

第8章 串行口通信

主要内容

- **8.1 串行通信的概念**
- **8.2 串行口1的有关寄存器**
- **8.3 串行口1的工作方式**
- **8.4 串行口1通信中波特率的设置**
- **8.5 串行口2的有关寄存器**
- **8.6 串行口2的工作方式**
- **8.7 串行口的应用**
- **8.8 RS-232 串行原则**
- **8.9 RS-485 串行接口**
- **8.10 SPI 串行接口概述**
- **8.11 SPI 串行数据通信**
- **8.12 SPI 接口的应用**

8.1 串行通信的概念

串行通信是按一位一位的顺序传送数据，串行通信方式又分为**同步通信**和**异步通信**两种。

同步通信:发送方和接受方用同一种时钟控制信号

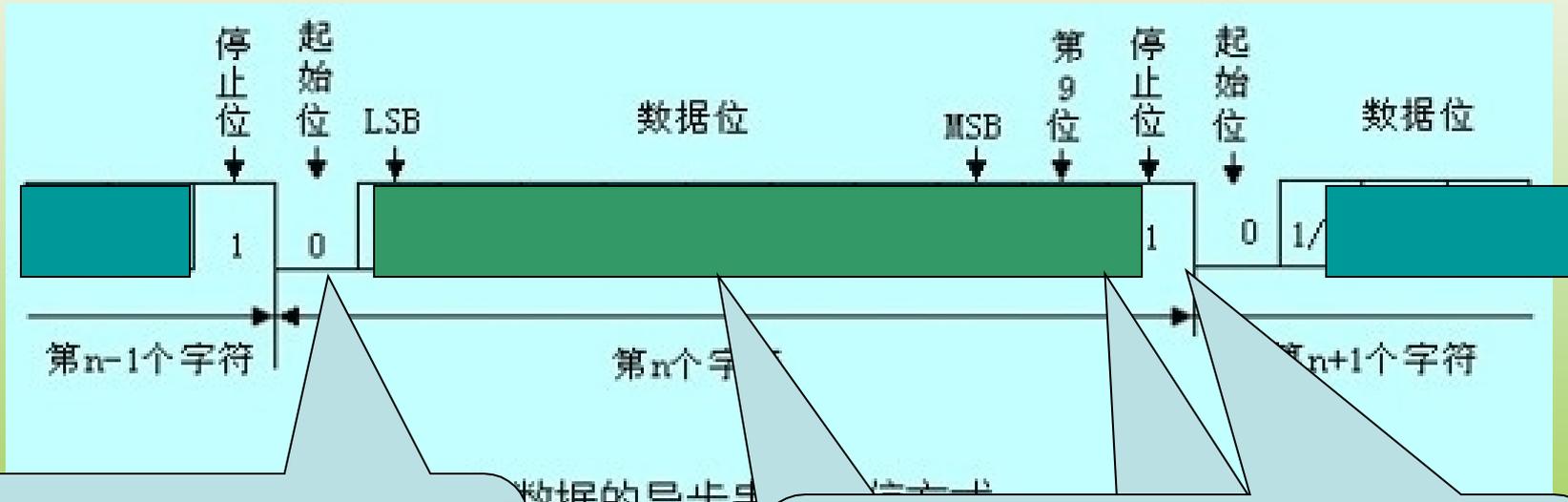


8.1 串行通信的概念

异步通信是一种将数据以一种字符为单位进行传送的通信方式

传送一种字符又称为一帧信息

发送与接受之间无时钟信号控制，**发送和接受的速率要求相同**



起始位: 位于一帧数据的开头，逻辑“0”电平信号，用来告知接受设备开始接受一帧信息

数据位: 紧跟在起始位之后的是数据位，低位在前，高位在后

停止位: 逻辑“1”电平信号，应不小于或等于一种位数据的时间。在发送完一帧信息后，用于发送的数据线一直处于高电平状态，等待再次发送

8.1 串行通信的概念

串行通信可分为单工、半双工和全双工3种方式

单工方式只有一根数据线，数据只能按照一种固定的方向传送

半双工方式的数据能够由甲向乙传送，也能够由乙向甲传送，但是不能在两个方向上同步传送

全双工方式能够在同一时刻实现双向传送。

在串行通信中，用波特率衡量通信速度的快慢。波特率是指每秒钟传送的二进制数的位数

单位：**位/秒**（bps） **字节/秒**（Bps）

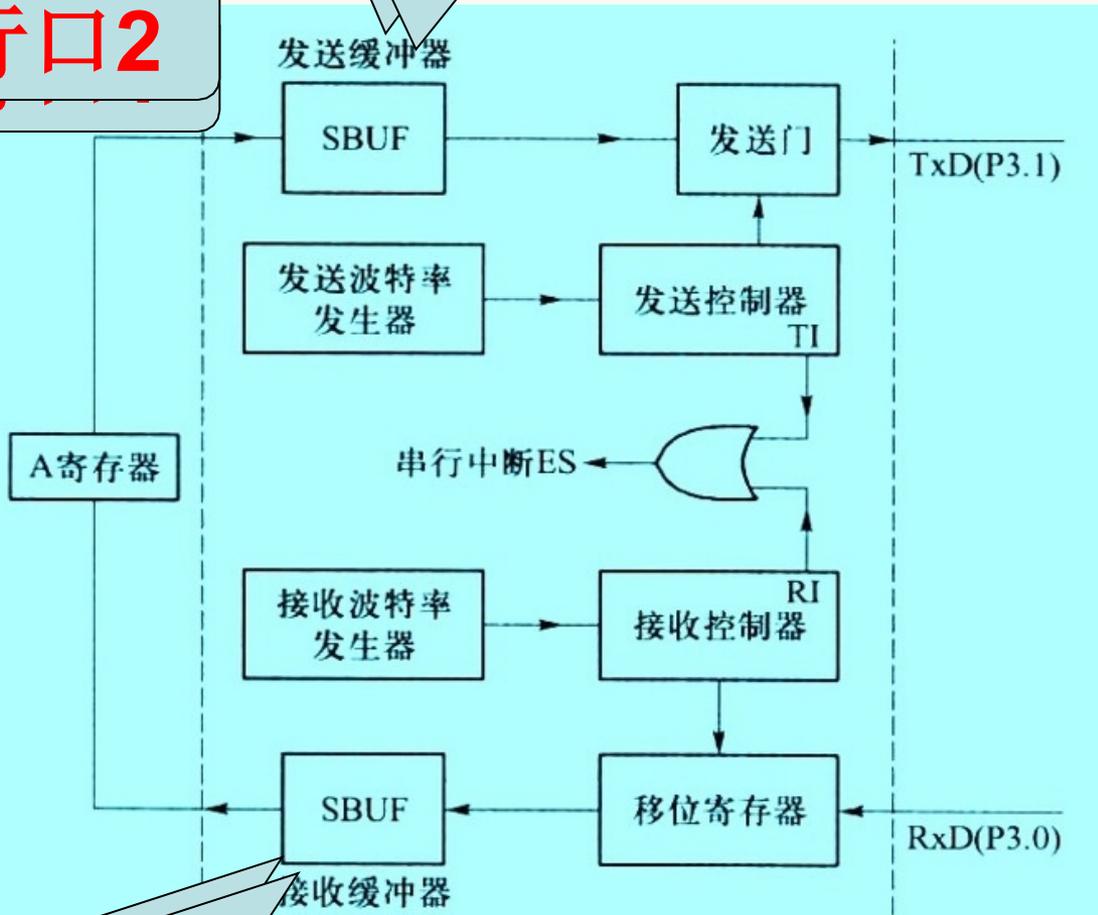
波特率（bps）=1个字符的二进制位数 × Bps

8.1 串行通信

发送缓冲器S2BUF
地址: 9BH

STC12C5A60S2单片机具有UART工作方式的串行口

串行口2



串行口1的内部结构框图

接受缓冲器S2BUF
地址: 9BH

8.2 串行口1的有关寄存器

