



哈爾濱工業大學 远程教育学院

第7章 MCS-51的串行口



- (1) 串行通讯的基本概念及**MCS-51**串行口的**SFR**
- (2) 串行口的工作方式
- (3) 串行口波特率的制定
- (4) 串行口的编程方法
- (5) 双机通讯的实现

提高:

MCS51的多机通讯

MCS51与**PC**机的通讯

7.1 串行通讯的基本概念



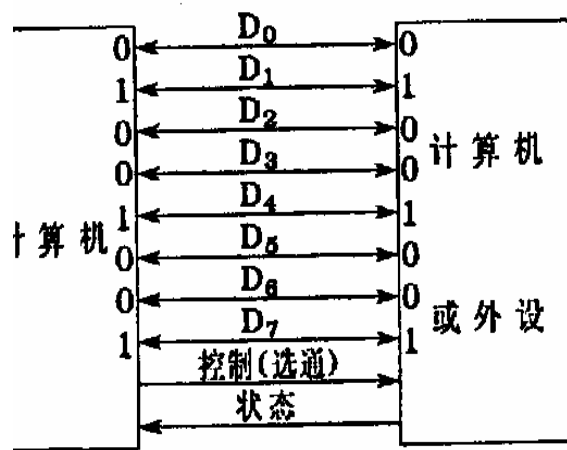
通讯的基本方式:

串行通讯与并行通讯

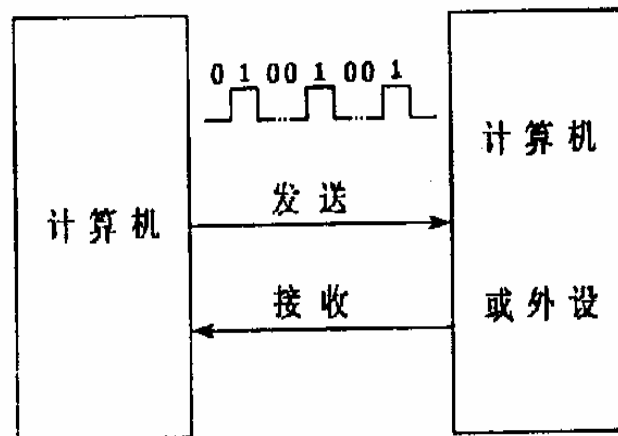
串行同步通讯、串行异步通讯

单工、半双工和全双工串行通讯

并行通讯、串行通讯



并行通讯

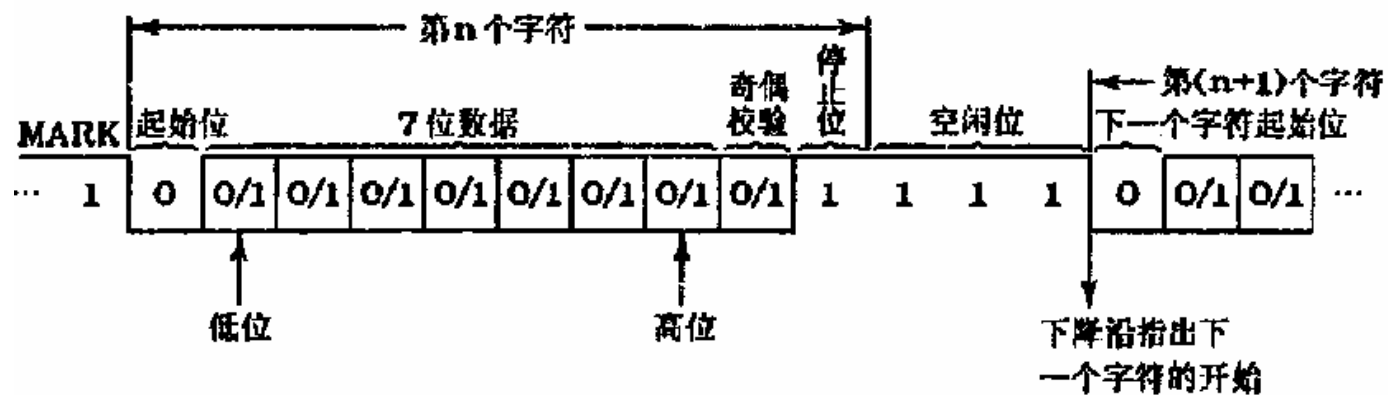


串行通讯

并行通讯：数据各位同时传送

串行通讯：数据一位一位顺序传送

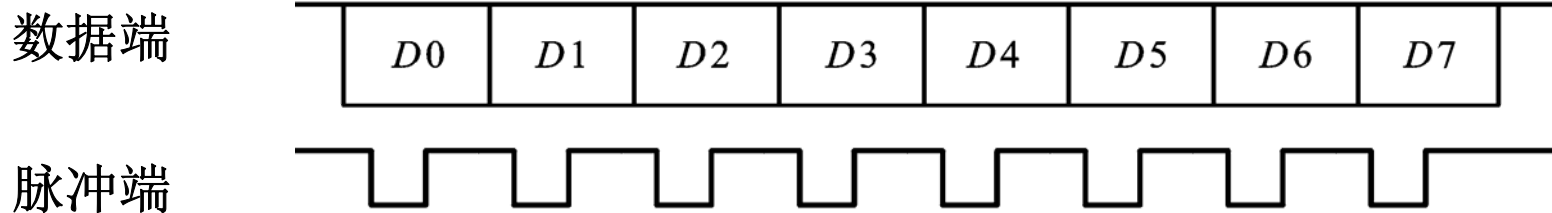
串行异步通讯



异步通讯的数据传送格式

串行同步通讯

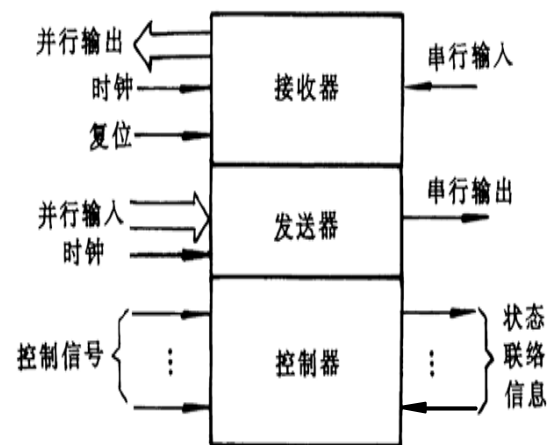
一种串行同步通讯的方法：需要有数据端、同步时钟端



7.2 MCS-51串行口的基本结构

发送器：将CPU送来的并行数据，通过发送移位寄存器变为串行数据逐位发送，并自动添加起始位、停止位和奇偶校验位。

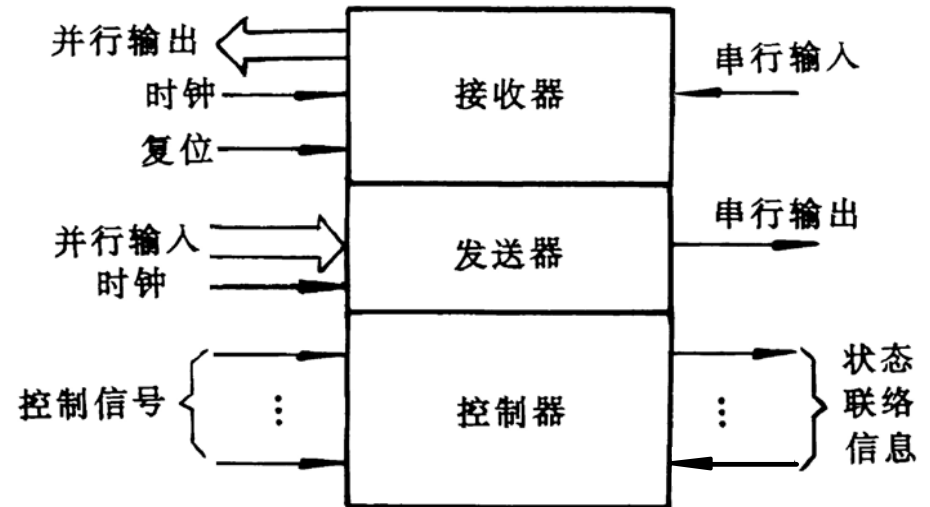
接收器：将接收到的串行数据，经接收移位寄存器变为并行数据，去掉起始位、停止位后，将正确数据送到接收数据缓冲器，供CPU读取。



