

本章分为三节，主要介绍：

6.1 计算机串行通信基础

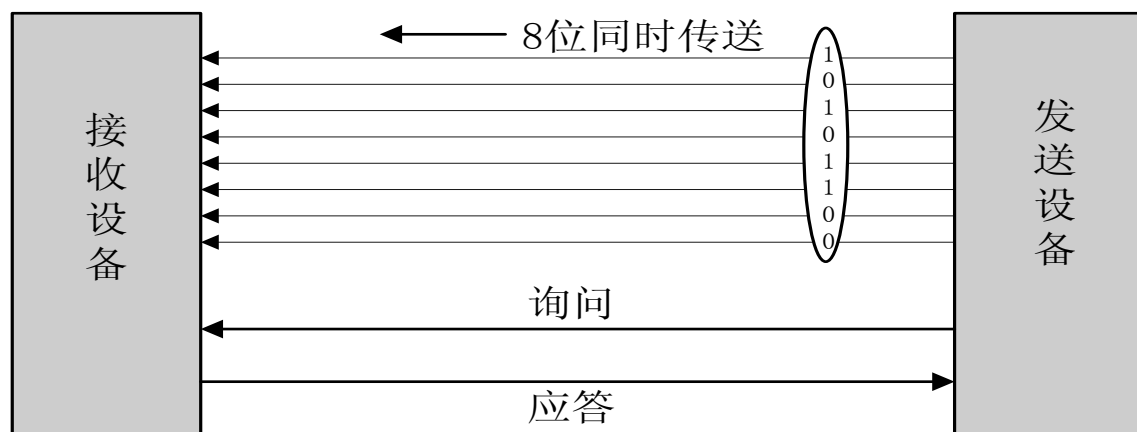
6.2 80C51的串行口

6.3 单片机串行口应用举例

6.1 计算机串行通信基础

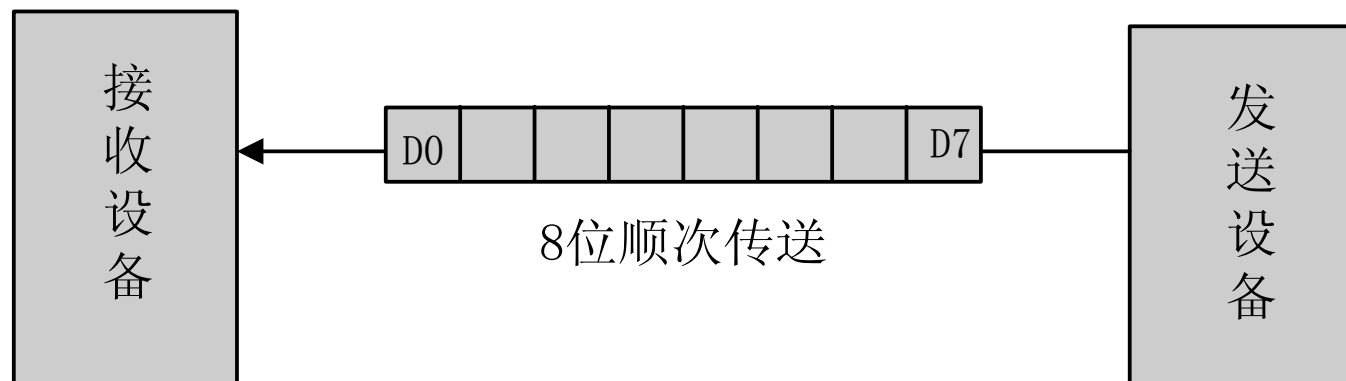
- 随着多微机系统的广泛应用和计算机网络技术的普及，计算机的通信功能愈来愈显得重要。**计算机通信**是指计算机与外部设备或计算机与计算机之间的信息交换。
- 通信有**并行通信**和**串行通信**两种方式。在多微机系统以及现代测控系统中信息的交换多采用串行通信方式。

- **计算机通信**是将计算机技术和通信技术的相结合，完成计算机与外部设备或计算机与计算机之间的信息交换。可以分为两大类：并行通信与串行通信。
- **并行通信**通常是将数据字节的各位用多条数据线同时进行传送。



并行通信控制简单、传输速度快；由于传输线较多，长距离传送时成本高且接收方的各位同时接收存在困难。

串行通信是将数据字节分成一位一位的形式在一条传输线上逐个地传送。



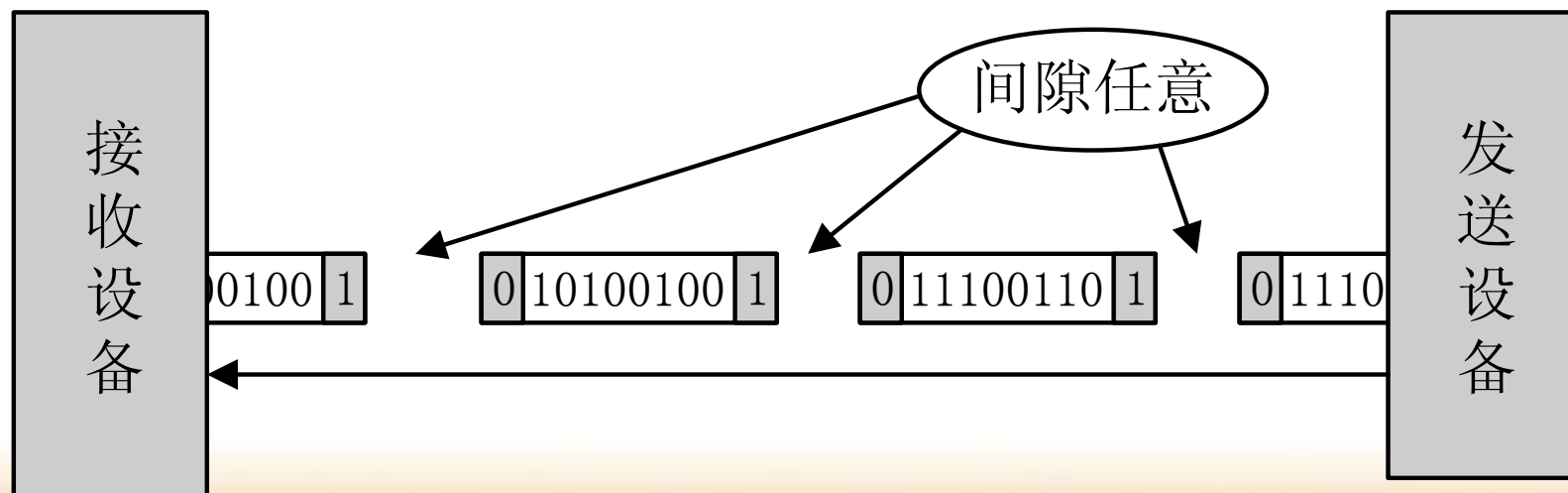
串行通信的特点：传输线少，长距离传送时成本低，且可以利用电话网等现成的设备，但数据的传送控制比并行通信复杂。

6.1.1 串行通信的基本概念

一、异步通信与同步通信

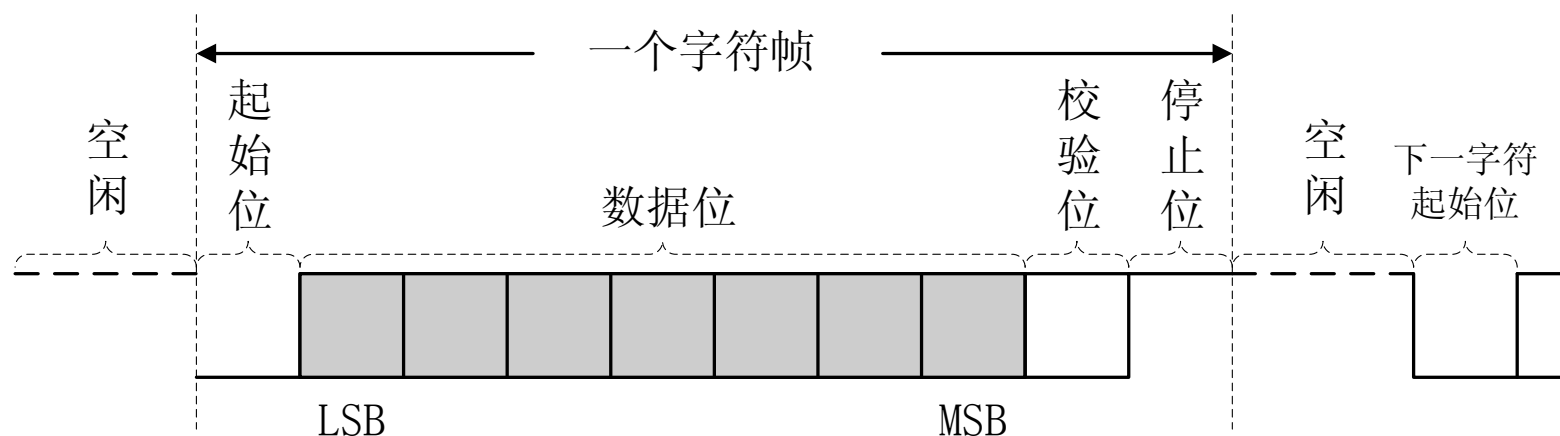
1、异步通信

异步通信是指通信的发送与接收设备使用各自的时钟控制数据的发送和接收过程。为使双方的收发协调，要求发送和接收设备的时钟尽可能一致。



异步通信是以字符（构成的帧）为单位进行传输，字符与字符之间的间隙（时间间隔）是任意的，但每个字符中的各位是以固定的时间传送的，即字符之间是异步的（字符之间不一定有“位间隔”的整数倍的关系），但同一字符内的各位是同步的（各位之间的距离均为“位间隔”的整数倍）。

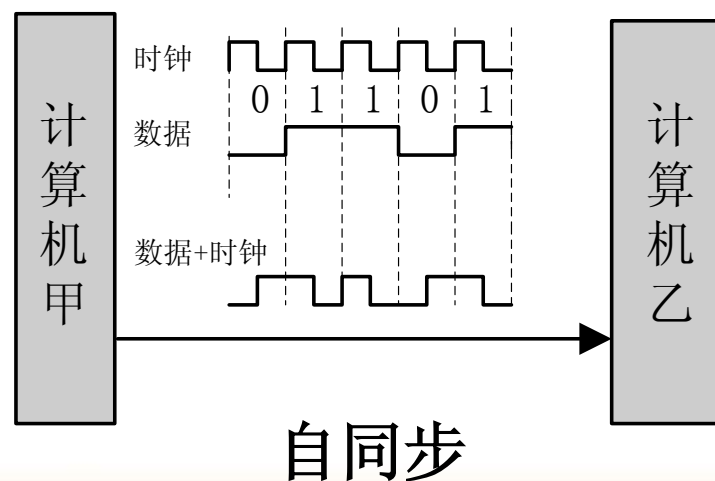
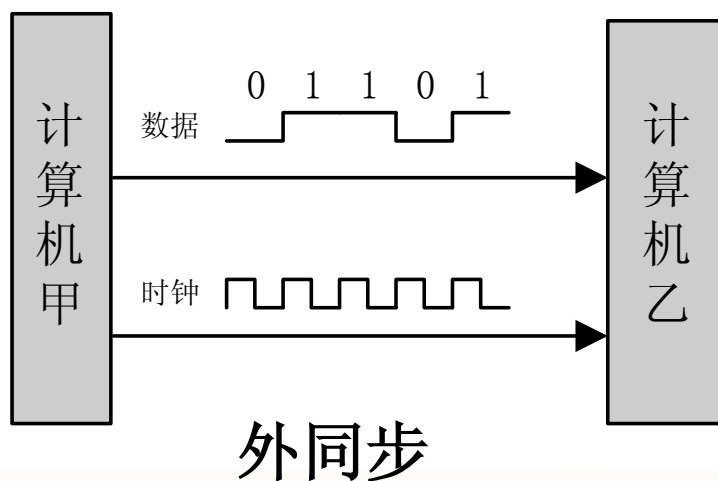
异步通信的数据格式：



