



第八章 80C51单片机串行通信

帅千钧

Email:sqj@cuc.edu.cn

办公室：主楼813



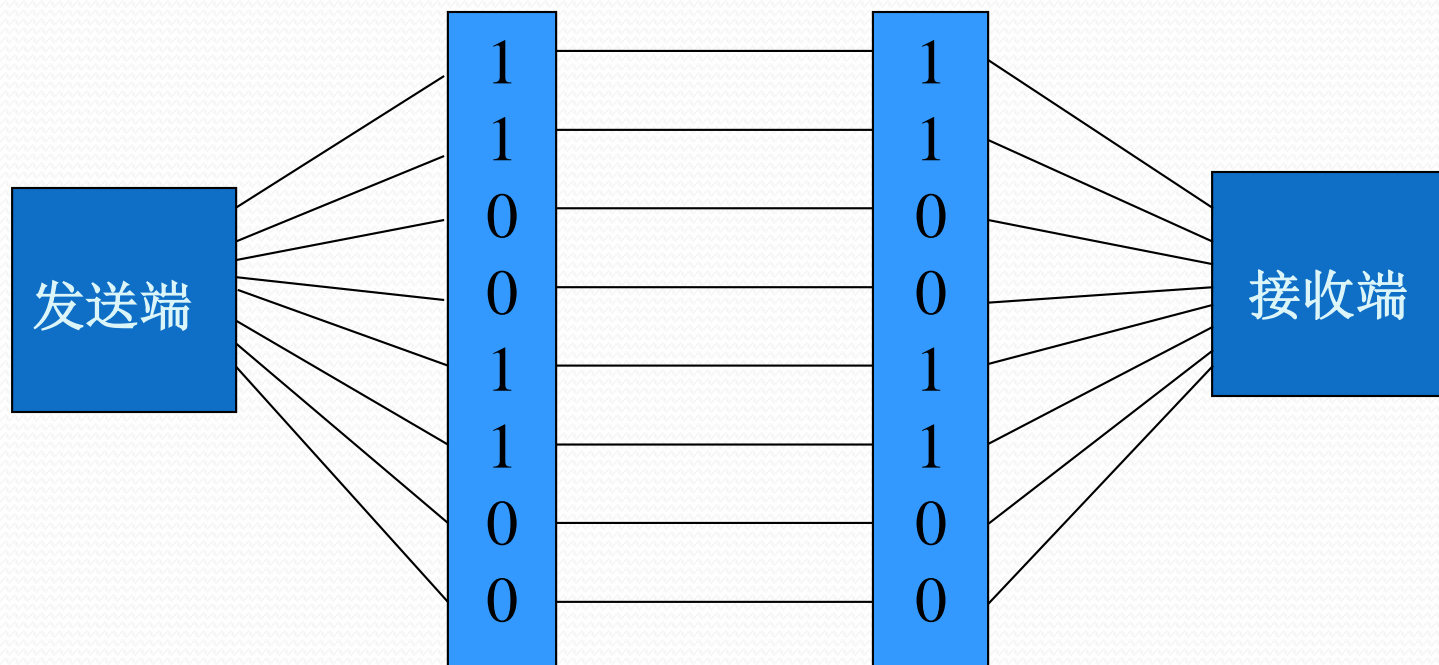
本章内容

- 串行通信概述
- MCS-51单片机的全双工串口
- 单片机双机通信

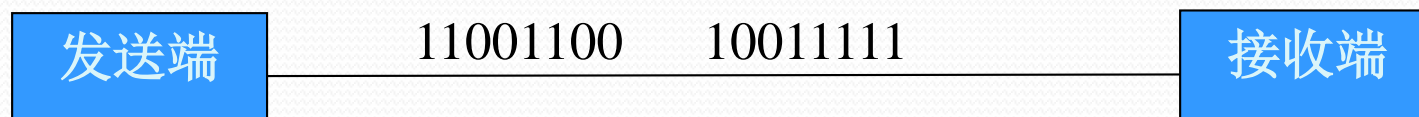


8.1 串行通信概述

并行通信



串行通信





8.1 串行通信和并行通信的比较

■ 串行通信

就是将构成字符的每个二进制数据位，按一定的顺序逐位进行传送的通信方式

- ◆ 优点：节省数据传输线，硬件简单、传送距离较远
- ◆ 缺点：传送速度慢
- ◆ 适合距离远，速度要求不高的场合

■ 并行通信

各位数据线上的数据同时传送的通信方式

- ◆ 优点：传输速度快
- ◆ 缺点：传输线多
- ◆ 只适合近距离传输



8.1 串行通信分类

■ 按字符的同步方式分：

- ◆ 异步通信
- ◆ 同步通信

■ 按通信的数据传输方向分：

- ◆ 单工
- ◆ 半双工
- ◆ 全双工



8.1.1 串行通信的同步

- 同步问题是串行通讯的一个重要问题。

接收端需要知道什么地方是一个字符的开始位，从而能从一串位流中将每个字符按规定编码格式分离出来。

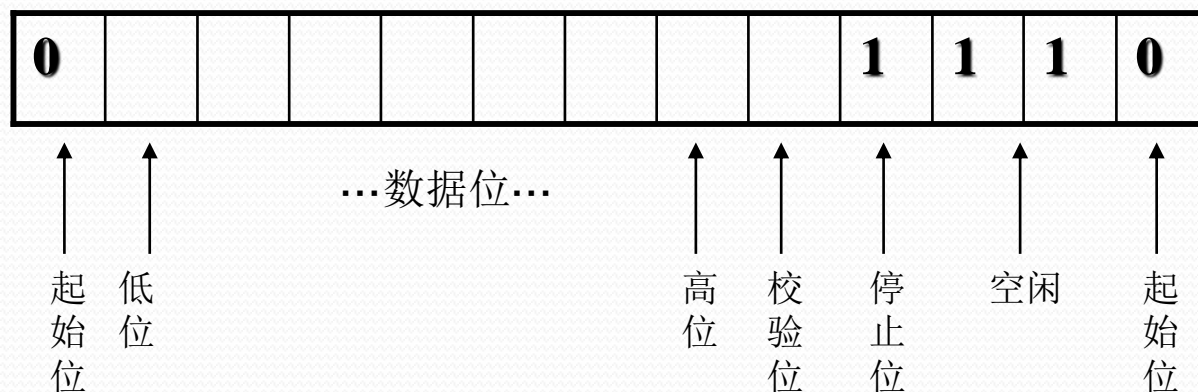
- ◆ 异步方式，每帧信息内部按规定的波特率发送接收，其时间间隔是固定，但是帧与帧之间的间隔是不固定的，可以改变。
- ◆ 同步方式，每帧信息以同步字符开始，大量连续、无间隔的字符形成一个数据块，可以连续进行收发，故速度较快。



8.1.1 异步通信

■ 异步通信

异步通信以帧格式传送，每一帧信息由起始位、数据位、奇偶校验位和停止位组成。





■ 异步通信

◆ 起始位

通信线上，没有数据传送时处于逻辑“1”状态；当发送设备要发送一个字符数据时，首先发送一个逻辑“0”，当接收设备检测到这个低电平时，就开始准备接收数据。

◆ 数据位

起始位之后，紧接着接收的就是数据位；数据位的个数可以是5、6、7或8位。

◆ 奇偶校验位

数据位之后有一位的奇偶校验位，用于检验差错；该位也可以没有。

◆ 停止位

停止位可以是1位、1位半或2位的高电平。



■ 异步通信的特点：

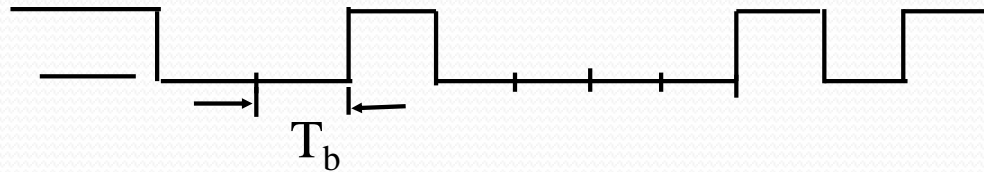
- ◆ 每帧信息内部按规定的波特率发送接收，其时间间隔是固定，即每个字符中的各位按固定时钟来传输；但是帧与帧之间的间隔是不固定的，可以改变。
- ◆ 可以认为异步数据传输在字符内部是同步的，而在字符间是异步的。
- ◆ 起始位和停止位起着重要的作用，它们使得接收器将其局部时钟与每个新字符同步起来。
- ◆ 帧格式允许有差别，可由用户自己定义，比如：数据位可以是5位、6位、7位或8位，可以没有奇偶校验位等。



