

# KF8F3132——IIC 主从模式样例程序

## 引言

本应用笔记提供了 KF8F3132—IIC 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<br>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<br>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

TR1: P1口方向控制寄存器(地址: 27H)

|                  |      |   |      |      |      |      |      |      |
|------------------|------|---|------|------|------|------|------|------|
| 复位值<br>--11 1111 | bit7 |   |      |      | bit0 |      |      |      |
|                  | -    | - | TR15 | TR14 | TR13 | TR12 | TR11 | TR10 |
|                  | U    | U | 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P1LR: P1口输出锁存寄存器(地址: 47H)

|                  |      |   |       |       |       |       |       |       |
|------------------|------|---|-------|-------|-------|-------|-------|-------|
| 复位值<br>--xx xxxx | bit7 |   |       |       | bit0  |       |       |       |
|                  | -    | - | P1LR5 | P1LR4 | P1LR3 | P1LR2 | P1LR1 | P1LR0 |
|                  | U    | U | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   |

T0: 定时/计数器 1 寄存器

SSCICTL0: SSCI 控制寄存器 0

寄存器10.1: SSCICTL0: SSCI控制寄存器0(地址:128H)

|                  |          |        |        |         |              |              |              |              |
|------------------|----------|--------|--------|---------|--------------|--------------|--------------|--------------|
| 复位值<br>0000 0000 | bit7     |        |        |         | bit0         |              |              |              |
|                  | SSCIWCFI | SSCIOV | SSCIEN | SSCICKP | SSCI<br>MOD3 | SSCI<br>MOD2 | SSCI<br>MOD1 | SSCI<br>MOD0 |
|                  | R/W      | R/W    | R/W    | R/W     | R/W          | R/W          | R/W          | R/W          |

SSCICTL1: SSCI 控制寄存器 1

**寄存器10.2: SSCICTL1: SSCI控制寄存器1(地址:12AH)**

|                  | bit7           |                |                |               |              |        | bit0          |         |
|------------------|----------------|----------------|----------------|---------------|--------------|--------|---------------|---------|
| 复位值<br>0000 0000 | SSCI<br>CALLEN | SSCI<br>ACKSTA | SSCI<br>ACKDAT | SSCI<br>ACKEN | SSCI<br>RCEN | STOPEN | RESTART<br>EN | STARTEN |
|                  | R/W            | R              | R/W            | R/W           | R/W          | R/W    | R/W           | R/W     |

SSCISTA: SSCI 状态寄存器

**寄存器10.3: SSCISTA: SSCI状态寄存器(地址:12BH)**

|                  | bit7   |       |        |          | bit0          |        |        |         |
|------------------|--------|-------|--------|----------|---------------|--------|--------|---------|
| 复位值<br>0000 0000 | SAMPLE | CKEGE | SSCIDA | SSCISTOP | SSCI<br>START | SSCIRW | SSCIUA | SSCIBUF |
|                  | R/W    | R/W   | R      | R        | R             | R      | R      | R       |

SSCIADD: IIC 地址寄存器

**寄存器10.5: SSCIADD: I2C地址寄存器(地址:12EH)**

|                  | bit7         |              |              |              |              |              |              | bit0         |
|------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| 复位值<br>0000 0000 | SSCI<br>ADD7 | SSCI<br>ADD6 | SSCI<br>ADD5 | SSCI<br>ADD4 | SSCI<br>ADD3 | SSCI<br>ADD2 | SSCI<br>ADD1 | SSCI<br>ADD0 |
|                  | R/W          | R/W          | R/W          | R/W          | R/W          | R/W          | R/W          | R/W          |

SSCIBUFR: SSCI 数据接收/发送缓冲寄存器

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:

T0IF:T0 中断标志位

SSCIPIN: SSCI 模块功能引脚切换位

SSCIEN: 同步串行端口使能位

STARTEN: 启动条件使能位

SSCIIF: SSCI 中断标志位

STOPEN: 停止条件使能位

SSCIBUF: 缓冲器满状态位

SSCIACKSTA: 应答状态位

SSCIRCEN: 接受使能位

SSCIACKDAT: 应答数据位

SSCIACKEN: 应答序列使能位

## IIC 主从模式样例程序框图



**注：** T0 定时器是用来进行中断延时的，流程图中未给出延时函数部分的流程。主机和从机的工作状态是由相关性的因此将两者的流程图放在一起来看。

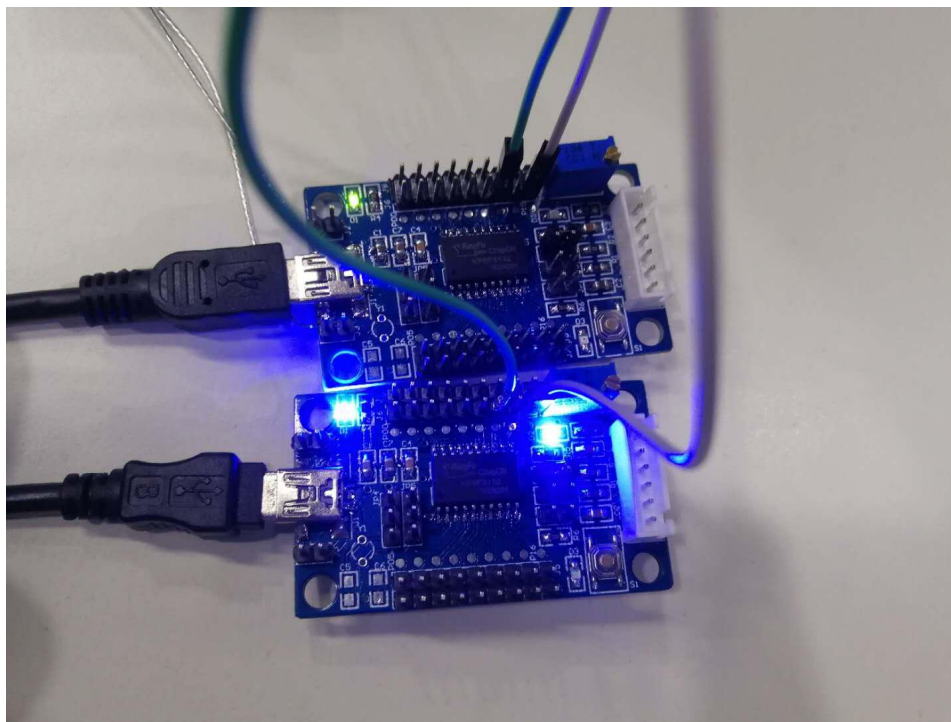
## IIC 主从模式样例简述：

开发环境：ChipON IDE

功能简述：IIC 主机模式，主机连续读、连续写。IIC 通信速率设置为 100Kb/s，如果连续读取的内容是正确的，则 LED2 闪烁。如果读取的内容是错的，则 LED 熄灭。

硬件连接：P13 用作 SDA、P15 用作 SCL，都接 5.1K 上拉电阻，并且与从机的 SDA、SCL 连接。JP2 用跳线帽连接。

下面为 IIC 通信主机和从机硬件连接实物图（图中显示的状态为读取内容正确）：



IIC 主机样例程序:

MCU 初始化:

```
void Init_mcu()  
{  
    OSCCTL=0x70; //时钟配置为16M  
  
    TR1=0x28; //P13、P15配置为输入，IIC口必须设置为输入模式  
    P1LR=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;  
        TOIF=0;  
        while(!TOIF);  
    }  
}
```

