实验十三:24C02EEPROM 实验

AT24C01/02/04/08/16/32/64/128/256 的读写操作
1K/2K/4K/8K/16K/32K/64K/128K/256K 位 E2PROM

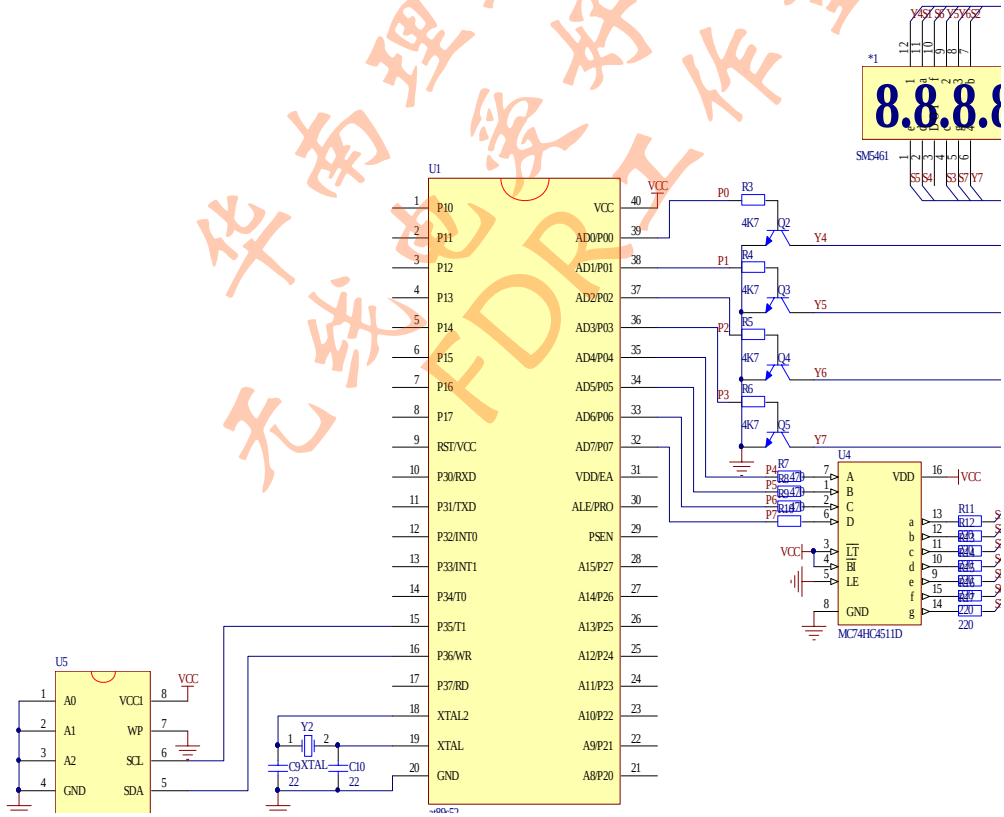
一、器件特性

与400KHz I2C 总线兼容
1.8 到6.0 伏工作电压范围
低功耗CMOS 技术
写保护功能当WP 为高电平时进入写保护状态
页写缓冲器
自定时擦写周期
噪声保护的施密特触发输入
零待机电流
1,000,000 编程/擦写周期
可保存数据100 年
8 脚DIP SOIC 封装
温度范围商业级工业级和汽车级

