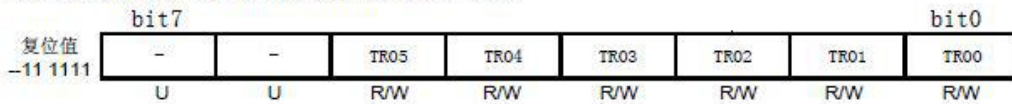
寄存器OSCCTL: 系统频率控制寄存器(地址:2FH)



图注: R = 可读 W = 可写 P = 可编程 U = 未使用
 - = 读为0 x = 状态未知

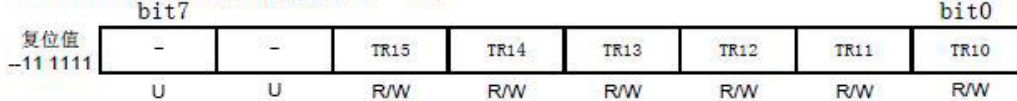
TR0 (P0 方向控制寄存器)

寄存器TR0: P0口方向控制寄存器(地址: 25H)



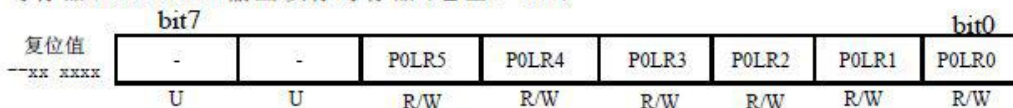
TR1 (P1 口方向控制寄存器)

TR1: P1口方向控制寄存器(地址: 27H)



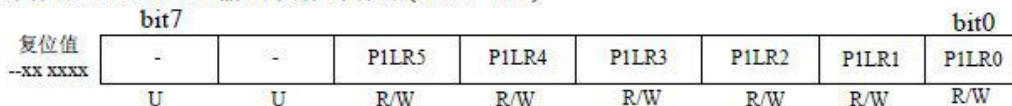
POLR (P0 口输出锁存控制寄存器)

寄存器POLR: P0口输出锁存寄存器(地址: 45H)



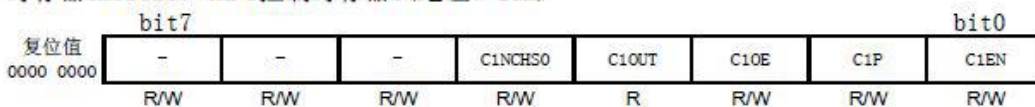
P1LR (P1 口输出锁存控制寄存器)

寄存器P1LR: P1口输出锁存寄存器(地址: 47H)



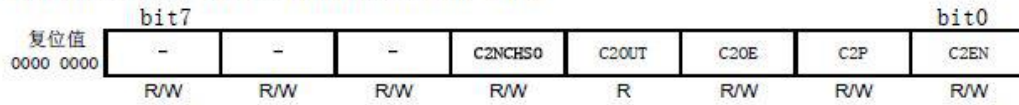
CM1CTL0 (CMP1 控制寄存器 0)

寄存器CM1CTL0: CMP1控制寄存器0(地址: 19H)



CM2CTL0 (CMP2 控制寄存器 0)

寄存器CM2CTL0: CMP2控制寄存器0(地址: 1AH)



位使用说明:

ANS<7:0>: 将对应的引脚配置为需要的模式 (数字口/模拟口/特殊功能脚)

VREEN: 内部参考电压使能位

C2IE: CMP2 中断允许位

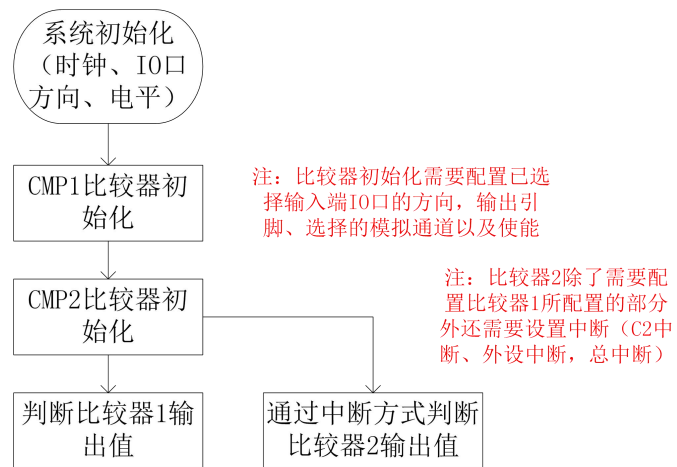
PUIE: 外设中断使能位

AIE: 总中断使能位

C2IF: CMP2 中断标志位

C1OUT: 比较器 1 输出

CMP 样例程序框图



注： 比较器 1 输出值的判断方式为条件判断，比较器 2 输出值的判断方式为设置相应中断，在中断中进行输出值的判断，用户使用时可以自行选择不同的方式来进行输出值的判断。