STC12C5A60S2单片机的串行口1 与8个特殊功能寄存器有关

串行控制寄存器SCON

电源控制寄存器PCON

辅助寄存器AUXR

独立波特率发生器寄存器BRT

唤醒控制寄存器WAKE_CLKO

中断寄存器IE、IP、IPH。

8.2 串行口1的有关寄存器

1. 串行控制寄存器SCON，地址为98H，可进行位寻址

D7	D6	D5	D4	D3	D2	D1	D0
SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI

SM0	SM1	工作	波特
-----	-----	----	----

SM2: 在方式2和3中，只有接受到停止位时，接受才有效。位为1时，只有正确接受到停止位时，接受才干有效。

REN: 接收允许标志位。当该位为1时，允许接收信息；当该位为0时，禁止接受信息。

TB8: 发送8位标志位。

RB8: 接收8位标志位。

RI: 串行口1接受中断祈求标志位。

0	1	方式1	8位异步通	波特率等于 $f_{osc}/2$
---	---	-----	-------	-------------------

8.2 串行口1的有关寄存器

2.电源控制寄存器PCON，地址为87H

D7	D6	D5	D4	D3	D2	D1	D0
SMOD	SMOD0	LVDF	POF	GF1	GF0	PD	IDL

S **SMOD0**: 帧错误检测有效控制位。
该位置1, **SCON**寄存器中的**SM0/FE**位用于**FE**功能;
该位为0, **SCON**寄存器中的**SM0/FE**位用于方式选择。

8.2 串行口1的有关寄存器

3. 辅助寄存器AUXR，地址为8EH

D7	D6	D5	D4	D3	D2	D1	D0
T0x12	T1x12	UART_M0x6	BRTR	S2SMOD	BRTx12	EXTRAM	S1BRS

UART_M0x6: 串行口1的通信速度控制位。
UART_M0x6=0，波特率为系统时钟频率。
UART_M0x6=1，波特率为 $f_{osc}/2$ 。

S1BRS: 串行口1的波特率发生器选择位。
S1BRS=0时，选择定时/计数器T1作为串行口1的波特率发生器；
S1BRS=1时，选择独立波特率发生器作为串行口1的波特率发生器。