异步通信的特点：不要求收发双方时钟的严格一致，实现容易，设备开销较小，但每个字符要附加**2~3**位用于起止位，各帧之间还有间隔，因此传输效率不高。

2、同步通信

同步通信时要建立发送方时钟对接收方时钟的直接控制，使双方达到完全同步。此时，传输数据的位之间的距离均为“位间隔”的整数倍，同时传送的字符间不留间隙，即**保持位同步**关系，也**保持字符同步**关系。发送方对接收方的同步可以通过两种方法实现。



面向字符的同步格式：

SYN	SYN	SOH	标题	STX	数据块	ETB/ETX	块校验
-----	-----	-----	----	-----	-----	---------	-----

此时，传送的数据和控制信息都必须由规定的字符集（如**ASCII**码）中的字符所组成。图中帧头为1个或2个同步字符**SYN**（**ASCII**码为**16H**）。**SOH**为序始字符（**ASCII**码为**01H**），表示标题的开始，标题中包含源地址、目标地址和路由指示等信息。**STX**为文始字符（**ASCII**码为**02H**），表示传送的数据块开始。数据块是传送的正文内容，由多个字符组成。数据块后面是组终字符**ETB**（**ASCII**码为**17H**）或文终字符**ETX**（**ASCII**码为**03H**）。然后是校验码。典型的面向字符的同步规程如**IBM**的二进制同步规程**BSC**。

面向位的同步格式：

8位	8位	8位	≥ 0 位	16位	8位
01111110	地址场	控制场	信息场	校验场	01111110

此时，将数据块看作数据流，并用序列**01111110**作为开始和结束标志。为了避免在数据流中出现序列**01111110**时引起的混乱，发送方总是在其发送的数据流中每出现**5**个连续的**1**就插入一个附加的**0**；接收方则每检测到**5**个连续的**1**并且其后有一个**0**时，就删除该**0**。

典型的面向位的同步协议如**ISO**的高级数据链路控制规程**HDLC**和**IBM**的同步数据链路控制规程**SDLC**。

同步通信的特点是以特定的位组合“**01111110**”作为帧的开始和结束标志，所传输的一帧数据可以是任意位。所以传输的效率较高，但实现的硬件设备比异步通信复杂。

二、串行通信的传输方向

1、单工

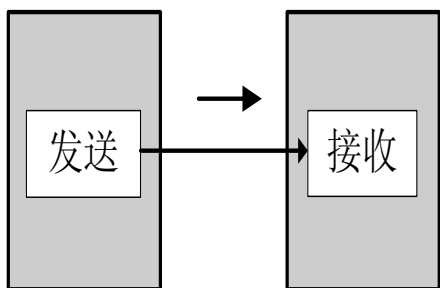
单工是指数据传输仅能沿一个方向，不能实现反向传输。

2、半双工

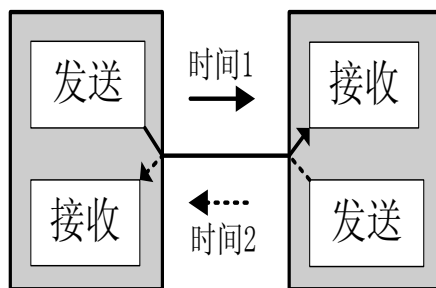
半双工是指数据传输可以沿两个方向，但需要分时进行。

3、全双工

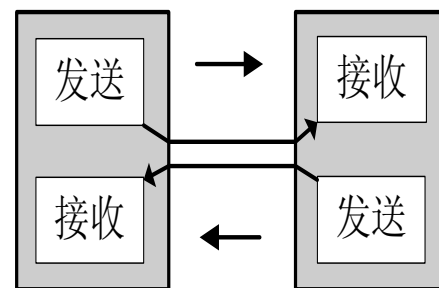
全双工是指数据可以同时进行双向传输。



单工



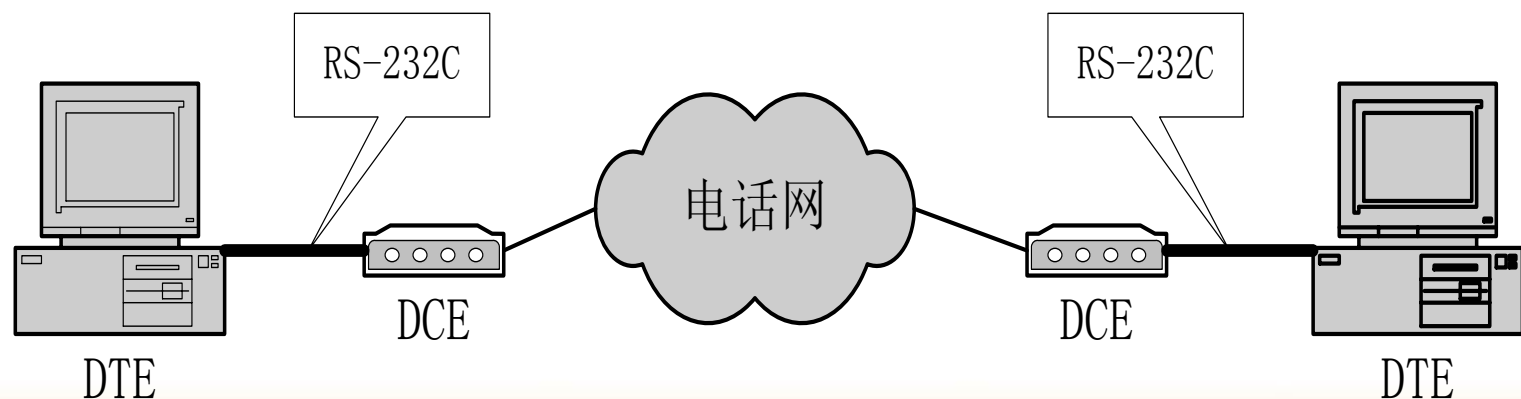
半双工



全双工

三、信号的调制与解调

利用调制器（**Modulator**）把数字信号转换成模拟信号，然后送到通信线路上去，再由解调器（**Demodulator**）把从通信线路上收到的模拟信号转换成数字信号。由于通信是双向的，调制器和解调器合并在一个装置中，这就是调制解调器 **MODEM**。



四、串行通信的错误校验

1、奇偶校验

在发送数据时，数据位尾随的1位为奇偶校验位（1或0）。奇校验时，数据中“1”的个数与校验位“1”的个数之和应为奇数；偶校验时，数据中“1”的个数与校验位“1”的个数之和应为偶数。接收字符时，对“1”的个数进行校验，若发现不一致，则说明传输数据过程中出现了差错。

2、代码和校验

代码和校验是发送方将所发数据块求和（或各字节异或），产生一个字节的校验字符（校验和）附加到数据块末尾。接收方接收数据同时对数据块（除校验字节外）求和（或各字节异或），将所得的结果与发送方的“校验和”进行比较，相符则无差错，否则即认为传送过程中出现了差错。

3、循环冗余校验

这种校验是通过某种数学运算实现有效信息与校验位之间的循环校验，常用于对磁盘信息的传输、存储区的完整性校验等。这种校验方法纠错能力强，广泛应用于同步通信中。

五、传输速率与传输距离

1、传输速率

比特率是每秒钟传输二进制代码的位数，单位是：位/秒（**bps**）。如每秒钟传送**240**个字符，而每个字符格式包含**10**位（**1**个起始位、**1**个停止位、**8**个数据位），这时的比特率为：

$$10\text{位} \times 240\text{个/秒} = 2400\text{ bps}$$

波特率表示每秒钟调制信号变化的次数，单位是：波特（**Baud**）。

波特率和比特率不总是相同的，对于将数字信号**1**或**0**直接用两种不同电压表示的所谓基带传输，**比特率和波特率是相同的**。所以，我们也经常用波特率表示数据的传输速率。

2、传输距离与传输速率的关系

串行接口或终端直接传送串行信息位流的最大距离与传输速率及传输线的电气特性有关。当传输线使用每0.3m（约1英尺）有50PF电容的非平衡屏蔽双绞线时，传输距离随传输速率的增加而减小。当比特率超过1000 bps 时，最大传输距离迅速下降，如9600 bps 时最大距离下降到只有76m（约250英尺）。

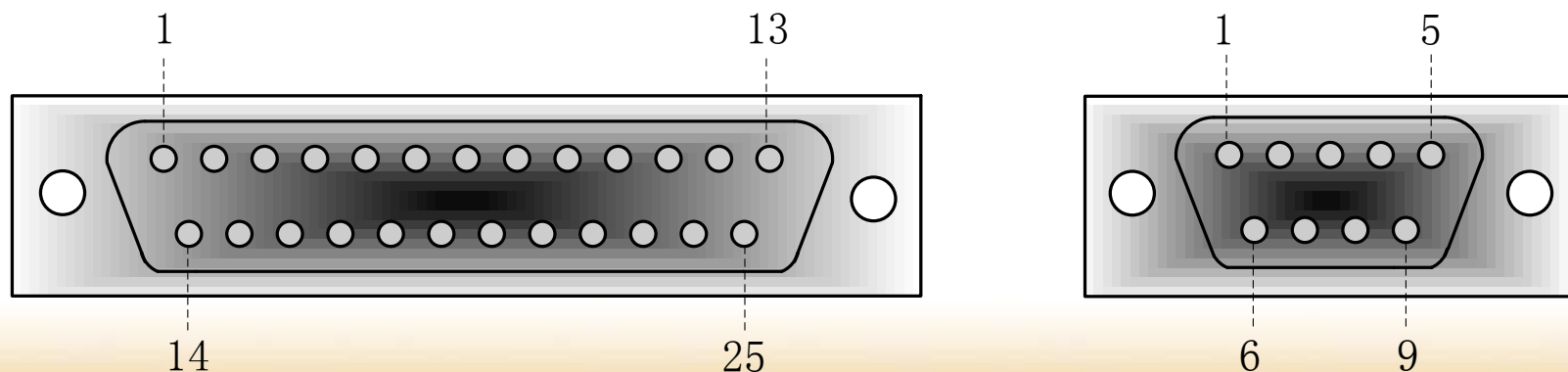
6.1.2 串行通信接口标准

一、RS-232C接口

RS-232C是EIA（美国电子工业协会）1969年修订**RS-232C**标准。**RS-232C**定义了数据终端设备（**DTE**）与数据通信设备（**DCE**）之间的物理接口标准。

1、机械特性

RS-232C接口规定使用**25**针连接器，连接器的尺寸及每个插针的排列位置都有明确的定义。（阳头）



2、功能特性

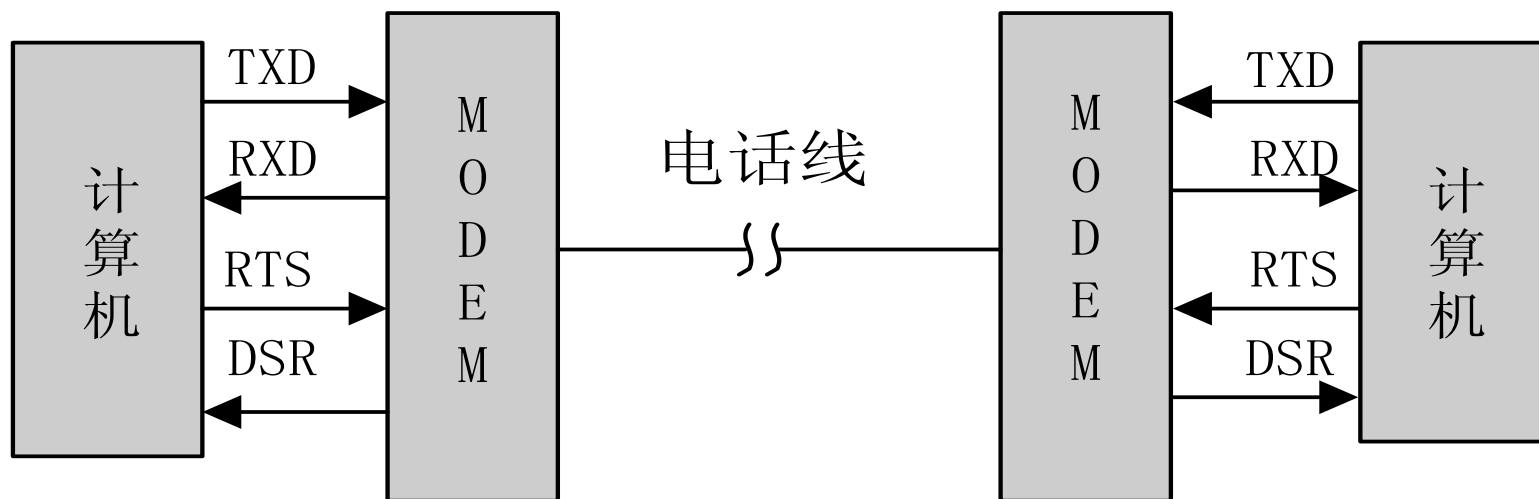
RS-232C 标准接口主要引脚定义

插针序号	信号名称	功能	信号方向
1	PGND	保护接地	
2 (3)	TXD	发送数据 (串行输出)	DTE→DCE
3 (2)	RXD	接收数据 (串行输入)	DTE←DCE
4 (7)	RTS	请求发送	DTE→DCE
5 (8)	CTS	允许发送	DTE←DCE
6 (6)	DSR	DCE 就绪 (数据建立就绪)	DTE←DCE
7 (5)	SGND	信号接地	
8 (1)	DCD	载波检测	DTE←DCE
20 (4)	DTR	DTE 就绪 (数据终端准备就绪)	DTE→DCE
22 (9)	RI	振铃指示	DTE←DCE

