第8章　串行口的工作原理及应用

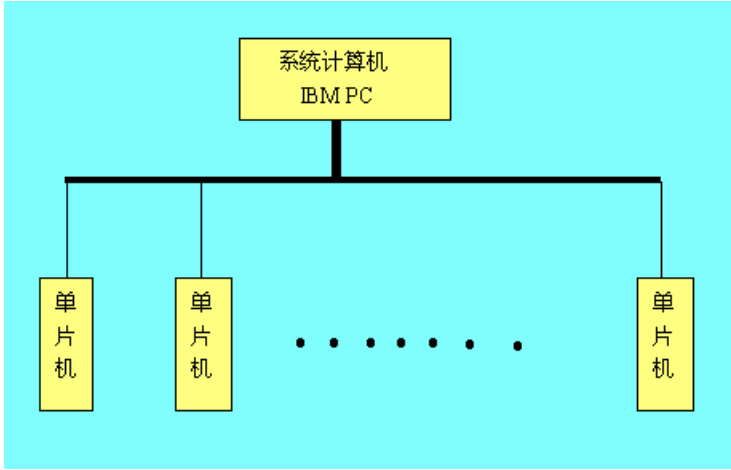
学习目的及要求

- 熟悉数据通信中的并行/串行、同步/异步、单工/双工以及波特率等概念；
- 熟悉89C51串行接口的基本结构，熟练掌握串行接口控制寄存器SCON各个位的含义及其控制功能；
- 熟练掌握89C51串行接口的4种工作方式及其实际应用，熟悉不同工作方式下的波特率公式。

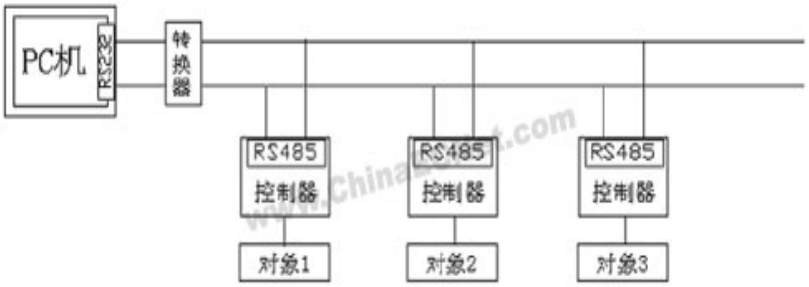
8.1 串行通信

- 805l单片机除具有四个8位并行口外，还具有串行接口。此串行接口是一个全双工串行通信接口，即能同时进行串行发送和接收。它可以作UART(通用异步接收和发送器)用，也可以作同步位移寄存器用。应用串行接口可以实现89C51单片机系统之间点对点的单机通信、多机通信和89C51与系统机(如IBM—PC机等)的单机或多机通信。