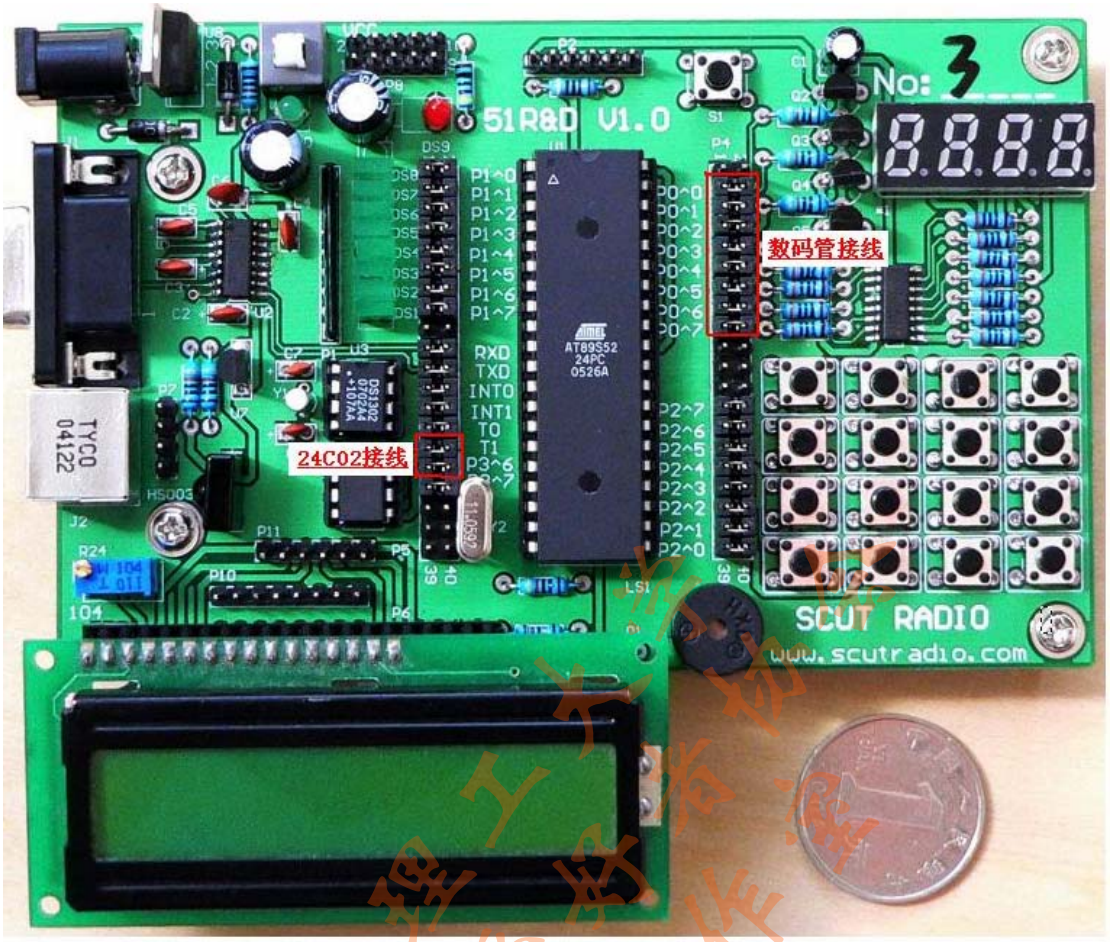
二、实验任务:

写数据进 AT24C02，再读出来。并在数码管上显示。

三、硬件原理图:



四、硬件连线图:



五、实验原理：

首先简单的说明以下 I2C 总线，I 2 C(Inter－Integrated Circuit)总线是一种由 PHILIPS 公司开发的两线式串行总线,用于连接微控制器及其外围设备。I 2 C 总线产生于在 80 年代,最初为音频和视频设备开发,如今主要在服务器管理中使用,其中包括单个组件状态的通信。例如管理员可对各个组件进行查询,以管理 系统的配置或掌握组件的功能状态,如电源和系统风扇。可随时监控内存、硬盘、网络、系统温度等多个参数,增加了系统的安全性,方便了管理。