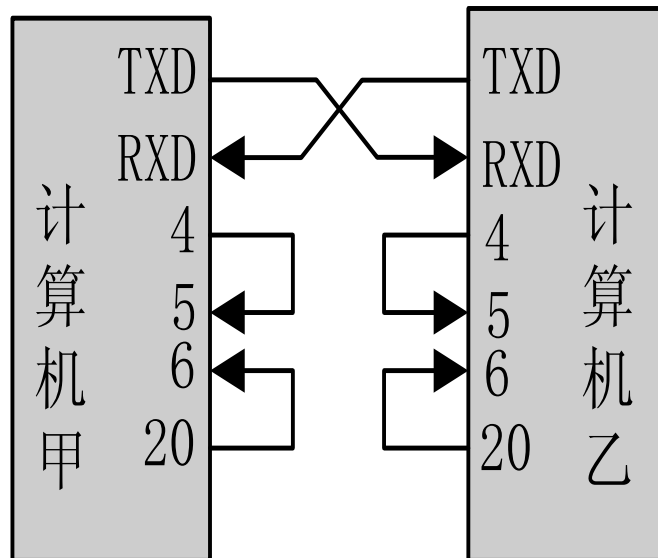
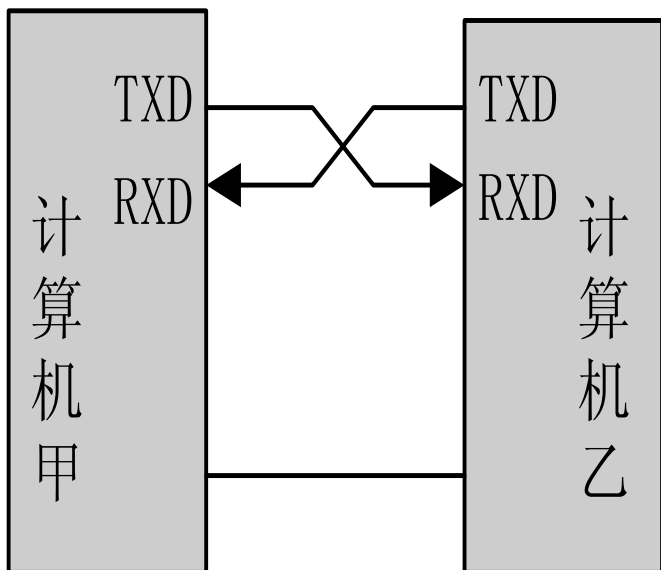
注：插针序号 () 内为 9 针非标准连接器的引脚号

4、过程特性

过程特性规定了信号之间的时序关系，以便正确地接收和发送数据。

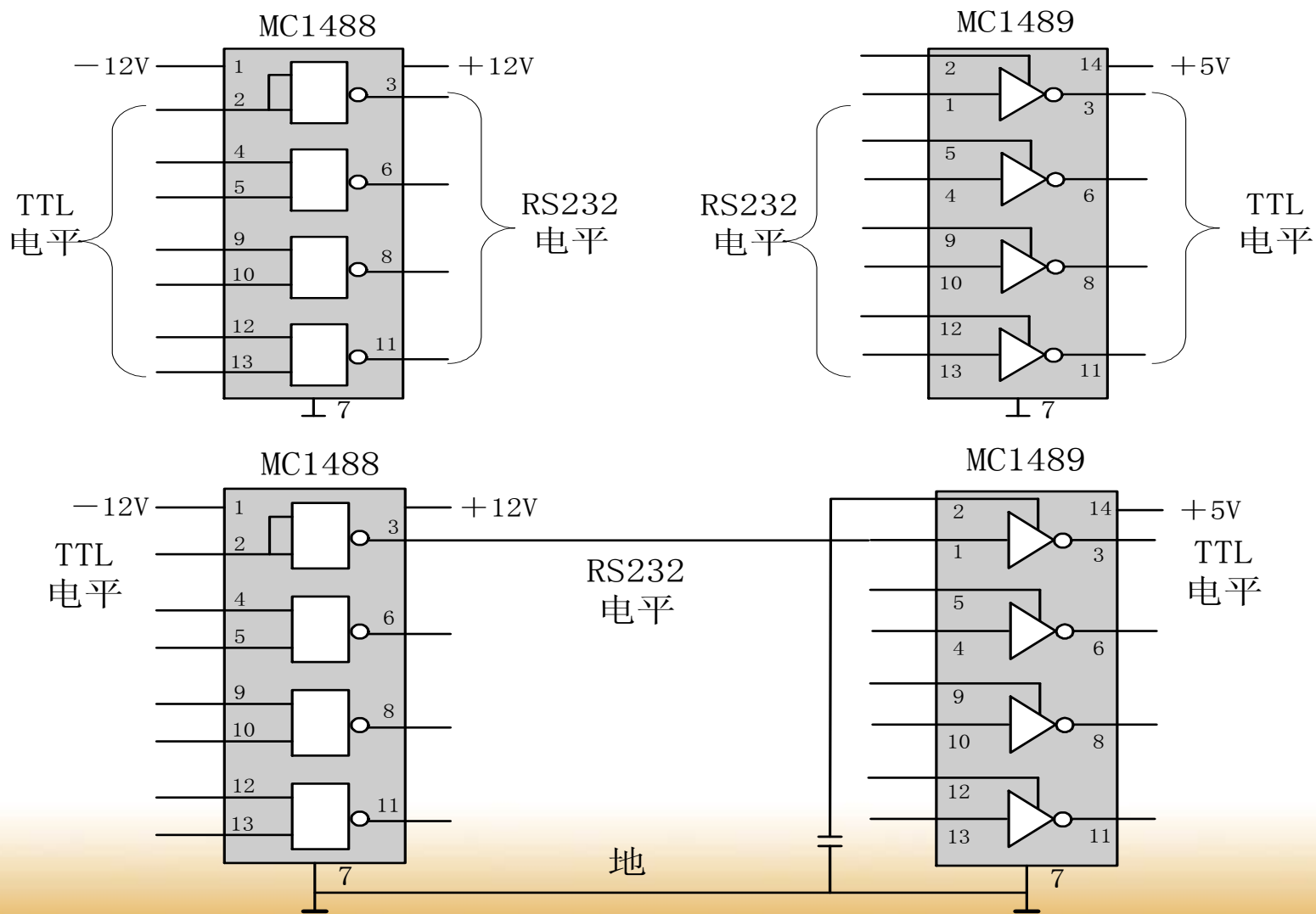


远程通信连接



近程通信连接

5、RS-232C电平与TTL电平转换驱动电路



6、采用RS-232C接口存在的问题

1、传输距离短，传输速率低

RS-232C总线标准受电容允许值的约束，使用时传输距离一般不要超过**15米**（线路条件好时也不超过几十米）。最高传送速率为**20Kbps**。

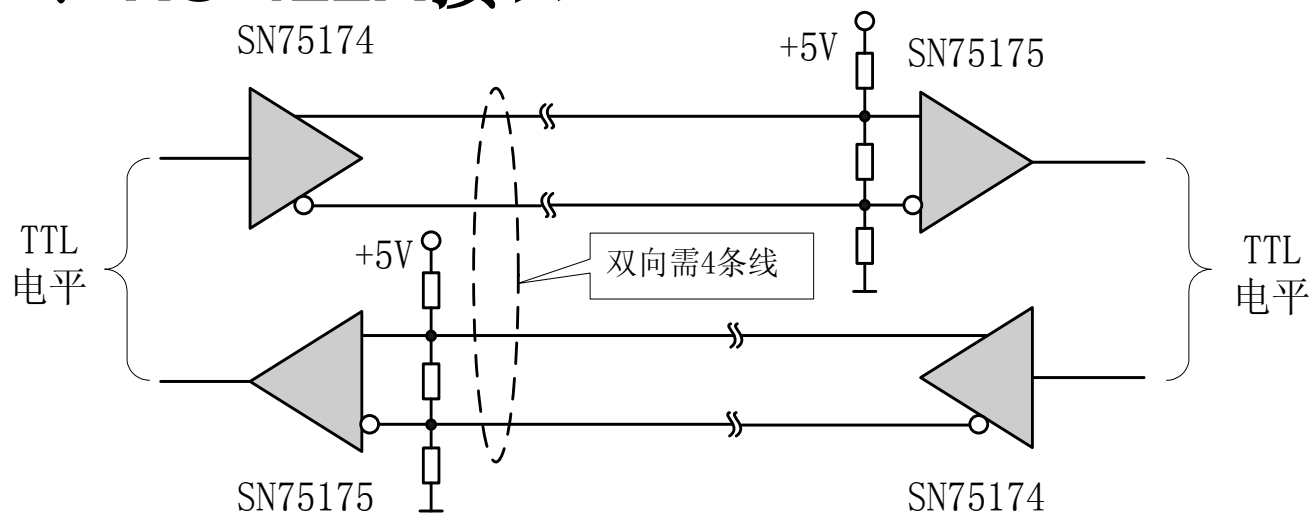
2、有电平偏移

RS-232C总线标准要求收发双方共地。通信距离较大时，收发双方的地电位差别较大，在信号地上将有比较大的地电流并产生压降。

3、抗干扰能力差

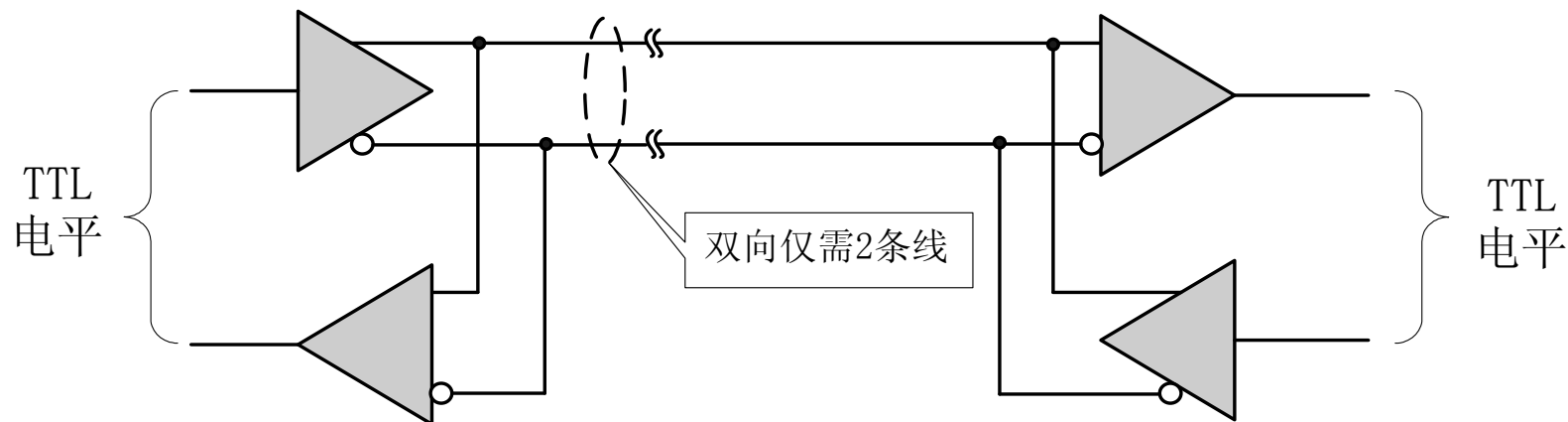
RS-232C在电平转换时采用单端输入输出，在传输过程中当干扰和噪声混在正常的信号中。为了提高信噪比，**RS-232C**总线标准不得不采用比较大的电压摆幅。

