

# I2C 总线规范与 I2C 器件 C51 读写程序

刘波 清华东路 27 号 1022 邮编：100083

**摘要：**本文简要介绍了 I2C 总线，并给出了 I2C 器件的 C51 读写程序，极大的方便了大家对 I2C 总线的学习和设计应用。程序设计采用模块化设计，方便向目标板的移植。

**关键字：**I2C 总线，C51，模块化设计

## 1、I2C 总线简介：

I2C 总线是一双线串行总线，它提供一小型网络系统，为总线上的电路共享公共的总线。总线上的器件有单片机、LCD 驱动器以及上 E2PROM 器等等。自从菲利普公司提出 I2C 总线规范以来，I2C 器件得到了广泛的应用。I2C 器件的应用大大减少了电路间连线，减小了电路板尺寸，降低了硬件成本，并提高了系统可靠性。

## 2、I2C 规范：

I2C 总线规范可以参考有关资料，本文仅给出有关硬件操作的基本概念。

### 2.1、I2C 总线名词解释：

**发送器：**发送数据到总线上的器件

**接收器：**从总线上接收数据的器件

**主器件：**启动数据传送并产生时钟信号的器件

**从器件：**被主器件寻址的器件

### 2.2、I2C 总线时序定义：

**起始位：**SCL=1 时，在 SDA 上有下降延

**停止位：**SCL=1 时，在 SDA 上有上升延

\*主器件发停止信号给从器件，作用在于使从器件处为准备状态(或是已知的状态)

**从应答：**SDA=0

\*写数据时，应答的器件在第九个时钟周期将 SDA 线拉低，表示已收到一个 8 位数据，并表示可继续接收。主机在向从机写一字节后接收从应答，再进行后续操作(一般不考虑从应答位的具体值，仅在应答检测时用于判断从器件是否完成其内部写周期)

**主应答：**SDA=0(用)

\*读数据时，主机每接收从机一个字节数据(不是最后一个)，返回主应答 ACK(SDA=0)是最后一个字节时返回无需应答 NO\_ACK (SDA=1)

当从机工作于读模式时，在向主机发送一个 8 位数据后释放 SDA 线并监视一个应答信号，一旦接收到主机的应答信号，从机继续发送数据；如主器件没有发送应答信号，从机停止传送数据并等待一个停止信号，主器件必须发一个停止信号给从机使其进入备用电源模式并使器件处于已知的状态

**应答检测：**采用应答检测读命令测试从机是否<页写>结束(通过从应答来识别)

\*当从器件完成内部写周期后将发送一个应答信号(从应答)给主器件，以便可以继续进行下一次读操作

## 3、I2C 器件 C51 读写程序

### 3.1、数据定义说明：

```
sbit sda=P0^0; //I2C 器件 SDA 数据线
```

```
sbit scl=P0^1; //I2C 器件 SCL 时钟线
```

```
#define nops() {_nop_();_nop_();_nop_();_nop_();_nop_();} //nop 指令个数定义
```

```
uchar i2c_ad_main; //器件从地址
```

```
uchar i2c_ad_sub; //器件子地址
```

```
uchar *i2c_buf; //发送/接收数据缓冲区
```

```
uchar i2c_num; //发送/接收数据个数
```

说明：可以在你的应用程序中修改上述值，使其适应你的 demo 设计

### 3.2、子函数说明：（私有函数，不能调用）

```
void i2c_start(void) //送起始位
```

```
void i2c_stop(void) //送停止位
```

```
void i2c_ack_main(bit ack_main) //主应答
```

说明：ack\_main=1:ack 主应答；ack\_main=0:no\_ack 无需应答

```
void send_byte(uchar c) //发送单个字节
```

```
uchar read_byte(void) //接收单个字节
```

说明：此函数需要配合主应答来用

### 3.3、模块函数说明：（I2C 外部接口函数，可被应用程序调用）

```
void send_to_byte(uchar ad_main,uchar c) //向无子地址器件发送单字节数据
```

```
void send_to_nbyte(uchar ad_main,uchar ad_sub,uchar *buf,uchar num)
```

```
//向有子地址器件发送多字节数据
```

```
void read_from_byte(uchar ad_main,uchar *buf) //从无子地址器件读单字节数据
```

```
void read_from_nbyte(uchar ad_main,uchar ad_sub,uchar *buf,uchar num)
```

```
//从有子地址器件读多个字节数据
```

