

# KF8F3132——PWM 模块样例程序

## 引言

本应用笔记提供了 KF8F3132—PWM 模块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 KF8F3132 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

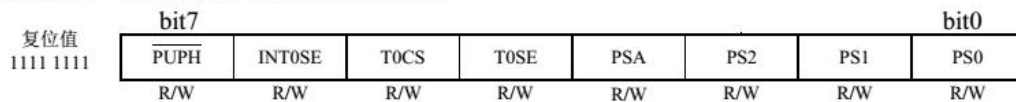
寄存器OSCCTL: 系统频率控制寄存器(地址:2FH)



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

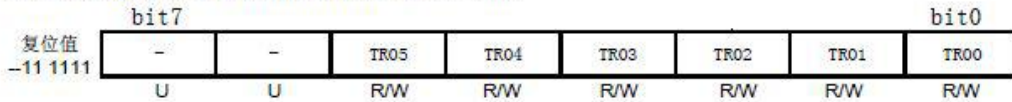
OPTR: 选择寄存器

寄存器6.1: OPTR: 选择寄存器(地址: 21H)



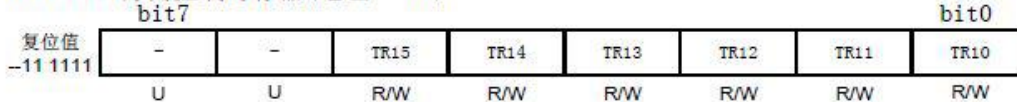
TR0: P0 方向控制寄存器

寄存器TR0: P0口方向控制寄存器(地址: 25H)



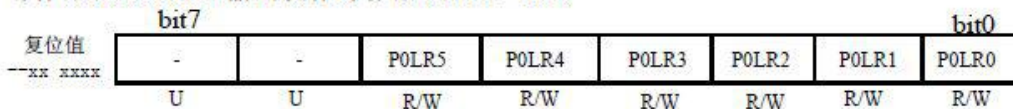
TR1: P1 口方向控制寄存器

TR1: P1口方向控制寄存器(地址: 27H)



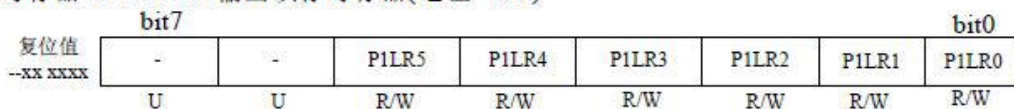
POLR: P0 口输出锁存控制寄存器

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

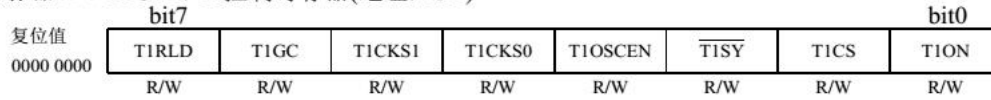
寄存器P1LR: P1口输出锁存寄存器(地址: 47H)



T0: 定时/计数器 1 寄存器

T1CTL: T1 控制寄存器

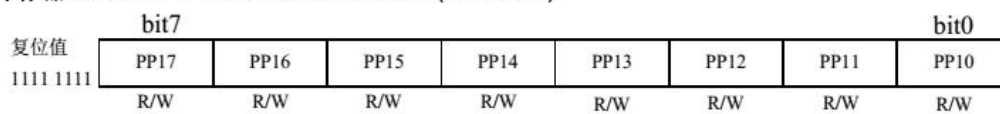
**寄存器6.2: T1CTL: T1控制寄存器(地址: 10H)**



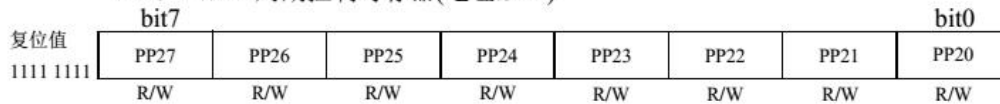
PP1: PWM1 周期寄存器

PP2: PWM2 周期寄存器

**寄存器8.2: PP1: PWM1周期控制寄存器(地址:16H)**



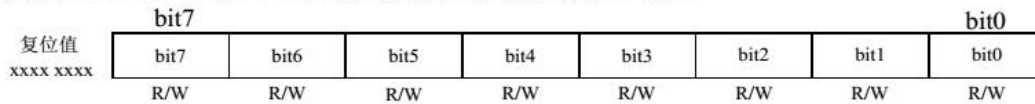
**PP2: PWM2周期控制寄存器(地址:32H)**



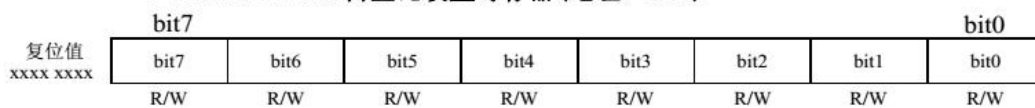
PWM1L: PWM1 占空比设置寄存器

PWM2L: PWM2 占空比设置寄存器

**寄存器8.3: PWM1L: PWM1占空比设置寄存器(地址: 13H)**



**PWM2L: PWM2占空比设置寄存器(地址: 33H)**



位操作使用说明:

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:

T1CLKEN: T1 定时模式时钟源选择位

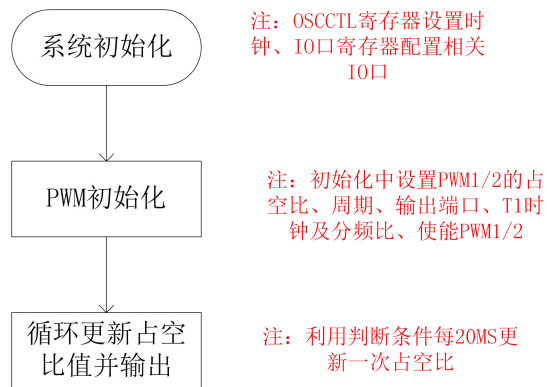
T0IF:T0 中断标志位

PWMPIN: PWM1/2 模块输出引脚选择位

PWM1ON: PWM1 启动控制位

PWM2ON: PWM2 启动控制位

## PWM 样例程序框图



注：PWM 的波形可以通过示波器查看配置的输出端口来检测正确性。

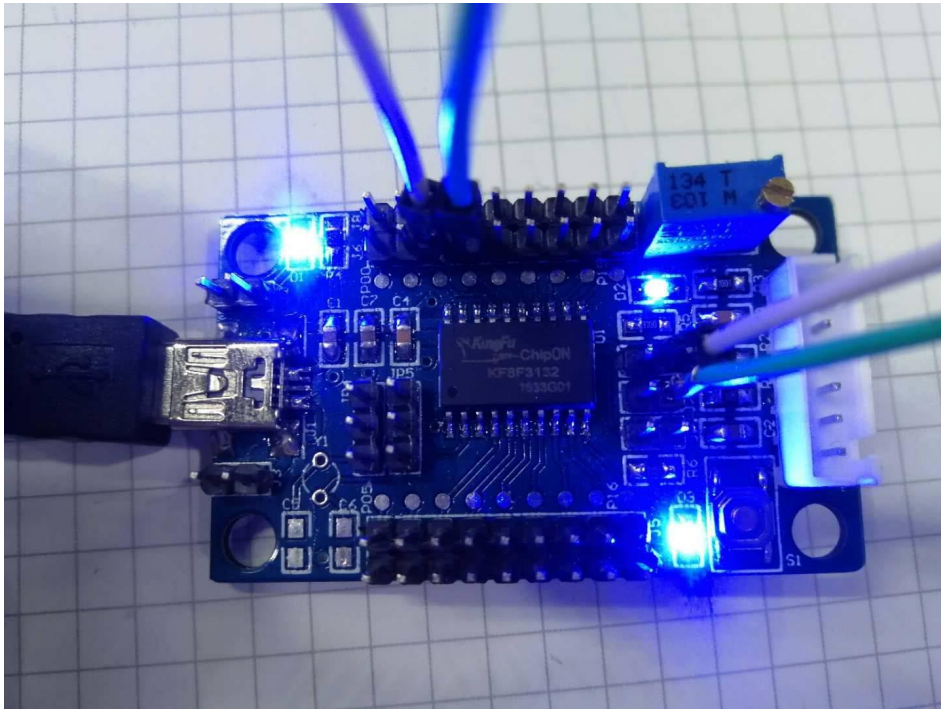
程序中用来更新占空比条件的代码可以查看后边的样例程序。

## PWM 样例简述:

开发环境: ChipON IDE

功能简述: PWM1 和 PWM2 输出占空比互补的 PWM 波形, 频率是 10K, 输出脚是 P02, P10。每隔 20ms 占空比变化 1%, 满占空比输出后, 更新占空比为 0, 依次循环。

硬件连接: P02 接 LED2 的 JP2 端, P10 接 LED3 的 JP3 端。



图为 PWM 硬件连接循环点亮 LED 灯实物图。

PWM 样例程序:

MCU 初始化:

```
void Mcu_Init()
{
    OSCCTL =0x70;//系统时钟16M

    TR02 =0;//PWM配置为输出口
    P0LR2=0;

    TR10 =0;//PWM配置为输出口
    P1LR0 =0;
}
```