二、RS-422A接口



RS-422A输出驱动器为双端平衡驱动器。如果其中一条线为逻辑“1”状态，另一条线就为逻辑“0”，比采用单端不平衡驱动对电压的放大倍数大一倍。**差分电路**能从地线干扰中拾取有效信号，差分接收器可以分辨**200mV**以上电位差。若传输过程中混入了干扰和噪声，由于差分放大器的作用，可使干扰和噪声相互抵消。因此可以避免或大大减弱地线干扰和电磁干扰的影响。**RS-422A传输速率（90Kbps）**时，**传输距离可达1200米。**

三、RS-485接口



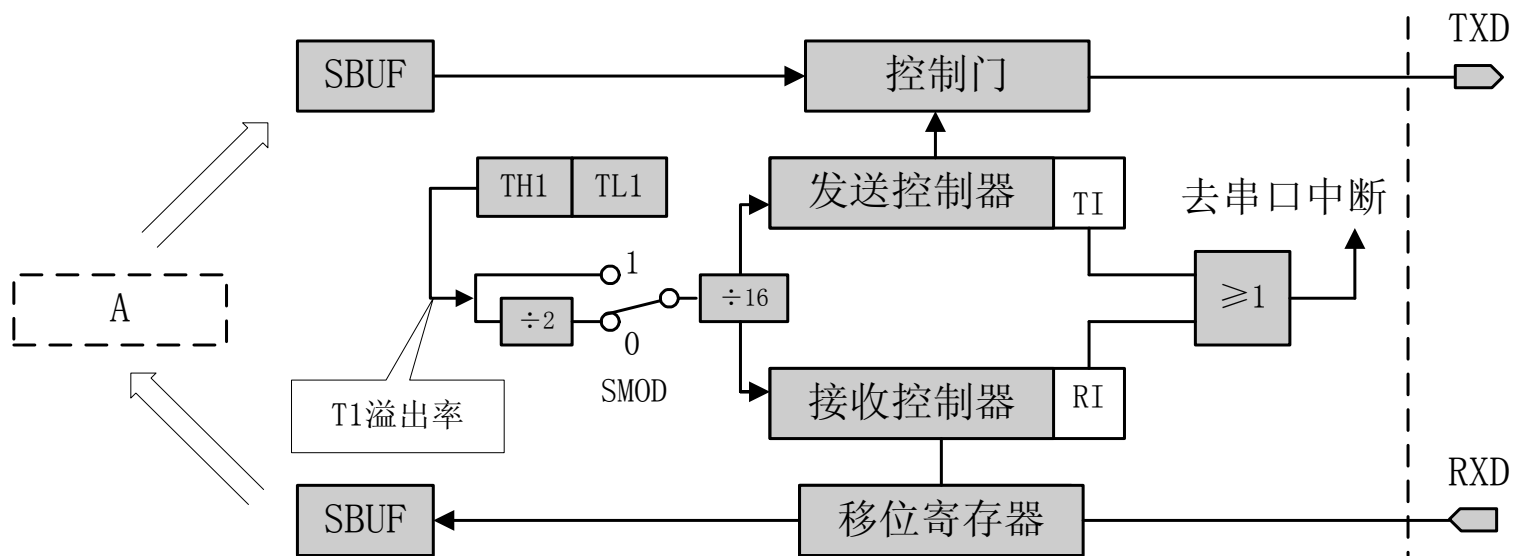
RS-485是RS-422A的变型：RS-422A用于全双工，而RS-485则用于半双工。RS-485是一种多发送器标准，在通信线路上最多可以使用32对差分驱动器/接收器。如果在一个网络中连接的设备超过32个，还可以使用中继器。

RS-485的信号传输采用两线间的电压来表示逻辑1和逻辑0。由于发送方需要两根传输线，接收方也需要两根传输线。传输线采用差动信道，所以它的干扰抑制性极好，又因为它的阻抗低，无接地问题，所以传输距离可达1200米，传输速率可达1Mbps。

RS-485是一点对多点的通信接口，一般采用**双绞线**的结构。普通的**PC**机一般不带**RS485**接口，因此要使用**RS-232C/RS-485**转换器。对于单片机可以通过芯片**MAX485**来完成**TTL/RS-485**的电平转换。在计算机和单片机组成的**RS-485**通信系统中，下位机由单片机系统组成，上位机为普通的**PC**机，负责监视下位机的运行状态，并对其状态信息进行集中处理，以图文方式显示下位机的工作状态以及工业现场被控设备的工作状况。系统中各节点（包括上位机）的识别是通过设置不同的站地址来实现的。

6.2 80C51的串行口

6.2.1 80C51串行口的结构



有两个物理上独立的接收、发送缓冲器**SBUF**，它们占用同一地址**99H**；接收器是双缓冲结构；发送缓冲器，因为发送时**CPU**是主动的，不会产生重叠错误。

6.2.2 80C51串行口的控制寄存器

SCON 是一个特殊功能寄存器，用以设定串行口的工作方式、接收/发送控制以及设置状态标志：

位	7	6	5	4	3	2	1	0	
字节地址：98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	SCON

SM0和**SM1**为工作方式选择位，可选择四种工作方式：

I[≡]

串行口的工作方式

SM0	SM1	方式	说明	波特率
0	0	0	移位寄存器	$f_{osc}/12$
0	1	1	10位异步收发器（8位数据）	可变
1	0	2	11位异步收发器（9位数据）	$f_{osc}/64$ 或 $f_{osc}/32$
1	1	3	11位异步收发器（9位数据）	可变

● **SM2**，多机通信控制位，主要用于方式2和方式3。当接收机的**SM2=1**时可以利用收到的**RB8**来控制是否激活**RI**（**RB8=0**时不激活**RI**，收到的信息丢弃；**RB8=1**时收到的数据进入**SBUF**，并激活**RI**，进而在中断服务中将数据从**SBUF**读走）。当**SM2=0**时，不论收到的**RB8**为**0**和**1**，均可以使收到的数据进入**SBUF**，并激活**RI**（即此时**RB8**不具有控制**RI**激活的功能）。通过控制**SM2**，可以实现多机通信。

在方式**0**时，**SM2**必须是**0**。在方式**1**时，若**SM2=1**，则只有接收到有效停止位时，**RI**才置**1**。

● **REN**，允许串行接收位。由软件置**REN=1**，则启动串行口接收数据；若软件置**REN=0**，则禁止接收。

● **TB8**，在方式**2**或方式**3**中，是发送数据的第九位，可以用软件规定其作用。可以用作数据的奇偶校验位，或在多机通信中，作为地址帧/数据帧的标志位。

在方式**0**和方式**1**中，该位未用。

● **RB8**，在方式**2**或方式**3**中，是接收到数据的第九位，作为奇偶校验位或地址帧/数据帧的标志位。在方式**1**时，若**SM2=0**，则**RB8**是接收到的停止位。

● **TI，发送中断标志位**。在方式0时，当串行发送第8位数据结束时，或在其它方式，串行发送停止位的开始时，由内部硬件使TI置1，向CPU发中断申请。在中断服务程序中，必须用软件将其清0，取消此中断申请。

● **RI，接收中断标志位**。在方式0时，当串行接收第8位数据结束时，或在其它方式，串行接收停止位的中间时，由内部硬件使RI置1，向CPU发中断申请。也必须在中断服务程序中，用软件将其清0，取消此中断申请。

PCON中只有一位SMOD与串行口工作有关：

位 [Ⓟ]	7 [Ⓟ]	6 [Ⓟ]	5 [Ⓟ]	4 [Ⓟ]	3 [Ⓟ]	2 [Ⓟ]	1 [Ⓟ]	0 [Ⓟ]	
字节地址：97H [Ⓟ]	SMOD	[Ⓟ]	[Ⓟ]	[Ⓟ]	[Ⓟ]	[Ⓟ]	[Ⓟ]	[Ⓟ]	PCON [Ⓟ]

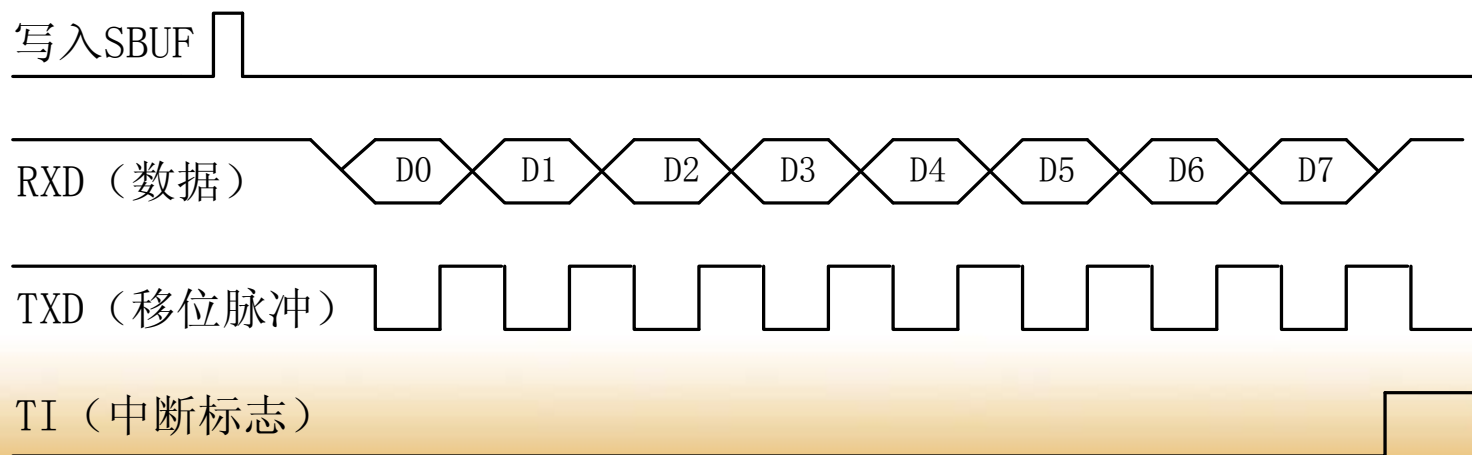
SMOD (PCON.7) 波特率倍增位。在串行口方式1、方式2、方式3时，波特率与**SMOD**有关，当**SMOD=1**时，波特率提高一倍。复位时，**SMOD=0**。