AT24C02 只有二根信号线，一根是双向的数据线 SDA，另一根是时钟线 SCL。在 I2C 总线上传送的一个数据字节由八位组成。总线对每次传送的字节数没有限制，但每个字节后必须跟一位应答位。数据传送首先传送最高位(MSB)，数据传送按图 1 所示格式进行。首先由主机发出启动信号“S”(SDA 在 SCL 高电平期间由高电平跳变为低电平)，然后由主机发送一个字节的数据。启动信号后的第一个字节数据具有特殊含义：高七位是从机的地址，第八位是传送方向位，0 表示主机发送数据(写)，1 表示主机接收数据(读)。被寻址到的从机设备按传送方向位设置为对应工作方式。标准 I2C 总线的设备都有一个七位地址，所有连接在 I2C 总线上的设备都接收启动信号后的第一个字节，并将接收到的地址与自己的地址进行比较，如果地址相符则为主机要寻访的从机，应在第九位答时钟脉冲时向 SDA 线送出低电平作为应答。除了第一字节是通用呼叫地址或十位从机地址之外第二字节开始即数据字节。数据传送完毕，由主机发出停止信号“P”(SDA 在 SCL 高电平期间由低电平跳变为高电平)。（注：如要了解更多，请看《AT24c02 串行储存器中文官方资料手册》）

六、 实验代码：

```
//以下是一个通用的 24C01－24C256 共 9 种 EEPROM 的字节读写操作程序例子：
#include <reg52.h>
#include <intrins.h>
#define ERROR 10
#define LEDPORT P0            //数码管接线脚
sbit SCL=P3^5;                //AT24C02-256 的接线脚
sbit SDA=P3^6;
enum eepromtype {AT2401,AT2402,AT2404,AT2408,AT2416,AT2432,AT2464,AT24128,AT24256};
enum eepromtype enumer;
unsigned char code buf1[]={1,3,5,7,9,10,11,12,13,15};/* 发送缓冲区 */
unsigned char buf2[10];                              /* 接收缓冲区 */
/* 一个通用的 24C01－24C256 共 9 种 EEPROM 的字节读写操作程序，
此程序有五个入口条件，分别为读写数据缓冲区指针，
进行读写的字节数，EEPROM 首址，EEPROM 控制字节，
以及 EEPROM 类型。此程序结构性良好，具有极好的容错性，程序机器码也不多:
DataBuff 为读写数据输入 / 输出缓冲区的首址
Length 为要读写数据的字节数量
Addr 为 EEPROM 的片内地址 AT24256 为 0～32767
Control 为 EEPROM 的控制字节，具体形式为(1)(0)(1)(0)(A2)(A1)(A0)(R/W),其中 R/W=1,
表示读操作,R/W=0 为写操作,A2,A1,A0 为 EEPROM 的页选或片选地址;
enumer 为枚举变量,需为 AT2401 至 AT24256 中的一种,分别对应 AT24C01 至 AT24C256;
函数返回值为一个位变量，若返回 1 表示此次操作失效，0 表示操作成功;
ERROR 为允许最大次数，若出现 ERRORCOUNT 次操作失效后，则函数中止操作，并返回 1
SDA 和 SCL 由用户自定义，这里暂定义为 P3^0 和 P3^1; */
/* －－－－－ AT24C01～AT24C256 的读写程序 －－－－－－－ */
bit RW24xx(unsigned char *DataBuff,unsigned char Length,unsigned int Addr,
unsigned char Control,enum eepromtype enumer)
{
    void Delay(unsigned int DelayCount); /* 延时 */
    void Start(void);                    /* 启动总线 */
    void Stop(void);                     /* 停止 IIC 总线 */
    bit RecAck(void);                    /* 检查应答位 */
    void NoAck(void);                    /* 不对 IIC 总线产生应答 */
    void Ack(void);                      /* 对 IIC 总线产生应答 */
    unsigned char Receive(void);         /* 从 IIC 总线上读数据子程序 */
    void Send(unsigned char sendbyte);   /* 向 IIC 总线写数据 */
    unsigned char data j,i=ERROR;
```

```
bit errorflag=1;                        /* 出错标志 */
while(i--)
{