典型的计算机测量与控制系统构成



典型的计算机测量与控制系统构成



补充：串行通信及基础知识

1．数据通信的概念

在实际工作中，计算机的CPU与外部设备之间常常要进行信息交换，一台计算机与其它计算机之间也往往要交换信息，所有这些信息交换均可称为通信。

串行通信及基础知识

数据通信方式：并行通信与串行通信

并行通信：一次传输8（16、32Bit）

8根数据线，1根控制线，1根状态线，地线，共11根；

特点：速度快，适合近距离传输

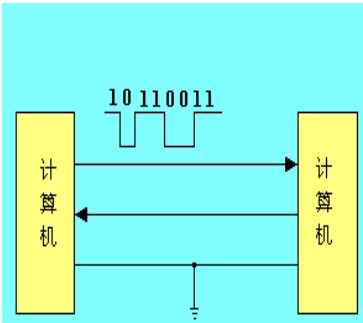
计算机并口，打印机，8255

串行通信：数据一位一位地发送，

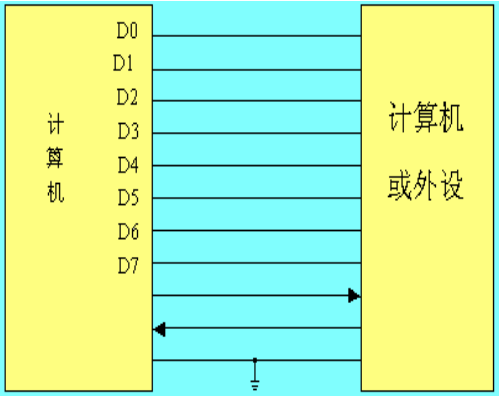
一根发送线，一根接受线，地线，共3根

特点：硬件方便，适合距离远，速度要求不高的场合

串行通信及基础知识



串行通信示意图



并行通信示意图

串行通信及基础知识

按串行通信的传送方向分为：单工、半双工、全双工通信方式

1．单工方式

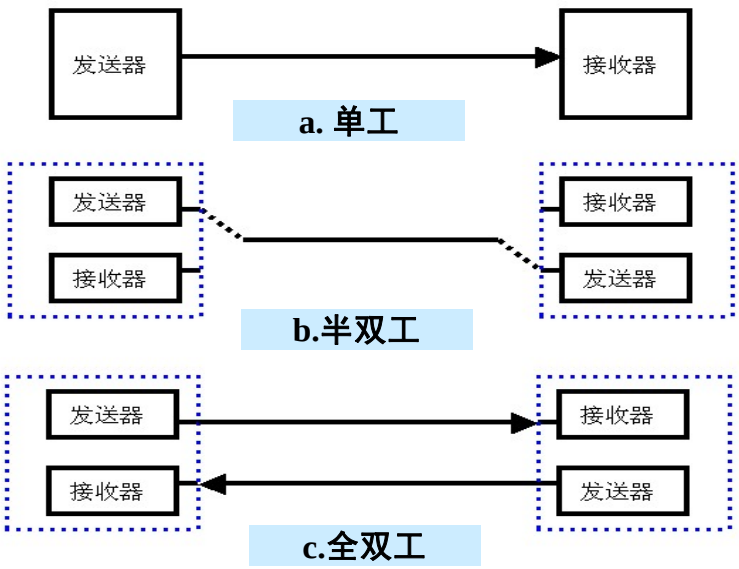
一端是发送端，另外一端是接收端：

2．半双工发式

每端口由一个发送器和接收器，通过开关连接在线路上，数据可以双方交换，但不能同时发送和接收.

3. 全双工方式

通信双方用两个独立的收发器单独连接,可以同时发送和接收数据,因而提高了速度。



串行通信及基础知识

3．异步通信和同步通信

串行通信有两种基本通信方式，即异步通信和同步通信。

(1)异步通信

在异步通信中，数据是一帧(Frame)一帧(包含一个字符代码或一字节数据)传送的，每一串行帧的数据格式如图7—31所示。

串行通信及基础知识

异步通信

1．起始位：在数据发送线上规定无数据时电平为1，当要发送数据时，首先发送一个低电平0，表示数据传送的开始，这就是起始位。

2．数据位：真正要传送的数据，可以是8位、10位等多位，数据位是由地位开始，高位结束；

3．奇偶校验：数据发送完后，发送奇偶校验位，以检验数据传送的正确性，这中方法是有限的，但是容易实现。

4．停止位：表示数据传送的结束，可以是一位或两位。

串行通信及基础知识

3．异步通信和同步通信

(2)同步通信

同步通信中，在数据开始传送前用同步字符来指示(常约定1—2个)，并由时钟来实现发送端和接收端同步，即检测到规定的同步字符后，下面就连续按顺序传送数据，直到通信告一段落。同步传送时，字符与字符之间没有间隙，也不用起始位和停止位，仅在数据块开始时用同步字符SYNC来指示，其数据格式如图7—32所示。

图7-31异步通信的一帧数据格式

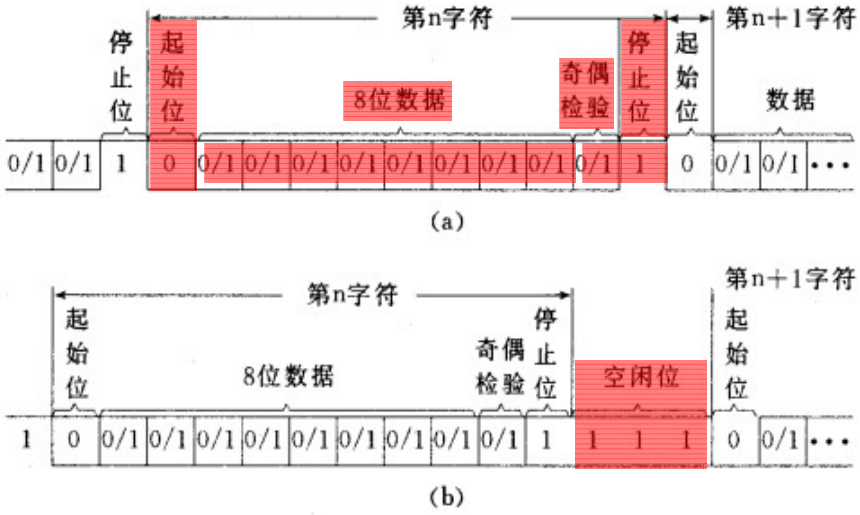


图7-31 异步通信的一帧数据格式

图7-32同步通信的数据格式

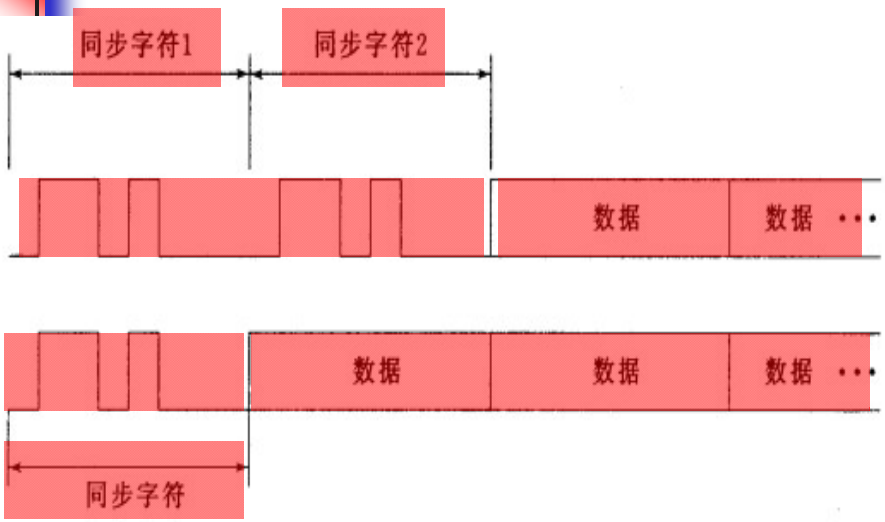


图7-32 同步通信的数据格式

同步和异步通信的区别

2)同步和异步都属于串行数据传送方式，但二者传送格式有所不同。

同步方式的一帧之内含有很多数据位，而异步方式的一帧内只含有几个数据位。换句话说，如果要传送一大串数据，同步方式只给这串数据进行一次外包装(即添加“头”帧、“尾”帧和“校验”帧)，