6.2.3 80C51串行口的工作方式

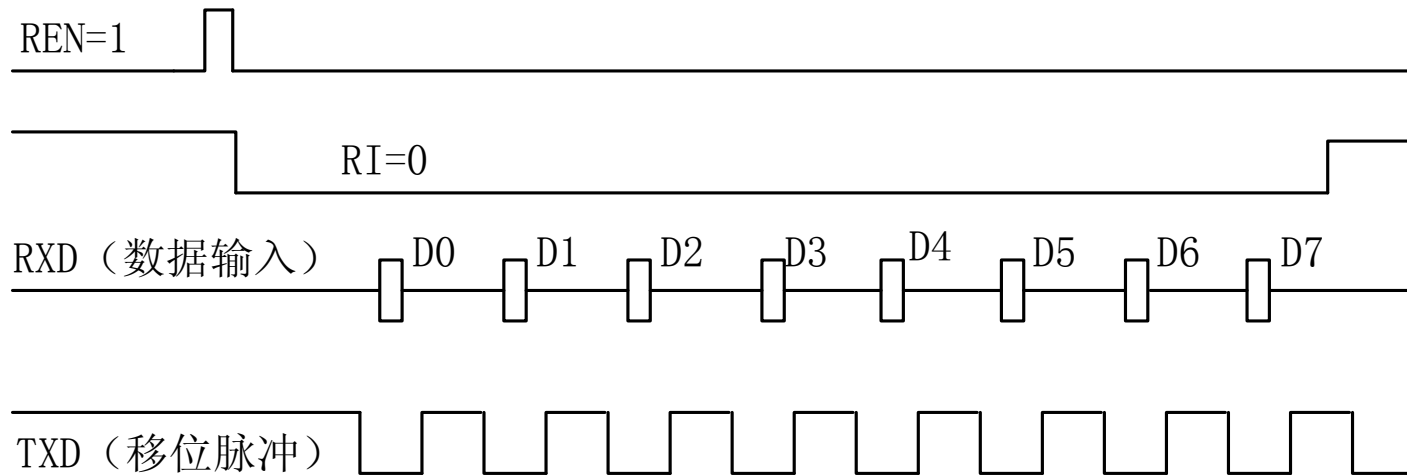
一、方式0

方式0时，串行口为同步移位寄存器的输入输出方式。主要用于扩展并行输入或输出口。数据由RXD（P3.0）引脚输入或输出，同步移位脉冲由TXD（P3.1）引脚输出。发送和接收均为8位数据，低位在先，高位在后。波特率固定为 $f_{osc}/12$ 。

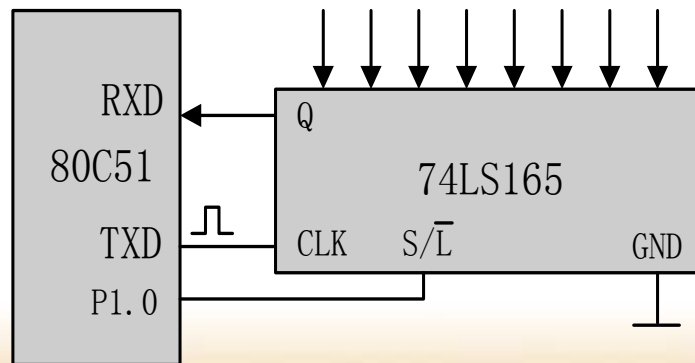
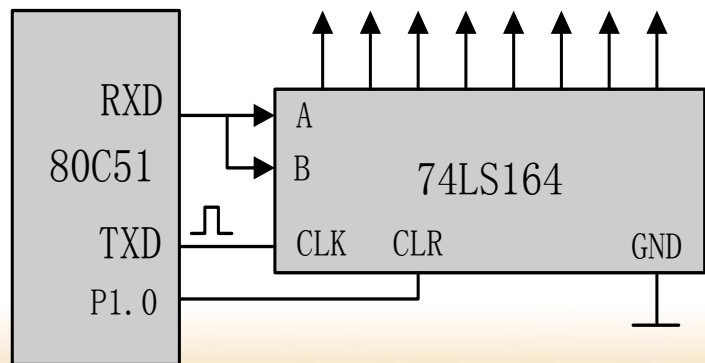
1、方式0输出



2、方式0输入

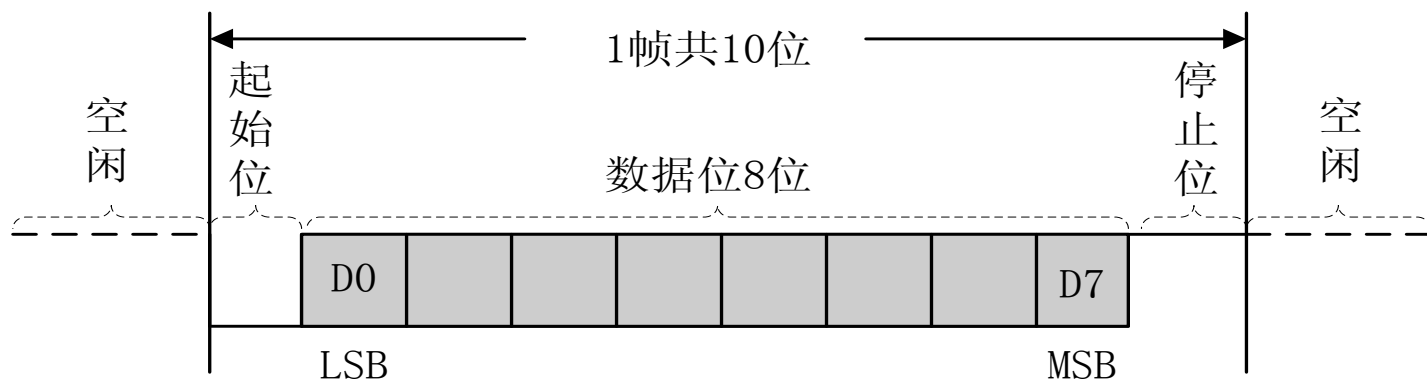


方式0接收和发送电路

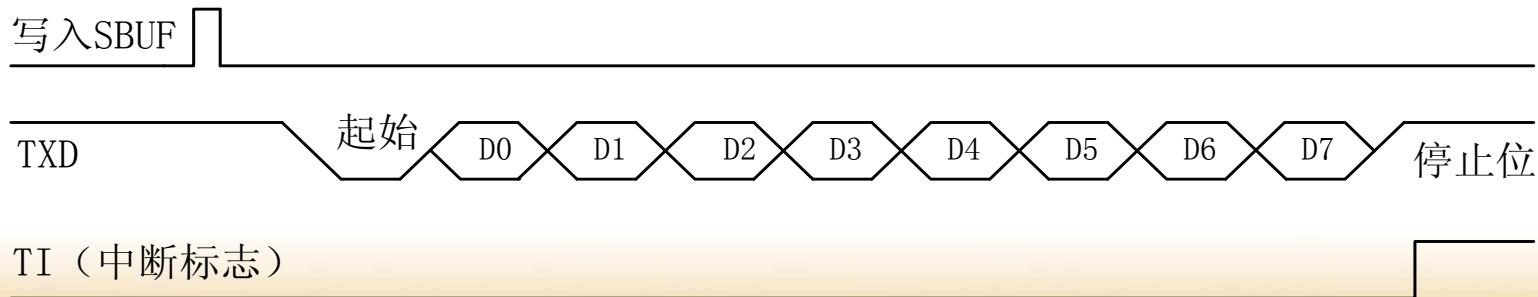


二、方式1

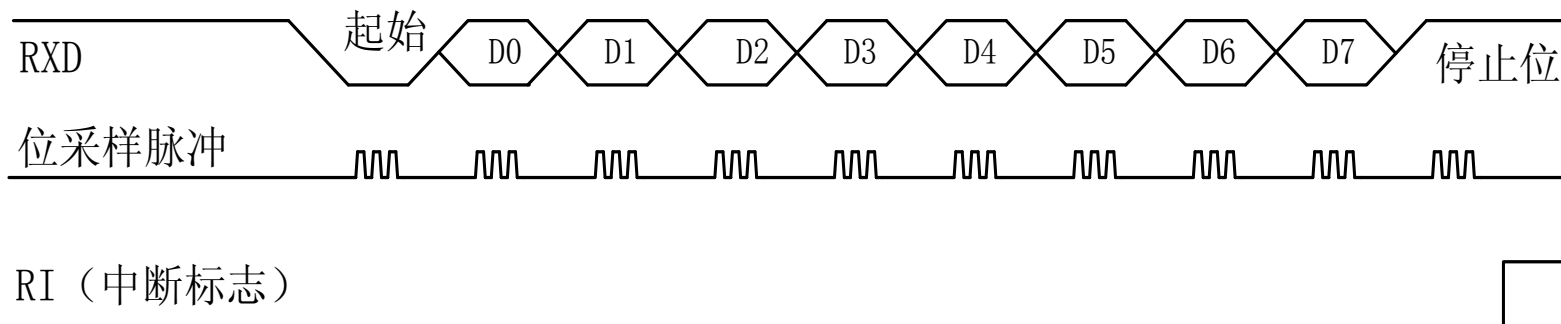
方式1是10位数据的异步通信口。**TXD**为数据发送引脚，**RXD**为数据接收引脚，传送一帧数据的格式如图所示。其中1位起始位，8位数据位，1位停止位。



1、方式1输出



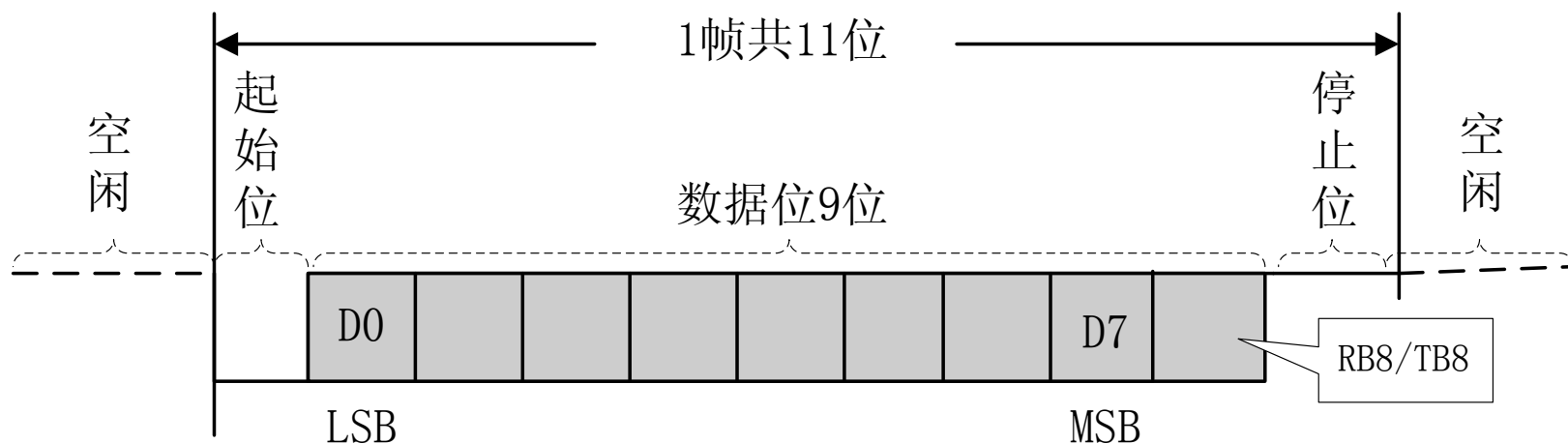
2、方式1输入



用软件置**REN**为**1**时，接收器以所选择波特率的**16**倍速率采样**RXD**引脚电平，检测到**RXD**引脚输入电平发生负跳变时，则说明起始位有效，将其移入输入移位寄存器，并开始接收这一帧信息的其余位。接收过程中，数据从输入移位寄存器右边移入，起始位移至输入移位寄存器最左边时，控制电路进行最后一次移位。当**RI=0**，且**SM2=0**（或接收到的停止位为**1**）时，将接收到的**9**位数据的前**8**位数据装入接收**SBUF**，第**9**位（停止位）进入**RB8**，并置**RI=1**，向**CPU**请求中断。

