

KF8F3132——P0_Intrupt

引言

本应用笔记提供了 KF8F3132—P0_Intrupt 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

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
	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P0: P0 寄存器

寄存器P0: P0口状态寄存器(地址:05H)

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

TR0: P0 口方向控制寄存器 (输入/输出)

寄存器TR0: P0口方向控制寄存器(地址:25H)

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

TR1: P1 口方向控制寄存器 (输入/输出)

TR1: P1口方向控制寄存器(地址:27H)

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

T0: 定时/计数器 0 寄存器

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：

TOIF：T0 中断标志位

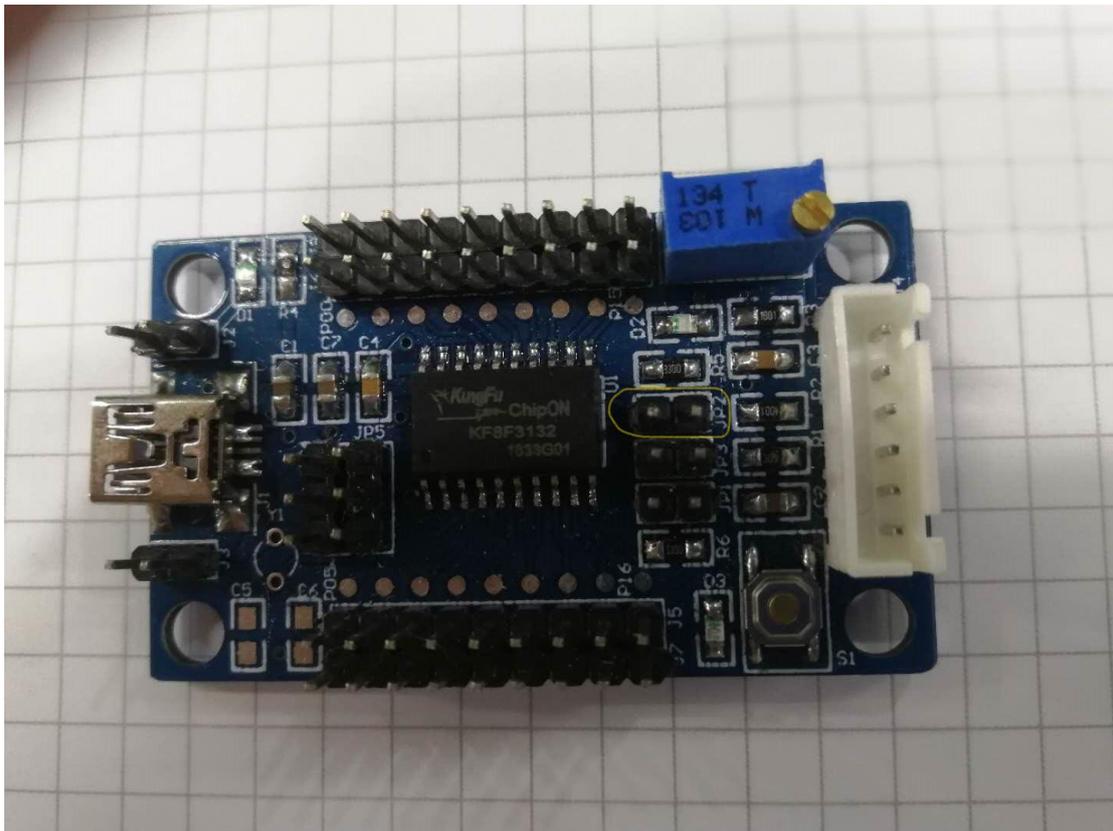
POIE：P0 中断使能位

AIE： 总中断使能位

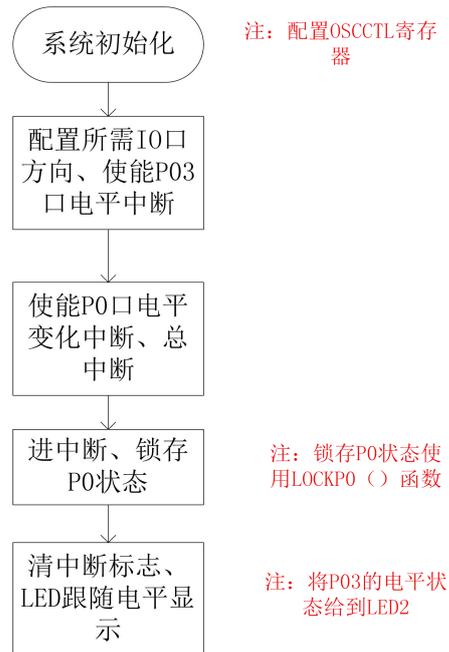
开发环境: chipon IDE

功能简述: 使用 P0 口的电平变化中断, 控制 LED2 的亮灭。按下 S1, LED2 点亮, 松开 S1, LED2 熄灭。

硬件说明: JP2 需带上跳线帽 (图中黄色框中的插针接上跳线帽)



P0_Intrupt 样例流程图:



注: 上电之后需要消抖操作之后再去检测 P0 口电平状态, 以防误触发电平中断。

样例程序如下：

时钟及 I/O 口初始化部分：

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

    TR03 =1; //P03设置为输入
    IOCL3=1; //使能P03的电平变化中断

    TR1=0;
    P1LR2 =1;

    OPTR =0x03; //T0 16分频，单计数周期是4us
}
```

延时函数：

```
void Delay_ms(uint j)
{
    uint k=0;
    for(k=0;k<j;k++)
    {
        T0 =6;
        T0IF=0;
        while(!T0IF);
    }
}
```

中断函数：

```
void int_fun0() __interrupt (0)
{
    if(P0IF)
    {
        LOCKP0(); //锁存P0口的状态
        P0IF=0; //清零中断标志量
        P1LR2 =P03; //让LED控制口的状态跟随P03的电平
    }
}
```

主函数：

```
void main()
{
    Init_mcu();
    Delay_ms(200); //上电等待P03接口电路稳定
    POIE=1; //打开P0电平变化中断
    AIE =1; //使能总中断
    while(1)
    {

    }
}
```