串行接口逻辑部件框图

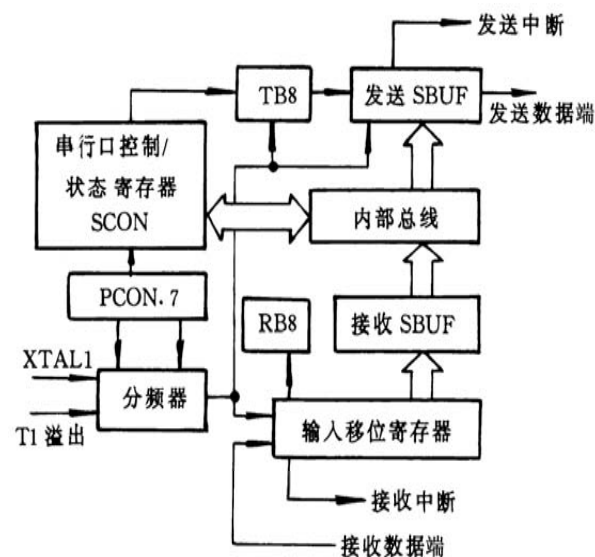
串行口的基本结构

控制器：接收CPU送来的编程命令，按选定方式实现对串行接口的控制。主要控制参数有：传送波特率的选择、数据格式的选择等。另外还与CPU交换状态信号和中断信号。



串行口的基本结构

- **SCON**为串行口控制/状态寄存器，通过编程写入**SCON**的控制位可选择串行口的工作方式，读出**SCON**的状态位可查询串行口的工作状态。
- 定时/计数器**T1**被用做串行口的波特率发生器，由可控分频电路决定数据传送的波特率。
- 串行口信息的发送/接收是通过写/读数据缓冲器**SBUF**来实现的。



7.3 串行口的SFR



串行数据缓冲器SBUF(99H)

串行口有两个8位数据缓冲器，一个是发送数据缓冲器SBUF，一个是接收数据缓冲器SBUF。它们共用一个地址99H，发送SBUF只能写入，而接收SBUF只能读出。

通过指令区分：*MOV SBUF,A* *MOV A,SBUF*

控制寄存器SCON (98H)

SM0 SM1 SM2 REN TB8 RB8 TI RI

节电控制寄存器PCON(87H)

SMOD (D7)

复位后状态均为00H

7.3.1 串行口控制寄存器SCON

串行口控制寄存器 SCON(字节地址 98H)的格式如下:

	D7	D6	D5	D4	D3	D2	D1	D0
	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
位地址	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H

SM0、SM1: 串行口4种工作方式选择位

SM2: 方式2和方式3中的多机通信控制位。在方式2和方式3中, 如**SM2=1**, 则当接收到的第9位数据(**RB8**)为0时, 不激活接收中断标志**RI**(**RI=1**)。在方式1中, 如**SM2=1**, 则只有当接收到有效停止位时才激活**RI**, 若没有接收到有效停止位, 则**RI**清0。在方式0中, **SM2**必须为0。

REN: 允许串行接收位。**REN=1**允许接收;

REN=0禁止接收, 由软件置位或清零。

TB8: 是工作在方式2和3时, 要发送的第9位数据。可由软件置1或清0。在许多的通讯协议中该位是奇偶校验位。在多机通讯中用来表示是地址帧还是数据帧, **TB8=1**为地址帧, **TB8=0**为数据帧。

串行口控制寄存器SCON



RB8: 当工作在方式2和3时，为接收到的第9位数据。

TI: 发送中断标志位，在方式0串行发送第8位数据结束时由硬件置1，或在其它方式串行发送停止位后置1，并申请中断。**CPU**响应中断后，可发送下一帧数据，**TI**必须由软件清0。

RI: 为接收结束中断标志。在方式0中，当接收到一帧数据的第8位结束时，**RI=1**。在其他方式中，接收到停止位后，**RI=1**，表示可读取接收**SBUF**中的内容。**RI**必须用软件清0。

每次发送和接收前，要注意先将**TI**和**RI**标志位清0。

串行口作为一个中断源，中断入口地址只有一个(0023H)，当用中断方式同时进行发送和接收时，可通过查询**TI**和**RI**状态，确定是发送还是接收结束引起的中断。

7.3.2 节电控制寄存器PCON

特殊功能寄存器 PCON 没有位寻址功能,字节地址为 87H:



SMOD:波特率系数选择位。

PCON主要用于节电运行方式控制，PCON.7用做串行口波特率加倍控制位。当设定SMOD=1时，波特率加倍，当SMOD=0时，波特率不加倍。

7.4 串行口的工作方式

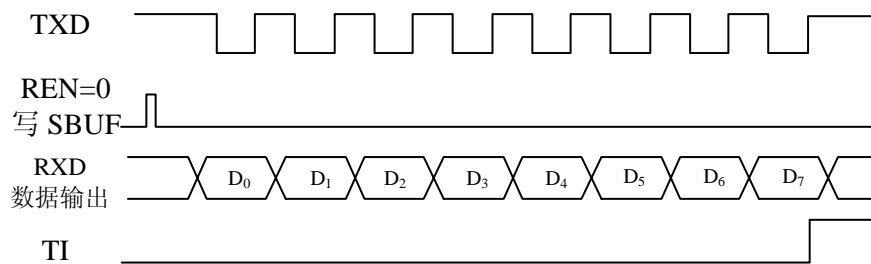
SM0、SM1工作方式选择位

串行口工作方式

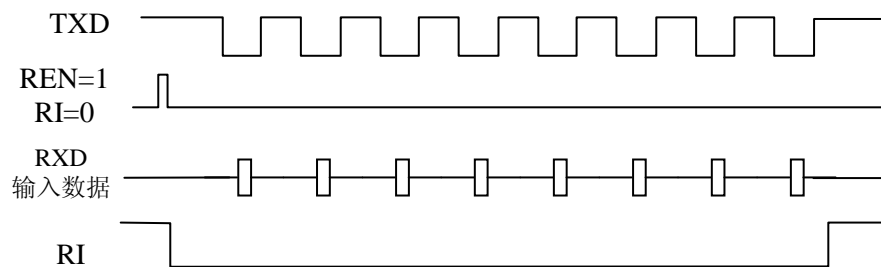
SM0	SM1	方式	功能说明
0	0	0	移位寄存器方式(用于扩展 I/O 口)
0	1	1	8 位 UART, 波特率可变(T1 溢出率/n)
1	0	2	9 位 UART, 波特率为 $f_{osc}/64$ 或 $f_{osc}/32$
1	1	3	9 位 UART, 波特率可变(T1 溢出率/n)

7.4.1 工作方式 0 : (SM0=0 SM1=0)

串行口方式0为8位数据同步移位寄存器方式。**RXD**引脚为同步数据的输入/输出端，**TXD**引脚为同步移位脉冲输出端。一个数据帧为8位，每一个移位脉冲对应数据帧的一个数据位。方式0的波特率固定为 $f_{osc}/12$ 。



(a) 发送时序图



(b) 接收时序图

串行口方式 0 工作时序图

方式0的说明:



写入**SCON**使**REN=0**, **TI=0**; 然后执行写入**SBUF**的指令, 8位数据装入发送**SBUF**, 硬件启动串行口发送器进行一次发送。通过发送移位寄存器将8位数据逐位送到**RXD**引脚, 每个机器周期内发送一位数据。对应发送数据的每一位, 由**TXD**引脚同步输出一个移位脉冲。发送完一帧, 自动置位**TI**标志, 并申请串行口中断。若**CPU**响应中断, 则将从**0023H**单元开始执行串行口的中断服务程序。

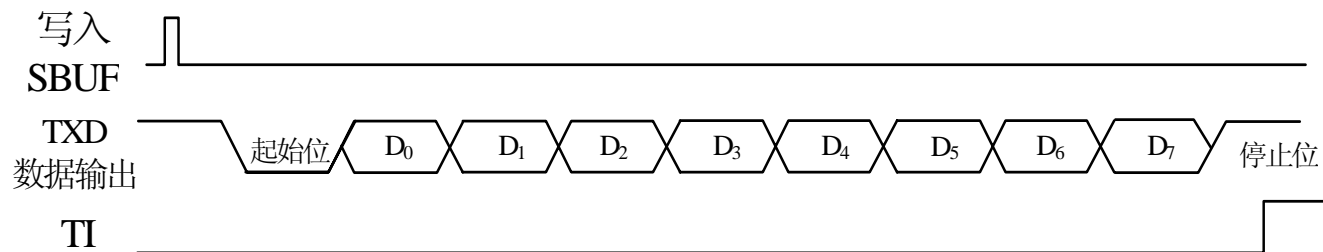
写入**SCON**使**REN = 1**、**RI = 0**, 将启动串行口接收器开始接收。由**TXD**输出移位脉冲, 每对应一个移位脉冲, 采样一次**RXD**引脚信号。每个机器周期采样一次**RXD**引脚, 采到的数据位送入接收移位寄存器后, 接收移位寄存器左移一位。接收完一帧, 自动置位**RI**标志, 申请串行口中断, 并将接收下来的8位数据装入接收**SBUF**。若**CPU**响应中断, 则从**0023H**单元开始执行串行口的中断服务程序。

*串行口方式0常用于扩展单片机的并行I/O口, 也可外接串行同步I/O设备。用方式0输入/输出数据皆通过**RXD**端, 因此方式0为半双工方式, 只能分时进行发送和接收。*