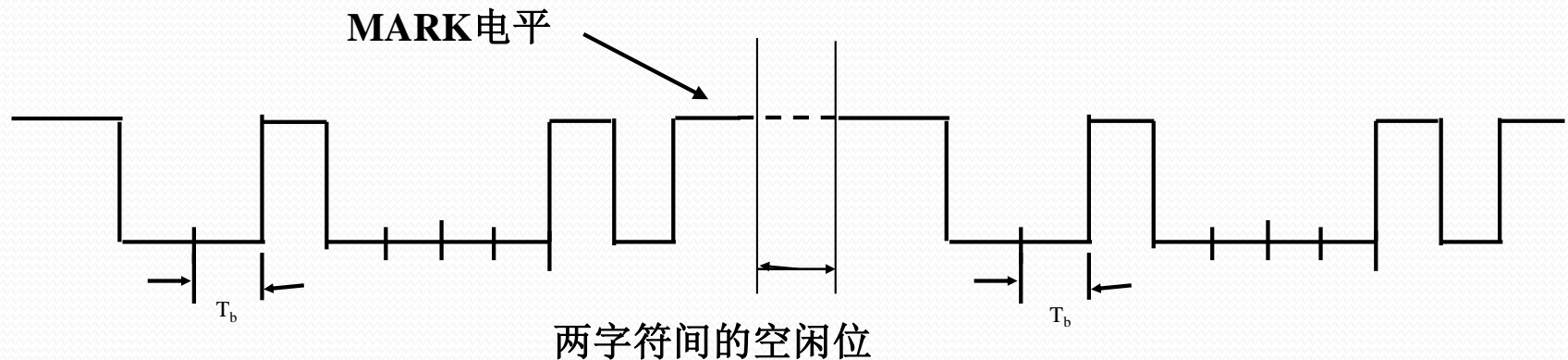
例：要求7位数据，1位停止位，用偶校验，通信速率为300波特
($T_b = 1/300\text{bps}$)
画出字符 B (ASCII码为42H 01000010B) 的字符格式。

MARK

SPACE



- 相邻两个字符之间的时间间隔即空闲时间可为任意长。





■ 异步通信双方在通信之前必须事先约定:

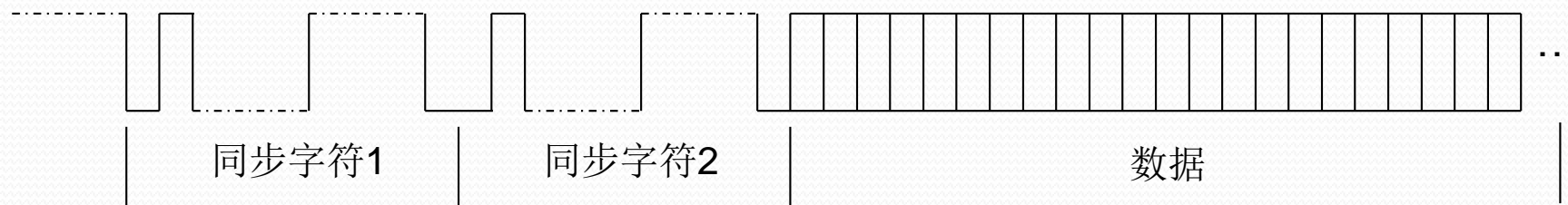
1. **字符格式**,即一个字符包含多少位数据位、停止位以及采用何种校验形式;
 2. **比特率** (bit/s kbit/s),比特率即数据传送速率,也就是比特间隔时间 T_b 的倒数,它也是衡量传输通道频宽的指标。
- 常用的比特率有 50、110、300、600、1200、2400、4800、9600(bps)。近几年通信发展,传送速率增加很快,新增加的速率有: 14.4、19.2、38.4、57.6、115.2(Kbps)等。



8.1.1 同步通信

■ 同步通信

- ◆ 采用**数据块**成帧进行传送而不需要起始位和停止位；
 - ◆ 需要时钟和同步字符使得每个数据块在收/发端同步；
- 如下：





■ 数据帧以同步字符SYNC开始

- ◆ 同步字符的作用是通知接收器 “一帧消息到达”。
- ◆ 同步字符的格式和同步字符的个数可协商确定（协议）。
- ◆ 同步字符后的第一个数据是消息头或称报头，它包含有助于接收器如何处理以后收到的消息的控制信息(如一帧内的字符数据计数)
- ◆ 数据块就是字符间不允许有间隙的连续的数据字符组合。
- ◆ 最后是校验字符，常用CRC 循环冗余校验码。



同步串行通信成帧格式



■ 同步通信特点

- 同步通信不需要起始位和停止位，节省了冗余，增加了带宽利用率。
- 以同步字符作为传送的开始，之后便连续发送数据，直到数据发送完毕。
- 字符数据之间不允许有空隙，当线路空闲时则发送同步字符。
- 同步数据按数据块定位，而不是按字符定位。
- 可在较长的线路上以较高的速率传输。



8.1.1 异步通信与同步通信的比较

■ 异步通信：

1. 以**字符帧**为单位传送，当接收方收到起始位后，只要在一个字符的传输时间内能和发送器保持同步，就能完成正确的接收。
 2. 传送时对每个被传送字符有一定的**格式**要求 (如：要求每个字符传送都有起始位和停止位)，因此控制信息至少占总信息的20%。这是额外的“开支”，因此传送效率较差。
 3. 对时钟同步的要求不太严格，只要在一个字符的传输的短暂时间内能收发送保持同步，可以容忍收发两端的时钟略有误差，两个字符之间的停止间隔将为这种误差提供一种缓冲。
- 对时钟同步的要求不太严格，这是异步通信的突出优点。



■ 同步传送:

1. 以**数据块帧**为单位进行传送，每帧中有多个字符。
2. 数据传输效率高。
3. **同步字符**作为传送的开始
4. 硬件电路复杂：
 1. 保持数据字符内各位以预定的固定时钟频率传送。
 2. 整个数据块的每个字符也都按预定的固定时钟频率传送。
 3. 数据块间不允许有间隙，若有间隙必须用专用字符填充。

收、发双方时钟严格同步是同步通信的基本要求。

➤ 因此，在技术上，必须用硬件从接收到的数据中分离出同步时钟来实现。
5. 固有检错能力比异步通信强得多。



- “异步” ——

主要体现在通信过程中字符与字符之间没有严格的定时要求，通信双方是通过收/发双方事先约定的收/发波特率和字符格式、且在收/发时钟信号作用下，实现被传送字符的位同步的。

- “同步” ——

通信过程中，要求字符与字符之间以及字符内部的位与位之间都必须同步。为达到这一目的，要求收、发双方必须使用同一时钟对被传输信息定位。



8.1.2 单工、半双工、全双工通信

■ 单工、半双工、全双工通信

串行通信中，数据通常是在二个站（点对点）之间进行传送，将数据从一个地方传送到另一个地方，须使用通信线路，数据在通信线路的两端，即两工作站之间传送，按其通信方式，可将数据传输线路分成3种：