    Start();                            /* 启动总线 */
    Send(Control & 0xfe);               /* 向 IIC 总线写数据 */
    if(RecAck()) continue;              /* 如写正确结束本次循环 */
    if(enumer > AT2416)
    {
        Send((unsigned char)(Addr >> 8));
        if(RecAck()) continue;
    }
    Send((unsigned char)Addr);          /* 向 IIC 总线写数据 */
    if(RecAck()) continue;              /* 如写正确结束本次循环 */
    if(!(Control & 0x01))
    {
        j=Length;
        errorflag=0;                    /* 清错误特征位 */
        while(j--)
        {
            Send(*DataBuff++);          /* 向 IIC 总线写数据 */
            if(!RecAck()) continue;     /* 如写正确结束本次循环 */
            errorflag=1;
            break;
        }
        if(errorflag==1) continue;
        break;
    }
    else
    {
        Start();                        /* 启动总线 */
        Send(Control);                  /* 向 IIC 总线写数据 */
        if(RecAck()) continue;
        while(--Length)                 /* 字节长为 0 结束 */
        {
            *DataBuff ++= Receive();
            Ack();                      /* 对 IIC 总线产生应答 */
        }
        *DataBuff=Receive();            /* 读最后一个字节 */
        NoAck();                /* 不对 IIC 总线产生应答 */
        errorflag=0;
        break;
    }
}
Stop();                                 /* 停止 IIC 总线 */
```

```
    if(!(Control & 0x01))
    {
        Delay(255); Delay(255); Delay(255); Delay(255);
    }
    return(errorflag);
}
/* * * * * 以下是对 IIC 总线的操作子程序 * * * * */
/* * * * * * 启动总线 * * * * */
void Start(void)
{
    SCL=0;          /* SCL 处于高电平时,SDA 从高电平转向低电平表示 */
    SDA=1;              /* 一个"开始"状态,该状态必须在其他命令之前执行 */
    SCL=1;
    _nop_(); _nop_(); _nop_();
    SDA=0;
    _nop_(); _nop_(); _nop_(); _nop_();
    SCL=0;
    SDA=1;
}
/* * * * * 停止 IIC 总线 * * * * */
void Stop(void)
{
    SCL=0;              /*SCL 处于高电平时,SDA 从低电平转向高电平 */
    SDA=0;              /*表示一个"停止"状态,该状态终止所有通讯 */
    SCL=1;
    _nop_(); _nop_(); _nop_();      /* 空操作 */
    SDA=1;
    _nop_(); _nop_(); _nop_();
    SCL=0;
}
/* * * * * 检查应答位 * * * * */
bit RecAck(void)
{
    SCL=0;
    SDA=1;
    SCL=1;
    _nop_(); _nop_(); _nop_(); _nop_();
    CY=SDA;                     /* 因为返回值总是放在 CY 中的 */
    SCL=0;
    return(CY);
}
/* * * * *对 IIC 总线产生应答 * * * * */
void Ack(void)
{
```