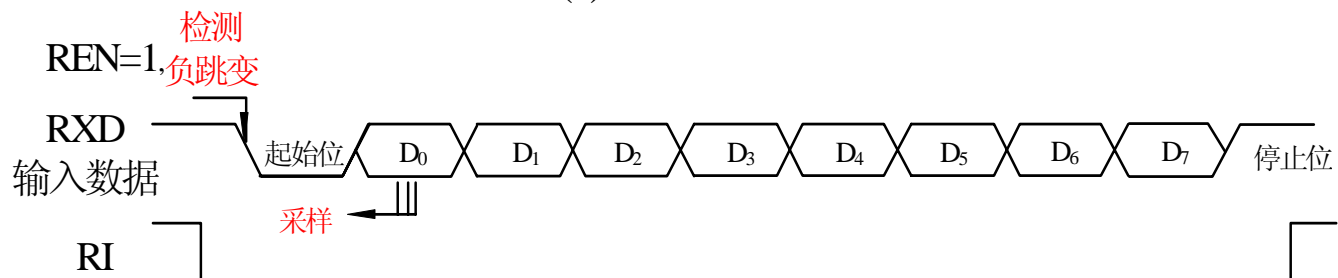
7.4.2 方式1 (SM0=0 SM1=1)

功能说明：串行口方式1为8位数据全双工异步通信方式。TXD为发送数据端，RXD为接收数据端。一个数据帧为10位，包含8位数据位，1个起始位“0”和1个停止位“1”。串行口用方式1工作时，使用定时器T1作为波特率发生器，波特率的设定以T1的溢出率为基准，波特率计算式为： $(2^{\text{SMOD}}/32) \times \text{T1溢出率}$ 。

方式1的时序图



(a) 发送时序图



(b) 接收时序图

串行口方式1 工作时序图

方式1功能说明

执行写入SBUF指令，将8位数据装入发送SBUF，并启动发送器进行一次发送：先将起始位“0”送到TXD引脚，再从低到高逐位发送8位数据，最后发送停止位“1”。每位发送占用的时间由设定的波特率决定。发送完一帧数据，置位TI标志，申请串行口中断。

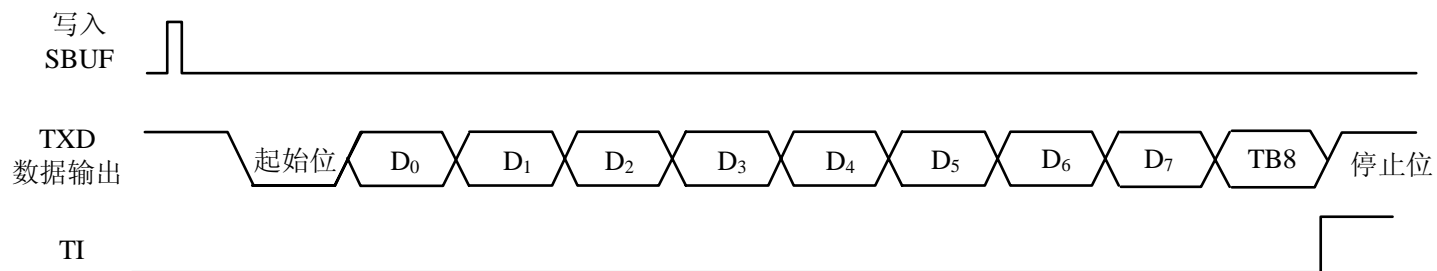
当REN=1时，允许接收器准备接收。开始，以16倍于波特率的速率检测RXD引脚的负跳变，当检测到负跳变后，启动一次接收，接收完一帧信息，判断是否满足条件： $RI = 0$ ，而且停止位为1(或SM2=0)。若同时满足这两个条件，则置位RI标志，申请串行口中断，并将接收的8位数据装入接收SBUF，停止位装入RB8；否则接收无效，丢失所接收的一帧信息，且不影响RI标志。一位时间以后，接收器重新开始检测RXD端的负跳变，以准备接收下一帧数据。

7.4.3 方式2、方式3

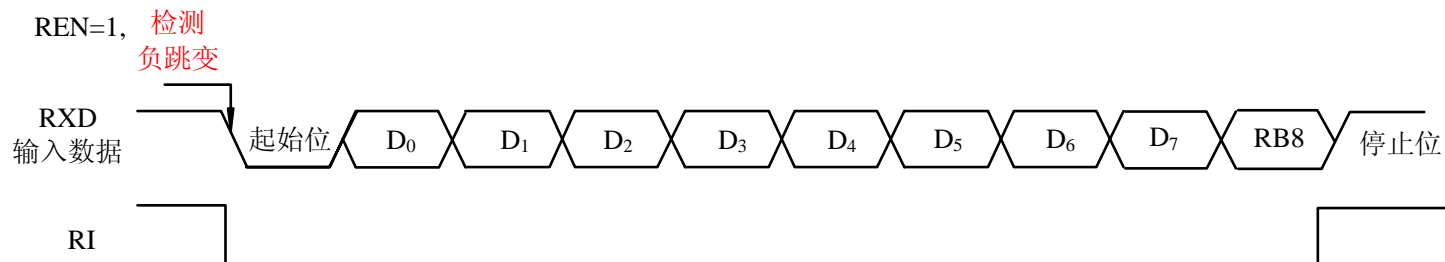
串行口方式2和方式3皆为9位数据全双工异步通信方式，比串行口方式1增加了第9位数据位。TXD为发送数据端，RXD为接收数据端。一帧信息有11位，包括9位数据位，1个起始位“0”和1个停止位“1”。发送数据的第9位在TB8中，接收数据的第9位存入RB8中。第9位数据位也称为可编程位，常用做奇偶校验位或多机通信中的地址/数据识别位。

串行口的方式2与方式3的唯一差别是波特率设定方法不同，方式2的波特率以系统时钟为基准，计算式为： $(2^{\text{SMOD}}/64) \times f_{\text{osc}}$ 。方式3的波特率与方式1的相同，计算式为： $(2^{\text{SMOD}}/32) \times \text{T1溢出率}$ 。

方式2、3的时序



(a) 发送时序图



(b) 接收时序图

串行口方式 2、方式 3 工作时序图

方式2、3的功能说明



发送时，应先将发送数据的第9位送入TB8，再写入8位数据到发送SBUF，使发送器启动一次发送，将一帧数据逐位送到TXD引脚：发送起始位“0”之后，再发送SBUF中的8位数据，接着发送TB8中的第9位，最后发送停止位“1”。一帧发送结束，置位TI标志，申请串行口中断。

方式2和方式3的接收过程和方式1的接收过程基本相同，但RB8中装入的是接收数据的第9位，而不是停止位。接收有效必须同时满足的两个条件改为：RI=0，且接收的第9位数据位为1(或SM2=0)。当正确接收到一帧数据后，前面8位数据进入接收SBUF，第9位数据进入RB8。

7.5 波特率的制定

波特率的定义：串行口每秒钟发送或者接收的位数。

串行口方式0的波特率是固定的，为系统时钟的12分频 ($f_{osc}/12$)，即每个机器周期传送一位数据位。

串行口用方式2工作时，波特率为 $(2^{SMOD}/64) \times f_{osc}$ 。SMOD 为PCON中的D₇位，即波特率加倍选择位。方式2可有两种波特率供选择，当SMOD=0时，波特率为 $f_{osc}/64$ ；当SMOD=1时，波特率则为 $f_{osc}/32$ 。

串行口方式1和方式3用定时器T1作为波特率发生器，其波特率有多种选择，与T1的溢出率有关。

$$\text{波特率} = (2^{SMOD} / 32) \times T1 \text{溢出率}$$

T1溢出率：T1溢出时间的倒数。

T1用于波特率发生器时一般工作于非门控定时器方式2，即常数自动重装入方式。

$$\text{T1 溢出率} = \frac{f_{\text{osc}}}{12 \times (2^8 - \text{TH1})}$$

$\text{TH1} = \text{TL1}$ 为T1的时间常数

$$\text{串行口方式1和方式3的波特率} = \frac{2^{\text{SMOD}}}{32} \times \frac{f_{\text{osc}}}{12 \times (256 - \text{TH1})}$$

定时器T1产生波特率的计算

串行口用方式1和方式3工作时，当需要很低的波特率时，也可选择T1工作于方式1，但需利用T1溢出中断来实现软件重装初值，T1用方式1工作时溢出率为：

$$\frac{f_{\text{osc}}}{12[2^{16} - (\text{TH1TL1}) + N]}$$

N为T1溢出后用软件重装初值所需的机器周期数。

T1产生的常用波特率

串 行 口方式	波特率 /(b/s)	f_{osc}/MHz	SMOD	定时/计数器 T1		
				C/T	方式	计数初值
0	1M	12	×	×	×	×
2	187.5K	12	0	×	×	×
	375K	12	1	×	×	×
1(3)	62.5K	12	1	0	2	FFH
	19.2K	11.059	1	0	2	FDH
	9.6K	11.059	0	0	2	FDH
	4.8K	11.059	0	0	2	FAH
	2.4K	11.059	0	0	2	F4H
	1.2K	11.059	0	0	2	E8H
	137.5	11.059	0	0	2	1DH
	110	6	0	0	2	72H
	110	12	0	0	1	FEEDH