- ◆ 单工(Single Duplex)
- ◆ 半双工(Half Duplex)
- ◆ 全双工(Full Duplex)

1. 单工(Single Duplex)



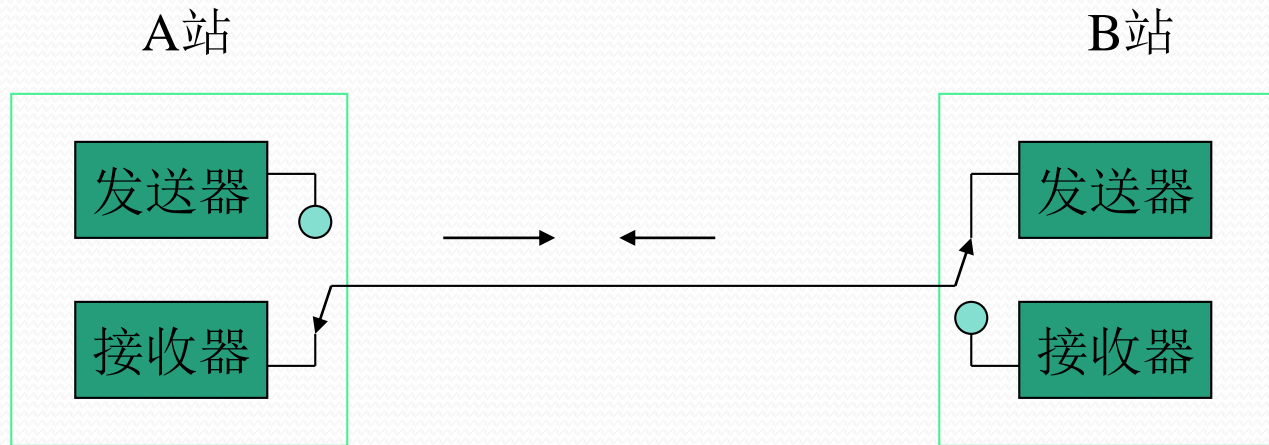
单工方式示意图

特点： 信息只能沿一个方向传送，使用一根传输线。

应用： 电视发射台

2. 半双工 (Half Duplex)

使用同一根传输线，既可发送数据又可接收数据，但不能同时。



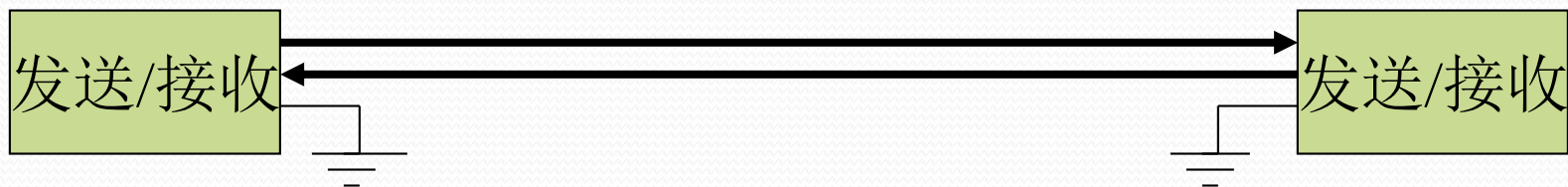
半双工方式示意图

特点:

- ① 通信双方各有一个收/发切换电子开关，双方均可进行数据的接收和发送。
- ② 只需要一根传输线。
- ③ 因有切换，会产生时间延迟。

应用: 对讲机、单向传送设备，发送器→接收器

3. 全双工 (Full Duplex)



全双工方式示意图

特点：①每一端都有发送器和接收器

②有二条传送线

应用：电话、交互式应用，远程监测控制



8.1.3 串行通信相关概念

- 传输速率
- 发送/接收时钟



■ 传输速率

每秒钟传送的二进制数的位数，单位为：**bps**

1bps means 1bit/s

- ◆ 在二进制的情况下，传输速率通常等于**波特率**
- ◆ 接收设备和发送设备必须保持相同的数据传输速率，才能保证数据正确的传送。

■ 发送/接收时钟

- ◆ 无论是发送还是接收，都必须通过时钟信号对传送的数据进行定位，发送/接收时钟就是用来控制通信设备发送/接收数据的速率的。
- ◆ 发送/接收时钟频率与波特率的关系：

时钟频率 = $n \times$ 波特率 (n 为1、16、64等)



8.2 MCS-51单片机的全双工串口

- 51单片机的的串行通信通过引脚**RXD(P3.0)**和**TXD(P3.1)**与外界通信。
- **RXD(P3.0)**是串行输入线，**TXD(P3.1)**是串行输出线。这是一个**全双工**的串行口。



8.2.1 51单片机的串行通信控制寄存器

■ SCON(98H)

用于定义串口的操作方式和进行一些功能控制。

■ SBUF(99H)

数据缓冲寄存器，在物理上，它对应着两个寄存器，一个发送寄存器，一个接收寄存器。

■ PCON(87H)

主要是为了实现单片机的电源控制而设置的，可实现对串行通信波特率的控制。



SCON(98H)

9FH	9EH	9DH	9CH	9BH	9AH	99H	98H
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

┌───┐
↑
串口
工作方式
选择位

8051串行口工作方式

SM0	SM1	方式	说明	波特率
0	0	0	移位寄存器方式	$F_{osc}/12$
0	1	1	10位UART	可变
1	0	2	11位UART	$F_{osc}/64$ 或 $F_{osc}/32$
1	1	3	11位UART	可变



SCON(98H)

- SM2: 多机通信控制位
 - ◆ 方式0, SM2一定要为0
 - ◆ 方式1, SM2=1, 则收到有效停止位, RI才置1。
 - ◆ 方式2、3, SM2=1, 且第九位RB8=1, 则将接收到的前8位数据送入SBUF, 并将RI置1。
- REN: 接收允许控制位, 置1允许, 清0禁止
- TB8: 方式2、3中发送数据的第9位
- RB8: 方式2、3种接收数据的第9位(方式1中, RB8接收到的是停止位; 方式0中, 不使用RB8位)
- TI: 发送中断标志位, 须软件清0
- RI: 接收中断标志位, 须软件清0