8.2 串行口1的有关寄存器

4.独立波特率发生器寄存器BRT，地址为9CH

独立波特率发生器是专门用于控制串行通信速度的定时器，与定时/计数器T1工作于方式2的情况相同。**BRT用于保存重装时间常数的一种8位寄存器**



注意：

串行口2：只能使用独立波特率发生器控制发送与接受，

串行口1：既能够选择T1作为波特率发生器，也能够选择独立波特率发生器。

串行口1和串行口2不能同步使用独立波特率发生器，

8.2 串行口1的有关寄存器

5. 唤醒控制寄存器WAKE_CLKO，地址为8FH

D7	D6	D5	D4	D3	D2	D1	D0
PCAWAKEUP	RXD_PIN_IE	T1_PIN_IE	T0_PIN_IE	LVD_WAKE	BRTCLKO	T1CLKO	T0CLKO

RXD_PIN_IE=0时，禁止P3.0/RxD引脚的下降沿置位RI，也禁止RxD引脚唤醒掉电模式；

RXD_PIN_IE=1时，允许P3.0/RxD引脚的下降沿将RI置1，而且RxD引脚的下降沿能够唤醒掉电模式

8.3 串行口1的工作方式

方式0

串行口1工作在方式0，作同步移位寄存器使用

D7	D6	D5	D4	D3	D2	D1	D0
T0x12	T1x12	UART_M0x6	BRTR	S2SMOD	BRTx12	EXTRAM	S1BRS

UART_M0x6: 串行口1方式0的通信速度控制位。

UART_M0x6=0，方式0的波特率为系统时钟频率的12分频；

UART_M0x6=1，方式0的波特率为 $f_{osc}/2$ 。

8.3 串行口1的工作方式

方式0

串行口数据由**RxD**引脚输入或输出，
同步移位脉冲经过**TxD**引脚输出，
发送、接受均为**8位数据**

发送过程： 在**TI=0**的条件下，
执行**MOV SBUF,A**指令，开启发送



注意：

发送顺序： 低位在前，高位在后

发送完一帧数据后，由**硬件将发送中断标志位TI置1**，向CPU发出中断祈求。

若要再次发送数据，**必须先经过程序将TI位清0。**

8.3 串行口1的工作方式

方式0

串行口数据由**RxD**引脚输入或输出，
同步移位脉冲经过**TxD**引脚输出，
发送、接受均为**8位数据**

接受过程： 在**RI=0**的条件下，
接受允许控制位**REN**置**1**时，处于接受状态
接受完一帧数据后，硬件将接受到的数据存入接
受缓冲器**SBUF**，并将中断标志位**RI**置**1**，向CPU
发出中断祈求



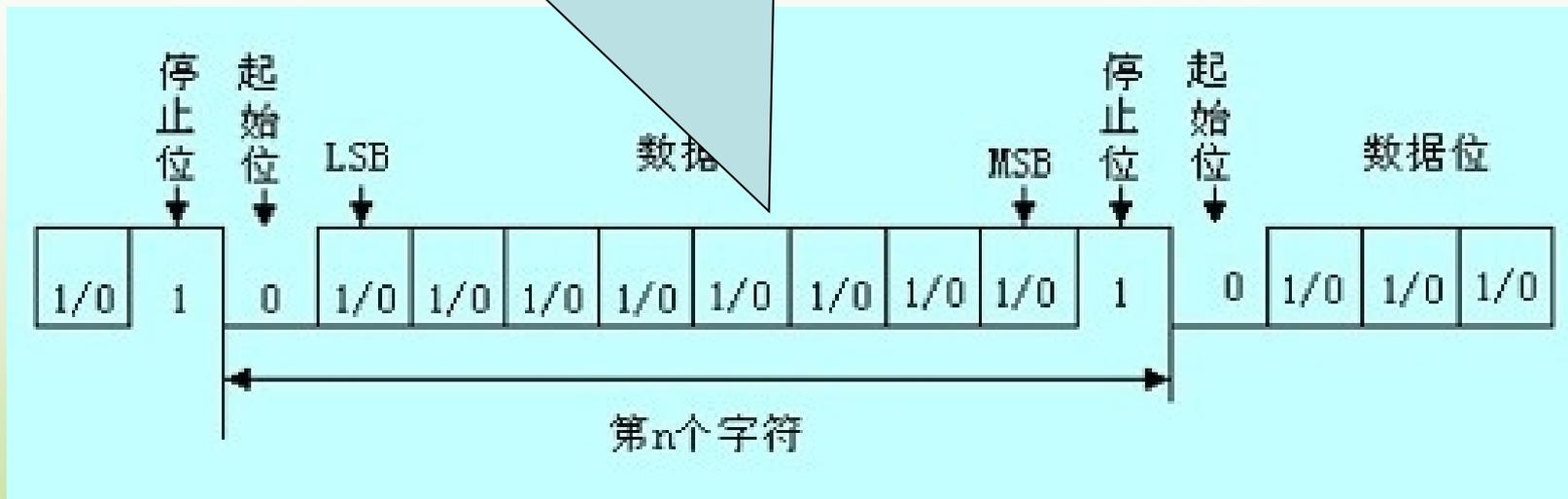
注意：

接受顺序： 先收到的是低位，后收到的是高位
若要继续接受数据，**必须先经过程序将RI位清0。**

8.3 串行口1的工作方式

方式1

8位数据，低位在前高位在后



发送过程: 在TI为0的条件下，执行将数据写入SBUF的指令时，开启发送过程。在内部移位脉冲的控制下，TxD引脚首先发送起始位、8位数据、停止位，发送完毕后TI置1，向CPU发出中断祈求

8.3 串行口1的工作方式

方式1



接受过程: 在**RI为0**、接受允许控制位**REN被置1**的条件下，CPU处于接受状态。当硬件检测到Rx_D引脚由高电平变为低电平时，开始接受一帧数据，并同步波特率发生器。

8.3 串行口1的工作方式

方式1



接受过程： (1) RI=0
(2) SM2=0或接受到的停止位为1时

注意： 不论满足这两个条件是否，接受控制器将继续检测RxD引脚，接受下一帧数据，所以在接受的过程中，应及时清接受中断标志位RI，以免数据丢失

8.3 串行口1的工作方式

方式2、方式3



方式2 与方式3唯一的区别在于方式2的波特率是固定的，而方式3的波特率与波特率发生器有关

方式1与方式3的区别是：方式1是8位的串行异步通信、方式3是9位串行异步通信

8.3 串行口1的工作方式

方式2、方式3



发送过程:

将发送中断标志位**TI清0**，并经过位传送指令将**第9位数据写入TB8**中，然后执行**MOV SBUF,A**指令，开启发送过程。一帧数据发送完毕，硬件将发送中断标志位**TI置1**