波特率计算的几点注意事项：



1) 波特率的相对误差不大于**2.5%**

2) 注意**SMOD**位对波特率的影响

例如：通讯波特率设为**2400**，晶振频率为**6MHz**，计算**T1**的计时常数。

当 **SMOD=0** 时，计时常数约为 **249**，相对误差 **7%**。

当 **SMOD=1** 时，计时常数约为 **243**，相对误差 **0.16%**。

7.6 串行口的编程举例



设定波特率

设定串行口的工作方式


设计中断服务程序或查询程序

注意对TI、RI的处理（硬件置位、软件清零）

例1： 方式2发送50H~5FH中内容

	D7	D6	D5	D4	D3	D2	D1	D0
	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
位地址	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H

```
ORG    0000H; 复位入口
AJMP   MAINT
ORG    0023H; 串行口中断入口
AJMP   TRANI
ORG    0100H; 主程序入口
MAINT: MOV    SCON, #80H      ; 串行口初始化, 用
                                   ; 方式2发送, TI=0
MOV    PCON, #80H           ; 波特率=  $f_{osc} / 32$ 
SETB   EA
SETB   ES    ; 允许串行口中断
MOV    R0, #50H    ; 设数据指针
MOV    R7, #10H    ; 数据长度
```



```

LOOP:  MOV    A, @R0; 取一个字符
        MOV    C, P    ; 加奇偶校验
        MOV    TB8, C
        MOV    SBUF, A    ; 发送一个字符
HERE:  SJMP    HERE    ; CPU执行其他任务, 同时等待发送结束
TRANI: CLR    TI        ; 中断服务程序, 发送结束标志清0
        DJNZ   R7, NEXT    ; 发送完全部字符?
        CLR    ES        ; 已发送完, 关闭串行口中断
        SJMP   TEND
NEXT:  INC     R0        ; 未发送完, 修改指针
        MOV    A, @R0; 取下一个字符
        MOV    C, P    ; 加奇偶校验
        MOV    TB8, C
        MOV    SBUF, A    ; 再次发送一个字符
TEND:  RETI            ; 中断返回主程序

```

7.7 MCS-51双机串行通讯的实现



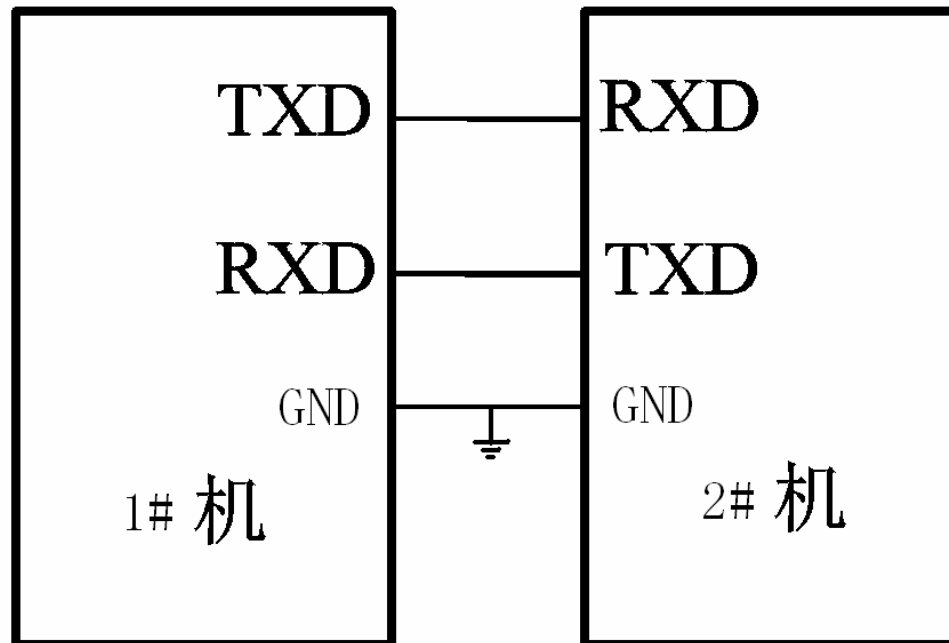
如何进行物理连线？

波特率的选择？

工作方式的选择？

如何编程实现？

7.7.1 正确进行物理接线





设定一致的波特率

设定一致的串行口工作方式

分别设计中断服务程序或查询程序

注意对TI、RI的处理

(硬件置位、软件清零)

7.7.2 方式3发送50H~5FH的内容



待发送的16个字节存放在片内RAM中

地址为50H~5FH

串行口波特率为2400

波特率发生器T1工作于方式2， 且SMOD=0

晶振为11.0592MHz

计算得到TL1的初值为F4H。

串行口工作于方式3

第9位发送奇偶校验位

例2： 方式3发送50H~5FH中内容-1

```

        ORG    0000H    ; 复位入口
        AJMP   MAIN
        ORG    0023H    ; 串行口中断入口
        AJMP   L1
        ORG    0100H    ; 主程序入口
MAIN:    MOV    SCON, #0C0H    ; 串行口初始化, 用方式2发送, TI=0
        MOV    PCON, #00H    ; SMOD=0
        MOV    TMOD, #20H    ; T1方式2定时
        MOV    TL1, #0F4H    ; 写入T1时间常数
        MOV    TH1, #0F4H
        SETB   TR1          ; 启动T1
        SETB   EA
        SETB   ES          ; 允许串行口中断
        MOV    R0, #50H    ; 设数据指针
        MOV    R7, #10H    ; 数据长度
LOOP:    MOV    A, @R0      ; 取一个字符
        MOV    C, P        ; 加奇偶校验
        MOV    TB8, C
        MOV    SBUF, A     ; 发送一个字符
HERE:    SJMP   HERE       ; CPU原地循环, 等待发送结束
    
```

	D7	D6	D5	D4	D3	D2	D1	D0
	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
位地址	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H

D7	D6	D5	D4	D3	D2	D1	D0
GATE	$\overline{C/T}$	M1	M0	GATE	$\overline{C/T}$	M1	M0
← T1 方式字段 →				← T0 方式字段 →			

例2： 方式3发送50H~5FH中内容-2



```
L1:
    CLR        TI                ; 中断服务程序，
                                发送结束标志位清0
    DJNZ       R7, NEXT        ; 发送完全部字符？
    CLR        ES                ; 已发送完，关闭串行口中断
    SJMP       TEND
NEXT:
    INC        R0                ; 未发送完，修改指针
    MOV        A, @R0           ; 取下一个字符
    MOV        C, P             ; 加奇偶校验
    MOV        TB8, C
    MOV        SBUF, A          ; 再次发送一个字符
TEND:
    RETI                        ; 中断返回主程序
```

7.7.3 方式3接收

接收到的16个字节存放在片内RAM中

地址为50H~5FH

串行口波特率为2400

波特率发生器T1工作于方式2， 且SMOD=0

晶振为11.0592MHz

计算得到TL1的初值为F4H。

串行口工作于方式3

奇偶校验方式判断接收数据是否出错

设校验出错标志位为F0

方式3接收-查询

```
MAIN: MOV     SCON, #0D0H ; 串行口方式3允许接收, RI=0
      MOV     PCON, #00H  ; SMOD=0
      MOV     TMOD, #20H  ; T1方式2定时
      MOV     TL1, #0F4H  ; 写入T1时间常数
      MOV TH1, #0F4H
      SETB    TR1         ; 启动T1
      LCALL   AAAA
      SJMP    $
```

	D7	D6	D5	D4	D3	D2	D1	D0
	SM0	SM1	SM2	REN	TB8	RB8	T1	RI
位地址	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H

D7	D6	D5	D4	D3	D2	D1	D0
GATE	C/T	M1	M0	GATE	C/T	M1	M0
← T1 方式字段 →				← T0 方式字段 →			

方式3接收-查询

```
AAAA: MOV      R0, #50H ; 设数据指针
      MOVR7, #10H ; 数据长度
WAIT: JBC      RI, NEXT ; 查询RI标志, 等待串行口接收结束
      SJMP
NEXT: MOV      A, SBUF ; 读取一个接收字符
      JNB      P, COMP ; 进行奇偶校验
      JNB      RB8, ERR ; P≠RB8, 数据出错
      SJMP      RIGHT ; P=RB8, 数据正确
COMP: JB       RB8, ERR
RIGHT: MOV     @R0, A ; 保存一个字符到片内RAM
      INC R0 ; 修改指针
      DJNZ R7, WAIT ; 全部字符接收完?
      CLR F0 ; F0 =0, 接收数据全部正确
      RET ; 返回主程序
ERR: SETB      F0 ; F0 =1, 接收数据出错
      RET ; 返回主程序
```