中断相关寄存器

TCON
88H

D7	D6	D5	D4	D3	D2	D1	D0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

位地址

8FH 8EH 8DH 8CH 8BH 8AH 89H 88H

SCON
98H

D7	D6	D5	D4	D3	D2	D1	D0
						TI	RI

位地址

99H 98H

IE
A8H

D7	D6	D5	D4	D3	D2	D1	D0
EA			ES	ET1	EX1	ET0	EX0

位地址

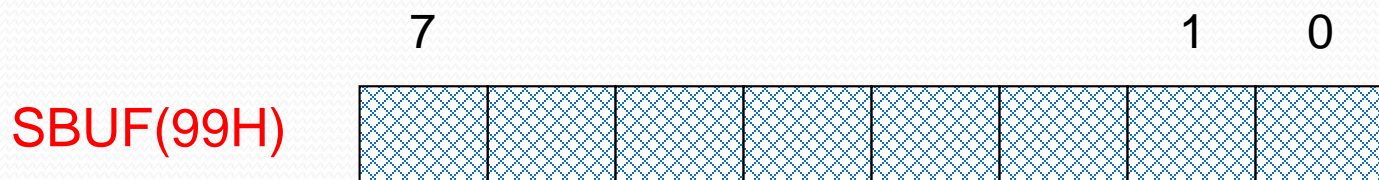
AFH ACH ABH AAH A9H A8H

IP
B8H

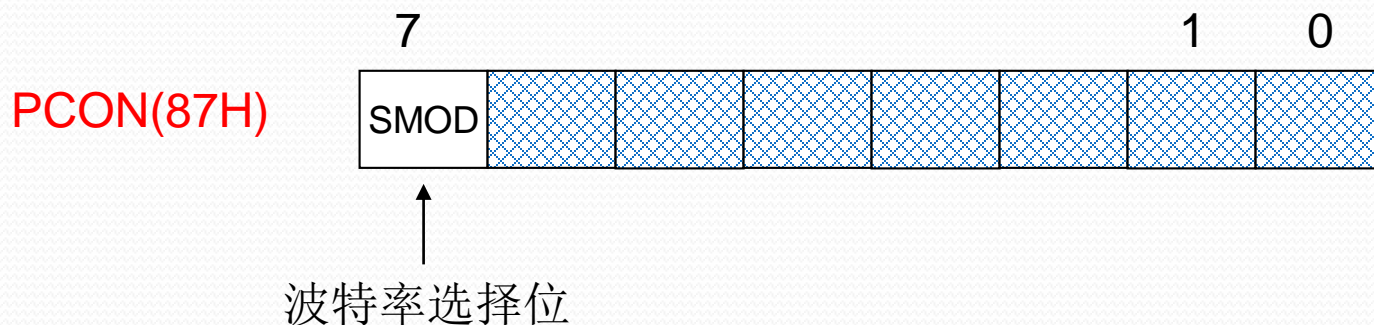
D7	D6	D5	D4	D3	D2	D1	D0
			PS	PT1	PX1	PT0	PX0

位地址

BCH BBH BAH B9H B8H



- CPU写SBUF就是修改发送寄存器。
- CPU读SBUF就是读接收寄存器。



- SMOD=1，在方式1、2、3时，波特率增加一倍。
- 复位时，SMOD=0。
- 不具有位寻址功能，对SMOD的置1和清0操作为：

MOV PCON, #80H	； 置1
MOV 87H, #80H	； 置1
MOV PCON, #00H	； 清0
MOV 87H, #00H	； 清0



8.2.2 51单片机的串行口工作方式

- 51单片机有四种串行口工作方式
 - ◆ 方式0，主要用于并行扩展
 - ◆ 串行通信一般使用方式1、2、3。

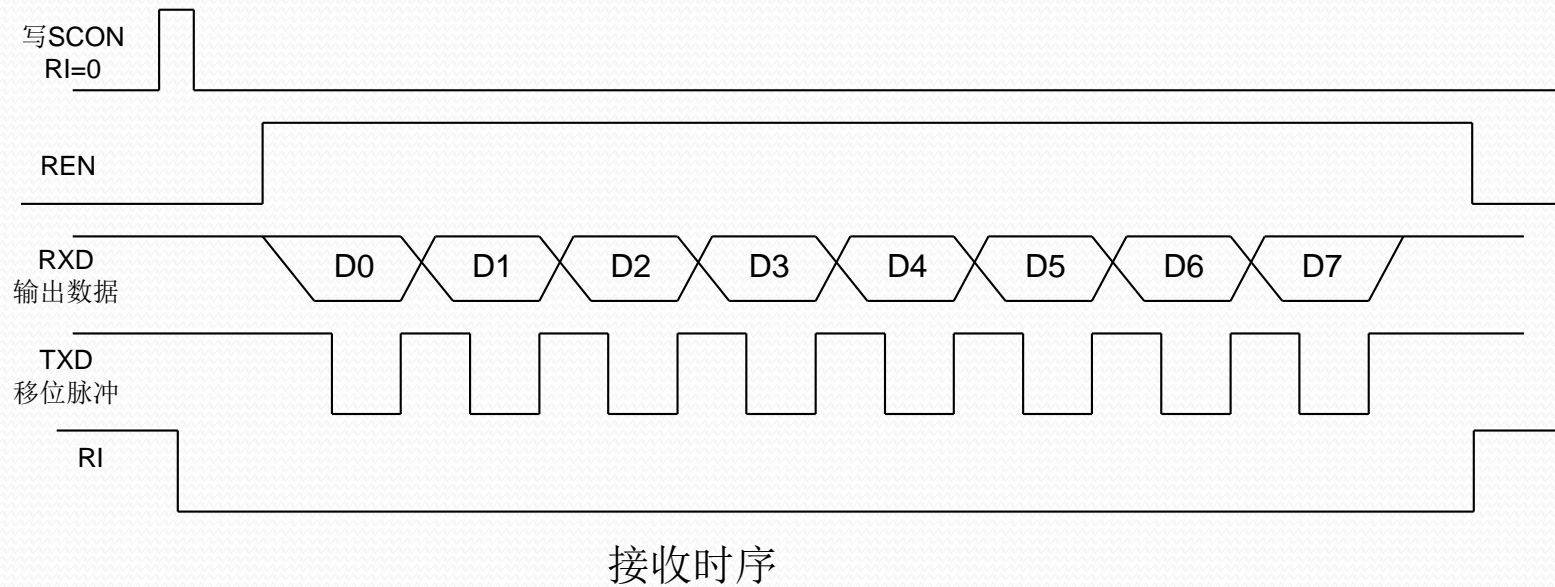
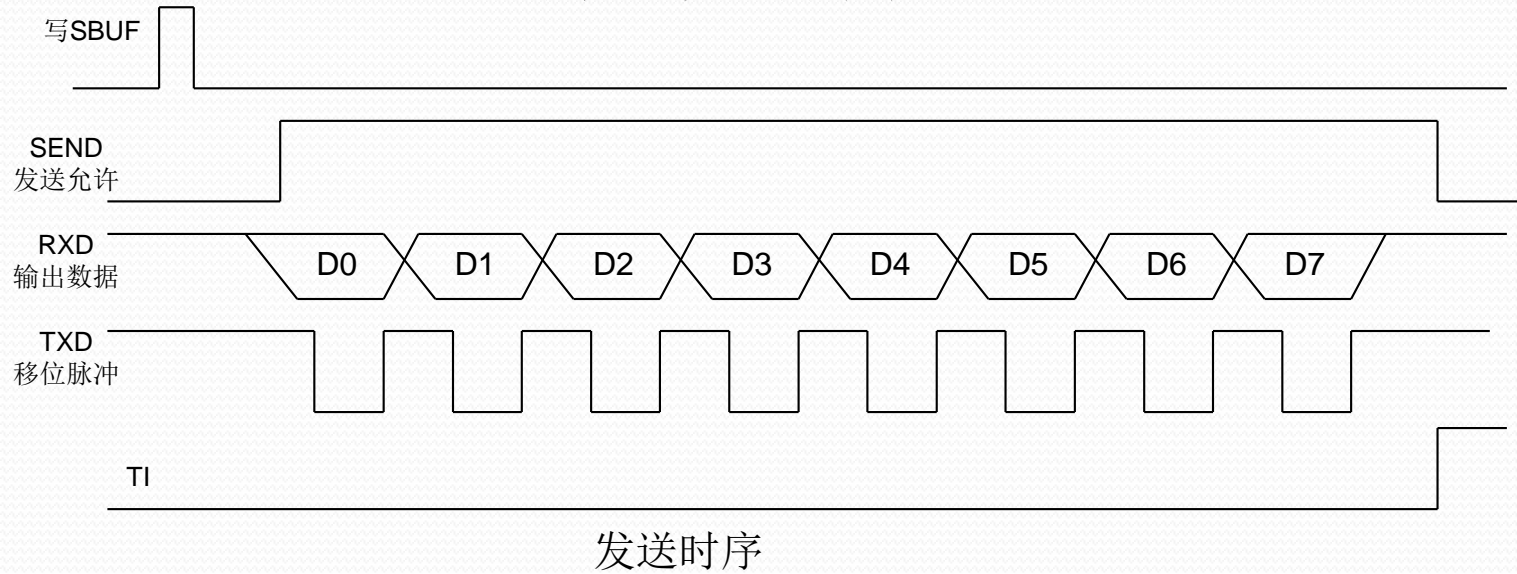