8.3 串行口1的工作方式

方式2、方式3



接受过程: 允许控制位**REN**被置**1**后处于接受状态。接受到的第**9**位（而不是停止位）装入到**RB8**位中



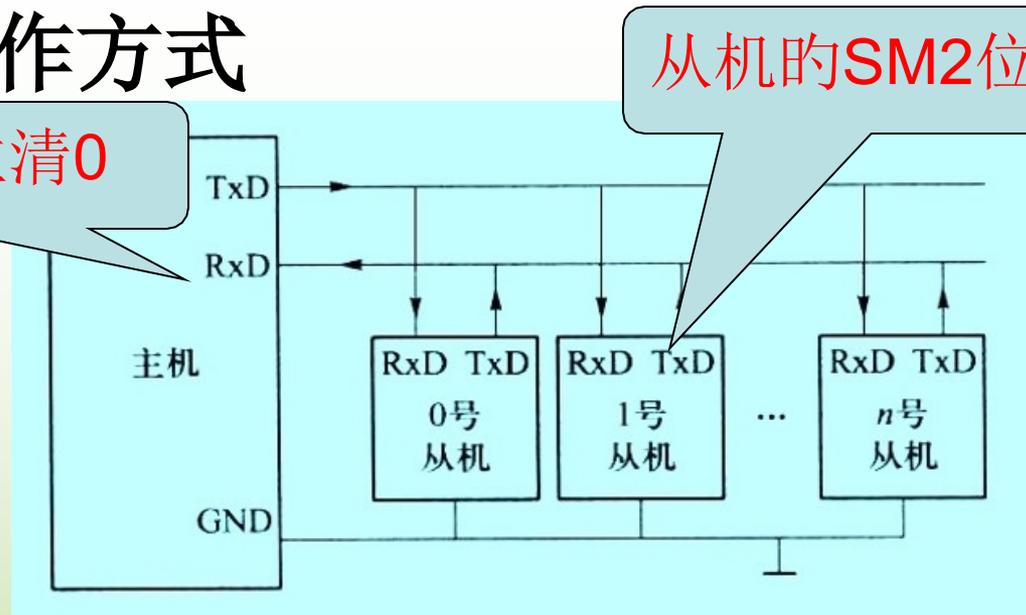
注意: 在方式2和方式3中，SM2为1时，只有接受到的第9位数据RB8位为1时，接受才干有效

8.3 串行口1的工作方式

8.3.4 主机的SM2位清0

在多机通信中**方式2和方式3**更适合主从通信

通信过程举例



(1) 主机向从机发送某一从机的地址，发送前将**TB8**位置1，然后进入接受状态

(2) 各从机接受到主机送出的地址信息后，**与本机地址相符时**，表达被选中，将本机地址回发给主机（**TB8**位为0），执行**CLR SM2**指令。

未被选中的从机，**SM2**位依然为1，接受不到**TB8**位为0的数据信息。

8.3 串行口1的工作方式

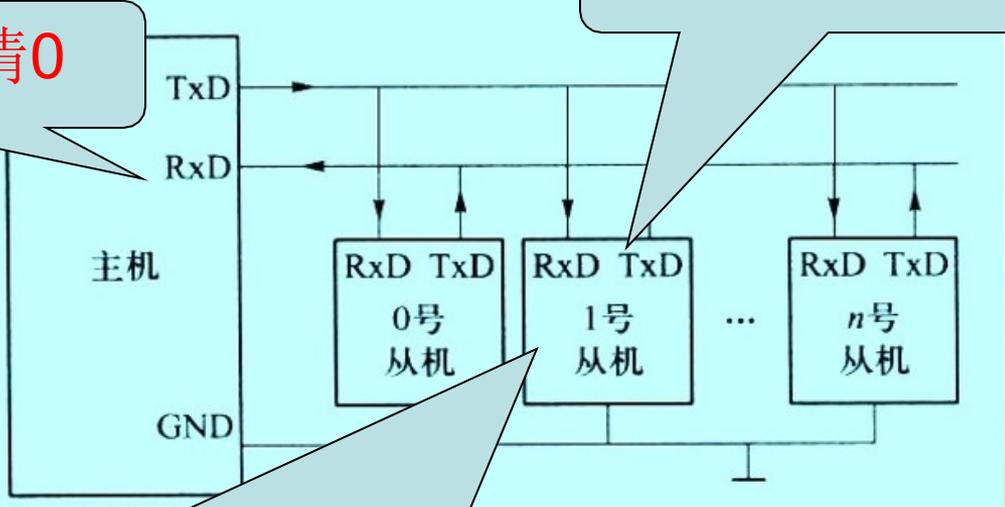
8.3.4 主机的SM2位清0

在多机通信中**方式2和方式3**更适合主从通信

通信过程举例

(3) 主机收到从机的地址一致，若不同，**异步**，若相同，**能够发出**

(4) 从机正确地接**回发先前约定好的应答**便从机能够接受主机再次**通信过程结束。**



从机的SM2位置1

理论上，一种主机能够与**256个从机通信**，但因为每个单片机引脚的输入阻抗及总线上阻抗的存在，总线上所允许并联从机的数量要受到限制。假如主机与过多的从机连接时，波特率应设置的低某些以降低通信错误

8.4 串行口1通信中波特率的设置

(1) 方式0

UART_M0x6=0时
，波特率是
 $f_{osc}/12$

辅助寄存器AUXR

D7	D6	D5	D4	D3	D2	D1	D0
T0x	T1x	UART_M0x6	BRTR	S2SMO	BRTx1	EXTR	S1BR
12	12			D	2	AM	S

UART_M0x6=1时
，波特率是 $f_{osc}/2$

8.4 串行口1通信中波特率的设置

SMOD=0时,
波特率 = $f_{osc}/64$

电源控制寄存器PCON

D7	D6	D5	D4	D3	D2	D1	D0
SMOD	SMOD0	LVDF	POF	GF1	GF0	PD	IDL

波特率 = $(2^{SMOD}/64) f_{osc}$

当SMOD=1时,
波特率 = $f_{osc}/32$

8.4 串行口1通信中波特率的设置

SMOD=0时,

波特率 = T1的溢出率 / 32

波特率 = $f_{osc} \times \text{SMOD} / (2^{\text{MOD}} \times 32)$ (定时/计数器T1的溢出率 / 32)

D7	D6	D5	D4	D3	D2	D1	D0
					GF0	PD	IDL

T1x12=0时,

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

D7	D6	D5	D4	D3	D2	D1	D0
T0x12	T1x12	UART_M0x6	BRTR	S2SMOD	BRTx12	EXTRAM	S1BR

T1x12=1时,

T1的溢出率 = $f_{osc} / (256 - \text{TH1})$

8.4 串行口1通信中波特率的设置

(3) 方式1和方式3

波特率= 2^{SMOD} (独立波特率发生器的溢出率 /32)