T0 初始化:

```
void T0_Init()
{
    OPTR =0x03;//T0定时模式，16分频
}
```

延时函数: (使用 T0 定时模式做延时)

```
void Delay_ms(unsigned int k)
{
    unsigned int i;

    for(i=0;i<k; i++)
    {
        T0=0;
        T0IF=0;
        while(!T0IF);
    }
}
```

PWM 初始化:

```
void PWM_Init()
{
    T1CLKEN =0;//T1时钟源选择Sysclk/4
    T1CTL =0x21;//T1分频器4分频，T1使能
    PP1=99;//PWM1周期设置为100us
    PP2=99;//PWM2周期设置为100us
    PWM1L=0;//占空比为0
    PWM2L=0;//占空比为0
    PWMPIN=0;//P20,P10设为PWM输出口。
    PWM1ON =1;
    PWM2ON =1;
}
```

主函数:

```
void main()
{
    unsigned char i=0;
    Mcu_Init();
    T0_Init();
    PWM_Init();

    while(1)
    {
        Delay_ms(20);
        if(i>99)//占空比循环变化
        {
            i=0;
        }
        else
        {
            i++;
        }

        PWM1L=i;//占空比更新
        PWM2L =100-i;
    }
}
```



## 注意事项:

1、程序中所使用到 T0 定时器是用来做延时用的。因此未出现在框图流程中。

2、任何复位都会将所有端口强制为输入模式，并强制 PWM1/2 使用的寄存器进入复位状态。

3、PWM 周期计算方式：(PP1/2 是一个 8 位寄存器，其值可以设置为 0~255)

$$\star \text{式 8.1: } \text{PWM}_x \text{周期} = (\text{PP}_x + 1) \cdot \text{Tpwm} \quad (x=1, 2)$$

注：当 T1CLKEN=0 时  $\text{Tpwm} = 4 \cdot \text{T}_{\text{sys}} \cdot (\text{T1 预分频比})$

当 T1CLKEN=1 时  $\text{Tpwm} = \text{T}_{\text{INTHF}} \cdot (\text{T1 预分频比})$

4、PWM1/2 占空比计算方式:

$$\star \text{式 8.2: } \text{脉冲宽度} = \text{PWM}_x \text{L} \cdot \text{Tpwm} \quad (x=1, 2)$$

注：当 T1CLKEN = 0 时， $\text{Tpwm} = 4 \cdot \text{T}_{\text{sys}} \cdot (\text{T1 预分频比})$

当 T1CLKEN = 1 时， $\text{Tpwm} = \text{T}_{\text{INTHF}} \cdot (\text{T1 预分频比})$

$$\star \text{式 8.3: } \text{占空比} = \frac{\text{脉冲宽度}}{\text{PWM 周期}} = \frac{\text{PWM}_x \text{L}}{\text{PP}_x + 1} \quad (x=1, 2)$$

5、PWM1/2 分辨率计算方式:

$$\star \text{式 8.4: } \text{分辨率} = \frac{\text{Log}[(\text{PP}_x + 1)]}{\text{log}2} \text{ 位} \quad (x=1, 2)$$

6、在休眠模式下，T1 寄存器将不会递增并且模块状态保持不变。PWM1/2 输出引脚电平保持不变（如果输出为高电平，则保持高电平，如果为低电平保持低电平）。当器件被唤醒时，T1 将从原来的状态继续工作。

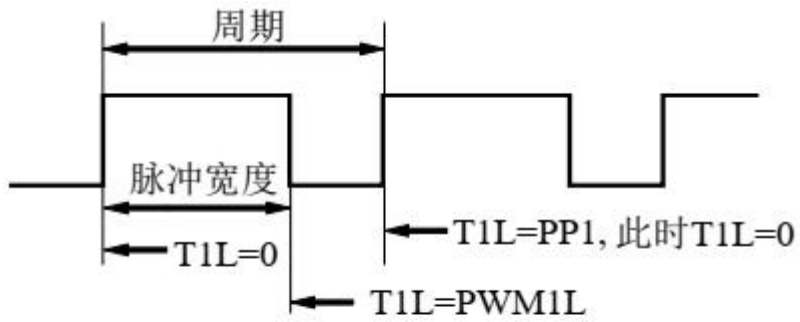
7、PWM1/2 输出引脚可以通过 PINSET 寄存器（156H）的 PWMPIN 位选择:

PWMPIN: PWM1/2 模块输出引脚选择位

0 = P0.2 作为 PWM1 的输出引脚, P1.0 作为 PWM2 的输出引脚

1 = P0.0 作为 PWM1 的输出引脚, P0.1 作为 PWM2 的输出引脚

## 8、PWM 输出波形图



T1L=PWM1L时PWM1输出变为低电平, T1L=PP1时PWM1输出变为高电平, 同时将T1L清零。