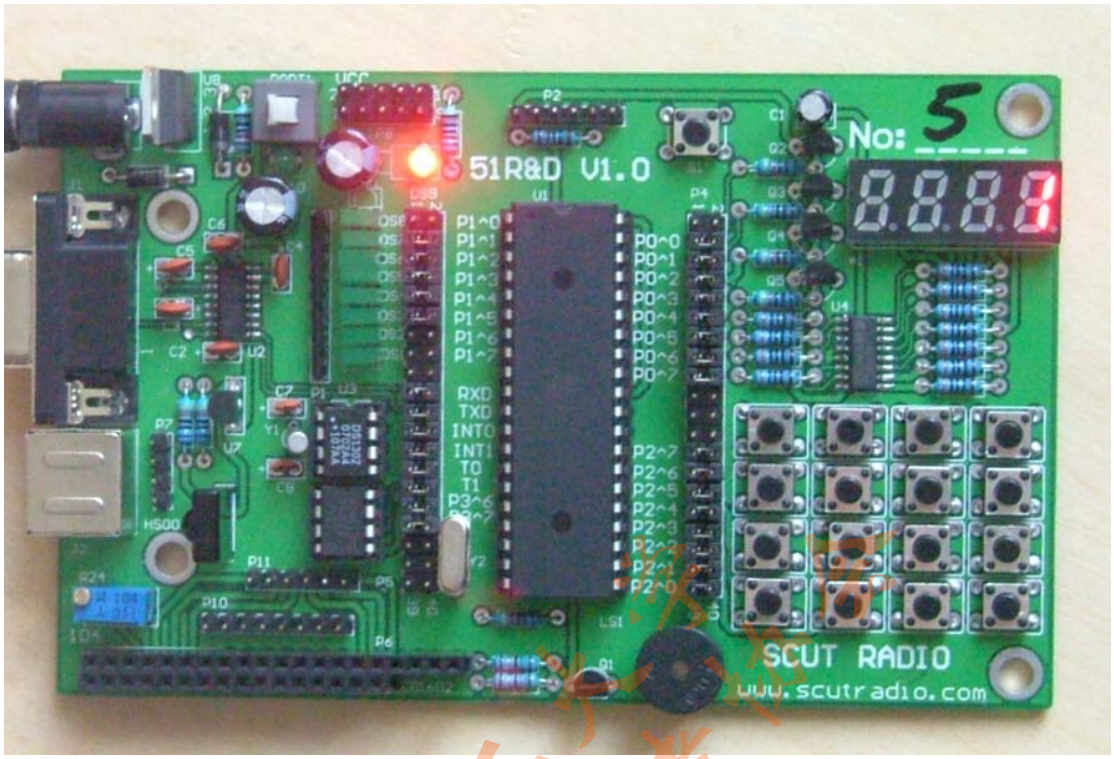
```
    SDA=0;                          /* EEPROM 通过在收到每个地址或数据之后, */
    SCL=1;                  /*  置 SDA 低电平的方式确认表示收到读 SDA 口状态  */
    _nop_(); _nop_(); _nop_(); _nop_();
    SCL=0;
    _nop_();
    SDA=1;
}

/* * * * * * * * * 不对 IIC 总线产生应答 * * * * */
void NoAck(void)
{
    SDA=1;
    SCL=1;
    _nop_(); _nop_(); _nop_(); _nop_();
    SCL=0;
}
/* * * * * * * * * 向 IIC 总线写数据 * * * * */
void Send(unsigned char sendbyte)
{
    unsigned char data j=8;
    for(;j>0;j--)
    {
        SCL=0;
        sendbyte <<= 1;             /*  使 CY=sendbyte^7; */
        SDA=CY;                     /* CY  进位标志位  */
        SCL=1;
    }
    SCL=0;
}
/* * * * * * * * * 从 IIC 总线上读数据子程序 * * * * */
unsigned char Receive(void)
{
    register receivebyte,i=8;
    SCL=0;
    while(i--)
    {
        SCL=1;
        receivebyte = (receivebyte <<1 ) | SDA;
        SCL=0;
    }
    return(receivebyte);
}
/*  以上为 AT24C01～AT24C256 的读写程序，各人可根据自己的需要应用。
在 buf1 中填入需要写入的内容，buf2 的大小可根据需要定义。
```