BRTx12 = 0时，独立波特率发生器的溢出率= $(f_{\text{osc}}/12)/(256-\text{BRT})$

D7	D6	D5	D4	D3	D2	D1	D0
T0x12	T1x12	UART_M0x6	BRTR	S2SMOD	BRTx12	EXTRAM	S1BR S

BRTx12 = 1时，独立波特率发生器的溢出率= $f_{\text{osc}}/(256-\text{BRT})$

8.4 串行口1通信中波特率的设置

(3) 方式1和方式3

$T1x12=0$, T1工作在方式2时, 几种波特率所相应T1的重装初值

波特率	系统时钟 (MHz)	SMOD	T1重装初值
19200	11.0592	1	FDH
9600	11.0592	1	FAH
9600	11.0592	0	FDH
4800	11.0592	1	F4H
4800	11.0592	0	FAH
2400	11.0592	1	E8H
2400	11.0592	0	F4H
1200	11.0592	1	D0H
1200	11.0592	0	E8H

8.5 串行口2的有关寄存器

与串行口2有关的寄存器有：

1. 串行口2的控制寄存器**S2CON**,地址为**9AH**,不能位寻址

D7	D6	D5	D4	D3	D2	D1	D0
S2SM0	S2SM1	S2SM2	S2SM3	S2SM4	S2SM5	S2TI	S2RI

S2SMOD位为0时,
串行口2的波特率不加倍

2. 辅助寄存器**AUXR**——波特率加倍控制位**S2SMOD**

D7	D6	D5	D4	D3	D2	D1	D0
T0x12	T1x12	UART_M0x6	BRTR	S2SMOD	BRTx12	EXTRAM	S1BR

S2SMOD位为1时,
串行口2的波特率加倍

8.5 串行口2的寄存器

与串行口2有关的

S2_P4位为0时，
TxD2为P1.3，RxD2为P1.2

3. 辅助寄存器AUXR1

D7	D6	D5	D4	D3	D2	D1	D0
-	PCA_P4	SPI_P4	S2_P4	GF2	ADRJ	-	DPS

4. 独立波特率

5. 中断寄存器

S2_P4位为1时，
TxD2为P4.3，RxD2为P4.2

6. 发送缓冲器S2BUF与接受缓冲器S2BUF
共用同一种地址9BH

8.6 串行口2的工作方式

8.6.1 方式0

当S2SM0、S2SM1为00时，串行口2工作在方式0

波特率= $f_{osc}/12$ 仅有这一种情况

8位数据低位在前，高位在后

经过RxD2 引脚接受或发送数据

TxD2引脚输出同步移位时钟

8.6 串行口2的工作方式

8.6.2 方式1

当S2SM0、S2SM1为01时，串行口2工作在方式1

TxD2引脚发送、**RxD2**引脚接受

一帧数据包括一种起始位、**8个数据位**和一种停止位

停止位存储在S2CON寄存器的**S2RB8**位

8.6 串行口2的工作方式

8.6.2 方式1

波特率 = $(2^{S2SMOD} / 32) \times$ 独立波特发生器的溢出率

BRTx12 = 0时，独立波特率发生器的溢出率 = $(f_{osc}/12) / (256 - BRT)$

D7	D6	D5	D4	D3	D2	D1	D0
T0x12	T1x12	UART_M0x6	BRTR	S2SMOD	BRTx12	EXTRAM	S1BR S

BRTx12 = 1时，独立波特率发生器的溢出率 = $f_{osc} / (256 - BRT)$

8.6 串行口2的工作方式

8.6.3 方式2

当S2SM0、S2SM1为10时，串行口2工作在方式2

TxD2引脚发送、**RxD2**引脚接受

一帧数据包括一种起始位、8位数据位、1位奇偶校验位、1位停止位

S2SMOD位为0时，

波特率 = $f_{osc} / 64$

发送时，第9位数据位来自S2TB8位，

接受时，第9位数据装入S2CC寄存器中的S2RB8位

D7	D6	D5	D4	D3	D2	D1	D0
T0x12	T1x12	UART_M0x6	BRTR	S2SMOD	BRTx12	EXTRAM	S1BR S

S2SMOD位为1时，

波特率 = $f_{osc} / 32$

8.6 串行口2的工作方式

8.6.4 方式3

当S2SM0、S2SM1为11时，串行口2工作在方式3

TxD2引脚发送、**RxD2**引脚接受

一帧数据包括一种起始位、**9个数据位**和一种停止位

发送时，第9位数据位来自S2CON寄存器中的S2TB8位，
接受时，第9位数据装入S2CON寄存器中的S2RB8位

波特率的设定与方式1相同

8.7 串行口的应用

1. 串行口1的初始化

(1) 设置**SCON**寄存器中各个有关位，由**SM0**、**SM1**两位拟定工作方式，将**TI**、**RI**清**0**，假如需要接受，将**REN**位置**1**；

(2) 设置合适的波特率，即设置定时/计数器**T1**的工作方式、给**TL1**、**TH1**赋初值，开启**T1**工作；使用独立波特率发生器时，给**BRT**赋初值，并初始化**AUXR**寄存器的**BRTR**及**BRTx12**两位；

(3) 根据需要拟定是否需要开放中断及中断优先级；

(4) 编写相应的中断或查询方式的服务程序。

8.7 串行口的应用

2. 串行口2的初始化

- (1) 即设置**S2CON**寄存器中的各个有关位的值，由**S2SM0**、**S2SM1**两位拟定工作方式，**S2RI**、**S2TI**两位清0。假如允许接受，将**S2REN**位置1；
- (2) 设置独立波特率发生器。初始化**AUXR**寄存器中的**BRTx12**、**S2SMOD**等位，将初值装入**BRT**寄存器；
- (3) 将**AUXR**中的**BRTR**位置1，开启独立波特率发生器工作；
- (4) 根据需要设置中断优先级，开放中断 **ES2**、 **EA**；
- (5) 编写相应的中断或查询方式的服务程序。