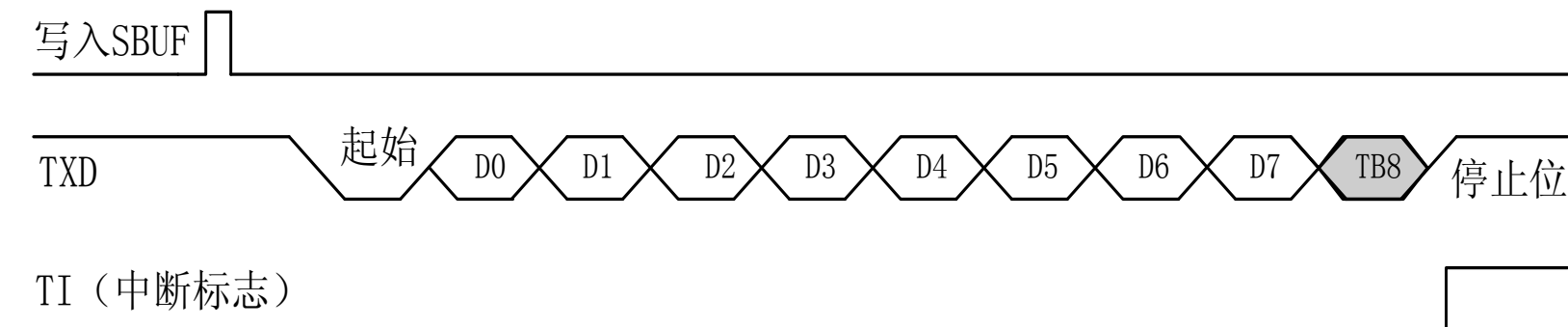
三、方式2和方式3

方式2或方式3时为11位数据的异步通信口。**TXD**为数据发送引脚，**RXD**为数据接收引脚。



方式2和方式3时起始位1位，数据9位（含1位附加的第9位，发送时为**SCON**中的**TB8**，接收时为**RB8**），停止位1位，一帧数据为11位。方式2的波特率固定为晶振频率的1/64或1/32，方式3的波特率由定时器T1的溢出率决定。

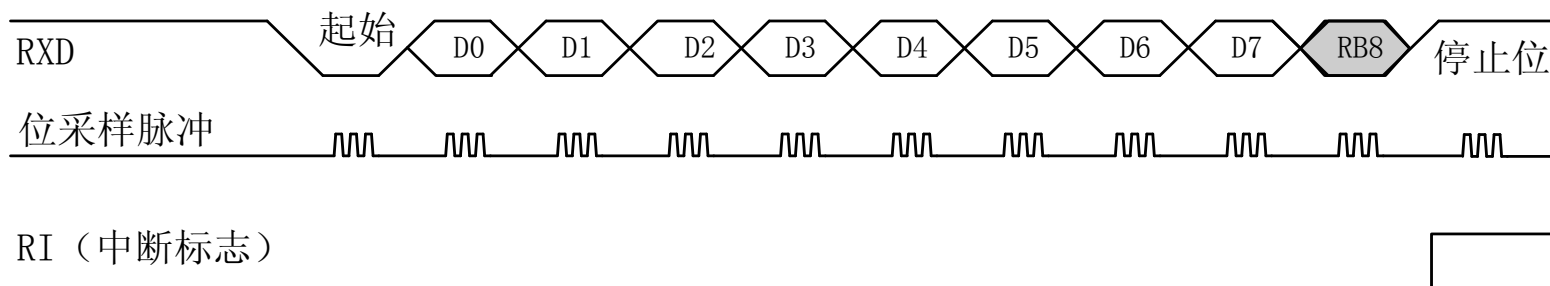
1、方式2和方式3输出



发送开始时，先把起始位**0**输出到**TXD**引脚，然后发送移位寄存器的输出位（**D0**）到**TXD**引脚。每一个移位脉冲都使输出移位寄存器的各位右移一位，并由**TXD**引脚输出。

第一次移位时，停止位“**1**”移入输出移位寄存器的第**9**位上，以后每次移位，左边都移入**0**。当停止位移至输出位时，左边其余位全为**0**，检测电路检测到这一条件时，使控制电路进行最后一次移位，并置**TI=1**，向**CPU**请求中断。

2、方式2和方式3输入



接收时，数据从右边移入输入移位寄存器，在起始位0移到最左边时，控制电路进行最后一次移位。当 $RI=0$ ，且 $SM2=0$ （或接收到的第9位数据为1）时，接收到的数据装入接收缓冲器SBUF和RB8（接收数据的第9位），置 $RI=1$ ，向CPU请求中断。如果条件不满足，则数据丢失，且不置位RI，继续搜索RXD引脚的负跳变。

四、波特率的计算

在串行通信中，收发双方对发送或接收数据的速率要有约定。通过软件可对单片机串行口编程为四种工作方式，其中方式0和方式2的波特率是固定的，而方式1和方式3的波特率是可变的，由定时器T1的溢出率来决定。

串行口的四种工作方式对应三种波特率。由于输入的移位时钟的来源不同，所以，各种方式的波特率计算公式也不相同。

方式0的波特率 = $f_{osc}/12$

方式2的波特率 = $(2^{SMOD}/64) \cdot f_{osc}$

方式1的波特率 = $(2^{SMOD}/32) \cdot (T1\text{溢出率})$

方式3的波特率 = $(2^{SMOD}/32) \cdot (T1\text{溢出率})$

当T1作为波特率发生器时，最典型的用法是使T1工作在自动再装入的8位定时器方式（即方式2，且TCON的TR1=1，以启动定时器）。这时溢出率取决于TH1中的计数值。

$$T1 \text{ 溢出率} = f_{osc} / \{12 \times [256 - (TH1)]\}$$

在单片机的应用中，常用的晶振频率为：12MHz和11.0592MHz。所以，选用的波特率也相对固定。常用的串行口波特率以及各参数的关系如表所示。

常用波特率与定时器 1 的参数关系

串口工作方式 及波特率/(b/s)		f _{osc} (MHz)	SMOD	定时器 T1		
				C/T	工作方式	初值
方式 1、3	62.5 k	12	1	0	2	FFH
	19.2 k	11.0592	1	0	2	FDH
	9600	11.0592	0	0	2	FDH
	4800	11.0592	0	0	2	FAH
	2400	11.0592	0	0	2	F4H
	1200	11.0592	0	0	2	E8H

串行口工作之前，应对其进行初始化，主要是设置产生波特率的定时器1、串行口控制和中断控制。具体步骤如下：

- 确定**T1**的工作方式（编程**TMOD**寄存器）；
- 计算**T1**的初值，装载**TH1**、**TL1**；
- 启动**T1**（编程**TCON**中的**TR1**位）；
- 确定串行口控制（编程**SCON**寄存器）；

串行口在中断方式工作时，要进行中断设置（编程**IE**、**IP**寄存器）。

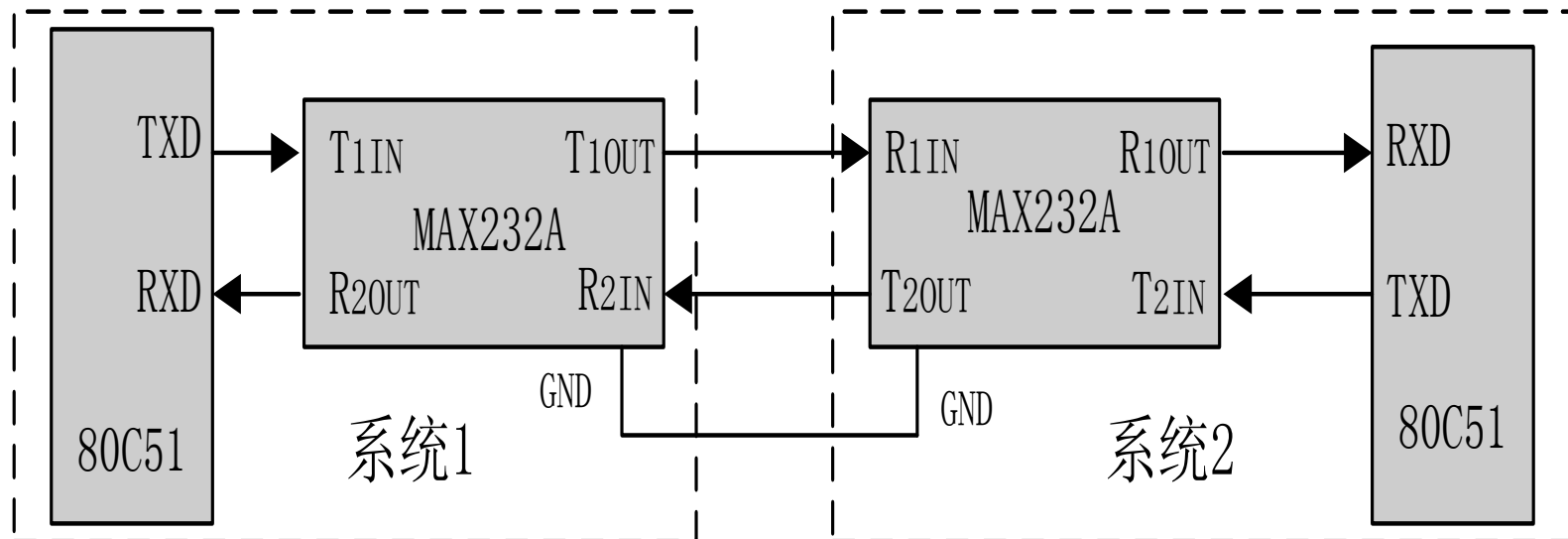
6.3 单片机串行口应用举例

在计算机分布式测控系统中，经常要利用串行通信方式进行数据传输。**80C51**单片机的串行口为计算机间的通信提供了极为便利的条件。利用单片机的串行口还可以方便地扩展键盘和显示器，对于简单的应用非常便利。这里仅介绍单片机串行口在通信方面的应用，关于键盘和显示器的扩展将在下一章介绍。

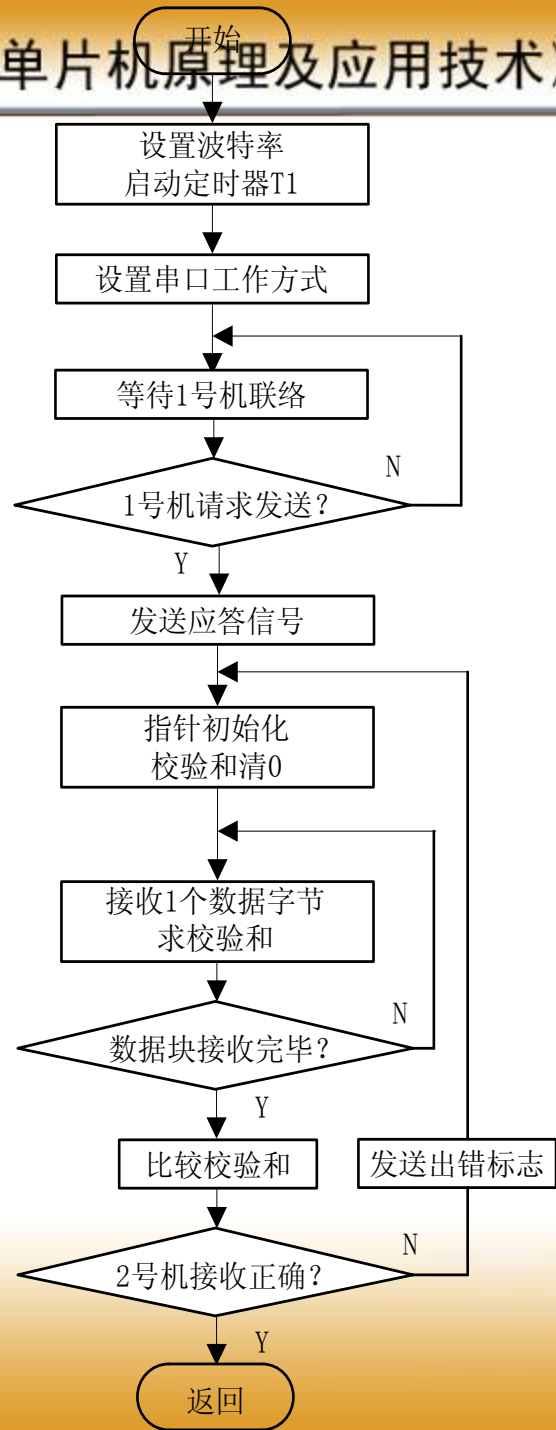
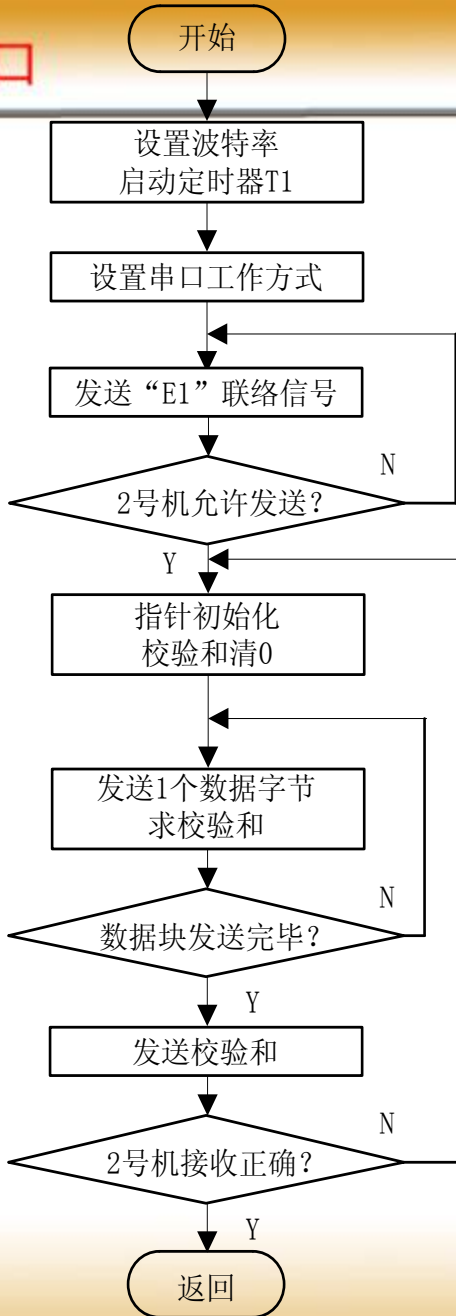
6.3.1 单片机与单片机的通信

