F28069M_外部中断

首先

> *今天开头没有鸡汤，直接开始进入正题。*

代码秀一波

```
#include "DSP28x_Project.h"

void gpio_setup(void);                  //端口初始化函数
interrupt void xint1_isr(void);         //中断函数
int a = 0;


void main()
{
    InitSysCtrl();
    DINT;
    InitPieCtrl();
    IER=0x0000;
    IFR=0x0000;
    InitPieVectTable();
    EALLOW;                             //使能写操作
    PieVectTable.XINT1=&xint1_isr;      //服务子函数存入相应向量地址
    EDIS;                               //禁止写操作
    PieCtrlRegs.PIECTRL.bit.ENPIE = 1;  //除Reset复位向量外的所有向量都从PIE向量列表
中提取
    PieCtrlRegs.PIEIER1.bit.INTx4 = 1;  //分配XINT1为PIE组1第4位
    IER |= M_INT1;                      //使能INT1中断
    EINT;                               //使能全局中断INTM
    gpio_setup();
    while(1)
    {
        //等待中断产生
    }

}

void gpio_setup(void)
{
    //初始化GPIO，上篇已详细介绍
    EALLOW;

    GpioCtrlRegs.GPBMUX2.bit.GPIO54=0;
```

```
    GpioCtrlRegs.GPBPUD.bit.GPIO54=0;
    GpioCtrlRegs.GPBDIR.bit.GPIO54=1;
    GpioDataRegs.GPBDAT.bit.GPIO54 = 1;
    EDIS;
    EALLOW;
    GpioCtrlRegs.GPAMUX2.bit.GPIO23=0;
    GpioCtrlRegs.GPAPUD.bit.GPIO23=0;
    GpioCtrlRegs.GPADIR.bit.GPIO23=1;
    GpioDataRegs.GPADAT.bit.GPIO23 = 1;
    EDIS;

    //外部中断输入I/O配置
    EALLOW;
    GpioCtrlRegs.GPAMUX2.bit.GPIO21=0;          //设置数字模式
    GpioCtrlRegs.GPADIR.bit.GPIO21=0;           //设置输出模式
    GpioCtrlRegs.GPAQSEL2.bit.GPIO21=0;         //与SYSCLKOUT同步
    GpioCtrlRegs.GPACTRL.bit.QUALPRD3 = 0xff;   //设置采样周期
    EDIS;
    EALLOW;
    GpioIntRegs.GPIOXINT1SEL.bit.GPIOSEL = 0x15;//配置GPIO21位XINT1输入引脚
    EDIS;
    XIntruptRegs.XINT1CR.bit.POLARITY = 0;      //配置为下降沿触发
    XIntruptRegs.XINT1CR.bit.ENABLE = 1;        //使能中断
}



interrupt void xint1_isr(void)                  //中断处理函数
{
    for(a=0;a<100;a++)
    {
    GpioDataRegs.GPADAT.bit.GPIO23=0;
    DELAY_US(100000);
    GpioDataRegs.GPADAT.bit.GPIO23=1;
    DELAY_US(100000);
    }
    PieCtrlRegs.PIEACK.all=PIEACK_GROUP1;       //中断已应答，可接收更多中断
}
```

怎么开始

> *首先你需要有前面章节的基础，本章节默认你已有基础，直接讲外部中断的初始化。*

1. 首先分析一下代码，感觉代码标注已经很明白了，这个就是基本初始化步骤

```
EALLOW;                                 //使能写操作
PieVectTable.XINT1=&xint1_isr;          //服务子函数存入相应向量地址
EDIS;                                   //禁止写操作
PieCtrlRegs.PIECTRL.bit.ENPIE = 1;      //使能PIE模块
PieCtrlRegs.PIEIER1.bit.INTx4 = 1;      //分配XINT1为PIE组1第4位
IER |= M_INT1;                          //使能INT1中断
EINT;                                   //使能全局中断INTM
```