而异步方式在传送这串数据时，则要对数据的每一字节分别加以包装(即添加“头”位“尾”位和校验位)。显然，同步方式比异步方式的传送速度快，但同步方式要求收发双方在整个数据传输过程中始终保持严格同步，这将增加硬件上实现的难度，而串行通信只要求在每帧的传送中短时间保持同步即可，实现起来要容易得多。

波特率

(3)波特率(Band Rate) (Baud Rate)

波特率，即数据传送速率，表示每秒钟传送二进制代码的位数，它的单位是波特(bps，位／s)。波特率对于CPU与外界的通信是很重要的。假设数据传送速率是120字符／s，而每个字符格式包含10个代码位(一个起始位、一个终止位、8个数据位)，这时传送的波特率为

10×120位／s＝1200bps

每一位代码的传送时间Td为波特率的倒数

Td＝1／1200＝0.833ms

波特率是衡量传送通道频宽的指标，它和传送数据的速率并不一致。如上例中，因为除掉起始位和终止位，每一个数据实际只占8位。所以数位的传送速率为

8×120＝960位／s

异步通讯的传送速度在50到64000bps之间。常用于计算机到终端和打印机之间的通信、直通电报以及无线电通讯的数据发送等。

AT89S51集成一个全双工通用异步收发（UART）串行口。

全双工：两个单片机之间串行数据可同时双向传输。

异步通信： 收、发双方使用各自时钟控制发送和接收，省去收、发双方的1条同步时钟信号线，使异步串行通信连接更简单且易实现。

8.1 串行口结构

AT89S51串行口内部结构见图8-1。有两个物理上独立的接收、发送缓冲器SBUF（特殊功能寄存器），可同时收发数据。发送缓冲器只写不读，接收缓冲器只读不写，两个缓冲器共用一个特殊功能寄存器字节地址（99H）。

控制寄存器共有2个：特殊功能寄存器SCON和PCON。下面详细介绍各位功能。

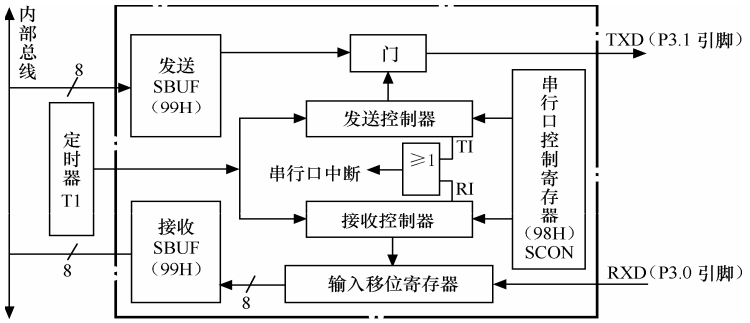


图8-1 串行口的内部结构

8.1.1 串行口控制寄存器SCON

串行口控制寄存器SCON，字节地址98H，可位寻址，位地址为98H～9FH，即SCON的所有位都可用软件来进行位操作清“0”或置“1”。SCON格式 见图8-2。

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

图8-2 串口控制寄存器SCON格式

寄存器SCON各位功能：

（1）SM0、SM1—串口4种工作方式选择

SM0、SM12位编码对应4种工作方式见表8-1。

表8-1

SM0	SM1	方　式	功 能 说 明
0	0	0	同步移位寄存器方式（用于扩展I/O口）
0	1	1	8位异步收发，波特率可变（由定时器控制）
1	0	2	9位异步收发，波特率为 $f_{osc}/64$ 或 $f_{osc}/32$
1	1	3	9位异步收发，波特率可变（由定时器控制）

（2）SM2—多机通信控制位

多机通信是在方式2和方式3下进行，因此SM2位主要用于方式2或方式3。

当串口以方式2或方式3接收时，如SM2=1，则只有当接收到的第9位数据（RB8）为“1”时，才使RI置“1”，产生中断请求，并将收到的前8位数据送入SBUF；当收到的第9位数据（RB8）为“0”时，则将收到的前8位数据丢弃。

当SM2=0时，则不论第9位数据是“1”还是“0”，都将接收的前8位数据送入SBUF中，并使RI置“1”，产生中断请求。

方式1时，如果SM2=1，则只有收到有效的停止位时才会激活RI。

方式0时，SM2必须为0。

（3）REN—允许串行接收位，由软件置“1”或清“0”。

REN=1，允许串行口接收数据。

REN=0，禁止串行口接收数据。

（4）TB8—发送的第9位数据

在方式2和方式3时，TB8是要发送的第9位数据，其值由软件置“1”或清“0”。

在双机串行通信时，TB8一般作为奇偶校验位使用；也可在多机串行通信中表示主机发送的是地址帧还是数据帧，TB8=1为地址帧，TB8=0为数据帧。

（5）RB8—接收的第9位数据

在方式2和方式3时，RB8存放接收到的第9位数据。在方式1，如果SM2=0，RB8是接收到的停止位。在方式0，不使用RB8。

（6）TI—发送中断标志位

方式0时，串行发送的第8位数据结束时，TI由硬件置“1”，在其他工作方式中，串行口发送停止位的开始时，置TI为“1”。TI=1，表示1帧数据发送结束。TI位状态可供软件查询，也可申请中断。CPU响应中断后，在中断服务程序向SBUF写入要发送的下一帧数据。注意：TI必须由软件清“0”。

