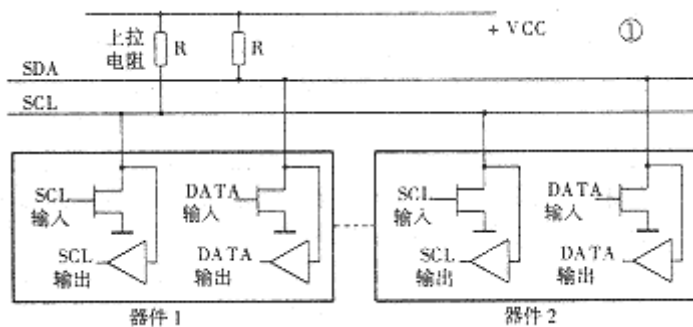


AT24C02 串行 E2PROM 的读写

I2C 总线是一种用于 IC 器件之间连接的二线制总线。它通过 SDA（串行数据线）及 SCL（串行时钟线）两根线在连到总线上的器件之间传送信息，并根据地址识别每个器件：不管是单片机、存储器、LCD 驱动器还是键盘接口。

1. I2C 总线的基本结构 采用 I2C 总线标准的单片机或 IC 器件，其内部不仅有 I2C 接口电路，而且将内部各单元电路按功能划分为若干相对独立的模块，通过软件寻址实现片选，减少了器件片选线的连接。CPU 不仅能通过指令将某个功能单元电路挂靠或脱离总线，还可对该单元的工作状况进行检测，从而实现对硬件系统的既简单又灵活的扩展与控制。I2C 总线接口电路结构如图 1 所示。



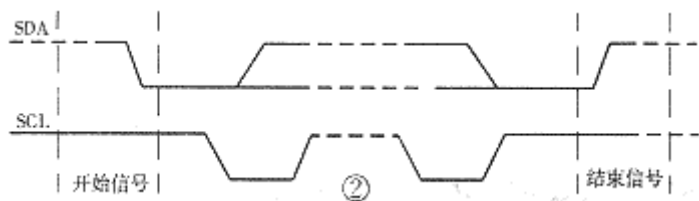
2. 双向传输的接口特性 传统的单片机串行接口的发送和接收一般都各用一条线，如 MCS51 系列的 TXD 和 RXD，而 I2C 总线则根据器件的功能通过软件程序使其可工作于发送或接收方式。当某个器件向总线上发送信息时，它就是发送器(也叫主器件)，而当其从总线上接收信息时，又成为接收器(也叫从器件)。主器件用于启动总线上传送数据并产生时钟以开放传送的器件，此时任何被寻址的器件均被认为是从器件。I2C 总线的控制完全由挂接在总线上的主器件送出的地址和数据决定。在总线上，既没有中心机，也没有优先机。

总线上主和从(即发送和接收)的关系不是一成不变的，而是取决于此时数据传送的方向。SDA 和 SCL 均为双向 I/O 线，通过上拉电阻接正电源。当总线空闲时，两根线都是高电平。连接总线的器件的输出级必须是集电极或漏极开路，以具有线“与”功能。I2C 总线的数据传送速率在标准工作方式为 100kbit/s，在快速方式下，最高传送速率可达 400kbit/s。

3. I2C 总线上的时钟信号 在 I2C 总线上传送信息时的时钟同步信号是由挂接在 SCL 时钟线上的所有器件的逻辑“与”完成的。SCL 线上由高电平到低电平的跳变将影响到这些器件，一旦某个器件的时钟信号下跳为低电平，将使 SCL 线一直保持低电平，使 SCL 线上的所有器件开始低电平期。此时，低电平周期短的器件的时钟由低至高的跳变并不能影响 SCL 线的状态，于是这些器件将进入高电平等待的状态。

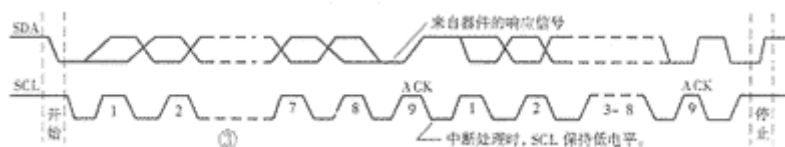
当所有器件的时钟信号都上跳为高电平时，低电平期结束，SCL 线被释放返回高电平，即所有的器件都同时开始它们的高电平期。其后，第一个结束高电平期的器件又将 SCL 线拉成低电平。这样就在 SCL 线上产生一个同步时钟。可见，时钟低电平时由时钟低电平期最长的器件确定，而时钟高电平时由时钟高电平期最短的器件确定。