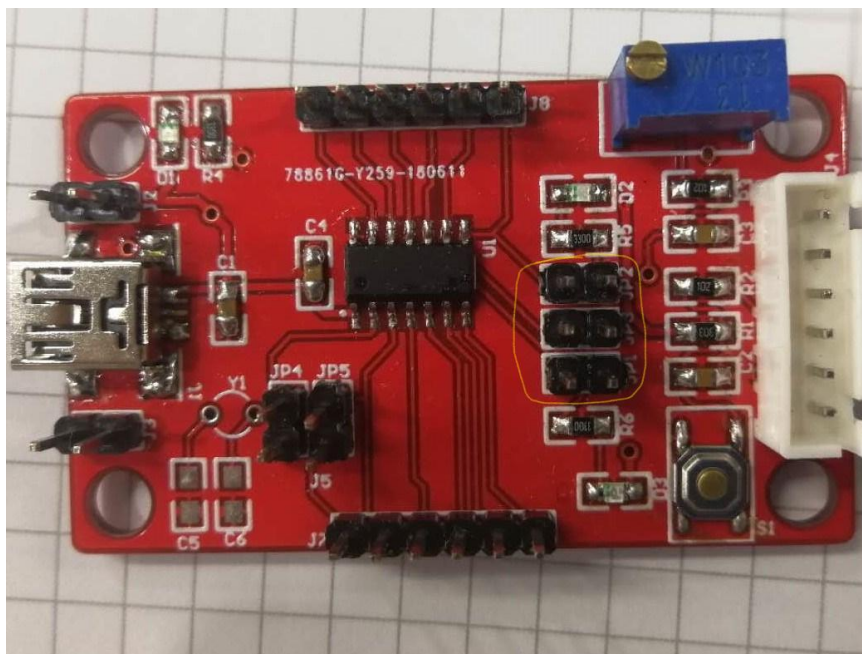
CMP2 比较器样例简述:

开发环境: ChipON IDE

功能简述: 比较器 1 负端配置为 I/O 口, 轮询比较器的结果并在 P02 输出比较结果。比较器 2 负端配置为内部参考电压, 比较器的结果发生变化, 触发中断, 比较结果在 P14 输出。

实验现象: 将 P10 和电位器的输出脚相连, 调节电位器, 如果电位器的电压大于 1.7V, LED2 点亮, P14 输出高电平。反之, P14 输出低电平。P00 接 VDD、P01 接 GND, 比较器 1 的输出脚 P02 将输出高电平, LED3 点亮。P00 接 GND、P01 接 VDD, 比较器 1 的输出脚 P02 将输出高电平, LED3 熄灭。

硬件连接: 连接 JP1、JP2、JP3 (用跳线帽连接黄色框里边的三组插针)



CMP2 比较器样例程序:

MCU 初始化:

```
void Init_fun()
{
    OSCCTL = 0x30;    //设置为500K

    TR12 =0;
    LED2=0;
    TR11=0;
    LED3=0;
}
```

CMP1 初始化:

```
void CMP1_Init()
{
    TR00=1;//比较器1的正向输入端—P00配置为输入模式
    ANS0=1;//打开P00的模拟通道
    TR01=1;//比较器1的负向输入端—P01配置为输入模式
    ANS1=1;//打开P01的模拟通道
    TR02=0;//配置比较器1的输出引脚,
    POLR2=0;
    CM1CTL0 = 0x0D;//比较器1的负端输入选择P01、输出使能、不反相、比较器使能
}
```

CMP2 初始化:

```
void CMP2_Init()
{
    TR10=1;//比较器2的正端输入—P10配置为输入模式
    ANS4 = 0;//P10对应的模拟通道4打开
    VREEN =1;//使能内部参考电压
    TR14 = 0;//配置比较器2的输出引脚,
    P1LR4 = 0;    //
    CM2CTL0 = 0x1D;//比较器2的负端输入选择内部参考电压、输出使能、不反相、比较器使能
    C2IE =1;
    PUIE=1;
    AIE=1;
}
```

中断函数:

```
void int_fun0() __interrupt (0)
{
// 比较器中断
    if(C2IF)
    {
        C2IF=0;
        LED2 = !C2OUT;
    }
}
```

主函数:

```
void main()
{
    Init_fun();//初始化晶振和公用IO口
    CMP1_Init();//比较器1的初始化
    CMP2_Init();//比较器2的初始化

    while(1)
    {
        if(C1OUT)
        {
            LED3 = 0;
        }
        else
        {
            LED3 = 1;
        }
    }
}
```

程序说明：比较器 1 采用条件判断的方式在主函数中实现，比较器 2 使用中断函数完成比较值的判断。

注意事项:

- 1、当电位器的电压值处于 1.7V 左右波动时，LED 灯会出现不断闪烁的状况，此现象为正常状况，因为比较器所得到的电压值也是有波动的就会影响比值。
- 2、对于字节的连续读写，只需要使用数组和循环语句来重复操作单字节的读写就可以达到目的。