（7）RI—接收中断标志位

串口在方式0时，接收完第8位数据时，RI由硬件置“1”。在其他工作方式中，串行接收到停止位时，该位置“1”。RI=1，表示一帧数据接收完毕，并申请中断，要求CPU从接收SBUF取走数据。该位状态也可供软件查询。

注意：RI必须由软件清“0”。

8.1.2 特殊功能寄存器PCON

字节地址为87H，不能位寻址。格式见图8-3。

仅最高位SMOD与串口有关，低4位功能在第2章中已介绍。

SMOD位：波特率选择位。

	D7	D6	D5	D4	D3	D2	D1	D0	
PCON	SMOD	—	—	—	GF1	GF0	PD	IDL	87H

图8-3 特殊功能寄存器PCON的格式

例如，方式1的波特率计算公式为

$$\text{方式1波特率} = \frac{2^{SMOD}}{32} * \text{T1的溢出率}$$

当SMOD=1时，比SMOD=0时波特率加倍，所以也称SMOD位为波特率倍增位。

8.2 串行口的4种工作方式

4种工作方式由特殊功能寄存器SCON中SM0、SM1位定义，编码见表8-1。

8.2.1 方式0

方式0为同步移位寄存器输入/输出方式。该方式并不用于两个AT89S51单片机间的异步串行通信，而是用于外接移位寄存器，用来扩展并行I/O口。

方式0以8位数据为1帧，没有起始位和停止位，先发送或接收最低位。波特率是固定的，为fosc/12。帧格式见图8-4。

…	D0	D1	D2	D3	D4	D5	D6	D7	…

图8-4 方式0帧格式

8.2.2 方式1

方式1为双机串行通信方式，如图8-9。

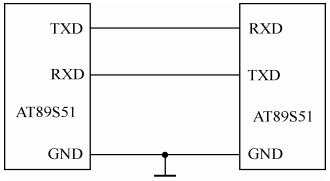


图8-9 方式1双机串行通信的连接电路

当SM0、SM1=01时，串行口设为方式1双机串行通信。TXD脚和RXD脚分别用于发送和接收数据。

方式1收发一帧数据为10位，1个起始位（0），8个数据位，1个停止位（1），先发送或接收最低位。方式1帧格式见图8-10。

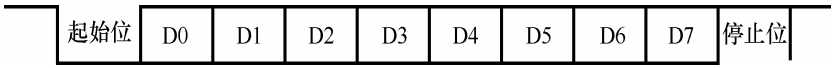


图8-10 方式1的帧格式

方式1为波特率可变的8位异步通信接口。波特率由下式确定：

$$\text{方式 1波特率} = \frac{2^{SMOD}}{32} * \text{定时器 1的溢出率}$$

式中，SMOD为PCON寄存器的最高位的值（0或1）。

1. 方式1发送

串口以方式1输出，数据位由TXD端输出，发送一帧信息为10位，1位起始位0，8位数据位（先低位）和1位停止位1，当CPU执行写数据到发送缓冲器SBUF的命令后，就启动发送。方式1发送时序如图8-11。

图8-11发送时钟TX时钟频率就是发送波特率。发送开始时，内部逻辑将起始位向TXD脚（P3.1）输出，此后每经1个TX时钟周期，便产生1个移位脉冲，并由TXD脚输出1个数据位。8位全发送完后，中断标志位TI置“1”。

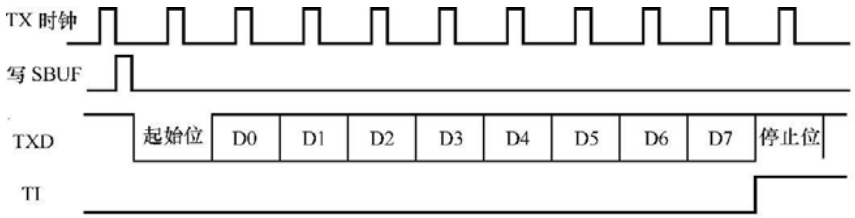


图8-11 方式1发送时序

2．方式1接收

串行口以方式1（SM0、SM1=01）接收时（REN=1），数据从RXD（P3.0）脚输入。当检测到起始位负跳变时，则开始接收。方式1接收时序如图8-12。

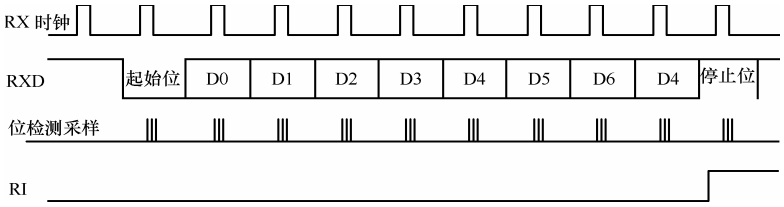


图8-12　方式1接收时序

接收时，定时控制信号有两种，一种是接收移位时钟（RX时钟），频率和传送的波特率相同，另一种是位检测器采样脉冲，它的频率是RX时钟的16倍。也就是在1位数据期间，有16个采样脉冲，以波特率的16倍速率采样RXD引脚状态。