1、方式0：移位寄存器方式

- 方式0，串口作为同步移位寄存器，数据传输速率固定为振荡频率的 $1/12(F_{osc}/12)$ ，即 一个机器周期发送或接收1位数据。
- 串行数据由RXD输入/输出，TXD输出同步移位脉冲。
- 数据的发送/接收以8位为一帧，低位在前，无起始位、奇偶校验位和停止位。



方式0时序图





■ 发送

当一个数据写入串行口发送寄存器SBUF时，串行口即将8位数据以 $F_{osc}/12$ 的波特率从RXD端输出(从低到高)，TXD输出端输出同步脉冲，发送完硬件将TI置1。

■ 接收

在 $RI=0$ 和 $REN=1$ 的条件下，就会启动一次接收过程，此时RXD为串行输入端，TXD为同步脉冲输出端，波特率为 $f_{osc} / 12$ 。收完8位数据后，控制信号复恢，中断标志RI置位，通知CPU取走数据。

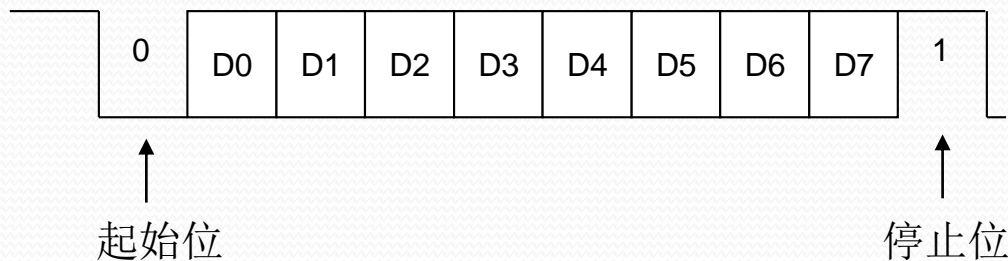


2、方式1：10位UART

- 方式1，是10位异步通信接口；传输速率可变，由定时器T1的溢出速率决定。

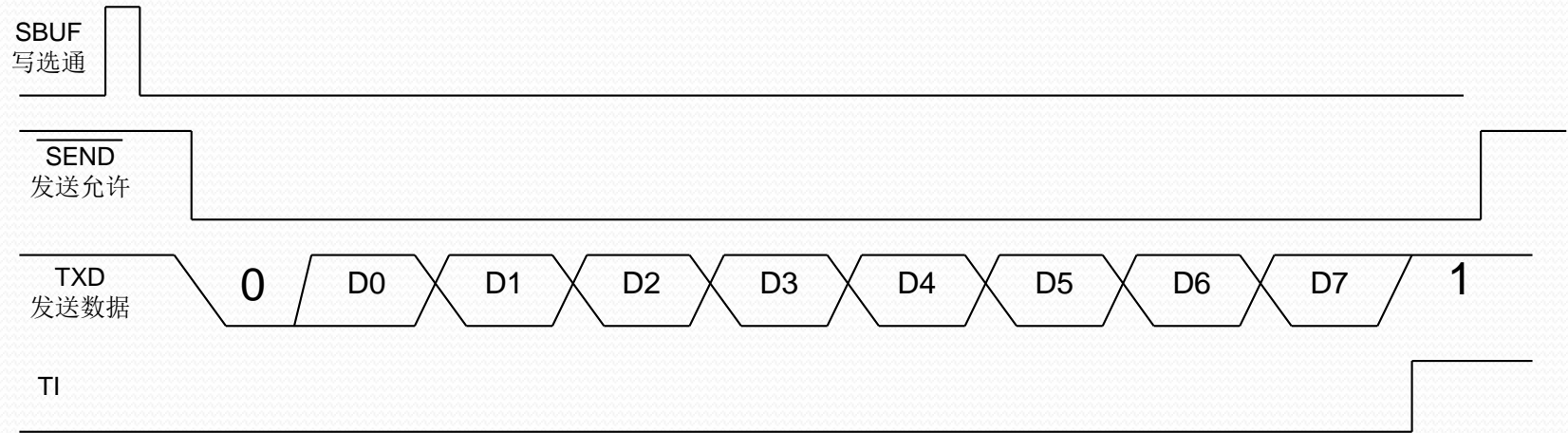
$$\text{波特率} = \frac{2^{\text{SMOD}}}{32} \times \text{T1溢出率}$$

- TXD为发送端，RXD为接收端
- 一帧数据包括10位：一位起始位(0)、8位数据位、一位停止位(1)。

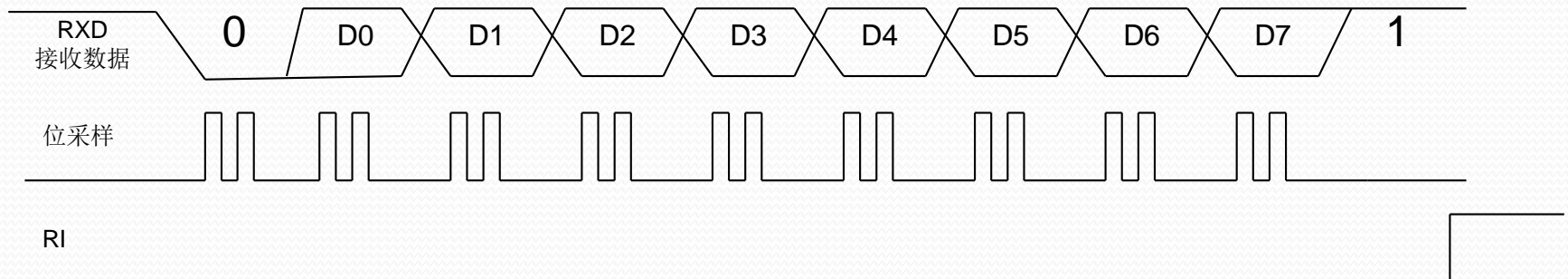




方式1时序图



发送时序



接收时序



■ 发送

- ◆ 执行一条**写SBUF**的指令，即启动串行输出。数据从TXD输出，发送一帧信息为10位，其中起始位0和停止位1是串行口自行加进去的。
- ◆ 发送完一帧信息，则硬件将TI置1。

■ 接收

- ◆ 用**指令使REN置1**，便开始串行输入工作。信息从RXD输入，这时串行口不停地检测RXD输入端，检测到由1到0的跳变，即起始位有效后，便开始接收一帧信息。
- ◆ 接收到停止位时，若数据有效条件满足，则硬件将8位数据装入接收寄存器SBUF、停止位装入RB8并将RI置1。

注意 数据接收有效的条件：方式1，通常将SM2置0

- ◆ $SM2=0$ 时： $RI=0$
- ◆ $SM2=1$ 时： $RI=0$ 且收到的停止位为1。



3、方式2：11位UART

- 方式2，11位异步通信接口，波特率固定

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

即：波特率为 $F_{\text{osc}}/64$ 或 $F_{\text{osc}}/32$

- TXD为发送端，RXD为接收端。
- 一帧数据包括11位：1位起始位(0)、8位数据位、1位可编程位(即第9数据位)、1位停止位(1)。



■ 发送

当向SBUF写入一个数据，便启动串行口输出。数据由TXD送出，一帧信息为11 位，其中1位起始位0，8位数据，第9位TB8，可由软件设置，最后是1位停止位1。

■ 接收

在REN=1时启动接收，串行口首先检测起始位是否有效，若起始位有效便接收一帧11 位信息，当接收到第9位数据时，若满足条件之一：

- ◆ SM2=0时：RI=0
- ◆ SM2=1时：RI=0且第9位数为1

8位数据便被送入接收寄存器SBUF，并由硬件将RI置1。



4、方式3：11位UART

- 方式3，11位异步通信接口。
- 波特率可变，由定时器T1的溢出速率决定

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

- TXD为发送端，RXD为接收端
- 一帧数据包括11位：1位起始位(0)、8位数据位、1位可编程位(即第9数据位)、1位停止位(1)