【例8-1】 STC12C5A60S2单片机的串行口1与一片8位串入并出的芯片74HC164连接，驱动一组发光二极管，使发光二极管从右至左延时轮番显示。

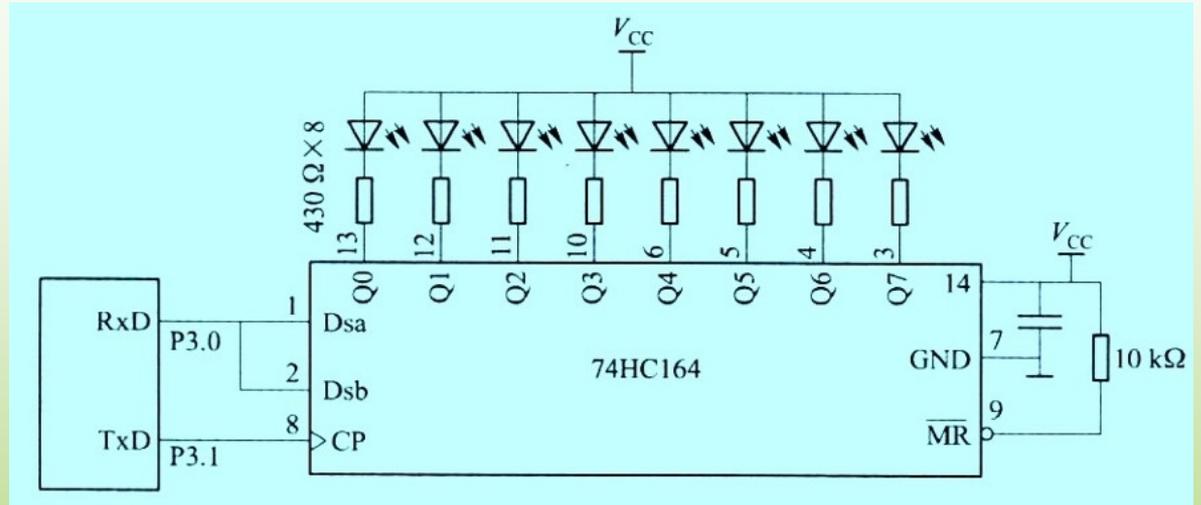
分析：74HC164是一种串行输入并行输出的移位寄存器，

Q0—Q7是并行输出端，

Dsa、Dsb为串行输入端。

CP为时钟输入端。

MR复位端，低电平有效，当复位端输入低电平时清除移位寄存器的值，强制全部的输出为低电平



设STC12C5A60S2单片机的系统时钟频率为12MHz，
串行口1工作在方式0。对SCON寄存器赋初值为00H

(1) 汇编程序:

```
    ORG 0000H
    LJMP MAIN
    ORG 0100H
MAIN: MOV SP,#0BFH      ; 初始化堆栈指针
      MOV SCON, #00H   ; 串行口1工作在方式0
      MOV A, #0FEH     ; LED显示码存A寄存器
START:MOV SBUF, A      ; 开启发送过程
LOOP:JNB TI, LOOP     ; 判断发送是否完毕
      CLR TI          ; 发送完毕, 将TI位清0
      ACALL DELAY     ; 延时
      RL A            ; 发光二极管显示码左移一位
      SJMP START
```

```
DELAY:  MOV R7,#0BDH ; 500ms的延时程序  
DL1:    MOV R6,#0E6H  
DL0:    MOV R5,#21H  
         DJNZ R5,$  
         DJNZ R6,DL0  
         DJNZ R7,DL1  
         RET  
         END
```

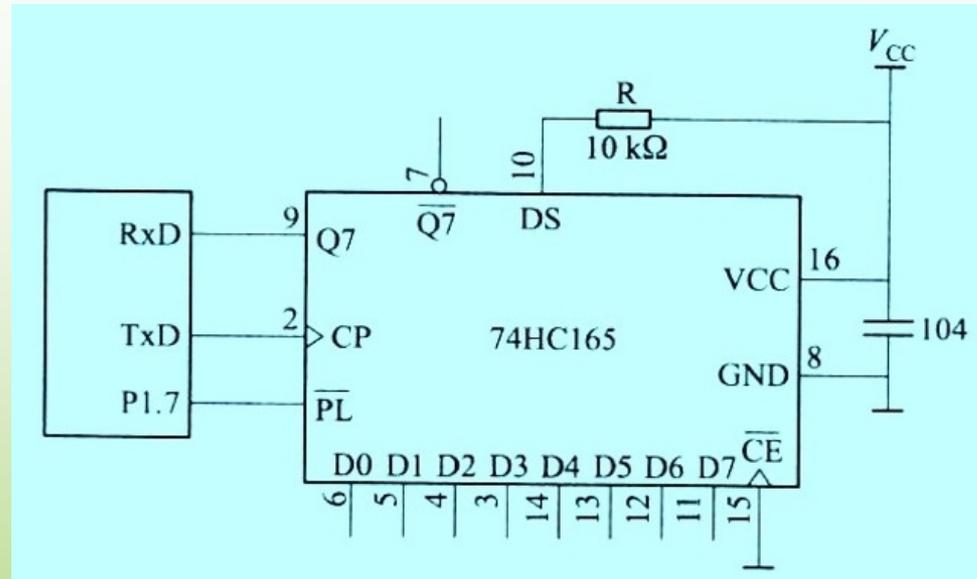
(2) C语言程序:

```
#include <STC12C5A.h>
#include<intrins.h>//申明本征函数库
void DELAY ( ) ; //延时函数
void main( )
{
unsigned char j;
SCON=0x00;//串行口1工作在方式0
j=0xFE; //LED显示码
for ( ; ; )
{
    SBUF=j;//开启发送过程
    while (!TI); //等待发送完毕
    TI=0;
    DELAY( ); //延时500ms
    j=_crol_(j,1);//循环左移一位
}
}
```

```
void DELAY( ) //延时500ms函数
{
    unsigned char a,b,c;
    for(c=189;c>0;c--)
        for(b=230;b>0;b--)
            for(a=33;a>0;a--);
}
```

