

KF8F3132——硬件除法器

引言

本应用笔记提供了 KF8F3132—硬件除法器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。

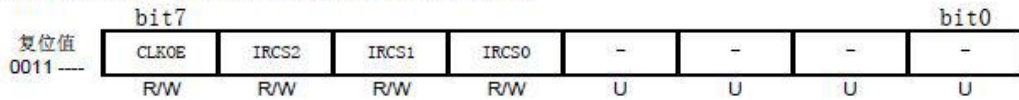
本应用笔记须与 KF8F3132 数据手册结合使用。

寄存器

寄存器使用说明：

OSCCTL： 系统控制寄存器

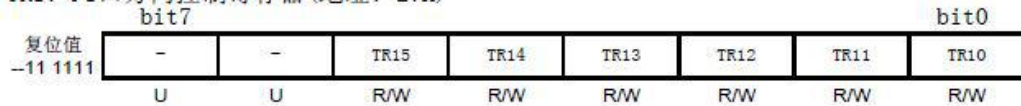
寄存器OSCCTL：系统频率控制寄存器(地址:2FH)



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

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)



DIVAH：被除数高 8 位寄存器

DIVAL：被除数低 8 位寄存器

DIVB：除数寄存器

DIVQH：商高 8 位寄存器

DIVQL：商低 8 位寄存器

DIVR：余数寄存器

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：

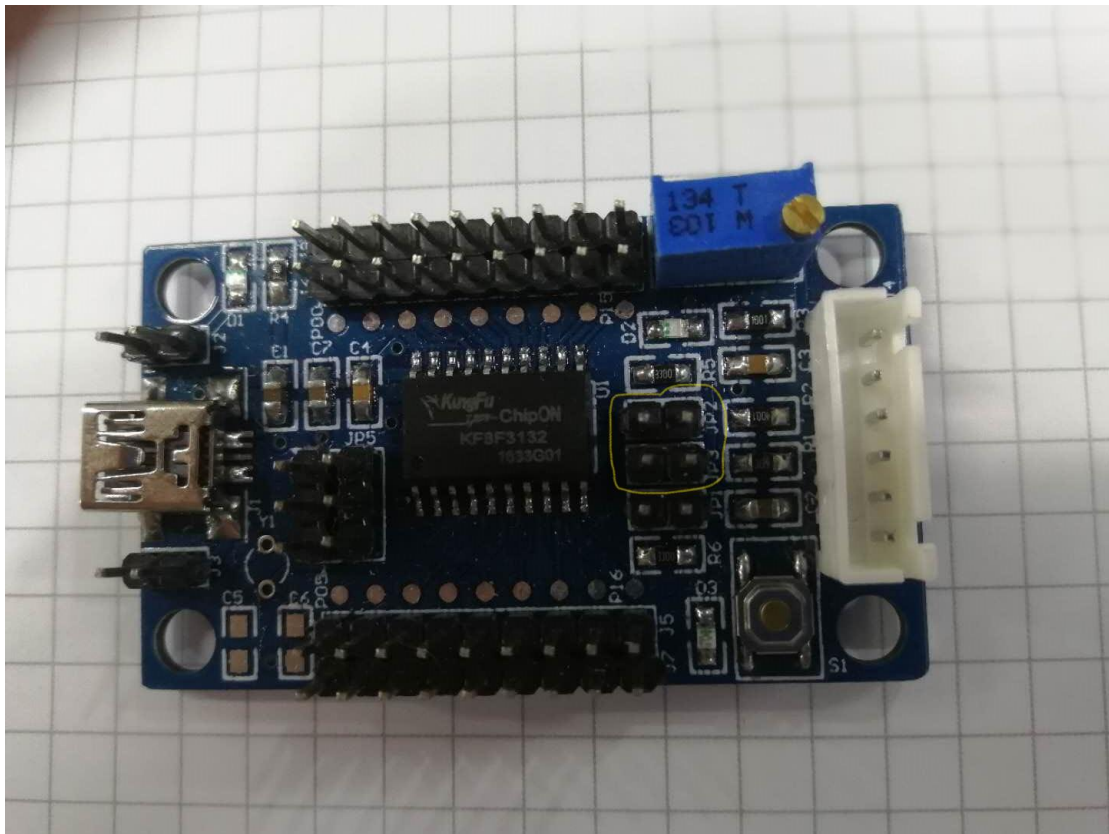
DIVEN: 除法运算使能标志

DIVOEN: 除法运算完成标志

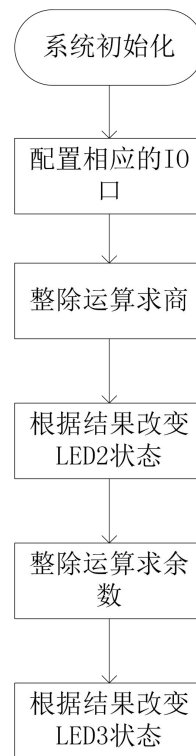
开发环境: chipon IDE

功能简述: 使用硬件除法器, 进行整数除法运算, 并使用计算结果驱动 LED2、LED3。

硬件说明: JP2、JP3 需带上跳线帽 (图中黄色框中的插针接跳线帽)



硬件除法器样例流程图：



注：配置IO口的电平状态也就是确定开始时LED灯的亮灭

注：除法运算中除数和最后的余数都是 8 位的值，被除数为 16 位值。

且都为无符号除法。

样例程序如下：

时钟及 IO 口初始化部分：

```
void Init_mcu()
{
    OSCCTL =0x70; //时钟配置为16M

    TR1=0; //IO配置为输出
    LED2 =1;
    LED3=1;
}
```

16 除法整除结果函数：

```
uint Get_div(uint NumA, uchar Numb)
{
    uint i=0;
    DIVAH=NumA>>8;
    DIVAL =NumA&0xFF;
    DIVB=Numb;
    DIVEN=1;
    while(!DIVOEN);
    DIVEN=0;
    i +=DIVQH;
    i <<=8;
    i +=DIVQL;
    return i;
}
```

16 位除法余数返回函数：

```

uchar Get_rem(uint NumA, uchar Numb)
{
    uchar i=0;
    DIVAH=NumA>>8;
    DIVAL =NumA&0xFF;
    DIVB=Numb;
    DIVEN=1;
    while(!DIVOEN);
    DIVEN=0;
    i=DIVR;
    return i;
}

```

主函数:

```

void main()
{
    uint k=0;
    Init_mcu();
    while(1)
    {
        k++;

        Value=Get_div(k, 100); //整除运算求取商
        if(Value>50) //商的结果决定LED2的输出
            LED2=0;
        else
            LED2=1;

        rem= Get_rem(k, 100); //整除运算求取余数
        if(rem>70) //余数的结果决定LED3的输出
            LED3=0;
        else
            LED3=1;
    }
}

```

模块使用注意事项：

1、DIVOEN 不能直接由软件清 0。对 DIVAH/DIVAL 或 DIVB 进行赋值操作会使 DIVOEN 硬件清 0，DIVEN 软件置 1 使能除法运算，当操作已经完成，DIVOEN 被硬件置 1，DIVEN 会被硬件清 0。