波特率的计算

■ 方式0:

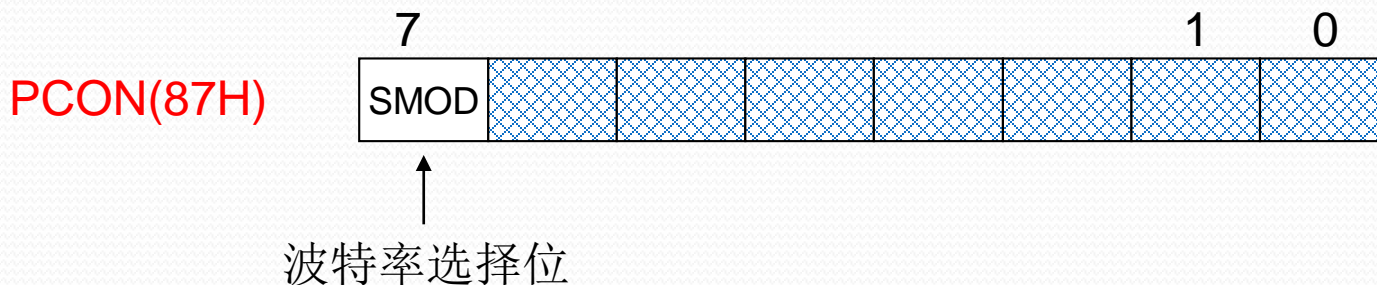
$$\text{波特率} = \frac{f_{\text{OSC}}}{12}$$

■ 方式1和3:

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

■ 方式2:

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





- 通常采用定时器T1工作方式2作为波特率发生器

$$\text{定时时间} = (2^8 - X) \times \frac{12}{f_{\text{osc}}}$$

$$T1\text{溢出率} = \frac{1}{\text{定时时间}} = \frac{f_{\text{osc}}}{12 \times (2^8 - X)}$$

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



- 若串口采用工作方式1，系统时钟 f_{osc} 为11.059MHz，设SMOD=1，波特率为2400bps，求定时器在定时方式2时的初值。

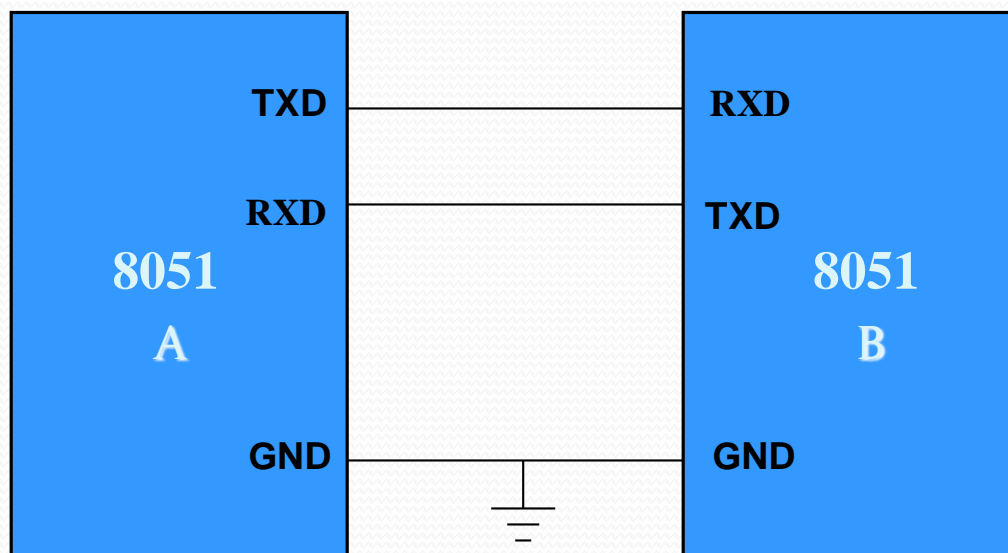
$$X = 2^8 - \frac{2^{SMOD} \times f_{osc}}{384 \times \text{波特率}}$$

$$X = E9H$$

8.3 单片机双机通信

■ 单片机双机通信的连接

- ◆ 当两个单片机系统相距很近(<1.5 米), 可以采用TTL电平直接相连。
- ◆ 当距离较远时, 可以采用RS电平进行相连。
- ◆ 当单片机应用系统和PC机进行双机通信时, 必须进行RS电平转换。





举例：A、B两个单片机应用系统直接相连进行双机通信。

■ 双方在软件上有如下一系列约定(通信协议):

- ◆ 时钟频率为11.059MHz，采用2400波特率传送，定时器T1工作方式2，SMOD=1。
- ◆ A机发送数据，B机接收数据。
- ◆ A发送呼叫信号用AA，B同意接收回答信号用BB，A机只有收到B的应答信号BB后才可把存放的数据发送给B机，否则继续向B机呼叫，直到B机同意接收。
- ◆ 发送数据格式为:

数据1	数据2	...	校验和
-----	-----	-----	-----

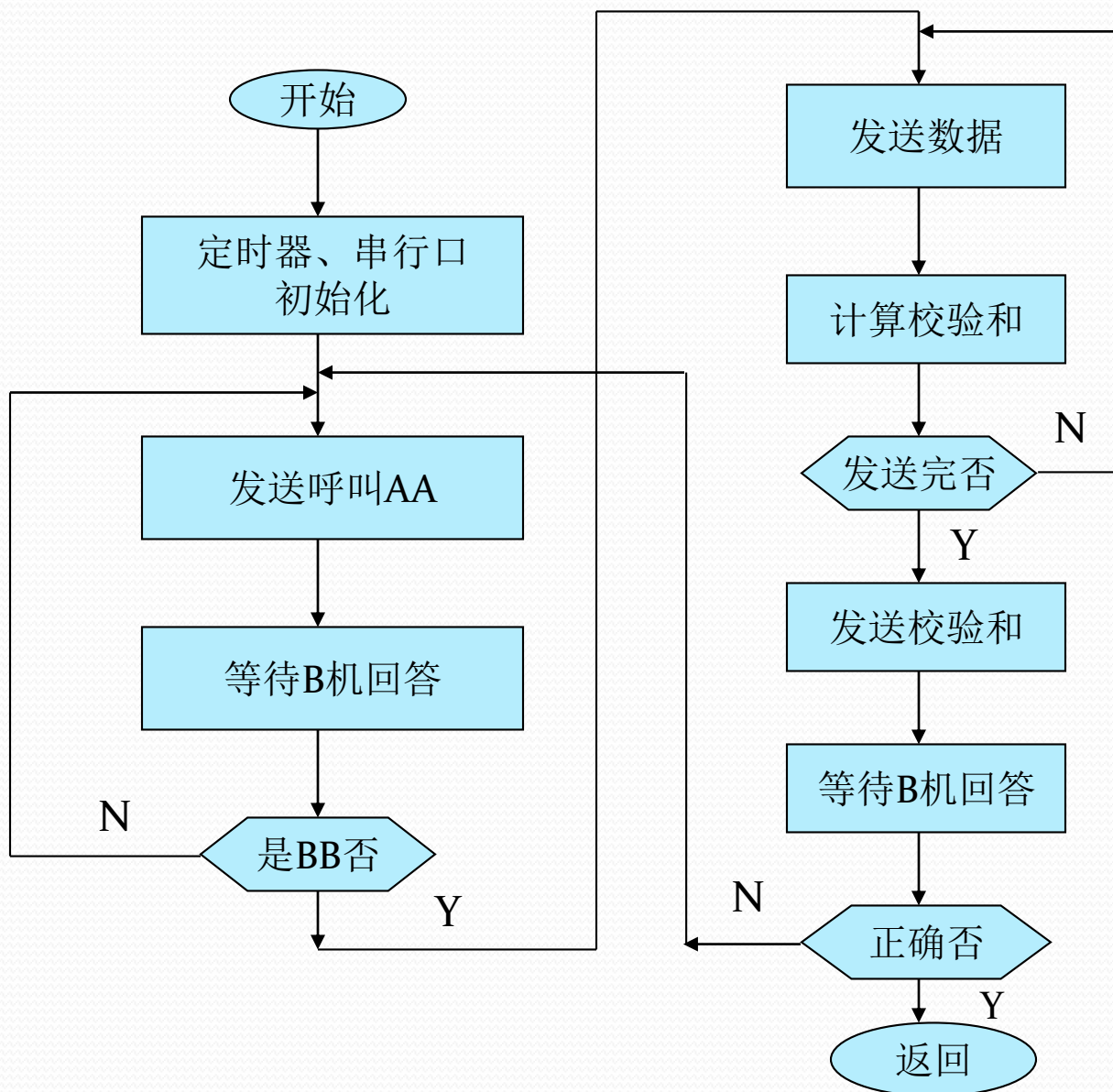
- ◆ 累加校验和是所有发送数据字节内容的算术累加和，存放在R6。



■ 约定(通信协议)-续:

- ◆ 发送的数据块长度为20个字节，存放在R7中，数据区起始地址存在R0中，数据发完接着发校验和。
- ◆ B机的校验和与A机发来的进行比较，两者相等，向A发送00，不等，发送FF并要求继续呼叫重发。

A机查询方式发送流程图



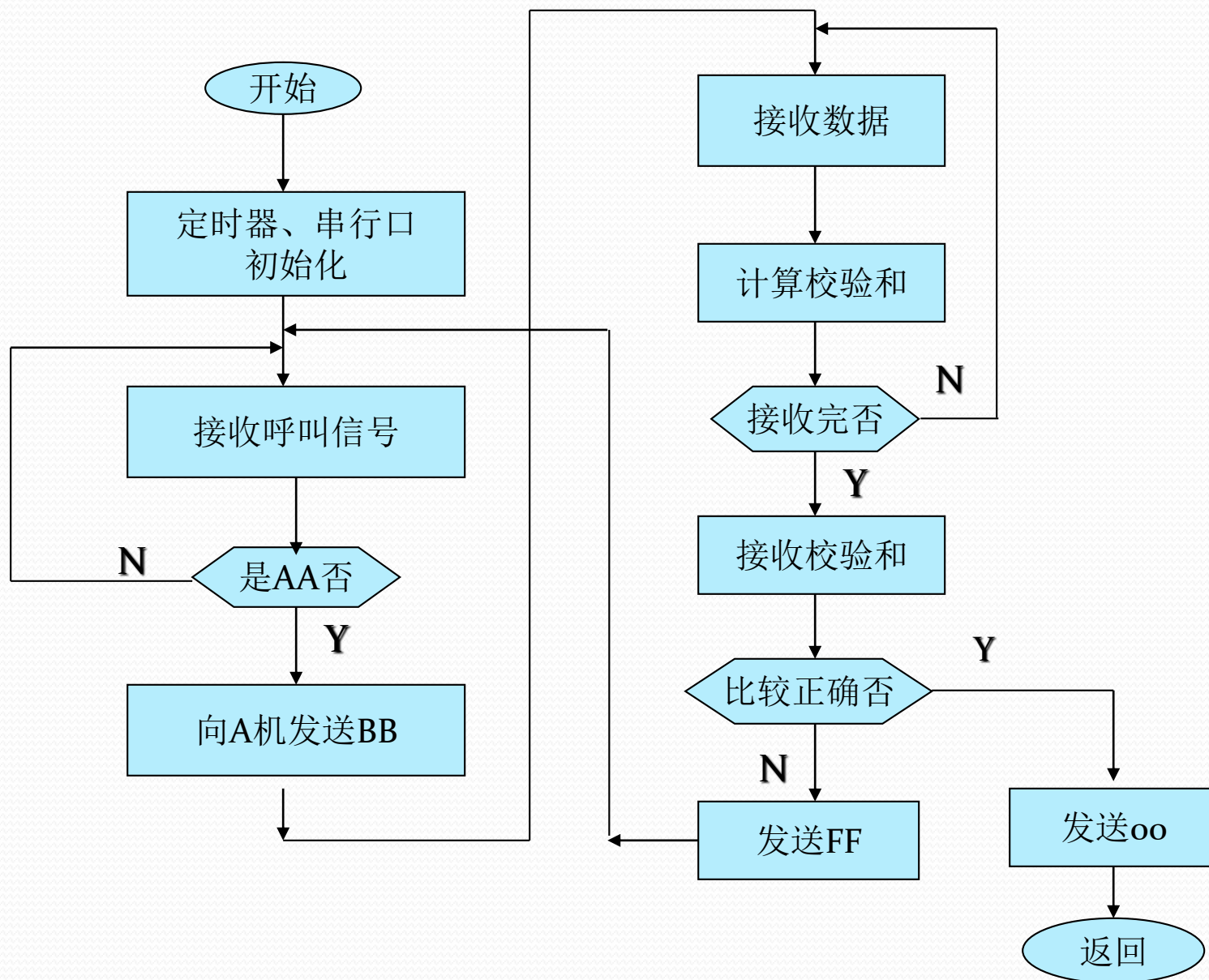


A机查询方式发送程序

```
INIT: MOV TMOD, #20H
      MOV TH1, #0E8H
      MOV TL1, #0E8H
      MOV SCON, #50H
      MOV PCON, #80H
      SETB TR1
ARAM: MOV R0, #30H
      MOV R7, #20
      MOV R6, #00H
AT1:  MOV SBUF, #0AAH
AW1:  JBC TI, AR1
      SJMP AW1
AR1:  JBC RI, AR2
      SJMP AR1
AR2:  MOV A, SBUF
      XRL A, #0BBH
```

```
      JNZ AT1
ATD:  MOV SBUF, @R0
      MOV A, R6
      ADD A, @R0
      MOV R6, A
      INC R0
ATDW: JBC TI, ATD1
      SJMP ATDW
ATD1: DJNZ R7, ATD
      MOV SBUF, R6
ATS:  JBC TI, ARS
      SJMP ATS
ARS:  JBC RI, ARS1
      SJMP ARS
ARS1: MOV A, SBUF
      JNZ ARAM
      RET
```