addr 可根据使用的芯片选择，可从任何位置读写，只要在该芯片的范围内。
enumer=ATxxx，根据使用的芯片赋值。各函数中的形式参数不需改变。
本程序只要在调用的程序中定义实际参数即可，上述各子程序不必改动。*/

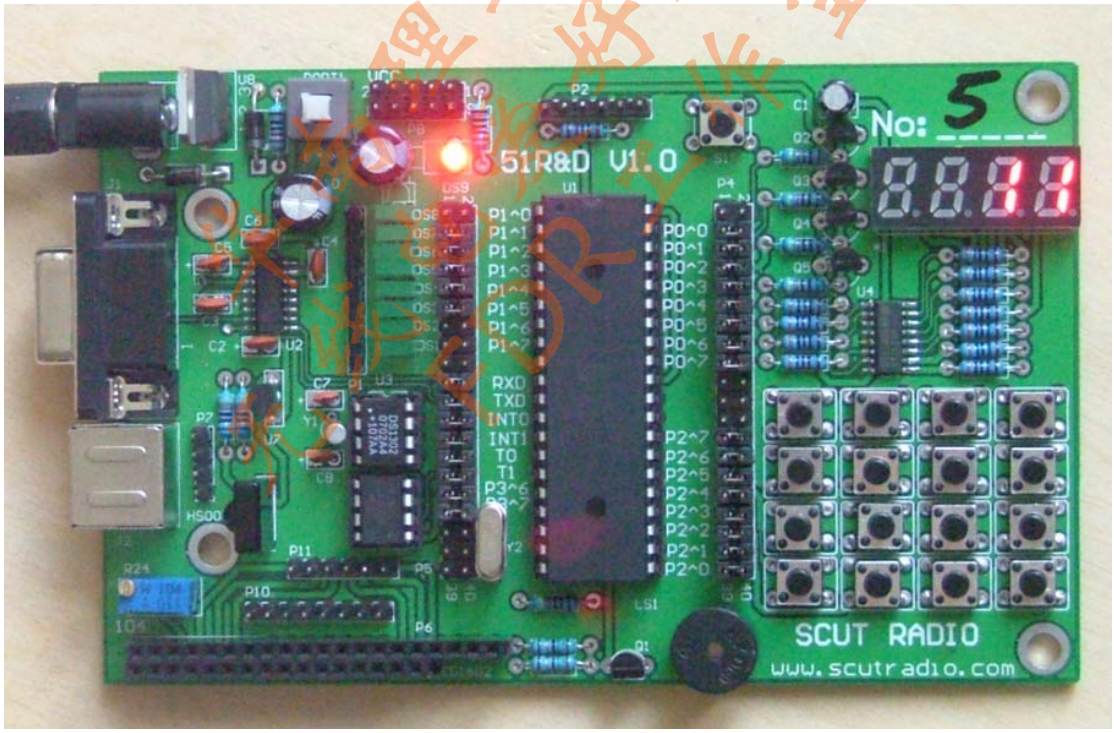
```
/* * * * * * * * 一个简单延时程序 * * * * * * * * * * * * */
void Delay(unsigned int DelayCount)
{
    while(DelayCount--);
}
/*-------------数码管显示函数--------------------
参数:number  要显示的数字
参数:pos     要显示的位置(从左到右为 1--4)
返回值:无
-----------------------------------------------*/
/* * * * * * * * * *数码管显示函数* * * * * * * * * * * */
void display(unsigned char number,unsigned char pos)//数码管显示函数
{
    unsigned char temp;                          //临时变量
    if(number<10 && pos<5 && pos>0)//确定数据合法
    {
        temp=(number<<4)&0xf0;          //获得要显示的数据
        temp+=1<<(pos-1);               //送位置位
        LEDPORT=temp;                   ////送显示数据,开始显示
    }
    else return;
}

void leddisplay(unsigned int count)
{
    unsigned int i=6000;
    while(i--)
    {
    if(count>=1000)
    {
        display(count/1000,1);
        display(count%1000/100,2);
        display(count%100/10,3);
        display(count%10,4);
    }
    else   if(count>=100)
    {
        display(count/100,2);
        display(count%100/10,3);
        display(count%10,4);
```

```
        }
        else    if(count>=10)
        {
            display(count/10,3);
            display(count%10,4);
        }
        else
        {
            display(count,4);
        }
        }
}
/* * * * * * * 主函数* * * * * * * * * * * * * * * */
void main()
{
        unsigned char Control,*p1,*p2;
        unsigned char Length,i=0;
        unsigned int addr ; /* 24Cxx 片内地址 */
        p1=buf1;p2=&buf2[0];
        addr=0;                 /* 片内地址 AT24C02 为 0～255 */
        Length=10;              /* 读写长度 */
        enumer=AT2402;          /* 读写 AT24C02 */
        Control=0xa0;           /* 写操作 */
        RW24xx(p1,Length,addr,Control,enumer); /* 写 */
        Control=0xa1;           /* 读操作 */
        RW24xx(p2,Length,addr,Control,enumer); /* 读 */
        while(1)
        {
            leddisplay(buf2[i]);//数码管显示读出来的数据
            i++;
            if(i==10){i=0;}
        }
}
```

七、效果图：


51R&D V1.0
No: 5
SCUT RADIO
www.scutradio.com

图一


51R&D V1.0
No: 5
SCUT RADIO
www.scutradio.com

图二