IIC 初始化:

```
void Init_IIC()  
{  
    SSCIPIN = 0;           //SDA/SDI=P13, SCK/SCL=P15  
    SSCIEN=0;             //关闭I2C模块，初始化完之后再打开  
  
    SSCICTL0=0X18;        //I2C主控模式，速度为Fosc/ ((SSCIADD+1) *4)  
    SSCICTL1=0X00;  
    SSCISTA=0X00;  
    SSCIADD = 0X27; //主控模式下，做为IIC波特率设置寄存器，79—50k; 39—100k  
    SSCIEN=1;  
}
```

IIC 起始信号函数:

```
void Start_I2C(void)
{
    STARTEN=1;           //发送START。由硬件自动清零
    while (STARTEN);
    while (!SSCIIF);
    SSCIIF=0;
}
```

IIC 停止信号函数:

```
void Stop_I2C()
{
    STOPEN=1;
    while (STOPEN);
    while (!SSCIIF);
    SSCIIF=0;
}
```

主机发送一个字节:

```
void SendB(uchar c )
{
    uchar j=0;
    SSCIBUFR=c;
    while (SSCIBUF);           //发送完成SSCIBUF清零
    j=0;
    while (SSCIACKSTA && j<200) //判断是否收到应答
    {j++;}
    while (!SSCIIF); //等待单个字节的数据发送完成
    SSCIIF=0;
}
```

主机接收一个字节:

```
uchar RcvB()
{
    uchar da=0;
    uint j=0;
    SSCIRCEN=1;
    while (SSCIRCEN); //接收完毕，硬件自动清零
    while (!SSCIIF && (j<500)) //接收完毕，SSCIIF置位
    {j++;}
    SSCIIF=0;

    da=SSCIBUFR;
    return da;
}
```



主机发送应答信号:

```
void ack(uchar a)
{
    if(a) SSCIACKDAT=1; //应答信号=1, 不应答; 应答信号=0, 应答
    else SSCIACKDAT=0;
    SSCIACKEN=1; //使能应答信号发送
    while(SSCIACKEN); //发送完ACKDAT, 硬件自动清零
    while(!SSCIIF);
    SSCIIF=0;
}
```

主机发送多个字节:

```
void IIC_SEND_DATAwrite(uchar cnt)
{
    uchar i=0;
    Start_I2C();
    SendB(Write_adr); //发送7位地址

    for(i=0;i<cnt;i++)
        SendB(Buf[i]);
    Stop_I2C();
}
```

主机读取多个字节:

```
void IIC_SEND_DATARead (uchar cnt)
{
    uchar i=0;

    Start_I2C();
    SendB(Read_adr); //发送7位地址

    for(i=0;i<(cnt-1);i++)
    {
        Buf[i]=RcvB();
        ack(0);
    }
    Buf[cnt-1] =RcvB();
    ack(1);

    Stop_I2C();
}
```

主函数:

```
void main()
{
    uchar i=0;
    uchar flag =1;

    Init_mcu();//初始化外设
    T0_Init();
    Init_IIC();

    while(1)
    {
        for(i=0;i<10;i++)//填写待发送的内容
        {
            Buf[i]=i;
        }
        IIC_SEND_DATAWrite(10);//向从机发送10个字节
        IIC_SEND_DATARead(10);//向从机读取10个字节
        for(i=0;i<10;i++)//验证读取的内容是否正确
        {
            if(Buf[i]== (i+10))
                flag =1;
            else
            {
                flag =0;
                break;
            }
        }

        if(flag==1)//读取的内容正确则LED2闪烁
        {
            flag =0;
            LED2 =0;
            Delay_ms(300);
            LED2=1;
            Delay_ms(300);
        }
        else//读取内容错误则LED2熄灭
        {
            LED2=1;
        }
    }
}
```

IIC 从机样例程序:

MCU 初始化:

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

    TR1=0x28;//P13设为SDA、P15设为SCL，IIC口必须设置为输入模式
    P1LR =0;
}
```

IIC 初始化:

```
void Init_IIC()
{
    SSCIPIN = 0;           //SDA=P13, SCK=P15
    SSCIEN=0;             //关闭I2C模块，初始化完之后再打开