B机查询方式接收流程图





B机查询接收程序

```
INIT:  MOV TMOD,#20H
        MOV TH1,#0E8H
        MOV TL1,#0E8H
        MOV SCON,#50H
        MOV PCON,#80H
        SETB TR1

BRAM:  MOV R0,#30H
        MOV R7,#20
        MOV R6,#00H

BR1:   JBC RI,BR2
        SJMP BR1

BR2:   MOV A,SBUF
        XRL A,#AAH
        JNZ BR1

BT1:   MOV SBUF,#0BBH

BW1:   JBC TI,BRD
        SJMP BW1
```

```
BRD:   JBC RI,BRD1
        SJMP BRD

BRD1:  MOV A,SBUF
        MOV @R0,A
        INC R0
        ADD A,R6
        MOV R6,A
        DJNZ R7,BRD

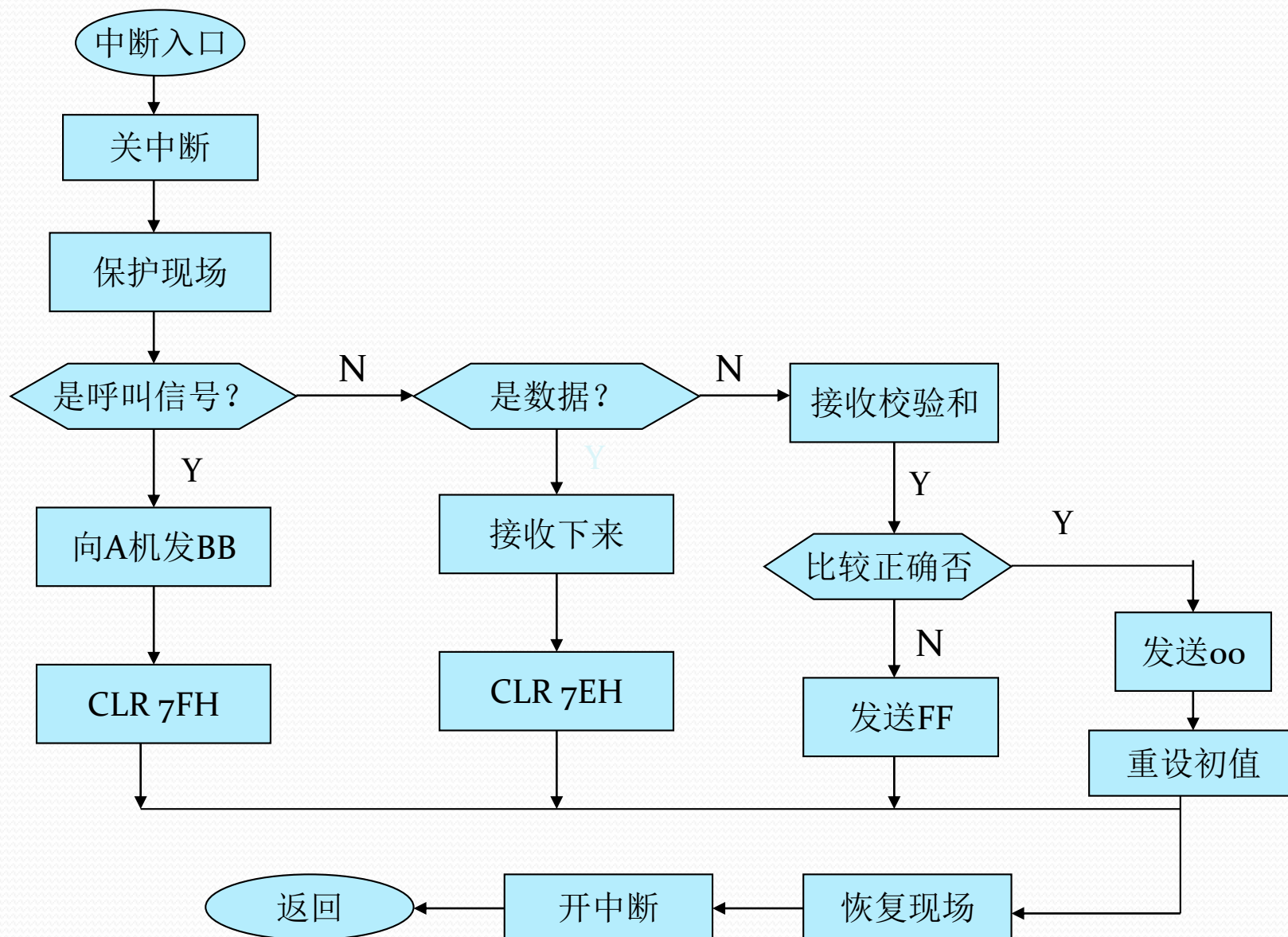
BRS:   JBC RI,BRS1
        SJMP BRS

BRS1:  MOV A, SBUF
        XRL A,R6
        JZ BEND
        MOV SBUF,#0FFH

BW2:   JBC TI,BRAM
        SJMP BW2

BEND:  MOV SBUF,#00H
        RET
```

B机中断方式接收流程图





B机中断方式接收程序

ORG 0000H	MOV R0, #30H
LJMP INIT	MOV R6, #00H
ORG 0023H	MOV R7, #20
LJMP SERVE	SETB EA
ORG 0050H	SETB ES
INIT: MOV TMOD,#20H	LJMP MAIN
MOV TH1,#0E8H	...
MOV TL1,#0E8H	
MOV SCON,#50H	
MOV PCON,#80H	
SETB TR1	
SETB 7FH	
SETB 7EH	



```
SERVE:      CLR EA
            CLR RI
            PUSH R0
            PUSH A
            JB 7FH,RX_ACK
            JB 7EH,RX_DATA
BR_SUM:     MOV A,SBUF
            XRL A,R6
            JZ TX_RIGHT
TX_ERR:     MOV SBUF,#0FFH
TX_ERRW:    JNB TI,TX_ERRW
            CLR TI
            SJMP AGAIN
TX_RIGHT:   MOV SBUF,#00H
TX_RTW:    JNB TI,TX_RTW
            CLR TI
            SJMP AGAIN
```



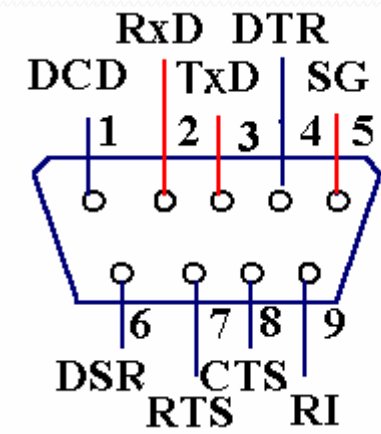
```
RX_ACK:      MOV A,SBUF
              XRL A,#0AAH
              JZ      TX_AGREE
              SJMP RETURN

TX_AGREE:    MOV SBUF,#0BBH
TX_AGW:      JNB TI,TX_AGW
              CLR TI
              CLR 7FH
              SJMP RETURN

RX_DATA:     MOV A,SUBF
              MOV @R0,A
              INC R0
              ADD A,R6
              MOV R6,A
              DJNZ R7,RETURN
              CLR 7EH
              SJMP RETURN
```

```
AGAIN:SETB 7FH
        SETB 7EH
        MOV R0,#30H
        MOV R6,#00H
        MOV R7,#20

RETURN:POP A
        POP R0
        SETB EA
        RETI
```



DB-9



常用波特率与其它参数的关系

串口工作方式	波特率	f_{osc}	SMOD	定时器T1		
				C/T	模式	定时器初值
方式0	1M	12MHz	-	-	-	-
方式2	375K	12MHz	1	-	-	-
	187.5K	12MHz	0	-	-	-
方式0或方式3	62.5K	12MHz	1	0	2	FFH
	19.2K	11.059MHz	1	0	2	FDH
	9.6K	11.059MHz	0	0	2	FDH
	4.8K	11.059MHz	0	0	2	FAH
	2.4K	11.059MHz	0	0	2	F4H
	1.2K	11.059MHz	0	0	2	F8H
	137.5K	11.059MHz	0	0	2	1DH
	110	12MHz	0	0	1	FEEDH
方式0	0.5M	6MHz	-	-	-	-
方式2	187.5K	6MHz	1	-	-	-
方式1.3	19.2K	6MHz	1	0	2	FEH
	9.6K	6MHz	1	0	2	FDH
	4.8K	6MHz	0	0	2	FDH
	2.4K	6MHz	0	0	2	FAH
	1.2K	6MHz	0	0	2	F4H
	0.6K	6MHz	0	0	2	E8H
	110	6MHz	0	0	2	72H
	55	6MHz	0	0	2	FEEDH



51单片机串行数据接口扩展

■ 主要包括

RS-232、RS-422、RS-485

USB

I2C

SPI



几种串口扩展技术

- 利用单片机空闲I/O引脚，通过专用芯片实现异步串口扩展，如：MAX233、MAX3110
- 利用单片机的并行数据接口，通过专用芯片实现异步串口扩展，如：TLC16C550、82C51
- 利用单片机空闲I/O引脚，通过软件编程模拟同步串行接口，如：I2C、SPI
- 利用单片机的并行数据接口，通过专用接口芯片实现USB(通用串行总线)扩展，如：PDIUSBD12
- 采用多串口单片机，如，W77E58