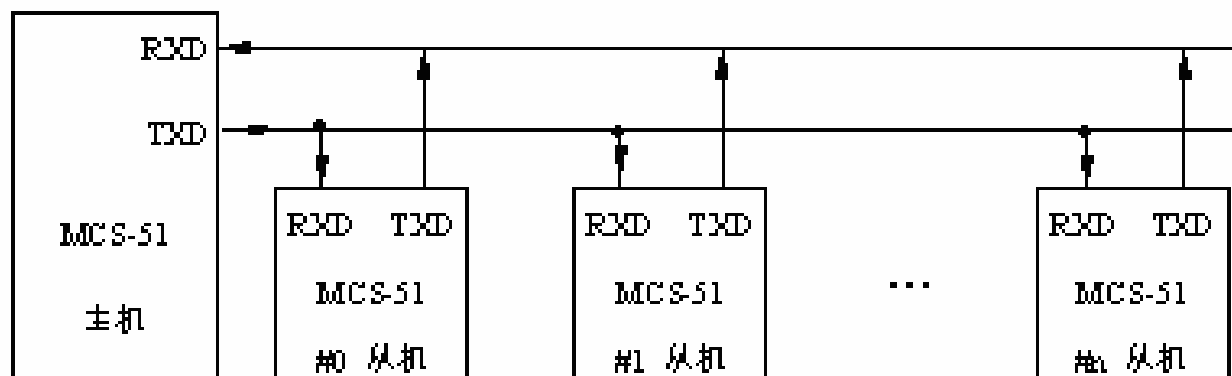


物理接线: TXD-RXD、RXD-TXD、共地
必须一致的波特率
必须一致的工作方式
各自的发送和接收程序可以不同

7.4 提高：多机通讯

主机与各从机实现全双工通信，而各从机之间只能通过主机交换信息。

下图是由MCS-51构成的的集散式分布系统



7.4.1 多机通讯的基本原理



多机通信控制位**SM2**的使用：

■ 当串行口以方式**2**(或方式**3**)接收时，若**SM2=1**，则只有接收到第**9**位数据为**1**，才将接收的数据装入接收**SBUF**，并置位**RI**标志；否则，不会置位**RI**，也不产生中断请求，一帧信息被丢弃。

■ 若**SM2=0**，则接收到的第**9**位不论为**0**还是为**1**，都会将数据装入**SBUF**，并置位**RI**标志。根据这一功能，可实现多机系统中的主从一对一通信。



现将多机通信步骤概述如下：

- 1) 令所有从机**SM2=1**，处于只接收地址帧状态（第9位为1）。
- 2) 主机令**TB8=1**，并发送地址呼叫信息。
- 3) 各从机接收到地址帧后，将接收到的地址与本机地址编号相比较，确认是否是被呼叫目标。
- 4) 被呼叫从机使**SM2清0**，变为接收数据帧状态，而未被呼叫从机仍保持**SM2=1**。
- 5) 主机使**TB8=0**，发送控制或数据信息。
- 6) 只有**SM2=0**的从机能接收到控制或数据信息，仍保持**SM2=1**的从机不予理睬。

7.4.2 编程实现

简单编程：主机向02号从机发送50H~5FH单元内的数据。发送程序如下：

```
TRM02:  MOV     SCON, #98H      ; 串口方式2, TB8=1
AGAIN:  MOV     SBUF, #02H      ; 呼叫02号从机
W1:     JBC     TI, W2          ; 等待发送一帧结束
        SJMP    W1
W2:     JBC     RI, ANS         ; 等待从机应答
        SJMP    W2
ANS:    MOV     A, SBUF         ; 取应答地址
        XRL     A, #02H        ; 是否02号从机应答
        JZ      DTR
        MOV     SBUF, #0FFH     ; 不是02号, 发复位信号
W3:     JBC     TI, AGAIN       ; 复位后, 重新呼叫
        SJMP    W3
DTR:    CLR     TB8            ; 联络成功, 换数据帧
        MOV     R0, #50H       ; 开始发送数据串
        MOV     R7, #10H
```

	D7	D6	D5	D4	D3	D2	D1	D0
	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
位地址	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H



```
LOOP:    MOV    A, @R0
          MOV    SBUF, A    ; 发送数据
W4:      JBC    TI, NEXT
          SJMP   W4
NEXT:    INC    R0
          DJNZ   R7, LOOP   ; 发送完全部数据?
          SETB   TB8    ; 发送结束, 使从机复位
          MOV    SBUF, #0FFH
W5:      JBC    TI, TEND
          SJMP   W5
TEND:    RET
```

从机程序

```

RECEV: MOV     SCON, #0B0H; 串口方式2接收, SM2=1
W1:     JBC     RI, ACOMP; 接收主机呼叫(监听)
        SJMP    W1
ACOMP:  MOV     A, SBUF ; 取呼叫地址
        XRL     A, #02H ; 判断是否呼叫本机
        JNZ     W1      ; 不是返回监听状态
        CLR     SM2     ; 是呼叫本机, SM2=0
        MOV     SBUF, #02H; 向主机应答
W2:     JBC     TI, DRE
        SJMP    W2
DRE:    MOV     R0, #50H      ; 准备开始接收主机数据
        MOV     R7, #10H
W3:     JBC     RI, NEXT      ; 等待接收
        SJMP    W3
NEXT:   JNB     RB8, RECB; 是否为复位信号?
        LJMP    RECEV ; 复位, 恢复监听状态
RECB:   MOV     A, SBUF ; 读取接收数据
        MOV     @R0, A
        INC     R0
        DJNZ    R7, W3
        LJMP    RECEV
    
```

	D7	D6	D5	D4	D3	D2	D1	D0
	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
位地址	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H