4. 数据的传送 在数据传送过程中，必须确认数据传送的开始和结束。在 I2C 总线技术规范中，开始和结束信号（也称启动和停止信号）的定义如图 2 所示。当时钟线 SCL 为高电平时，数据线 SDA 由高电平跳变为低电平定义为“开始”信号；当 SCL 线为高电平时，SDA 线发生低电平到高电平的跳变为“结束”信号。开始和结束信号都是由主器件产生。在开始信号以后，总线即被认为处于忙状态；在结束信号以后的一段时间内，总线被认为是空闲的。



I2C 总线的数据传输格式是：在 I2C 总线开始信号后，送出的第一个字节数据是用来选择从器件地址的，其中前 7 位为地址码，第 8 位为方向位(R/W)。方向位为“0”表示发送，即主器件把信息写到所选择的从器件；方向位为“1”表示主器件将从从器件读信息。开始信号后，系统中的各个器件将自己的地址和主器件送到总线上的地址进行比较，如果与主器件发送到总线上的地址一致，则该器件即为被主器件寻址的器件，其接收信息还是发送信息则由第 8 位(R/W)确定。

在 I2C 总线上每次传送的数据字节数不限，但每一个字节必须为 8 位，而且每个传送的字节后面必须跟一个认可位（第 9 位），也叫应答位（ACK）。数据的传送过程如图 3 所示。每次都是先传最高位，通常从器件在接收到每个字节后都会作出响应，即释放 SCL 线返回高电平，准备接收下一个数据字节，主器件可继续传送。如果从器件正在处理一个实时事件而不能接收数据时，（例如正在处理一个内部中断，在这个中断处理完之前就不能接收 I2C 总线上的数据字节）可以使时钟 SCL 线保持低电平，从器件必须使 SDA 保持高电平，此时主器件产生 1 个结束信号，使传送异常结束，迫使主器件处于等待状态。当从器件处理完毕时将释放 SCL 线，主器件继续传送。



当主器件发送完一个字节的的数据后，接着发出对应于 SCL 线上的一个时钟（ACK）认可位，在此时钟内主器件释放 SDA 线，一个字节传送结束，而从器件的响应信号将 SDA 线拉成低电平，使 SDA 在该时钟的高电平期间为稳定的低电平。从器件的响应信号结束后，SDA 线返回高电平，进入下一个传送周期。

I2C 总线还具有广播呼叫地址用于寻址总线上所有器件的功能。若一个器件不需要广播呼叫寻址中所提供的任何数据，则可以忽略该地址不作响应。如果该器件需要广播呼叫寻址中提供的数据，则应对地址作出响应，其表现为一个接收器。

5. 总线竞争的仲裁 总线上可能挂接有多个器件，有时会发生两个或多个主器件同时想占用总线的情况。例如，多单片机系统中，可能在某一时刻有两个单片机要同时向总线发送数据，这种情况叫做总线竞争。I2C 总线具有多主控能力，可以对发生在 SDA 线上的总线竞争进行仲裁，其仲裁原则是这样的：当多个主器件同时想占用总线时，如果某个主器件发送高电平，而另一个主器件发送低电平，则发送电平与此时 SDA 总线电平不符的那个器件将自动关闭其输出级。总线竞争的仲裁是在两个层次上进行的。首先是地址位的比较，如果主器件寻址同一个从器件，则进入数据位的比较，从而确保了竞争仲裁的可靠性。由于是利用 I2C 总线上的信息进行仲裁，因此不会造成信息的丢失。

6. I2C 总线接口器件 目前在视频处理、移动通信等领域采用 I2C 总线接口器件已经比较普遍。另外，通用的 I2C 总线接口器件，如带 I2C 总线的单片机、RAM、ROM、A/D、D/A、LCD 驱动器等器件，也越来越多地应用于计算机及自动控制系统中。

AT24C02 是美国 ATMEL 公司的低功耗 CMOS 串行 EEPROM，它是内含 256×8 位存储空间，具有工作电压宽（2.5~5.5V）、擦写次数多（大于 10000 次）、写入速度快（小于 10ms）等特点。

电子制作实验室网站提供的 AT89C51 试验开发板上带有一片 AT24C02 芯片可以提供试验。下面是它的电路图。

图中 AT24C02 的 1、2、3 脚是三条地址线，用于确定芯片的硬件地址。在 AT89S52 试验开发板上它们都接地，第 8 脚和第 4 脚分别为正、负电源。第 5 脚 SDA 为串行数据输入/输出，数据通过这条双向 I2C 总线串行传送，在 AT89S52 试验开发板上和单片机的 P2.1 连接。第 6 脚 SCL 为串行时钟输入线，在 AT89C51 试验开发板上和单片机的 P2.0 连接。SDA 和 SCL 都需要和正电源间各接一个 5.1K 的电阻上拉。第 7 脚

需要接地。

24C02 中带有片内地址寄存器。每写入或读出一个数据字节后，该地址寄存器自动加 1，以实现对下一个存储单元的读写。所有字节均以单一操作方式读取。为降低总的写入时间，一次操作可写入多达 8 个字节的数据。