当采样到RXD端从1到0的负跳变（有可能是起始位）时，就启动接收检测器。接收的值是3次连续采样（第7、8、9个脉冲时采样），取其中两次相同的值，以确认是否是真正起始位（负跳变）开始，这样能较好消除干扰引起的影响，以保证可靠无误地开始接收数据。

当确认起始位有效时，开始接收一帧信息。接收每一位数据时，也都进行3次连续采样（第7、8、9个脉冲时采样），接收的值是3次采样中至少两次相同的值，以保证接收到的数据位的准确性。当一帧数据接收完毕后，必须同时满足以下两个条件，这次接收才真正有效。

（1）RI=0，即上一帧数据接收完成时，RI=1发出的中断请求已被响应，SBUF中的数据已被取走，说明“接收SBUF”已空。

（2）SM2=0或收到的停止位=1（方式1时，停止位已进入RB8），则将接收到的数据装入SBUF和RB8（装入的是停止位），且中断标志RI置“1”。

若不同时满足这两个条件，收到的数据不能装入SBUF，这意味着该帧数据将丢失。

8.2.3　方式2

串口工作于方式2和方式3时，为9位异步通信接口。每帧数据均为11位，1位起始位0，8位数据位（先低位），1位可程控为1或0的第9位数据及1位停止位。方式2、方式3帧格式如图8-13。

图8-13　方式2、方式3的帧格式

方式2的波特率由下式确定：

$$\text{方式2波特率} = \frac{2^{SMOD}}{64} \times f_{osc}$$

1．方式2发送

发送前，先由通信协议由软件设置TB8（如奇偶校验位或多机通信的地址/数据的标志位），然后将要发送的数据写入SBUF，即可启动发送过程。串行口能自动把TB8取出，并装入到第9位数据位的位置，再逐一发送出去。发送完毕，则使TI位置“1”。

方式2和方式3发送时序如图8-14。

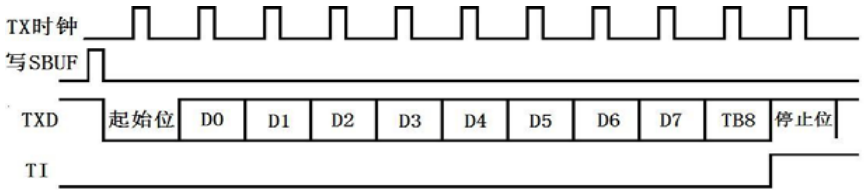


图8-14 方式2和方式3发送时序

2．方式2接收

当SCON寄存器SM0、SM1=10，且REN=1时，允许串行口以方式2接收数据。接收时，数据由RXD端输入，接收11位信息。当位检测逻辑采样到RXD引脚从1到0的负跳变，并判断起始位有效后，便开始接收一帧信息。在接收完第9位数据后，需满足以下两个条件，才将接收到的数据送入接收缓冲器SBUF。

（1）RI=0，意味着接收缓冲器为空。

（2）SM2=0或接收到的第9位数据位RB8=1。

当满足上述两个条件时，接收到的数据送入SBUF（接收缓冲器），第9位数据送入RB8，且RI置“1”。若不满足这两个条件，接收的信息将被丢弃。

串行口方式2和方式3接收时序如图8-15。

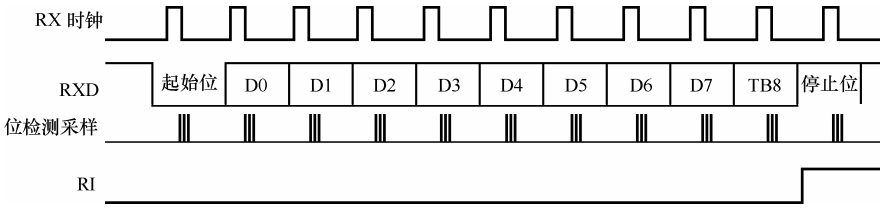


图8-15 方式2和方式3接收时序

8.2.4 方式3

当SM0、SM1两位为11时，串行口被定义工作在方式3。方式3为波特率可变的9位异步通信方式，除了波特率外，方式3和方式2相同。方式3发送和接收时序见图8-11和图8-12。

方式3波特率由下式确定：

$$\text{方式 3波特率} = \frac{2^{SMOD}}{32} \times \text{定时器 T1的溢出率}$$

8.3 多机通信(自学)

多个AT89S51单片机可利用串行口进行多机通信，经常采用图8-16主从式结构。该多机系统有1个主机（AT89S51单片机或其他具有串口的微机）和3个（也可为多个）AT89S51单片机组成的从机系统，如图8-16。主机RXD与所有从机TXD端相连，TXD与所有从机RXD端相连。从机地址分别为01H、02H和03H。

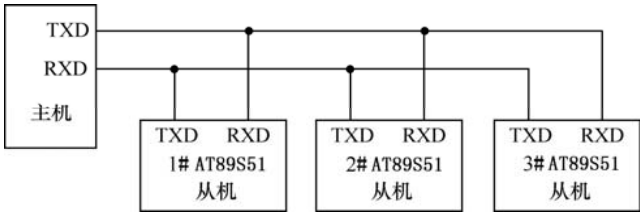


图8-16 多机通信系统示意图

8.4 波特率的制定方法

在串行通信中，收、发双方发送或接收的波特率必须一致。通过软件对串口可设定4种工作方式。其中方式0和方式2的波特率是固定的；方式1和方式3的波特率是可变的，由定时器T1的溢出率（T1每秒溢出的次数）来确定。