一、点对点的通信

1、硬件连接



二、应用程序



设1号机是发送方，2号机是接收方。当1号机发送时，先发送一个“E1”联络信号，2号机收到后回答一个“E2”应答信号，表示同意接收。当1号机收到应答信号“E2”后，开始发送数据，每发送一个数据字节都要计算“校验和”，假定数据块长度为16个字节，起始地址为40H，一个数据块发送完毕后立即发送“校验和”。2号机接收数据并转存到数据缓冲区，起始地址也为40H，每接收到一个数据字节便计算一次“校验和”，当收到一个数据块后，再接收1号机发来的“校验和”，并将它与2号机求出的校验和进行比较。若两者相等，说明接收正确，2号机回答00H；若两者不相等，说明接收不正确，2号机回答0FFH，请求重发。1号机接到00H后结束发送。若收到的答复非零，则重新发送数据一次。双方约定采用串行口方式1进行通信，一帧信息为10位，其中有1个起始位、8个数据位和一个停止位；波特率为2400波特，T1工作在定时器方式2，振荡频率选用11.0592MHz，查表可得TH1=TL1=0F4H，PCON寄存器的SMOD位为0。



发送程序清单如下：

ASTART: CLR EA

MOV TMOD, #20H ; 定时器1置为方式2

MOV TH1, #0F4H ; 装载定时器初值, 波特率2400

MOV TL1, #0F4H

MOV PCON, #00H

SETB TR1 ; 启动定时器

MOV SCON, #50H ; 设定串口方式1, 且准备接收应答信号

ALOOP1: MOV SBUF, #0E1H ; 发联络信号

JNB TI, \$; 等待一帧发送完毕

CLR TI ; 允许再发送

JNB RI, \$; 等待2号机的应答信号

CLR RI ; 允许再接收

MOV A, SBUF ; 2号机应答后, 读至A

XRL A, #0E2H ; 判断2号机是否准备完毕

JNZ ALOOP1 ; 2号机未准备好, 继续联络

ALOOP2: MOV R0, #40H ; 2号机准备好, 设定数据块地址指针初值

MOV R7, #10H ; 设定数据块长度初值

MOV R6, #00H ; 清校验和单元

```
ALOOP3: MOV SBUF, @R0    ; 发送一个数据字节
        MOV A, R6
        ADD A, @R0       ; 求校验和
        MOV R6, A        ; 保存校验和
        INC R0
        JNB TI, $
        CLR TI
        DJNZ R7, ALOOP3  ; 整个数据块是否发送完毕
        MOV SBUF, R6     ; 发送校验和
        JNB TI, $
        CLR TI
        JNB RI, $        ; 等待2号机的应答信号
        CLR RI
        MOV A, SBUF      ; 2号机应答, 读至A
        JNZ ALOOP2      ; 2号机应答“错误”, 转重新发送
        RET              ; 2号机应答“正确”, 返回
```



接收程序清单如下：

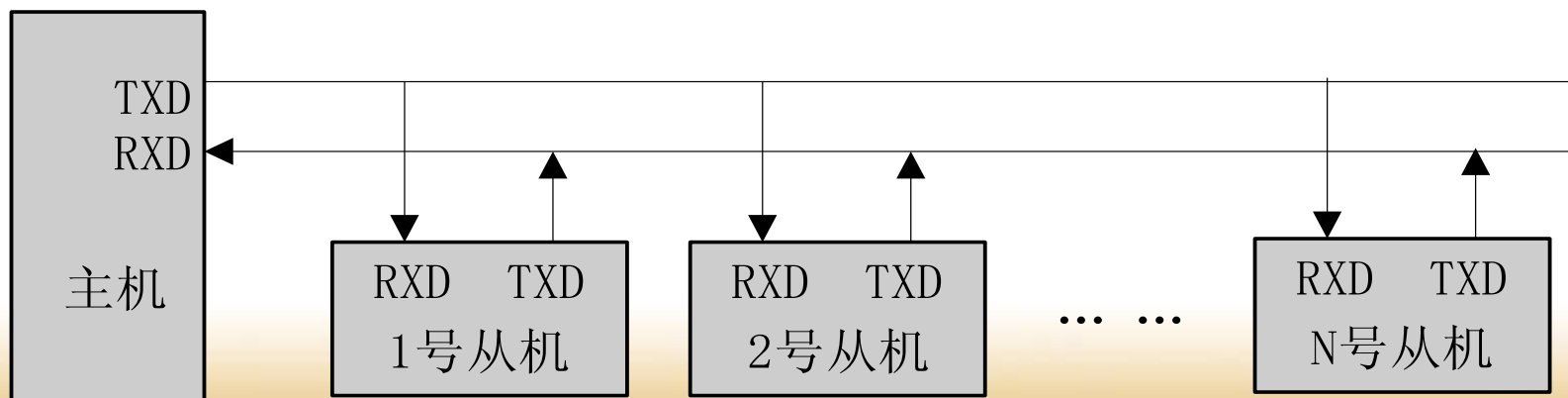
```
BSTART: CLR EA
        MOV TMOD, #20H
        MOV TH1, #0F4H
        MOV TL1, #0F4H
        MOV PCON, #00H
        SETB TR1
        MOV SCON, #50H ; 设定串口方式1, 且准备接收
BLOOP1: JNB RI, $      ; 等待1号机的联络信号
        CLR RI
        MOV A, SBUF    ; 收到1号机信号
        XRL A, #0E1H   ; 判是否为1号机联络信号
        JNZ BLOOP1    ; 不是1号机联络信号, 再等待
        MOV SBUF, #0E2H ; 是1号机联络信号, 发应答信号
        JNB TI, $
        CLR TI
        MOV R0, #40H   ; 设定数据块地址指针初值
        MOV R7, #10H   ; 设定数据块长度初值
        MOV R6, #00H   ; 清校验和单元
```

```
BLOOP2: JNB RI, $
        CLR RI
        MOV A, SBUF
        MOV @R0, A      ; 接收数据转储
        INC R0
        ADD A, R6       ; 求校验和
        MOV R6, A
        DJNZ R7, BLOOP2 ; 判数据块是否接收完毕
        JNB RI, $      ; 完毕, 接收1号机发来的校验和
        CLR RI
        MOV A, SBUF
        XRL A, R6      ; 比较校验和
        JZ END1        ; 校验和相等, 跳至发正确标志
        MOV SBUF, #0FFH ; 校验和不相等, 发错误标志
        JNB TI, $      ; 转重新接收
        CLR TI
END1: MOV SBUF, #00H
      RET
```


二、多机通信

1、硬件连接

单片机构成的多机系统常采用总线型主从式结构。所谓主从式，即在数个单片机中，有一个是主机，其余的是从机，从机要服从主机的调度、支配。**80C31**单片机的串行口方式**2**和方式**3**适于这种主从式的通信结构。当然采用不同的通信标准时，还需进行相应的电平转换，有时还要对信号进行光电隔离。在实际的多机应用系统中，常采用**RS-485**串行标准总线进行数据传输。



2、通信协议

- 所有从机的**SM2**位置**1**，处于接收地址帧状态。
- 主机发送一地址帧，其中**8**位是地址，第**9**位为地址/数据的区分标志，该位置**1**表示该帧为地址帧。
- 所有从机收到地址帧后，都将接收的地址与本机的地址比较。对于地址相符的从机，使自己的**SM2**位置**0**（以接收主机随后发来的数据帧），并把本站地址发回主机作为应答；对于地址不符的从机，仍保持**SM2=1**，对主机随后发来的数据帧不予理睬。
- 从机发送数据结束后，要发送一帧校验和，并置第**9**位（**TB8**）为**1**，作为从机数据传送结束的标志。

- 主机接收数据时先判断数据接收标志 (**RB8**)，若 **RB8=1**，表示数据传送结束，并比较此帧校验和，若正确则回送正确信号 **00H**，此信号命令该从机复位（即重新等待地址帧）；若校验和出错，则发送 **0FFH**，命令该从机重发数据。若接收帧的 **RB8=0**，则存数据到缓冲区，并准备接收下帧信息。
- 主机收到从机应答地址后，确认地址是否相符，如果地址不符，发复位信号（数据帧中 **TB8=1**）；如果地址相符，则清 **TB8**，开始发送数据。
- 从机收到复位命令后回到监听地址状态 (**SM2=1**)。否则开始接收数据和命令。

3、应用程序

- 主机发送的地址联络信号为：**00H, 01H, 02H,**（即从机设备地址），地址**FFH**为命令各从机复位，即恢复**SM2=1**。
- 主机命令编码为：**01H**，主机命令从机接收数据；**02H**，主机命令从机发送数据。其它都按**02H**对待。