7.5 使用MCS51串行口串行通讯的局限性



MCS-51串行口的输入、输出均为TTL电平, 抗干扰性差, 传输距离短。

为提高串行通讯的可靠性, 增大串行通讯的距离, 采用标准串行接口, 如RS-232、RS-422A、RS-485等标准。

7.5.1 RS-232C接口

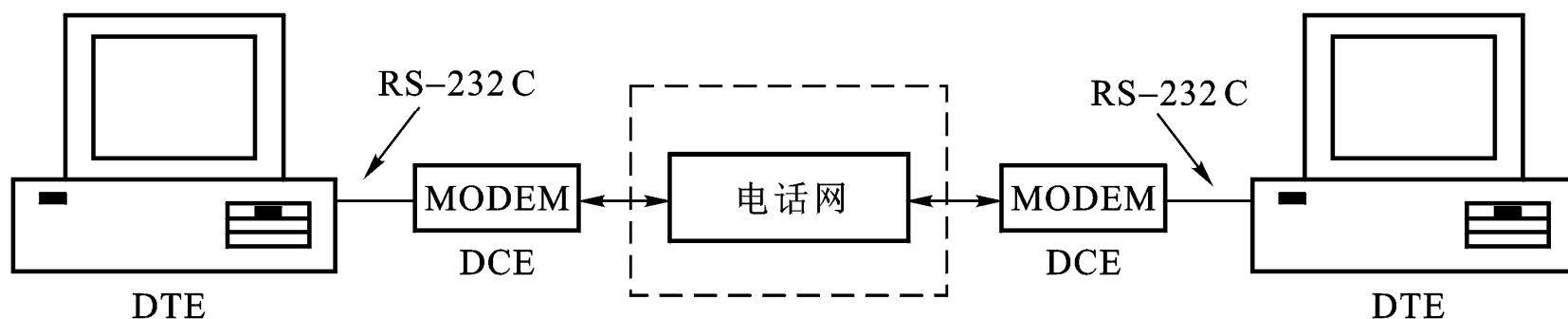


RS232接口定义数据终端设备（DTE）和数据通讯设备（DCE）之间的串行接口标准。

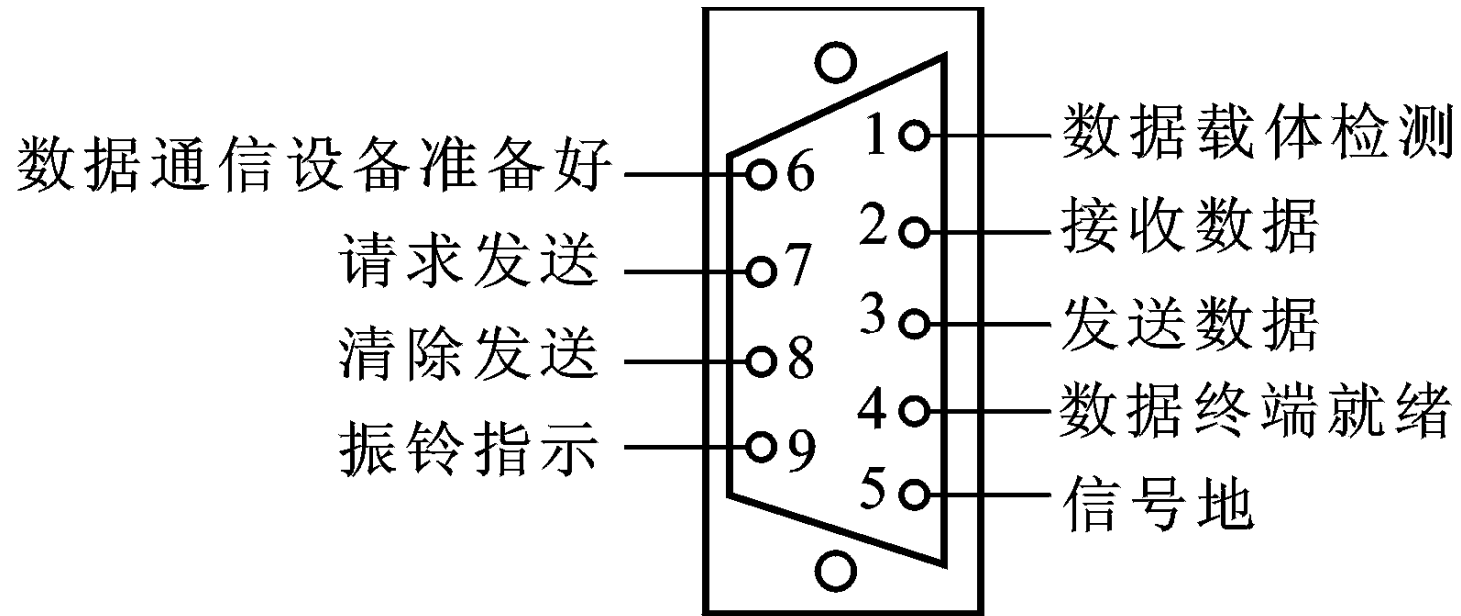
RS-232是由美国电子工业协会（EIA）于1962年制定的标准，台式机一般都具有标准的RS232接口。

RS232采用负逻辑，-12V表示逻辑“1”
+12V表示逻辑“0”

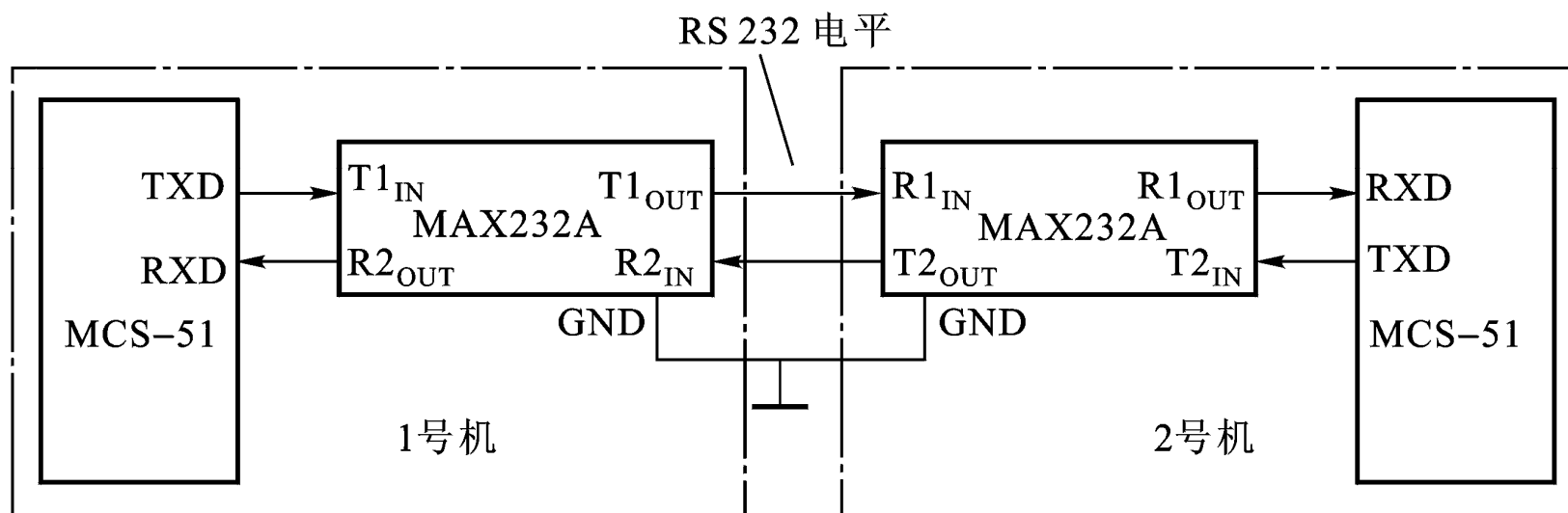
下图是两台计算机利用MODEM、电话线进行远距离串行通讯的示意图，DTE为计算机，DCE的典型代表是MODEM。