;这是将 0600H 地址中以下的 8 个数据写到 24C02 的 01H 为首址单元中去的汇编程序

ORG 0000H

SCL BIT P2.0;定义 24C02 的串行时钟线

SDA BIT P2.1;定义 24C02 的串行数据线

LJMP START

START:LCALL STAR;调用

MOV R2,#08H;一个数据有 8 位

MOV DPTR,#0600H;定义源数据的位置

LOOP:MOV A,#00H

MOVC A,@A+DPTR

LCALL SDATA

LCALL ACK

JC LOOP

INC DPTR

DJNZ R2,LOOP

LCALL STOP;调用停止子程序

STAR:SETB SDA

SETB SCL

NOP

NOP

NOP

NOP

CLR SDA

NOP

NOP

NOP

NOP

CLR SCL

RET

SDATA:MOV R0,#08H

LOOP0:RLC A

MOV SDA,C

NOP

NOP

SETB SCL

```

NOP
NOP
NOP
NOP
CLR SCL
DJNZ R0,LOOP0
RET
ACK:SETB SDA
NOP
NOP
SETB SCL
NOP
NOP
NOP
NOP
MOV C,SDA
CLR SCL
RET
STOP:CLR SDA
NOP
NOP
NOP
NOP
SETB SCL
NOP
NOP
NOP
NOP
SETB SDA
NOP
NOP
NOP
NOP
RET
ORG 0600H
DB 0A0H,10H,01H,02H,03H,04H,05H,06H
END

```

读写子程序如下：

```

;写串行 E2PROM 子程序 EEPW
; R3=10100000（命令 1010+器件 3 位地址+读/写。 器件地址一个芯片，是 000）
; (R4)=片内字节地址
; (R1)=欲写数据存放地址指针

```

; (R7)=连续写字字节数 n

EEPW: MOV P1,#0FFH

CLR P1.0 ;发开始信号

MOV A,R3 ;送器件地址

ACALL SUBS

MOV A,R4 ;送片内字节地址

ACALL SUBS

AGAIN: MOV A,@R1

ACALL SUBS ;调发送单字节子程序 INC R1

DJNZ R7,AGAIN;连续写 n 个字节

CLR P1.0 ;SDA 置 0, 准备送停止信号

ACALL DELAY ;延时以满足传输速率要求

SETB P1.1 ;发停止信号

ACALL DELAY

SETB P1.0

RET

SUBS: MOV R0,#08H;发送单字节子程序

LOOP: CLR P1.1

RLC A

MOV P1.0,C

NOP

SETB P1.1

ACALL DELAY

DJNZ R0,LOOP ;循环 8 次送 8 个 bit

CLR P1.1

ACALL DELAY

SETB P1.1

REP: MOV C,P1.0

JC REP ;判应答到否, 未到则等待

CLR P1.1

RET

DELAY: NOP

NOP

RET

;读串行 E2PROM 子程序 EEPR

; (R1)=欲读数据存放地址指针

;; R3=10100001 (命令 1010+器件 3 位地址+读/写。 器件地址一个芯片, 是 000)

; (R4)=片内字节地址

; (R7)=连续读字节数

```

EEPR: MOV  P1,#0FFH
      CLR   P1.0    ;发开始信号
      MOV   A,R3     ;送器件地址
      ACALL SUBS     ;调发送单字节子程序
      MOV   A,R4     ;送片内字节地址
      ACALL SUBS
      MOV   P1,#0FFH
      CLR   P1.0     ;再发开始信号
      MOV   A,R3
      SETB  ACC.0    ;发读命令
      ACALL SUBS
MORE: ACALL SUBR
      MOV   @R1,A
      INC   R1
      DJNZ  R7,MORE
      CLR   P1.0
      ACALL DELAY
      SETB  P1.1
      ACALL DELAY
      SETB  P1.0 ;送停止信号
      RET

SUBR: MOV   R0,#08H ;接受单字节子程序
LOOP2: SETB  P1.1
      ACALL DELAY
      MOV   C,P1.0
      RLC   A
      CLR   P1.1
      ACALL DELAY
      DJNZ  R0,LOOP2
      CJNE  R7,#01H,LOW
      SETB  P1.0    ;若是最后一个字节置 A=1
      AJMP SETOK
LOW: CLR   P1.0    ;否则置 A=0
SETOK: ACALL DELAY
      SETB  P1.1
      ACALL DELAY
      CLR   P1.1
      ACALL DELAY
      SETB  P1.0    ;应答毕,SDA 置 1
      RET

```

程序中多处调用了 DELAY 子程序(仅两条 NOP 指令),这是为了满足 I2C 总线上数据传送速率的要求,只有当 SDA 数据线上的数据稳定下来之后才能进行读写(即 SCL 线发出正脉冲)。另外,在读最后一数据字节时,置应答信号为“1”,表示读操作即将完成