【例8-2】 利用STC12C5A60S2单片机的串行口1外接一片8位的并入串出芯片74HC165，用于扩展并行输入，如图所示。

分析：74HC165是一种并入串出的芯片，其中D0—D7为并行数据输入端，DS为串行数据输入端，Q7为串行输出端。



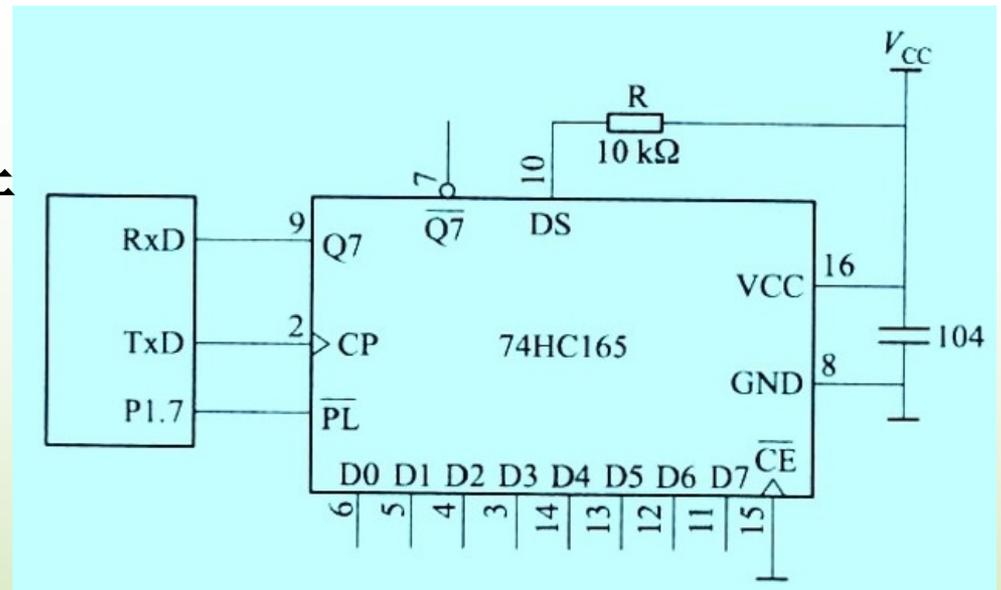
当并行输入允许引脚输入为低时，从D0至D7引脚并行输入的数据被输入到内部移位寄存器内，

而当为高时，并行输入禁止，移位寄存器的数据在CP引脚输入的时钟脉冲上升沿的控制下，向右移动一位DS→Q0→Q1→...→Q7，Q7的状态出现在第9引脚上。

设串行口1工作在方式0，SCON寄存器的内容为00H。

(1) 汇编程序段如下:

```
MOV SCON, #00000000B
CLR P1.7 ; 并行输入允许
NOP ; 等待引脚状态稳定
NOP
NOP
NOP
NOP
NOP
SETB P1.7 ; 禁止并行输入
NOP ; 等待引脚状态稳定
NOP
NOP
NOP
NOP
NOP
```



```
CLR RI ; 清除中断标志RI
SETB REN ; 允许接受
LOOP: JNB RI, LOOP ; 等待
CLR RI
MOV A, SBUF
```

(2) C语言程序

```
#include <STC12C5A.h>
```

```
#include<intrins.h>
```

```
sbit P1_7=P1^7;
```

```
void main( )
```

```
{
```

```
unsigned char temp;
```

```
SCON=0x00;
```

```
P1_7=0; //并行输入允许
```

```
_nop_(); //等待引脚状态稳定
```

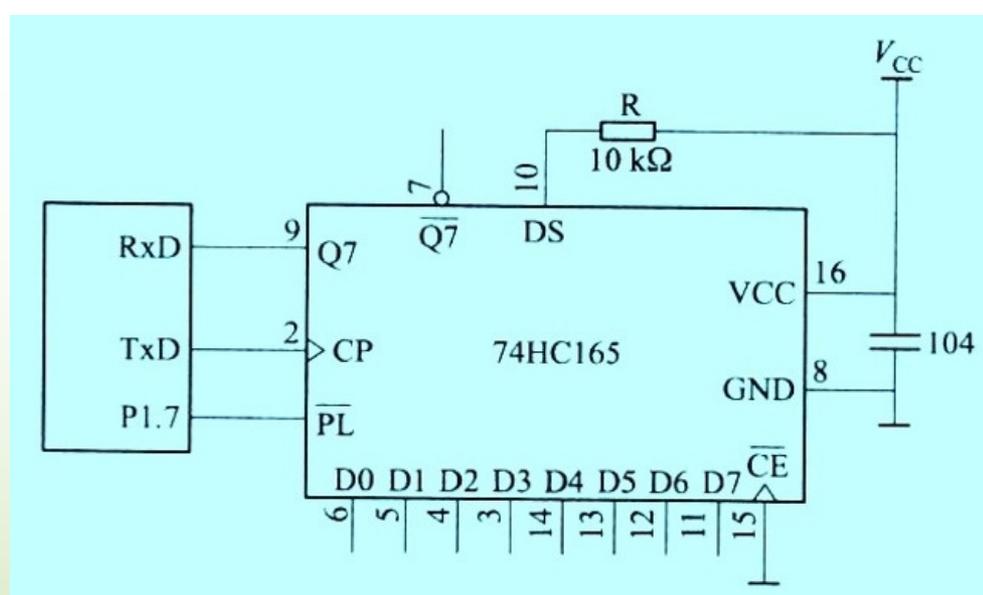
```
_nop_();_nop_();_nop_();
```

```
_nop_();_nop_();
```

```
P1_7=1; //并行输入禁止
```

```
_nop_();_nop_();_nop_();
```

```
_nop_();_nop_();_nop_();
```



```
RI=0; //清除接受中断标志RI
```

```
REN=1; //允许接受
```

```
while (!RI); //等待接受数据
```