8.4.1 波特率的定义

波特率的定义：串行口每秒钟发送（或接收）的位数称为波特率。设发送一位所需要的时间为T，则波特率为1/T。

对于定时器的不同工作方式，得到的波特率的范围是不一样的，这是由于定时器/计数器T1在不同工作方式下计数位数不同所决定。

8.4.2 定时器T1产生波特率的计算

波特率和串口工作方式有关。

（1）方式0

波特率固定为fosc的1/12，且不受SMOD位的影响。若fosc=12MHz，波特率为fosc/12，即1Mbit/s。

（2）方式2

波特率仅与SMOD位的值有关。

$$\text{方式2波特率} = \frac{2^{SMOD}}{64} \times f_{osc}$$

若fosc=12MHz：SMOD=0，波特率=187.5 kbit/s；SMOD=1，

波特率=375 kbit/s。

（3）方式1或方式3

常用T1作为波特率发生器，其关系式为

$$\text{波特率} = \frac{2^{SMOD}}{32} * \text{T1 溢出率} \qquad (8\text{-}1)$$

由式（8-1），T1的溢出率和SMOD的值共同决定波特率。

在实际设定波特率时，用 定时器方式2（自动装初值）确定波特率较理想，它不需用软件重装初值，可避免因软件重装初值带来的定时误差，且算出的波特率比较准确。即TL1作为8位计数器，TH1存放备用初值。

设定时器T1方式2的初值为X，则有

$$\text{定时器T1的溢出率} = \frac{\text{计数速率}}{256-X} = \frac{f_{osc}/12}{256-X} \quad (8\text{-}2)$$

将式（8-2）代入式（8-1），则有

$$\text{波特率} = \frac{2^{SMOD}}{32} \times \frac{f_{osc}}{12(256-X)} \quad (8\text{-}3)$$

由式（8-3），这种方式波特率随fosc、SMOD和初值X而变化。

在实际使用时，常根据已知波特率和时钟频率fosc来计算T1的初值X。为避免繁杂初值计算，常用波特率和初值X间关系常列成表8-2形式，供查用。

表 8-2　用定时器 T1 产生的常用波特率

波　特　率	f_{osc}	SMOD 位	方式	初值 X
62.5kbit/s	12 MHz	1	2	FFH
19.2kbit/s	11.059 2 MHz	1	2	FDH
9.6kbit/s	11.059 2 MHz	0	2	FDH
4.8kbit/s	11.059 2 MHz	0	2	FAH
2.4kbit/s	11.059 2 MHz	0	2	F4H
1.2kbit/s	11.059 2 MHz	0	2	E8H

表8-2有两点需要注意：

（1）在时钟振荡频率fosc为12MHz或6MHz时，将初值X和fosc带入式（8-3），不能整除，因此算出的波特率有一定误差。要消除误差可通过调整fosc实现，例如采用的时钟频率为11.0592MHz。因此，为减小波特率误差，应该使用的时钟频率必须为11.0592MHz。

（2）如果串行通信选用很低波特率（如波特率选为55），可将定时器T1设置为方式1定时。但在这种情况下，T1溢出时，需在中断服务程序中重新装入初值。中断响应时间和执行指令时间会使波特率产生一定的误差，可用改变初值的方法加以调整。

【例8-3】若AT89S51的时钟为11.0592MHz，选用T1的方式2定时作为波特率发生器，波特率为2 400bit/s，求初值。

设T1为方式2定时，选SMOD=0。

将已知条件带入式（8-3）

$$\text{波特率} = \frac{2^{SMOD}}{32} \times \frac{f_{osc}}{12(256-X)} = 2400$$

从中解 X=244=F4H。

只要把F4H装入TH1和TL1，则T1发出的波特率为2400bit/s。在实际编程中，该结果也可直接从表8-2中查到。

这里时钟振荡频率选为11.0592MHz，就可使初值为整数，从而产生精确的波特率。

8.5　串行口应用设计案例

单片机串行通信接口设计时，需考虑如下问题。

（1）确定串行通信双方的数传速率和通信距离。

（2）由串行通信的数传速率和通信距离确定采用的串行通信接口标准。

（3）注意串行通信的通信线选择，一般选用双绞线较好，并根据传输的距离选择纤芯的直径。如空间干扰较多，还要选择带有屏蔽层的双绞线。

下面首先介绍有关串行通信各种接口标准。

8.5.1　串行通信标准接口RS232、RS422与RS485简介（自学）

AT89S51串口输入、输出均为TTL电平。这种以TTL电平来串行传输数据，**抗干扰性差，传输距离短，传输速率低**。为提高串行通信可靠性，增大串行通信距离和提高传输速率，在实际设计中都采用标准串行接口，如RS-232、RS-422A、RS-485等。