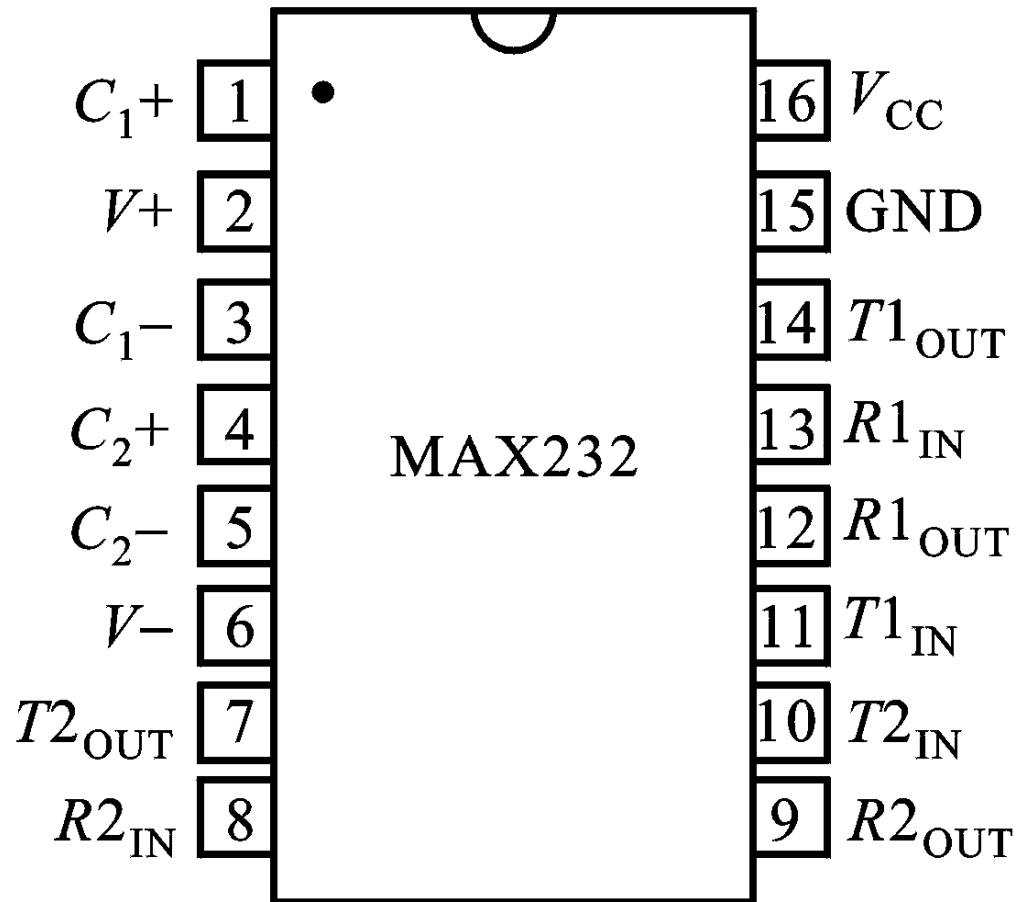
RS-232C的“D”型9针插头的引脚定义

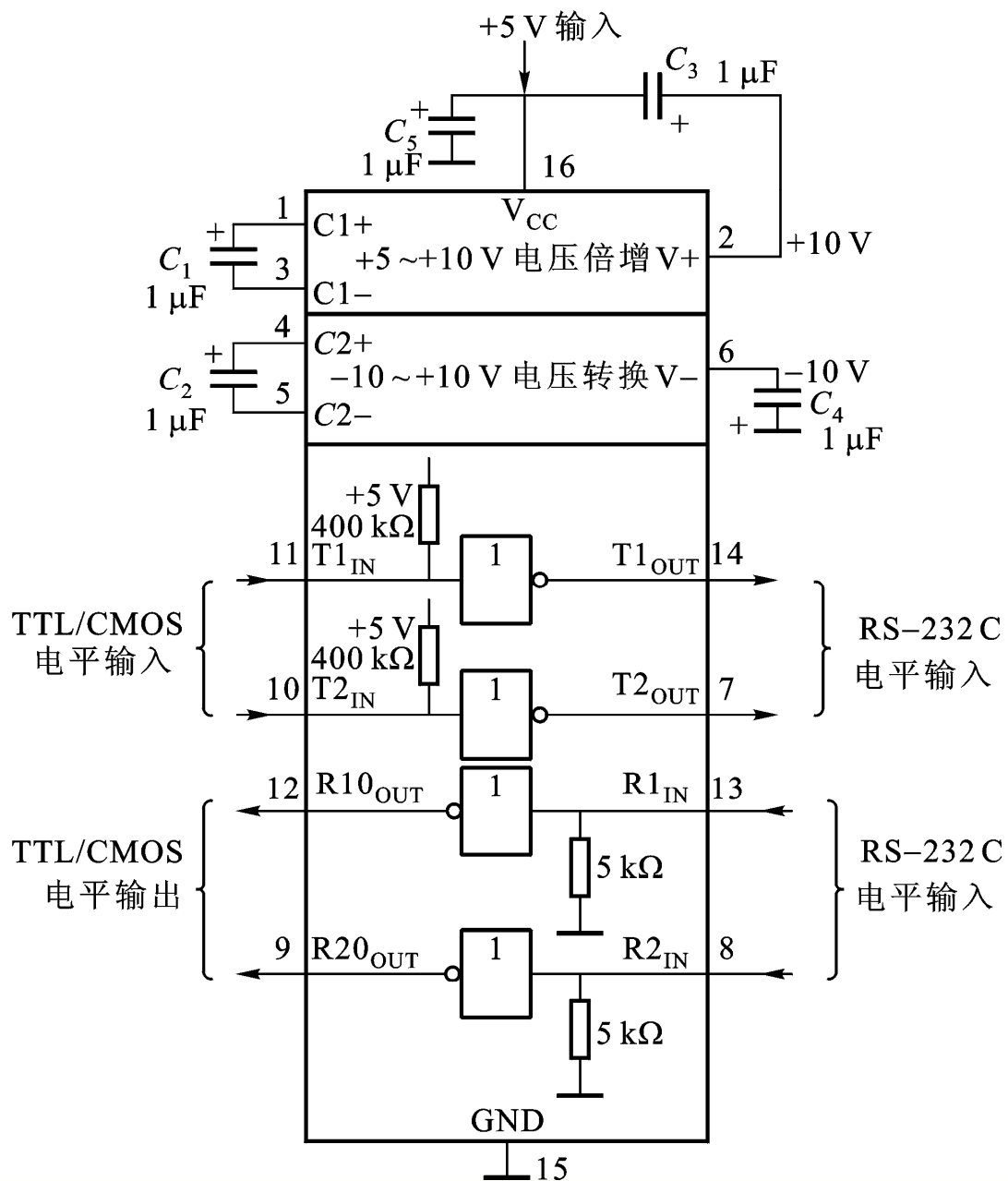


7.5.2 双MCS51通过RS232接口通讯

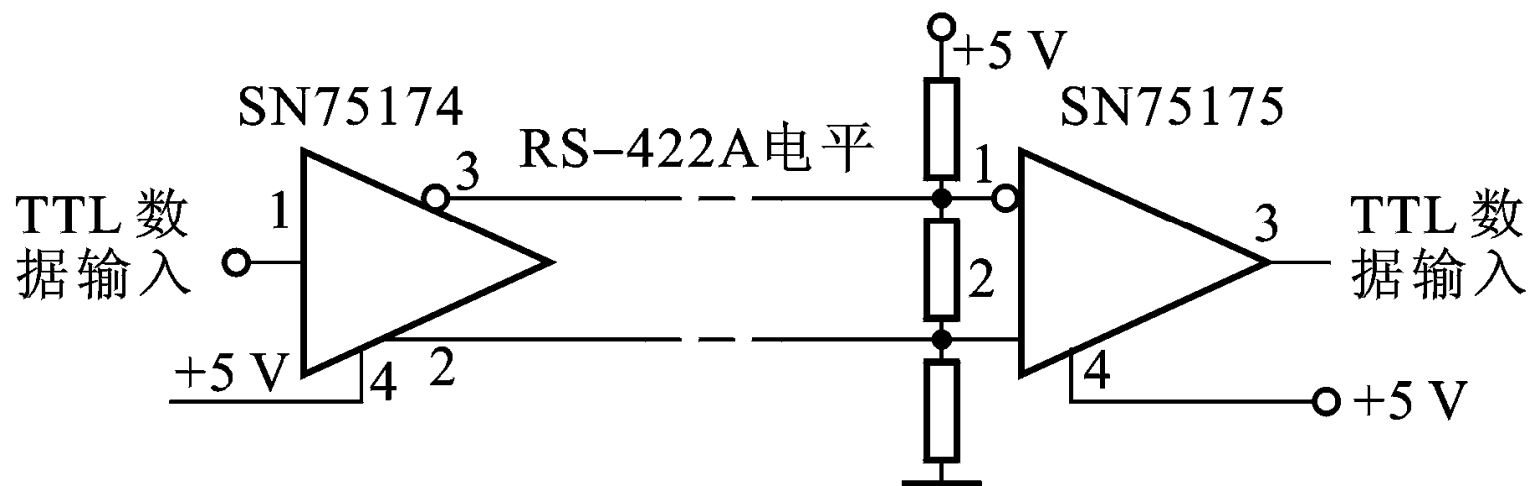


MAX232接口芯片

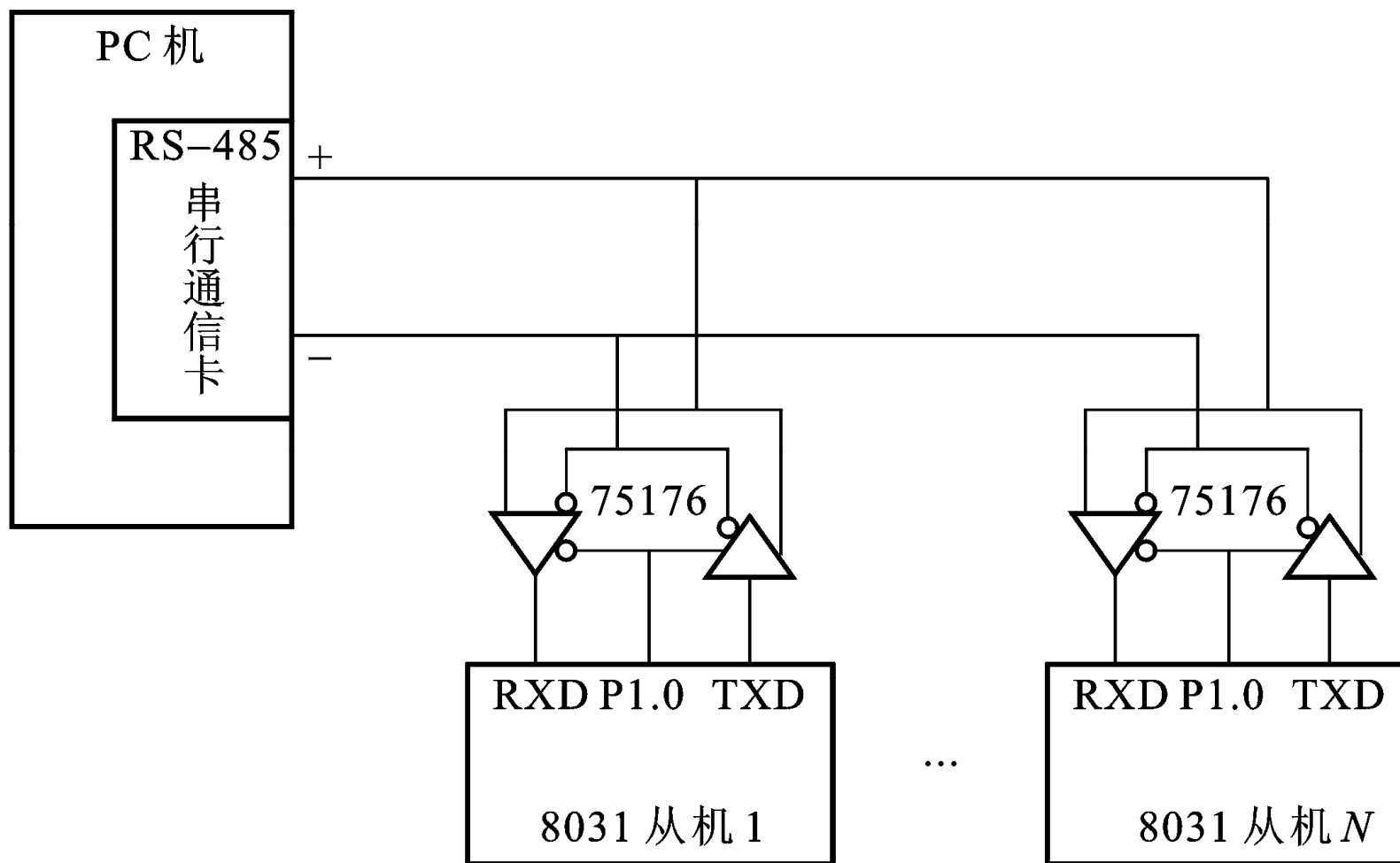




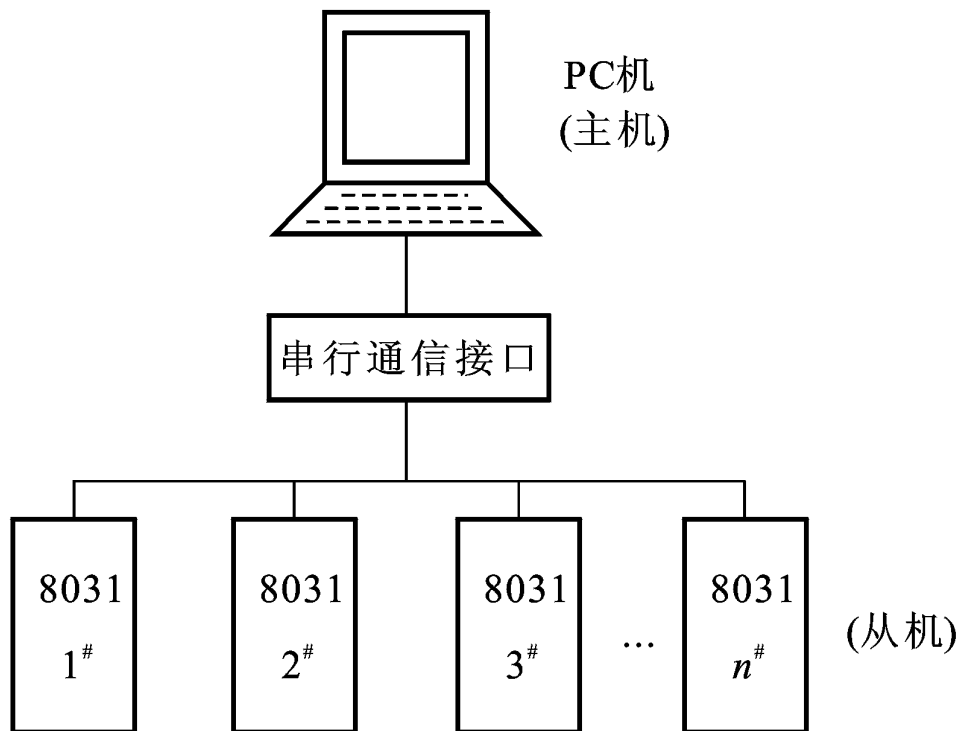
7.5.3 RS422接口在MCS51中的应用



7.5.4 RS485的应用（也可使用MAX485芯片）



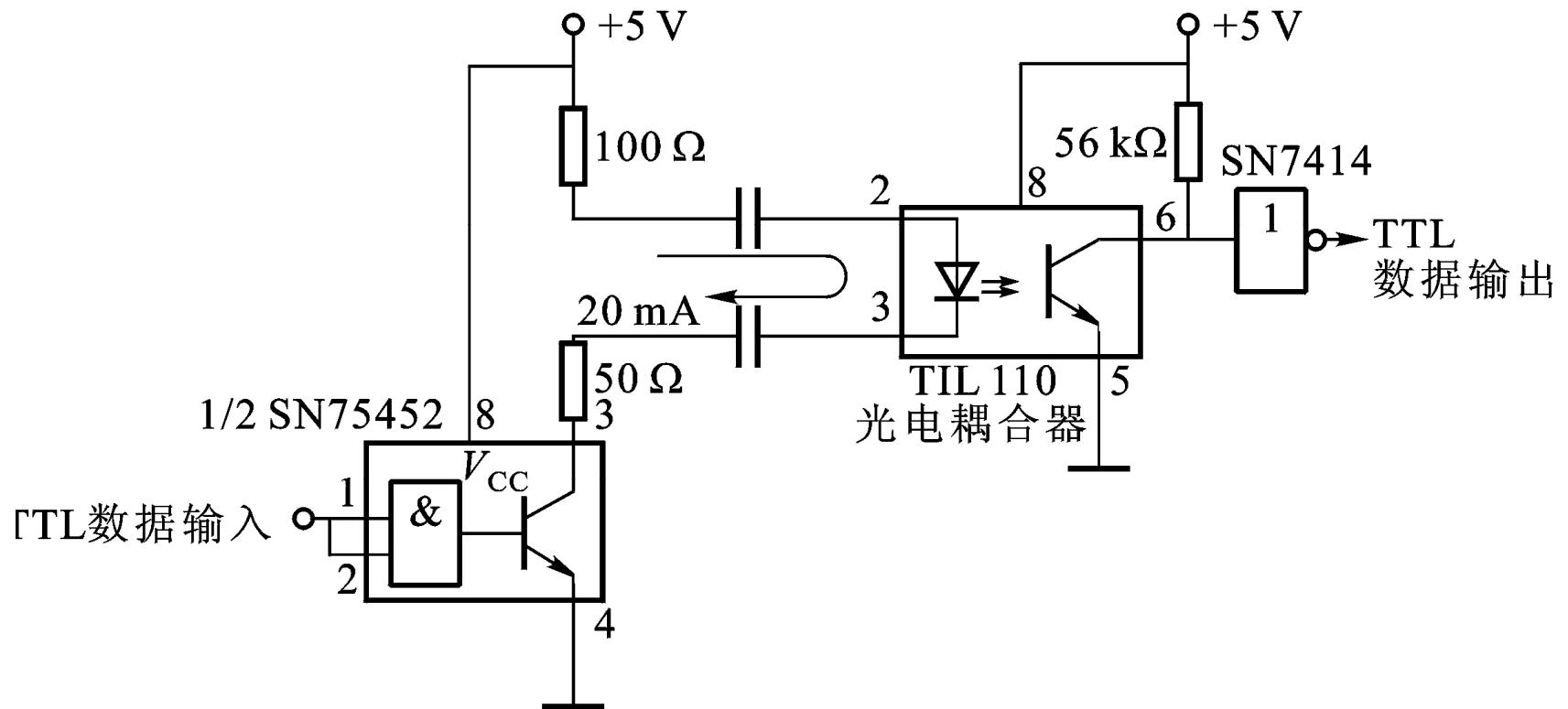
提高：MCS51与台式机的串行通讯



MCS51 串行口
需外加**MAX232**
芯片进行接口电
平转换，才能与
台式机串行接口
相连。

其他的设置，如
波特率、数据格
式的要求同双机
通讯。

7.5.5 20mA电流环的应用



习题



1. 帧格式为1个起始位，8个数据位和1个停止位的异步串行通讯方式是方式（ ）。

答：方式1

2. 假定串行口串行发送的字符格式为1个起始位，8个数据位，1个奇校验位，1个停止位，请画出传送字符“A”的帧格式。

答：从左向右：0，1，0，0，0，0，0，1，0，1，1

3. 判断下列说法是否正确：

(A) 串行口通讯的第9数据位的功能可由用户定义。

(B) 发送数据的第9数据位的内容在SCON寄存器的TB8位中预先准备好的。