    SSCIADD=IIC_adr;
    SSCICTL0=0X06;        //7位地址模式
    SSCICTL1=0X00;
    SSCISTA=0X00;

    SSCIEN=1;

    SSCIIF=0;            //清零I2C中断标志
    SSCIIE=1;           //打开I2C中断

    //中断初始化
    PUIE=1;              //使能所有未屏蔽的外设中断
    AIE = 1;             //打开总中断
}
```

T0 初始化:

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

延时函数: (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);
    }
}
```

中断处理函数:

```
void int_fun0() __interrupt (0)
{
    if(SSCIIF)
    {
        SSCIIF=0;

        if(SSCIRW==0) //接收数据状态
        {
            Temp=SSCIBUFR;
            if(SSCIDA) //接收到的是数据
            {
                if(Temp==0)
                    cnt=0;

                if (cnt == Temp) //验证接收的数据内容是否正确
                {
                    flag = 1;
                }
                else
                {
                    flag = 0;
                }
                cnt++;
            }
            else //收到的是地址
            {
                kk=9; //将待发送的数据初始化
            }
        }

        else //从模式发送数据
        {
            if(kk<19) //从10开始,连续发送10-19的10个数字
                kk++;
            else
                kk=10;
            SSCIBUFR=kk; //发送数据
            SSCICKP=1;
        }
    }
}
```

主函数:

```
void main()
{
    Init_mcu(); //初始化外设
    T0_Init();
    Init_IIC();
    while(1)
    {
        if(flag==1) //接收的数据正确
        {
            flag = 0;
            LED2 =0;
            Delay_ms(300);
            LED2=1;
            Delay_ms(300);
        }
        else //接收的数据错误
        {
            LED2=1;
        }
    }
}
```

## 注意事项:

1、IIC 数据格式有七位和十位地址两种模式，本样例中采用的发送格式为七位地址格式。

2、在发送数据的起始位、停止位和发送完成一帧数据后都需要清 0 SSCIIF 标志位以释放 SCL 时钟线。

3、当产生应答信号时，电平信号为低电平。

4、当发送为十位地址时会产生一个重复启动位，此时也需要对 SSCIIF 标志位清 0。

5、SSCI 模块可以通过 PINSET 寄存器的 SSCIPIN、SDOPIN、及 SSPIN 位选择 SSCI 模块相关功能引脚位置。

当 SSCIPIN=0 时，SDA/SDI=P1.3, SCK/SCL=P1.5;

当 SSCIPIN=1 时，SDA/SDI=P0.0, SCK/SCL=P0.1。

注：14 或 16 引脚芯片使用 SSCI 功能时 SSCIPIN 必须配置为 1

当 SDOPIN=0 时，SDO=P1.7;

当 SDOPIN=1 时，SDO=P0.2。

注：14 或 16 引脚芯片使用 SSCI 功能时 SDOPIN 必须配置为 1

当 SSPIN=0 时， $\overline{SS}$ =P2.3;

当 SSPIN=1 时， $\overline{SS}$ =P0.4。

注：14 引脚芯片使用 SSCI 功能时 SSPIN 必须配置为 1

6、当 SSCICTL0 位 SSCIMOD<3:0>=1001 时，不能对 SSCIADD 寄存器进行读和写，任何对 SSCIADD 寄存器（地址也是 12EH）的读和写操作均是对 SSCIMSK 寄存器操作。

7、在 IIC 模式下，不支持 SSCIADD 寄存器的值为 0、1、2 的情况。

8、IIC 主设备产生时钟，起始和停止信号。

9、任何 IIC 模式的选择，在 SSCIEN 置 1 后都会强制 SCL 和 SDA 引脚为漏极开路（假定通过编程将相应的 TR1 位置 1，使这些引脚成为输入引脚）。必须在 SCL 和 SDA 引脚外接上拉电阻，才能使 IIC 模块正常工作。