从机状态字节格式为：

位	7	6	5	4	3	2	1	0
	ERR	0	0	0	0	0	TRDY	RRDY

RRDY=1：表示从机准备好接收。

TRDY=1：表示从机准备好发送。

ERR=1：表示从机接收的命令是非法的。

程序分为主机程序和从机程序。约定一次传递数据为**16**个字节，以**01H**地址的从机为例。

主机程序清单：

设从机地址号存于40H单元，命令存于41H单元。

```
MAIN: MOV TMOD, #20H    ; T1方式2
      MOV TH1, #0FDH    ; 初始化波特率9600
      MOV TL1, #0FDH
      MOV PCON, #00H
      SETB TR1
      MOV SCON, #0F0H   ; 串口方式3, 多机, 准备接收应答
LOOP1: SETB TB8
      MOV SBUF, 40H     ; 发送预通信从机地址
      JNB TI, $
      CLR TI
      JNB RI, $        ; 等待从机对联络应答
      CLR RI
      MOV A, SBUF      ; 接收应答, 读至A
      XRL A, 40H       ; 判应答的地址是否正确
      JZ  AD_OK
```



```
AD_ERR: MOV SBUF, #0FFH ; 应答错误, 发命令FFH
        JNB TI, $
        CLR TI
        SJMP LOOP1 ; 返回重新发送联络信号
AD_OK:  CLR TB8 ; 应答正确
        MOV SBUF, 41H ; 发送命令字
        JNB TI, $
        CLR TI
        JNB RI, $ ; 等待从机对命令应答
        CLR RI
        MOV A, SBUF ; 接收应答, 读至A
        XRL A, #80H ; 判断应答是否正确
        JNZ CO_OK
        SETB TB8
        SJMP AD_ERR ; 错误处理
```

```
CO_OK: MOV A, SBUF      ; 应答正确, 判是发送还是接收命令
        XRL A, #01H
        JZ  SE_DATA     ; 从机准备好接收, 可以发送
        MOV A, SBUF
        XRL A, #02H
        JZ  RE_DATA     ; 从机准备好发送, 可以接收
        LJMP SE_DATA
RE_DATA: MOV R6, #00H   ; 清校验和接收16个字节数据
        MOV R0, #30H
        MOV R7, #10H
```

```
LOOP2: JNB RI, $
        CLR RI
        MOV A, SBUF
        MOV @R0, A
        INC R0
        ADD A, R6
        MOV R6, A
        DJNZ R7, LOOP2
        JNB RI, $
        CLR RI
        MOV A, SBUF      ; 接收校验和并判断
        XRL A, R6
        JZ XYOK          ; 校验正确
        MOV SBUF, #0FFH ; 校验错误
        JNB TI, $
        CLR TI
        LJMP RE_DATA
```




```
XYOK: MOV SBUF, #00H ; 校验和正确, 发00H
      JNB TI, $
      CLR TI
      SETB TB8 ; 置地址标志
      LJMP RETEND
SE_DATA: MOV R6, #00H ; 发送16个字节数据
        MOV R0, #30H
        MOV R7, #10H
LOOP3: MOV A, @R0
      MOV SBUF, A
      JNB TI, $
      CLR TI
      INC R0
      ADD A, R6
      MOV R6, A
      DJNZ R7, LOOP3
```

```
MOV A, R6
MOV SBUF, A      ; 发校验和
JNB TI, $
CLR TI
JNB RI, $
CLR RI
MOV A, SBUF
XRL A, #00H
JZ  RET_END      ; 从机接收正确
SJMP SE_DATA     ; 从机接收不正确, 重新发送
RET_END: RET
```

从机程序清单：

设本机号存于40H单元，41H单元存放“发送”命令，42H单元存放“接收”命令。

```
MAIN: MOV TMOD, #20H    ; 初始化串行口
      MOV TH1, #0FDH
      MOV TL1, #0FDH
      MOV PCON, #00H
      SETB TR1
      MOV SCON, #0F0H

LOOP1: SETB EA          ; 开中断
      SETB ES
      SETB RRDY        ; 发送与接收准备就绪
      SETB TRDY
      SJMP LOOP1
```

```
SERVE: PUSH PSW          ; 中断服务程序
        PUSH ACC
        CLR ES
        CLR RI
        MOV A, SBUF
        XRL A, 40H        ; 判断是否本机地址
        JZ SER_OK
        LJMP ENDI        ; 非本机地址, 继续监听
SER_OK: CLR SM2          ; 是本机地址, 取消监听状态
        MOV SBUF, 40H    ; 本机地址发回
        JNB TI, $
        CLR TI
        JNB RI, $
        CLR RI
        JB RB8, ENDI     ; 是复位命令, 恢复监听
        MOV A, SBUF      ; 不是复位命令, 判是“发送”还是“接收”
        XRL A, 41H
        JZ SERISE        ; 收到“发送”命令, 发送处理
        MOV A, SBUF
        XRL A, 42H
        JZ SERIRE        ; 收到“接收”命令, 接收处理
        SJMP FFML        ; 非法命令, 转非法处理
```



```
SERISE: JB TRDY, SEND      ; 从机发送是否准备好
        MOV SBUF, #00H
        SJMP WAIT01
SEND:   MOV SBUF, #02H     ; 返回“发送准备好”
WAIT01: JNB TI, $
        CLR TI
        JNB RI, $
        CLR RI
        JB RB8, ENDII     ; 主机接收是否准备就绪
        LCALL SE_DATA    ; 发送数据
        LJMP END
FFML:   MOV SBUF, #80H    ; 发非法命令, 恢复监听
        JNB TI, $
        CLR TI
        LJMP ENDII
SERIRE: JB RRDY, RECE     ; 从机接收是否准备好
        MOV SBUF, #00H
        SJMP WAIT02
```

```
RECE: MOV SBUF, #01H      ; 返回“接收准备好”
WEIT02: JNB TI, $
        CLR TI
        JNB RI, $
        CLR RI
        JB RB8, ENDII     ; 主机发送是否就绪
        LCALL RE_DATA    ; 接收数据
        LJMP END
ENDII: SETB SM2
ENDI: SETB ES
END: POP ACC
     POP PSW
     RETI
```

SE_DATA: CLR TRDY ; 发送数据块子程序

MOV R6, #00H

MOV R0, #30H

MOV R7, #10H

LOOP2: MOV A, @R0

MOV SBUF, A

JNB TI, \$

CLR TI

INC R0

ADD A, R6

MOV R6, A

DJNZ R7, LOOP2 ; 数据块发送完毕?

MOV A, R6

MOV SBUF, A

JNB TI, \$; 发送校验和

CLR TI

JNB RI, \$

CLR RI

MOV A, SBUF

XRL A, #00H ; 判发送是否正确

JZ SEND_OK

SJMP SE_DATA ; 发送错误, 重发



```
SEND_OK: SETB SM2      ; 发送正确，继续监听  
         SETB ES  
         RET  
RE_DATA: CLR RRDY      ; 接收数据块子程序  
         MOV R6, #00H  
         MOV R0, #30H  
         MOV R7, #10H
```



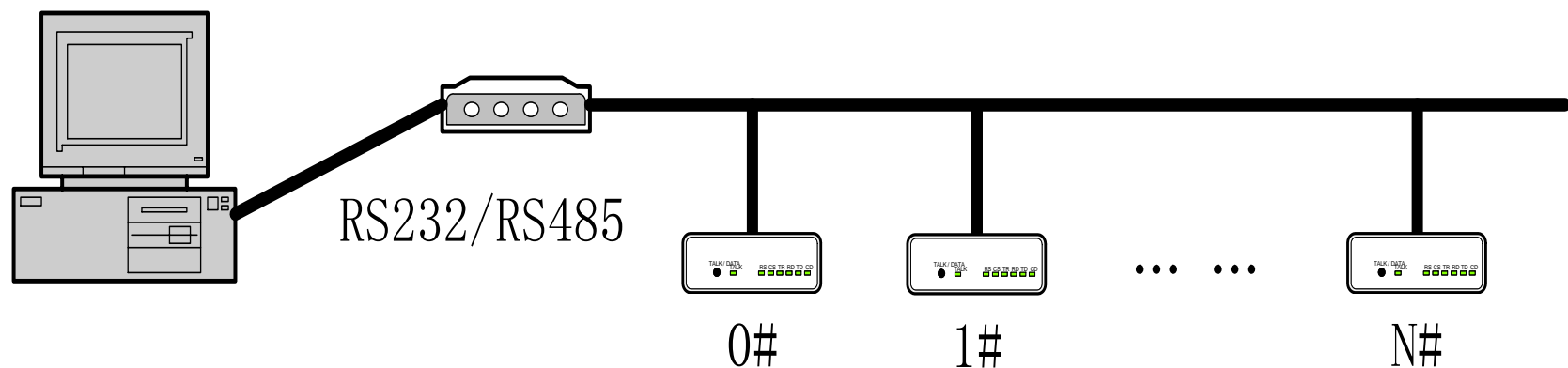
```
LOOP3: JNB RI, $  
       CLR RI  
       MOV A, SBUF  
       MOV @R0, A  
       INC R0  
       ADD A, R6  
       MOV R6, A  
       DJNZ R7, LOOP3    ; 接收数据块完毕?  
       JNB RI, $        ; 接收校验和  
       CLR RI  
       MOV A, SBUF  
       XRL A, R6        ; 判断校验和是否正确  
       JZ RECE_OK  
       MOV SBUF, #0FFH ; 校验和错误, 发FFH  
       JNB TI, $  
       CLR TI  
       LJMP RE_DATA    ; 重新接收
```



```
RECE_OK: MOV A, #00H           ; 校验和正确, 发00H
          MOV SBUF, A
          JNB TI, $
          CLR TI
          SETB SM2              ; 继续监听
          SETB ES
          RET
```

6.3.2 单片机与PC机的通信

在工控系统（尤其是多点现场工控系统）设计实践中，单片机与PC机组合构成分布式控制系统是一个重要的发展方向。分布式系统主从管理，层层控制。主控计算机监督管理各子系统分机的运行状况。子系统与子系统可以平等信息交换，也可以有主从关系。分布式系统最明显的特点是可靠性高，某个子系统的故障不会影响其它子系统的正常工作。



1台PC机即可以与1个80C31单片机应用系统通信，也可以与多个80C31单片机应用系统通信；可以近距离也可以远距离。单片机与PC机通信时，其硬件接口技术主要是电平转换、控制接口设计和远近通信接口的不同处理技术。

在DOS操作环境下，要实现单片机与微机的通信，只要直接对微机接口的通信芯片8250进行口地址操作即可。在WINDOWS的环境下，由于系统硬件的无关性，不再允许用户直接操作串口地址。如果用户要进行串行通信，可以调用WINDOWS的API应用程序接口函数，但其使用较为复杂，而使用VB通信控件（Mscomm）却可以很容易的解决这一问题。

VB是WINDOWS图形工作环境与Basic语言编程简便性的完美结合。它简明易用，实用性强。VB提供一个名为MSCOMM32.OCX的通信控件，它具备基本的串行通信能力：即通过串行口发送和接收数据，为应用程序提供串行通信功能。



MSComm控件有许多属性，主要的几个如下：

CommPort: 设置并返回通信端口号；

Settings: 以字符串的方式设置并返回波特率、奇偶校验、数据位、停止位；

PortOpen: 设置并返回端口的状态，也可以打开和关闭端口；

Input: 从接收缓冲区返回字符和删除字符；

Output: 向传输缓冲区写一个字符。

单片机程序清单如下：

```
ORG 3000H
MAIN: MOV TMOD, #20H ; 在11.0592MHz下, 串行口波特率
      MOV TH1, #0FDH ; 9600bps, 方式3
      MOV TL1, #0FDH
      MOV PCON, #00H
      SETB TR1
      MOV SCON, #0D8H
LOOP: JBC RI, RECEIVE ; 接收到数据后立即发出去
      SJMP LOOP
RECEIVE: MOV A, SBUF
         MOV SBUF, A
SEND: JBC TI, SENDEND
      SJMP SEND
SENDEND: SJMP LOOP
```

PC机程序清单：（VB语言）

```
Sub Form_Load ()
    MSComm1.CommPort=2
    MSComm1.PortOpen=TURE
    MSComm1.Settings="9600, N, 8, 1"
End Sub
Sub command1_Click ()
    Instring as string
    MSComm1.InBufferCount=0
    MSComm1.Output="A"
    Do
        Dummy=DoEvents ()
    Loop Until (MSComm1.InBufferCount>2)
    Instring=MSComm1.Input
End Sub
Sub command2_Click ()
    MSComm1.PortOpen=FALSE
    UnLoad Me
End Sub
```

思考题及习题

- 1、**80C51**单片机串行口有几种工作方式？如何选择？简述其特点？
- 2、串行通信的接口标准由那几种？
- 3、在串行通信中通信速率与传输距离之间的关系如何？
- 4、在利用**RS-422/RS-485**通信的过程如果通信距离（波特率固定）过长，应如何处理？
- 5、利用单片机串行口扩展**24**个发光二极管和**8**个按键，要求画出电路图并编写程序使**24**个发光二极管按照不同的顺序发光（发光的时间间隔为**1S**）。
- 6、编制图**6-30**的中断方式的数据接收程序。
- 7、简述**80C51**单片机多机通信的特点。
- 8、在微机与单片机构成的测控网络中，要提高通信的可靠性要注意哪些问题？