(C) 串行通讯帧发送时，指令把TB8位的状态送入发送SBUF中。

(D) 串行通讯接收到的第9位数据送SCON寄存器的RB8中保存。

(E) 串行口方式1的波特率是可变的，通过定时器/计数器T1的溢出率设定。

答：(A) 串行口通讯的第9数据位的功能可由用户定义。（对）

(B) 发送数据的第9数据位的内容在SCON寄存器的TB8位中预先准备好的。（对）

(C) 串行通讯帧发送时，指令把TB8位的状态送入发送SBUF中。（错）

(D) 串行通讯接收到的第9位数据送SCON寄存器的RB8中保存。（对）

(E) 串行口方式1的波特率是可变的，通过定时器/计数器T1的溢出率设定。（对）



4. 串行口工作方式1的波特率是：

- (A) 固定的，为 $f_{osc}/32$ 。
- (B) 固定的，为 $f_{osc}/16$ 。
- (C) 可变的，通过定时器/计数器T1的溢出率设定。
- (D) 固定的，为 $f_{osc}/64$ 。

答：(C) 可变的，通过定时器/计数器T1的溢出率设定。

5. 在串行通讯中，收发双方对波特率的设定应该是（ ）的。

答：一致的

6. 若晶体振荡器为11.0592MHz，串行口工作于方式1，波特率为4800b/s，写出用T1作为波特率发生器的方式控制字和计数初值。

答：方式控制字为13H，计数初值为FAH

7. 为什么MCS-51串行口的方式0帧格式没有起始位(0)和停止位(1)？

答：因为串行口的方式0是同步移位寄存器方式，不是用于异步串行通讯的，所以收发双方不需要确定数据帧的起始和终止，所以没有起始位(0)和停止位(1)。

END