2. 既然是外部中断，肯定要初始化GPIO啊

```
//外部中断输入I/O配置
EALLOW;
GpioCtrlRegs.GPAMUX2.bit.GPIO21=0;          //设置数字模式
GpioCtrlRegs.GPADIR.bit.GPIO21=0;           //设置输出模式
GpioCtrlRegs.GPAQSEL2.bit.GPIO21=0;         //与SYSCLKOUT同步
GpioCtrlRegs.GPACTRL.bit.QUALPRD3 = 0xff;   //设置采样周期
EDIS;
EALLOW;
GpioIntRegs.GPIOXINT1SEL.bit.GPIOSEL = 0x15;//配置GPIO21位XINT1输入引脚
EDIS;
XIntruptRegs.XINT1CR.bit.POLARITY = 0;      //配置为下降沿触发
XIntruptRegs.XINT1CR.bit.ENABLE = 1;        //使能中断
```

3. 怎么把代码贴上之后，感觉一切都那么明了那，不需要什么介绍了，~~逃~~

相关寄存器

> *这个是我写作中最慢的地方，但是也是最有用的地方，上面代码怎么根据自己的需要定制，全靠这里面的知识了啊。一般教程都有流程图什么的，那种任何书上都有，我就不重复了，写这个就是为了囤干货，不为造轮子而生，只为轮子找好归宿。*

首先说一下本代码用的寄存器

1. PIECTRL
   - PIE控制寄存器

字段	功能
PIEVECT	中断向量地址，判断CPU响应那个中断
ENPIE	PIE使能位，0：禁止，1：使能

2. PIEIERx(x=1~12)

- PIEx组使能寄存器
- PIE一共12组，每组8位，此处仅做演示

字段	功能
INTx.y(y=1~8)	使能组内中断，1：使能，0：禁止

3. GpioIntRegs.GPIOXINT1SEL.bit.GPIOSEL = 0x15
    - 配置GPIO21位XINT1输入引脚
    - 此处只是请查阅GPIO章节
4. XIntruptRegs.XINTnCR（n=1~7）
    - 外部中断使能，配置触发方式，

字符	功能
Polarity	触发方式选择，00：下降沿，01：上升沿，10：下降沿，11：上升和下降沿
Enable	中断使能，1：使能，0：禁止

5. IER
    - 中断使能寄存器

字段	功能
PTOSINT	RTOS实时系统中断使能位
DLOGINT	数据记录中断使能位
INT1~INT14	CPU级中断使能位

其他寄存器

1. PIEACK
    - PIE中断确认寄存器
    - 寄存器0~11位，每一位代表一个PIE组，例如bit11对应PIE12组（INT12）
    - x=0，读返回0，PIE组可以发送中断请求，写0无反应

    - x=1，读返回1，表明本组像CPU发送过中断请求，此时组内其他中断请求阻塞，写1将本位清除，从新允许组内再次向CPU发送中断请求
2. PIEIFRx(x=1~12)
    - 类似PIEIERx格式。INTx.y（y=1~8）
    - 指示其相应中断是否有效。
3. IFR
    - 标志位寄存器

字段	功能
RTOSINT	RTOS实时系统中断标志位，0：无中断，1：至少一个中断
DLOGINT	数据记录中断，0：无中断，1：至少一个中断
INT1~INT14	表示CPU中断状态，0：当前通道有中断，1：通道至少有一中断请求

4. DBGIER(仿真模式下中断)
    - 仿真模式下，CPU停止后，用这个寄存器控制中断，有需要在学。
    - 感觉这个不重要，需要的百度一下/Google啥都有。
5. XNMICR
    - 特殊寄存器，对应GPIO内的XNMI，配置同XINTxCR，有需要同上。

最后

> *每一次，后面都需要有一段鸡汤，这篇教程我写的比较长，写了好几天，真实的想法不能说，哎。这篇后面有几个不常用的，需要可以自己查，我写这个目的是让你更好的复习，记住上一篇说的这个是给有基础的你，学习这东西是需要真金白银投资的。什么书啦，开发板啦不该少就不能少，希望你可以把钱花在刀印上。*

还有好多话要说，感觉是时候写一篇随想了。