

## 第 29 课，本课试验写老鼠爱大米的乐谱到 24c02 中

这一课实际上纯粹是 24c02 的操作，是为了下一课的设计的。

从前我们只学习过写入一个字节到 24c02 中。在这一课中，我们将一首乐谱全部写入到 24c02 中。在下一课时，我们将会读出 24c02 中的乐谱播放。

这种工作方式，可以比作一个简单的 mp3，将乐曲写在可以重复写入的存储器中，播放时，读出来再播放。这样，只要换一个存有新的音乐的存储器，或者从外部将新的乐曲写入到存储器中，就可以播放新的乐曲，而单片机的程序并没有改变。

```
#define uchar unsigned char //定义一下方便使用
#define uint unsigned int
#define ulong unsigned long
#include <reg52.h> //包括一个 52 标准内核的头文件
//本课试验写入一个字节到 24c02 中

char code dx516[3] _at_ 0x003b;//这是为了仿真设置的

#define WriteDeviceAddress 0xa0 //定义器件在 IIC 总线中的地址
#define ReadDviceAddress 0xa1
sbit SCL=P2^7;
sbit SDA=P2^6;

sbit P10=P1^0;

//定时函数
void DelayMs(uint number)
{
    uchar temp;
    for(;number!=0;number--)
    {
        for(temp=112;temp!=0;temp--);
    }
}

//开始总线
void Start()
{
    SDA=1;
    SCL=1;
    SDA=0;
    SCL=0;
}

//结束总线
```

```

void Stop()
{
    SCL=0;
    SDA=0;
    SCL=1;
    SDA=1;
}

//测试 ACK
bit TestAck()
{
    bit ErrorBit;
    SDA=1;
    SCL=1;
    ErrorBit=SDA;
    SCL=0;
    return(ErrorBit);
}

//写入 8 个 bit 到 24c02
Write8Bit(uchar input)
{
    uchar temp;
    for(temp=8;temp!=0;temp--)
    {
        SDA=(bit)(input&0x80);
        SCL=1;
        SCL=0;
        input=input<<1;
    }
}

//写入一个字节到 24c02 中
void Write24c02(uchar ch,uchar address)
{
    Start();
    Write8Bit(WriteDeviceAddress);
    TestAck();
    Write8Bit(address);
    TestAck();

    Write8Bit(ch);
    TestAck();
}

```

```
    Stop();
    DelayMs(10);
}
```

//老鼠爱大米

```
uchar code dami[]={
"321-|3.2_1-|3231|3_6.5-|3_5_665|65-3_2_|122_3_2|12--|"
};
```

//本课试验写老鼠爱大米的乐谱到 24c02 中

```
void main(void) // 主程序
{
    uchar i;
    for(i=0;dami[i]!=0;i++)
    {
        Write24c02(dami[i],i);// 逐个写入到 24c02 的中
    }
    Write24c02(0x00,i); //写入最后一个 0

    P10=0; //指示运行完毕
    while(1); //程序挂起
}
```

通过一个 for 循环，我们就把乐谱全部写入到了 24c02 中。看到 for 循环的条件是 dami[i]!=0，这是因为字符串的最后一个字符是 0，可以作为结束的判断，这个方法在前面的音乐播放等例子中已经见到。

编译，运行，看到 P10 的 LED 亮了，乐谱已经写入。

本课结束。

作业：无