

//0xa0 写从地址， 0xa1 读从地址， 数据地址:0x00--0x2047,有 16 页， 页写一次 16 字节

```
#include <reg51.h>
```

```
#include <intrins.h>
```

```
#define uchar unsigned char
```

```
#define uint unsigned int
```

```
#define nop _nop_();_nop_();_nop_();_nop_();
```

```
sbit i2c_sck=P2^1;
```

```
sbit i2c_dat=P2^2;
```

```
bit flag_cck;
```

```
uchar w_i2c[10]={1,2,3,4,5,6,7,8,9,10};
```

```
uchar r_i2c[10]={0};
```

```
/*
```

```
bit i2c_start();//启动 i2c 总线
```

```
void i2c_stop();//停止 i2c 总线
```

```
void i2c_ack(); //对 i2c 总线应答
```

```
void i2c_nck(); //不对 i2c 总线应答
```

```
bit i2c_cck();//检查应答位
```

```
    w_i2c_1byte();//向 i2c 写 1byte 数据
```

```
    r_i2c_1byte();//从 i2c 读 1byte 数据
```

```
    w_i2c_1();//向 i2c 写 1byte 数据
```

```
    w_i2c_n();//向 i2c 写 nbyte 数据
```

```
    r_i2c_n();//从 i2c 读 nbyte 数据
```

```
*/
```

```
bit i2c_start();//启动 i2c 总线
```

```
{
```

```
    i2c_dat=1;
```

```
    i2c_sck=1;
```

```
    nop;
```

```
    i2c_dat=0;
```

```
    nop;
```

```
    i2c_sck=0;
```

```
}
```

```
void i2c_stop() //停止 i2c 总线
```

```
{  
    i2c_dat=0;  
    i2c_sck=1;  
    nop;  
    i2c_dat=1;  
    nop;  
    i2c_sck=0;  
}
```

```
void i2c_ack() //对 i2c 总线应答
```

```
{  
    i2c_dat=0;  
    i2c_sck=1;  
    nop;  
    i2c_sck=0;  
    nop;  
    i2c_dat=1;  
}
```

```
void i2c_nck() //不对 i2c 总线应答
```

```
{  
    i2c_dat=1;  
    i2c_sck=1;  
    nop;  
    i2c_sck=0;  
}
```

```
bit i2c_cck() //检查应答位
```

```
{  
    flag_cck=0;  
    i2c_dat=1;  
    i2c_sck=1;  
    nop;  
    flag_cck=i2c_dat;  
    nop;  
    i2c_sck=0;  
    return(flag_cck);  
}
```

```
w_i2c_byte(uchar da)//向 i2c 写 1byte 数据
```

```
{  
    uchar i=8;
```

```

while(i--)
{
    i2c_dat=da&0x80;
    i2c_sck=1;
    nop;
    i2c_sck=0;
    da=da<<1;
}
}

r_i2c_1byte()//从 i2c 读 1byte 数据
{
    uchar i=8,acda;
    i2c_sck=0;
    while(i--)
    {
        i2c_sck=1;
        acda=(acda<<1)|i2c_dat;
        i2c_sck=0;
    }
    return(acda);
}

w_i2c_1(uchar byte_adr,uint n,uchar w_dat[])//向 i2c 写 1byte 数据
{
    i2c_start();
    w_i2c_byte(0xa0);//写从地址
    i2c_ack(); //对 i2c 总线应答
    i2c_cck();
    if(flag_cck)
    {
        w_i2c_byte(byte_adr);
        w_i2c_byte(w_dat[n]);
        i2c_ack();
        i2c_stop();
    }
}

r_i2c_1(uchar byte_adr,uint n,uchar r_dat[])//向 i2c 读 1byte 数据
{
    i2c_start(); //启动总线
    w_i2c_byte(0xa0);//写从地址
    w_i2c_byte(byte_adr);//写数据地址
    i2c_ack(); //对 i2c 总线应答
}

```

```

i2c_cck(); //查询应答
if(flag_cck) //总线不忙
{
    i2c_start();//启动总线
    w_i2c_byte(0xa1);//写读命令
    w_i2c_byte(byte_adr);//读地址为 byte_adr 的数据
    i2c_ack(); //对 i2c 总线应答

    r_dat[n]=r_i2c_1byte();//将读出的数据放在数组
    i2c_ack(); //对 i2c 总线应答
    i2c_nck(); //不对 i2c 总线应答
    i2c_stop();//停止总线
}
}

w_i2c_n(uchar byte_adr,uint n,uchar w_dat[])//向 i2c 写 nbyte 数据
{
    uint k;
    i2c_start();//启动总线
    w_i2c_byte(0xa0);//写从地址
    i2c_ack(); //对 i2c 总线应答
    i2c_cck();//查询应答
    if(flag_cck)//总线不忙
    {
        w_i2c_byte(byte_adr);//写数据始地址
        for(k=0;k<n;k++)
        {
            w_i2c_byte(w_dat[k]);//将数组写入 i2c
            i2c_ack();//应答
        }

        i2c_stop();//停止
    }
}

r_i2c_n(uchar byte_adr,uint n,uchar r_dat[])//从 i2c 读 nbyte 数据
{
    uint k;
    i2c_start(); //启动总线
    w_i2c_byte(0xa0);//写从地址
    w_i2c_byte(byte_adr);//写数据地址
    i2c_ack(); //对 i2c 总线应答
    i2c_cck(); //查询应答

```

```

if(flag_cck) //总线不忙
{
    i2c_start();//启动总线
    w_i2c_byte(0xa1);//写读命令
    w_i2c_byte(byte_adr);//读地址为 byte_adr 的数据
    i2c_ack(); //对 i2c 总线应答
    for(k=0;k<n-1;k++)
    {
        r_dat[k]=r_i2c_1byte();//将读出的数据放在数组
        i2c_ack(); //对 i2c 总线应答
    }
    r_dat[k]=r_i2c_1byte();
    i2c_nck(); //不对 i2c 总线应答
    i2c_stop();//停止总线
}
}

main()
{

while(1)
{
    w_i2c_1(0xa4,0,w_i2c);
    r_i2c_1(0xa4,0,r_i2c);
    //bcd_turn_d();
    //x_595();
}
}

```