根据双机通信距离和抗干扰性要求，可选择TTL电平传输，或选择RS-232C、RS-422A、RS-485串行接口进行串行数据传输。

8.5.2　方式1的应用设计

【例8-4】图8-22为单片机甲、乙双机串行通信，双机RXD和TXD相互交叉相连，甲机P1口接8个开关，乙机P1口接8个发光二极管。甲机设置为只能发送不能接收的单工方式。要求甲机读入P1口的8个开关的状态后，通过串行口发送到乙机，乙机将接收到的甲机的8个开关的状态数据送入P1口，由P1口的8个发光二极管来显示8个开关的状态。双方晶振均采用11.0592MHz。

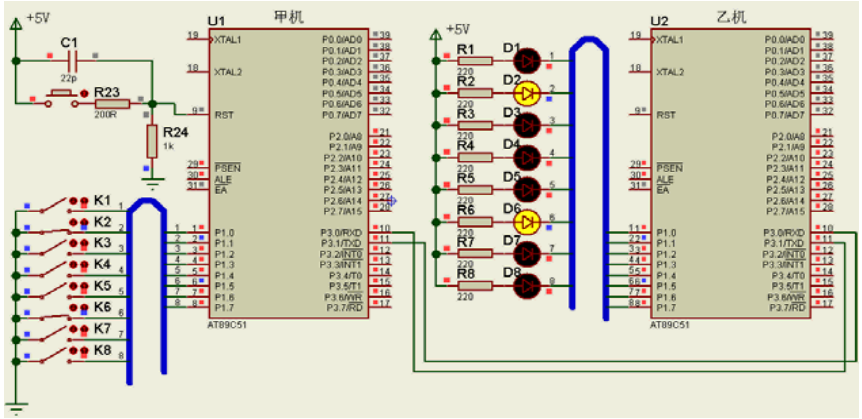


图8-22　　单片机方式1双机通信的连接

参考程序如下：

```
//甲机串行发送
#include <reg51.h>
#define uchar unsigned char
#define uint unsigned int
void main()
{   uchar temp=0;
    TMOD=0x20;                  //设置定时器T1为方式2
    TH1=0xfd;                   //波特率9600
    TL1=0xfd;
    SCON=0x40;                  //串口初始化方式1发送，不接收
    PCON=0x00;                  // SMOD=0
    TR1=1;                      //启动T1
    P1=0xff;                    //设置P1口为输入
    while(1)
    {
        while(TI==0);           //如果TI=0，未发送完，循环等待
        TI=0;                   //已发送完，把TI清0
        temp=P1;                //读入P1口开关的状态
        SBUF=temp;}             //数据送串行口发送
    }
```



```
//乙机串行接收
#include <reg51.h>
#define uchar unsigned char
#define uint unsigned int
void main( )
{
    uchar temp=0;
    TMOD=0x20;                  //设置定时器T1为方式2
    TH1=0xfd;                   //波特率9600
    TL1=0xfd;
    SCON=0x50;                  //设置串口为方式1接收，REN=1
    PCON=0x00;                  //SMOD=0
    TR1=1;                      //启动T1
    while(1)
    {
        while(RI==0);           // 若RI为0，未接收到数据
        RI=0;                   // 接收到数据，则把RI清0
        temp=SBUF;              // 读取数据存入temp中
        P1=temp;                //接收的数据送P1口控制8个LED的亮灭
    }
}
```


8.5.3 方式2和方式3的应用设计

方式2与方式1比两点不同之处：

（1）方式2接收/发送11位信息，第0位为起始位，第1～8位为数据位，第9位是程控位，由用户设置的TB8位决定，第10位是停止位1，这是方式2与方式1的一个不同点。

（2）方式2的波特率变化范围比方式1小，方式2的波特率=振荡器频率/n。

当SMOD=0 时，n = 64。

当SMOD=1时，n = 32。

而方式2和方式3相比，除了波特率的差别外，其它都相同，所以下面介绍的方式3应用编程，也适用于方式2。

【例8-6】如图8-25，甲、乙两单片机进行 方式3（或方式2）串行通信。甲机把控制8个流水灯点亮的数据发送给乙机并点亮其P1口的8个LED。方式3比方式1多了1个可编程位TB8，该位一般作奇偶校验位。乙机接收到的8位二进制数据有可能出错，需进行奇偶校验，其方法是将乙机的RB8和PSW的奇偶校验位P进行比较，如果相同，接收数据；否则拒绝接收。