### 3.4、C51 读写程序源代码：

#### 3.4.1 头文件<i2cbus.h>定义：

```
#define uchar unsigned char
```

```
#define uint unsigned int
```

```
//nop 指令个数定义
```

```
#define nops() {_nop();_nop();_nop();_nop();_nop();}
```

```
//端口位定义，可修改
```

```
sbit sda=P0^0;
```

```
sbit scl=P0^1;
```

```
//内部数据定义
```

```
uchar i2c_ad_main; //器件从地址
```

```
uchar i2c_ad_sub; //器件子地址
```

```
uchar *i2c_buf; //发送|接收数据缓冲区
```

```
uchar i2c_num; //发送|接收数据个数
```

```
#define ack 1 //主应答
```

```
#define no_ack 0 //从应答
```

#### 3.4.2C51 程序<i2cbus.c>代码：

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

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

```
#include <i2cbus.h>
```

```
/**/
```

```
//送起始位 sda=1->0
```

```
void i2c_start(void){
```

```
    scl=0;
```

```
    sda=1;
```

```

    _nop_();
    scl=1;
    nops();
    sda=0;
    nops();
    scl=0;
}
//*****
//送停止位 sda=0->1
void i2c_stop(void){
    scl=0;
    _nop_();
    sda=0;
    _nop_();
    scl=1;
    nops();
    sda=1;
    nops();
    scl=0;
}
//*****
//主应答(包含 ack:sda=0 和 no_ack:sda=0)
void i2c_ack_main(bit ack_main){
    scl=0;
    if(ack_main)sda=0; //ack 主应答
    else sda=1; //no_ack 无需应答
    nops();
    scl=1;
    nops();
    scl=0;
}
//*****
//字节发送程序
//发送 c(可以是数据也可是地址), 送完后接收从应答
//不考虑从应答位
void send_byte(uchar c){
    uchar i;
    for(i=0;i<8;i++){
        scl=0;
        if((c<<i) & 0x80)sda=1; //判断发送位
        else sda=0;
        _nop_();
        scl=1;
        nops();
    }
}

```

```

        scl=0;
    }
    nops();
    sda=1; //发送完 8bit, 释放总线准备接收应答位
    _nop_();
    scl=1;
    nops(); //sda 上数据即是从应答位
    scl=0; //不考虑从应答位|但要控制好时序
}
//*****
//字节接收程序
//接收器件传来的数据, 此程序应配合|主应答函数|i2c_ack_main()使用
//return: uchar 型 1 字节
uchar read_byte(void){
    uchar i;
    uchar c;
    c=0;
    scl=0;
    _nop_();
    sda=1; //置数据线为输入方式
    for(i=0;i<8;i++){
        _nop_();
        scl=0; //置时钟线为低, 准备接收数据位
        nops();
        scl=1; //置时钟线为高, 使数据线上数据有效
        _nop_();
        c<<=1;
        if(sda)c+=1; //读数据位, 将接收的数据存 c
    }
    scl=0;
    return c;
}
//*****
//向无子地址器件发送单字节数据
void send_to_byte(uchar ad_main,uchar c){
    i2c_start();
    send_byte(ad_main); //发送器件地址
    send_byte(c); //发送数据 c
    i2c_stop();
}
//*****
//向有子地址器件发送多字节数据
void send_to_nbyte(uchar ad_main,uchar ad_sub,uchar *buf,uchar num){
    uchar i;

```

```

i2c_start();
send_byte(ad_main); //发送器件地址
send_byte(ad_sub); //发送器件子地址
for(i=0;i<num;i++){
    send_byte(*buf); //发送数据*buf
    buf++;
}
i2c_stop();
}
//*****
//从无子地址器件读单字节数据
//function:器件地址，所读数据存在接收缓冲区当前字节
void read_from_byte(uchar ad_main,uchar *buf){
    i2c_start();
    send_byte(ad_main); //发送器件地址
    *buf=read_byte();
    i2c_ack_main(no_ack); //无需应答<no_ack=0>
    i2c_stop();
}
//*****
//从有子地址器件读多个字节数据
//function:
void read_from_nbyte(uchar ad_main,uchar ad_sub,uchar *buf,uchar num){
    uchar i;
    i2c_start();
    send_byte(ad_main);
    send_byte(ad_sub);
    for(i=0;i<num-1;i++){
        *buf=read_byte();
        i2c_ack_main(ack); //主应答<ack=1>
        buf++;
    }
    *buf=read_byte;
    buf++; //本次指针调整无意义，目的是操作后 buf 指向下一地址
    i2c_ack_main(no_ack); //无需应答<no_ack=0>
    i2c_stop();
}

```

#### 4、参考文献：

4. 1模拟I2C 与80C51 单片机的接口，李奇刚，[www.zlgmcu.com](http://www.zlgmcu.com)
4. 2模拟I2C总线C51程序软件包，广州周立功单片机发展有限公司 [www.zlgmcu.com](http://www.zlgmcu.com)
4. 3 单片机 C 程序设计及应用实例，胡伟\*季晓衡，北京，人民邮电出版社 2003/07