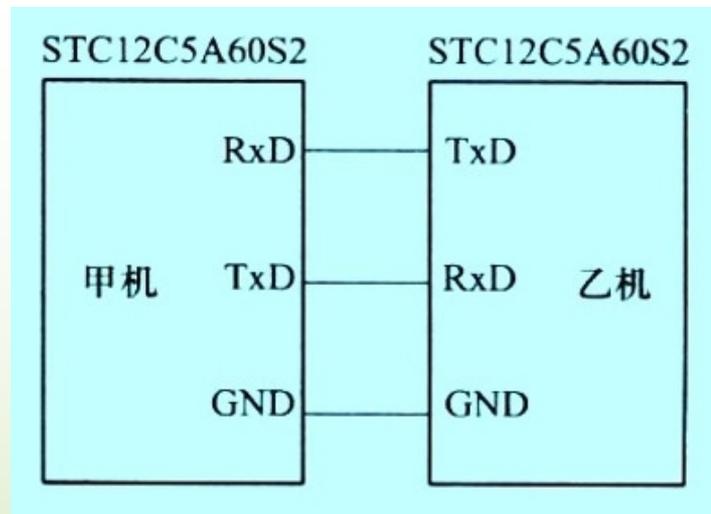
```
RI=0; //清除接受中断标志RI
```

```
temp=SBUF; //读入数据
```

```
while (1);
```

```
}
```

【例8—3】在图8—10所示的甲、乙两机间的串行通信中，单片机系统的时钟频率为11.0592MHz，要求波特率为1200bps，编程实现将甲机片内的30H—3FH单元的数据传送到乙机片内40H—4FH单元中。

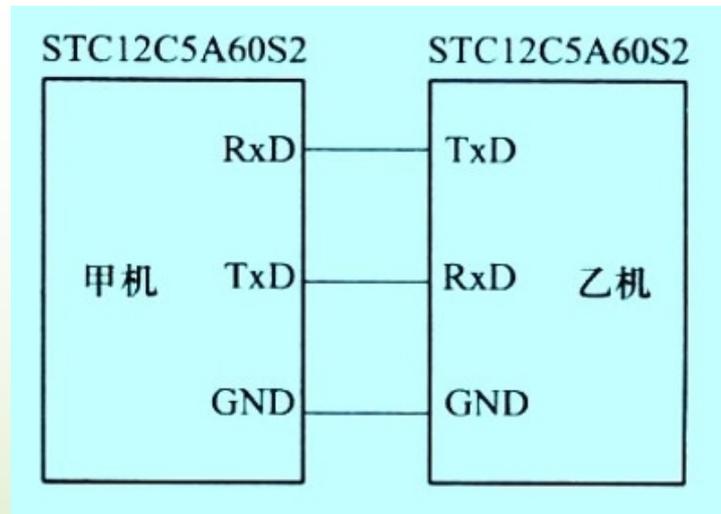


(1) 汇编子程序:

分析：甲、乙两机都选择串行口1工作在**方式3**，**9位异步通信**方式，**TB8**用作**奇偶校验位**，甲机串口控制寄存器**SCON**的初值为**0C0H**，乙机的控制字**0D0H**。

D7	D6	D5	D4	D3	D2	D1	D0
SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI

【例8-3】在图8-10所示的甲、乙两机间的串行通信中，单片机系统的时钟频率为11.0592MHz，要求波特率为1200bps，编程实现将甲机片内的30H-3FH单元的数据传送到乙机片内40H-4FH单元中。



(1) 汇编子程序:

分析: 甲、乙两机都选择串行口1工作在**方式3**，**9位异步通信**方式，**TB8用作奇偶校验位**，甲机串口控制寄存器SCON的初值为0C0H，乙机的控制字0D0H。

定时/计数器T1工作在方式2作为波特率发生器，TMOD为20H;

波特率 = $2^{\text{SMOD}} \times [(\text{定时/计数器T1的溢出率}) / 32]$,

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

T1的重装初值 = $256 - (\text{fosc} / 12) / \text{T1的溢出率} = 256 - (\text{fosc} / 12)$

$\times 2^{\text{SMOD}} / (\text{波特率} \times 32) = 256 - (11059200 / 12)$

$/ (1200 \times 32) = 232 = \text{E8H}$

甲机的发送子程序:

```

TRA_UART: MOV TMOD,#20H ; T1工作在方式2
ANL AUXR,#0BFH ; T1计数工作在系统时钟的12分频
MOV TL1,#0E8H ; 给T1的低8位赋初值
MOV TH1,#0E8H ; 给T1的高8位赋初值
ANL PCON,#7FH ; 波特率不加倍
MOV SCON,#0C0H ; 串行口1工作在方式3
MOV R0,#30H ; R0作为片内RAM指针
MOV R7,#10H ; R7作为发送计数器
SETB TR1 ; 波特率发生器开始工作
LOOP:MOV A,@R0 ; 读取片内30H—3FH单元的内容
MOV C,P
MOV TB8, C ; 将奇偶校验位写入TB8
MOV SBUF, A ; 开启发送过程
WAIT: JNB TI,WAIT ; 等待发送完毕
CLR TI ; 发送完毕, 将TI清0
INC R0 ; 指向下一种单元地址
DJNZ R7,LOOP ; 假如数据未发送完, 将继续发送
RET

```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/968142133111006140>