本例使用了一个虚拟终端来观察甲机串口发出的数据。

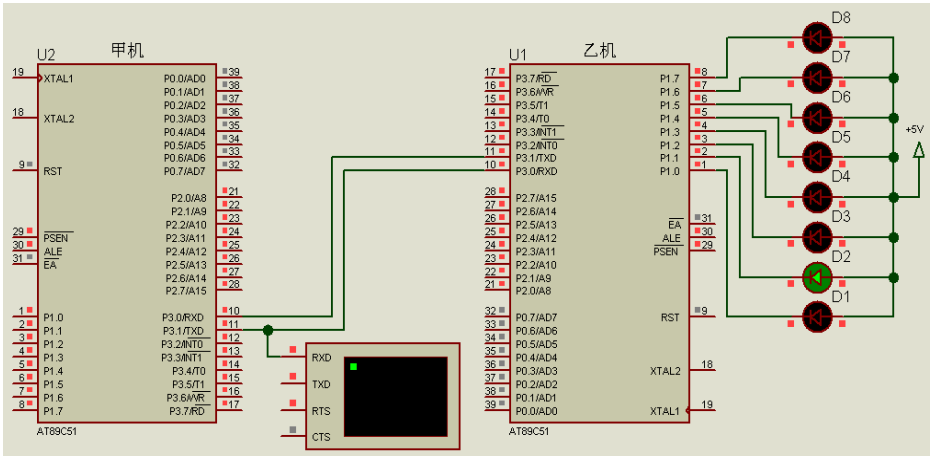


图8-25 甲乙两个单片机进行方式3（或方式2）串行通信

参考程序如下。

```
//甲机发送程序
#include <reg51.h>
sbit P=PSW^0;                    //P位为PSW寄存器的第0位，即奇偶校验位
unsigned char Tab[8]= {0xfe, 0xfd, 0xfb, 0xf7, 0xef, 0xdf,
                       0xbf, 0x7f};   //控制流水灯显示数据数组，为全局变量

void main(void)                                  //主函数
{
      unsigned char i;
      TMOD=0x20;                                 //设置定时器T1为方式2
      SCON=0xc0;                                 //设置串口为方式3
      PCON=0x00;                                 //SMOD=0
```



```
      TH1=0xfd;                  //给T1赋初值，波特率设置为9600
      TL1=0xfd;
      TR1=1;                     //启动定时器T1
      while(1)
      {
          for(i=0;i<8;i++)
          {
                Send(Tab[i]);
                delay( );        //大约200ms发送一次数据
          }
      }
      }
      void Send(unsigned char dat)        // 发送1字节数据的函数
      {
      TB8=P;                     // 将偶校验位作为第9位数据发送
      SBUF=dat;
      while(TI==0);              //检测TI，TI=0，未发送完
      ;                          // 空操作
      TI=0;                      // 1字节发送完，TI清0
      }
```



```
void delay (void)            // 延时约200ms的函数
  {
      unsigned char m,n;
      for(m=0;m<250;m++)
      for(n=0;n<250;n++);
  }

//乙机接收程序
    #include <reg51.h>
    sbit P=PSW^0;            // P位为PSW 寄存器的第0位，即奇偶校验位

    void main(void)          //主函数
    {
        TMOD=0x20;           //设置定时器T1为方式2
        SCON=0xd0;           //设置串口为方式3，允许接收REN=1
        PCON=0x00;           // SMOD=0
        TH1=0xfd;            //给定时器T1赋初值，波特率为9600
        TL1=0xfd;
        TR1=1;               //接通定时器T1
        REN=1;               //允许接收
```

```
while(1)
{
P1= Receive( );                    //将接收到的数据送P1口显示
}
}

unsigned char Receive(void)        //接收1字节数据的函数
{
unsigned char dat;
while(RI==0);                      //检测RI，RI=0，未接收完，则循环等待
;
RI=0;                              //已接收一帧数据，将RI清0
ACC=SBUF;                          //将接收缓冲器的数据存于ACC
if(RB8==P)                         //只有偶校验成功才能往下执行，接收数据
{
    dat=ACC;                       //将接收缓冲器的数据存于dat
    return dat;                    //将接收的数据返回
}
}
```


8.5.4 多机通信的应用设计（自学）

下面通过一个具体案例，介绍如何来实现单片机的多机通信。

【例8-7】实现主单片机分别与3个从单片机串行通信，原理电路见图8-26。用户通过分别按下开关k1、k2或k3来选择主机与对应1#、2#或3#从机串行通信，当黄色LED点亮，表示主机与相应的从机连接成功；该从机的8个绿色LED闪亮，表示主机与从机在进行串行数据通信。如果断开k1、k2或 k3，则主机与相应从机的串行通信中断。

8.5.5 单片机与PC机串行通信的设计（自学）

工业现场测控系统中，常用单片机进行监测点的数据采集，然后单片机通过串口与PC通信，把采集的数据串行传送到PC机上，再在PC机上进行数据处理。PC机配置都是RS-232标准串口，为9针“D”型插座，输入/输出为RS-232电平。“D”型9针插头引脚定义见图8-28。

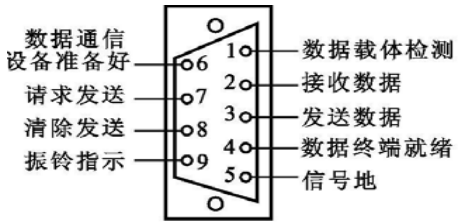


图8-28 “D”型9针插头引脚定义

8.5.6 PC机与单片机或与多个单片机的串行通信（自学）

一台PC与若干台AT89S51可构成小型分布式测控系统，见图8-33。

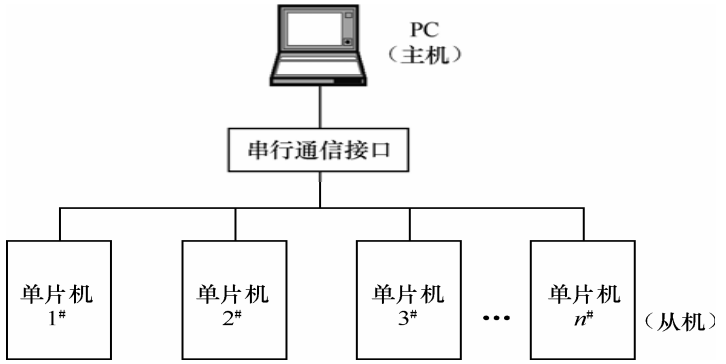


图8-33 PC机与多台单片机构成小型的分布式测控系统