

KF8F3132——CCP 模块

引言

本应用笔记提供了 KF8F3132—CCP 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)

复位值 0011	bit7							bit0
	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

OPTR: 选择寄存器

寄存器6.1: OPTR: 选择寄存器(地址: 21H)

复位值 1111 1111	bit7							bit0
	PUPH	INT0SE	T0CS	T0SE	PSA	PS2	PS1	PS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TR1: P1 口方向控制寄存器 (输入/输出)

寄存器2.9: TR1: P1口方向控制寄存器(地址: 27H)

复位值 1111 1111	bit7							bit0
	TR17	TR16	TR15	TR14	TR13	TR12	TR11	TR10
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TR2: P2 口方向控制寄存器 (输入/输出)

寄存器2.13: TR2: P2口方向控制寄存器(地址: 26H)

复位值 1111 1111	bit7							bit0
	-	-	-	-	TR23	TR22	TR21	TR20
	R/W	R/W	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2.8: P1LR: P1口输出锁存寄存器(地址: 47H)

复位值 xxxx xxxx	bit7							bit0
	P1LR7	P1LR6	P1LR5	P1LR4	P1LR3	P1LR2	P1LR1	P1LR0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

T0: 定时/计数器 0 寄存器

T1H: 定时/计数器 1 高字节寄存器

T1L: 定时/计数器 1 低字节寄存器

T1CTL: T1 控制寄存器

寄存器6.2: T1CTL: T1控制寄存器(地址: 10H)

		bit7						bit0	
复位值 0000 0000		T1RLD	T1GC	T1CKS1	T1CKS0	T1OSCEN	$\overline{\text{T1SY}}$	T1CS	T1ON
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PP1: PWM1 周期控制寄存器 PP2: PWM2 周期控制寄存器

寄存器8.2: PP1: PWM1周期控制寄存器(地址:16H)

		bit7						bit0	
复位值 1111 1111		PP17	PP16	PP15	PP14	PP13	PP12	PP11	PP10
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PP2: PWM2周期控制寄存器(地址:32H)

		bit7						bit0	
复位值 1111 1111		PP27	PP26	PP25	PP24	PP23	PP22	PP21	PP20
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CCPCTL: 捕捉/比较控制寄存器

寄存器9.1: CCPCTL: 捕捉/比较控制寄存器(地址: 152H)

		bit7						bit0	
复位值 ---0 0000		-	-	-	CCPPIN	CCMOD3	CCMOD2	CCMOD1	CCMOD0
		U	U	U	R/W	R/W	R/W	R/W	R/W

CCRH: 捕捉/比较寄存器高 8 位

CCRL: 捕捉/比较寄存器低 8 位

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

CCPIF: CCP 中断标志位

CCPIE: CCP 中断使能位

T1IF: T1 定时器中断标志位

T1IE: T1 定时器中断使能位

PUIE: 外设中断使能位

AIE: 总中断使能位

CCP 捕捉模式:

开发环境: chipon IDE

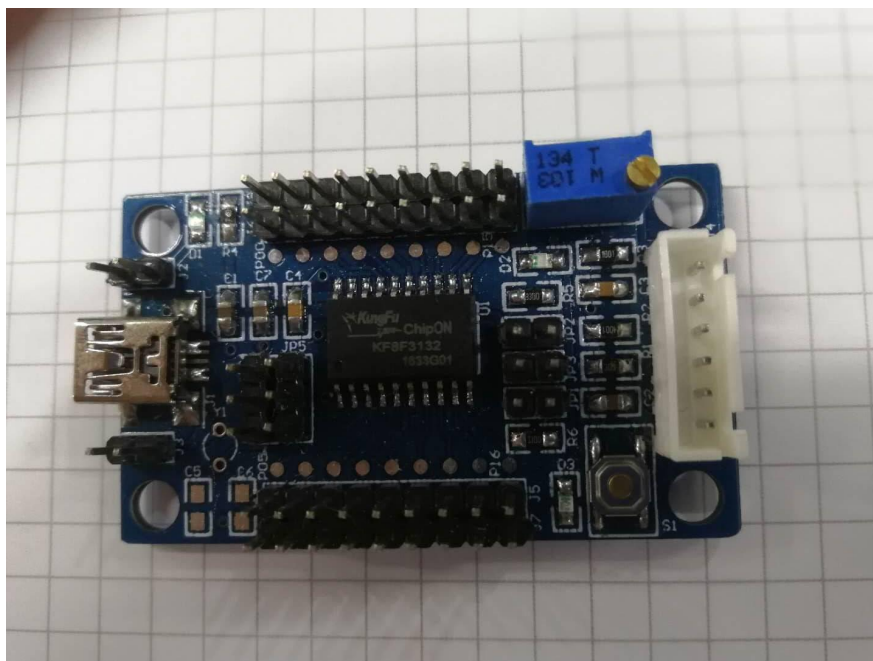
功能简述: 使用 CCP 的捕捉模式, P20 每个上升沿时, CCRH 和 CCRL 获取 T1 的值, P10 电平翻转。也可以在调试环境下查看 cnt_0 的值, 既 PWM 信号的周期值。

