

## 实验三 IIC 总线 EEPROM 读写

实验目的：

- (1) 掌握 IIC 总线的使用方式
- (2) 掌握 EEPROM 芯片的工作原理
- (3) 掌握 IIC 总线通信方式实现 MCU 与 24C02 的数据读写

实验内容：

学习 IIC 总线传输机制，通过单片机 MCU 的 I/O 实现 IIC 总线在 EEPROM 24C02 上读写数据并显示在数码管上。

参考资料：芯片手册文档，

实验步骤：

- (1) 编写 EEPROM 写入程序，将数据（学号后面 8 位）保存在芯片，下载到开发板进行调试，实现显示写入完毕功能。
- (2) 编写 EEPROM 读出程序，将数据（学号后面 8 位）显示在数码管。

实验要求：

编写实验报告，主要包括关键步骤的实现和效果截屏，并分析实验过程中出现的问题和分析解决方法。

### 一、实现代码

```
#include "reg51.h"
#include "intrins.h"
#define uchar unsigned char
#define uint unsigned int
#define out P2
#define SMG_PORT PO
```

```
sbit scl = P2^1;
sbit sda = P2^0;
sbit key1 = P3^2;
sbit key2 = P3^3;
```

```
sbit LSA = P2^2;
sbit LSB = P2^3;
sbit LSC = P2^4;
```

```
uchar data mem[4]_at_ 0x55;
uchar mem[4] = {21, 12, 10, 90};
uchar data rec_mem[4] _at_ 0x60;
uchar rec_mem[4] = {0x00, 0x00, 0x00, 0x00};
uchar gsmg_code[17]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,
                    0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71};
```

```

void start(void);
void stop(void);
void sack(void);
bit rack(void);
void ackn(void);
void send_byte(uchar);
uchar rec_byte();
void write(void);
void read(void);
void delay4us(void);
void show(void);
void smg_display(uchar dat[], uchar pos);

```

```

void main(void)
{
    EA=1;EXO=1;EX1=1;
    while(1)
    {
        show();
    }
}

```

```

void ext0()interrupt 0
{
    write();
}

```

```

void ext1()interrupt 2
{
    read();
    show();
}

```

```

void read(void)
{
    uchar i;
    bit f;
    start();
    send_byte(0xa0);
    f = rack();
    if(!f)
    {
        start();
    }
}

```

```

        send_byte(0xa0);
        f = rack();
        send_byte(0x00);
        f = rack();
    }
    if(!f)
    {
        start();
        send_byte(0xa1);
        f = rack();
    }
    if(!f)
    {
        for(i=0; i<3; i++)
        {
            rec_mem[i] = rec_byte();
            sack();
        }
        rec_mem[3] = rec_byte();
        ackn();
    }
    stop();
    out = rec_mem[3];
    while(!key2);
}

```

```

void write(void)
{
    uchar i;
    bit f;
    start();
    send_byte(0xa0);
    f = rack();
    if(!f)
    {
        send_byte(0x00);
        f = rack();
        if(!f)
        {
            for(i=0; i<4; i++)
            {
                send_byte(mem[i]);
            }
        }
    }
}

```

```

        f = rack();
        if(f) break;
    }
}
}
stop();
out = 0xc3;
while(!key1);
}

```

```

void start(void)
{
    scl = 1;
    sda = 1;
    delay4us();
    sda = 0;
    delay4us();
    scl = 0;
}

```

```

void stop(void)
{
    scl = 0;
    sda = 0;
    delay4us();
    scl = 1;
    delay4us();
    sda = 1;
    delay4us();
    sda = 0;
}

```

```

bit rack(void)
{
    bit flag;
    scl = 1;
    delay4us();
    flag = sda;
    scl = 0;
    return(flag);
}

```

```

void sack(void)
{

```

```

    sda = 0;
    delay4us();
    scl = 1;
    delay4us();
    scl = 0;
    delay4us();
    sda = 1;
    delay4us();
}

```

```

void ackn(void)
{
    sda = 1;
    delay4us();
    scl = 1;
    delay4us();
    scl = 0;
    delay4us();
    sda = 0;
}

```

```

uchar rec_byte(void)
{
    uchar i, temp;
    for(i=0; i<8; i++)
    {
        temp <<= 1;
        scl = 1;
        delay4us();
        temp |= sda;
        scl = 0;
        delay4us();
    }
    return(temp);
}

```

```

void send_byte(uchar temp)
{
    uchar i;
    scl = 0;
    for(i=0; i<8; i++)
    {
        sda = (bit)(temp&0x80);
        scl = 1;
    }
}

```

```

        delay4us();
        scl = 0;
        temp <= 1;
    }
    sda = 1;
}

```

```

void show(void)
{
    uchar i=0;
    uchar dat[8];//= {2,1, 1,2, 1,0, 9,0};
    for(i=0; i<4; i++)
    {
        dat[i*2+1] = rec_mem[i]%10;
        dat[i*2] = rec_mem[i]/10;
    }

    smg_display(dat, 1);
}

```

```

void smg_display(uchar dat[], uchar pos)
{
    uchar i=0;
    uchar pos_temp=pos-1;
    uint k=0;
    //out = 0xc3;
    for(i=pos_temp; i<8; i++)
    {
        switch(i)// 浣? 閫?
        {
            case 0: LSC=1;LSB=1;LSA=1;break;
            case 1: LSC=1;LSB=1;LSA=0;break;
            case 2: LSC=1;LSB=0;LSA=1;break;
            case 3: LSC=1;LSB=0;LSA=0;break;
            case 4: LSC=0;LSB=1;LSA=1;break;
            case 5: LSC=0;LSB=1;LSA=0;break;
            case 6: LSC=0;LSB=0;LSA=1;break;
            case 7: LSC=0;LSB=0;LSA=0;break;
        }
        SMG_PORT=gsmsg_code[dat[i-pos_temp]];// 浣? 閫? 变? 閫? 鏃? 鍒?
        for(k=0; k<250; k++)
        {
            //delsy4us();
        }
    }
}

```

```

        SMG_PORT=0x00;// 娑? 闊?
    }
}
void delay4us(void)
{
    _nop();_nop();_nop();_nop();
}

```

## 二、演示效果