硬件连接: P20 接 PWM 信号

CCP 比较模式:

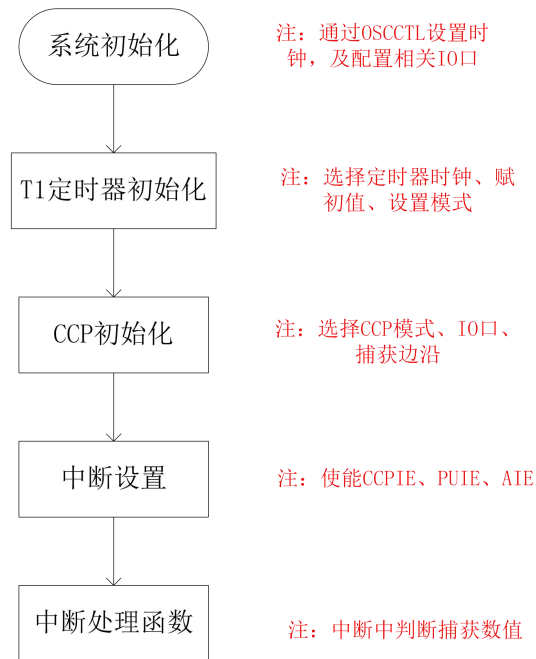
功能简述: 使用 CCP 的比较模式, 模拟产生 PWM 波形, 用于驱动 LED2 形成呼吸灯效果。利用 T1 的重载功能产生 PWM 的周期, CCP 的比较功能产生占空比。T1 的重载和 CCP 的匹配时间产生中断, 在中断里切换 IO 口的电平, 以模拟 PWM。

硬件连接: P10 接 LED2 的 JP2

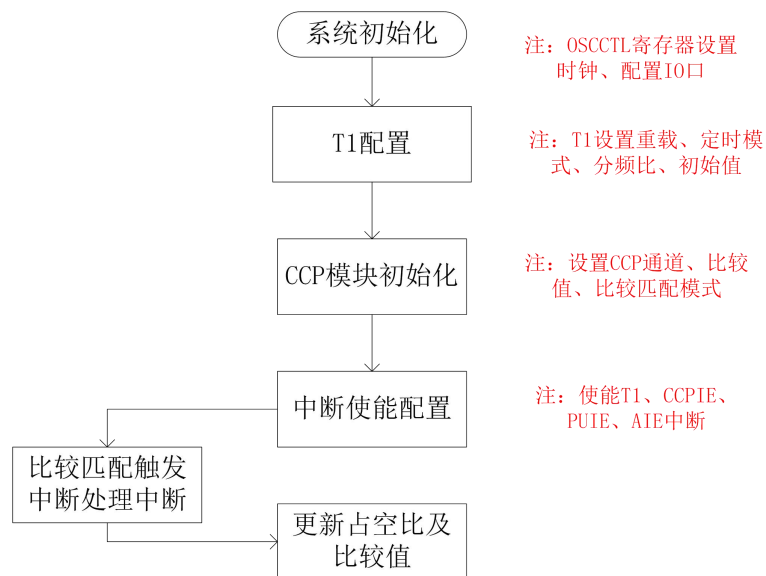


图片为 KF8F3132 DEMO 板。

CCP 捕捉模式样例流程图：



CCP 比较模式样例流程图：



注：1、T1 定时器设置重载模式，可以调整重载值。

2、CCP 的比较值可以自行的改变，如果需要进行操作可在匹配发生后再改变。样例程序中有更新比较值的设置，可进行参考。

CCP 捕捉模式样例程序:

时钟及 IO 口初始化部分:

```
void Init_mcu()
{
    OSCCTL=0x70;//16M晶振

    TR20 =1;//捕捉的CCP管脚设置为输入

    TR10=0;
    P1LR0=0;
}
```

T1 定时器初始化:

```
void Init_T1()
{
    T1CLKEN =0;//时钟源选择系统时钟4分频
    T1H=0;
    T1L=0;
    T1CTL =0x01;//T1分频器不分频, T1定时模式, 使能T1
}
```

CCP 模块初始化:

```
void Init_CCP()
{
    // CCPCTL=0x04;//P20做为CCP通道, 捕获模式, 下降沿捕获
    CCPCTL=0x05;//P20做为CCP通道, 捕获模式, 上升沿捕获
}
```

中断设置函数:

```
void Init_intrpt()
{
    CCPIE=0;
    CCPIE=1;

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

中断执行函数:

```

void int_fun0() __interrupt (0)
{
    if(CCPIF) //上升沿发生捕捉
    {
        CCPIF=0;
        T1H=0;
        T1L=0;
        cnt_0=CCRH;
        cnt_0 =cnt_0<<8;
        cnt_0 +=CCRL;
        P1LRC=!P10;
    }
}

```

主函数:

```

void main()
{
    cnt_0=0;

    Init_mcu();
    Init_T1();
    Init_CCP();
    Init_intrpt();

    while(1)
    {
    }
}

```

CCP 比较模式样例程序:

T0 初始化:

```

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

```

MS 延时函数: (使用 T0 计数器)


```

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

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

```

MCU 初始化:

```

void Init_mcu()
{
    OSCCTL=0x70;//16M晶振

    TR20 =0;

    TR10=0;
    P1LR0=0;
}

```

T1 初始化配置函数:

```

void Init_mcu()
{
    OSCCTL=0x70;//16M晶振

    TR20 =0;

    TR10=0;
    P1LR0=0;
}

```

CCP 模块初始化:

```

void Init_CCP()
{
    CCPCTL=0x02;//P20做为CCP通道，比较模式，匹配时电平翻转。

    CCRL=0x58;
    CCRH=0x02;//比较值设置为600
}

```

中断设置函数:

```

void Init_intrpt()
{
    T1IF=0;
    T1IE=1;

    CCP1F=0;
    CCP1E=1;

    PU1E=1;
    A1E=1;
}

```

中断处理函数:

```

void int_fun0() __interrupt (0)
{
    if(CCP1F) //比较值匹配时
    {
        CCP1F=0;
        P1LRC=0; //
    }

    if(T1IF) //达到重载值后输出高电平
    {
        T1IF=0;
        P1LRC=1;
    }
}

```

主函数:

```

void main()
{
    unsigned char i=1, j=1;
    unsigned int k=0;

    Init_mcu();
    Init_T1();
    Init_CCP();
    Init_intrpt();
    T0_Init();

    while(1)
    {
        Delay_ms(10);
        if(j) //占空比增加时
        {
            i++;
        }
        else //占空比减少
        {
            i--;
        }

        if((i<2) || (i>99)) //占空比变化方向改变
            j= !j;

        k=i*20;
        CCRH =k>>8; //更新比较值
        CCRL =k&0xFF;
    }
}

```

模块使用注意事项：

1、T1 必须运行在定时模式或同步计数模式下 CCP 模块才能使用捕捉功能。

2、当需要对捕捉模式下 CCP 模块的相关配置进行改变时为避免误操作，应关闭该模块

3、任何的复位条件以及 CCP 模块状态（关闭和改变模式）的改变都会清零预分频计数器，所以需要在改变状态后重新对预分频计数器进行设置。

4、T1CLKEN 必须配置为 0。

5、CCP 选择相应的 IO 口后必须将其设置为输出模式。