

单片机应用技术

项目十

AVR 单片机 USART
串行通信应用

【知识目标】

- 了解通信基础知识
- 掌握异步串行通信协议
- 了解 ATmega16 单片机串行通信接口结构
- 了解与串行通信有关的寄存器的功能

【能力目标】

- 掌握 ATmega16 单片机的串行接口相关寄存器的配置方法
- 掌握 RS—232C 与 TTL 电平转换方法
- 掌握 ATmega16 单片机的串行接口及相关寄存器的配置方法
- 掌握简单的单片机串行通信系统程序的编写、调试方法
- 掌握 PC 机串口调试软件的使用方法

【任务 1】 项目知识点学习

❖ 一、通信知识概述

- 在通信领域，有两种数据通信方式：**并行通信**、**串行通信**

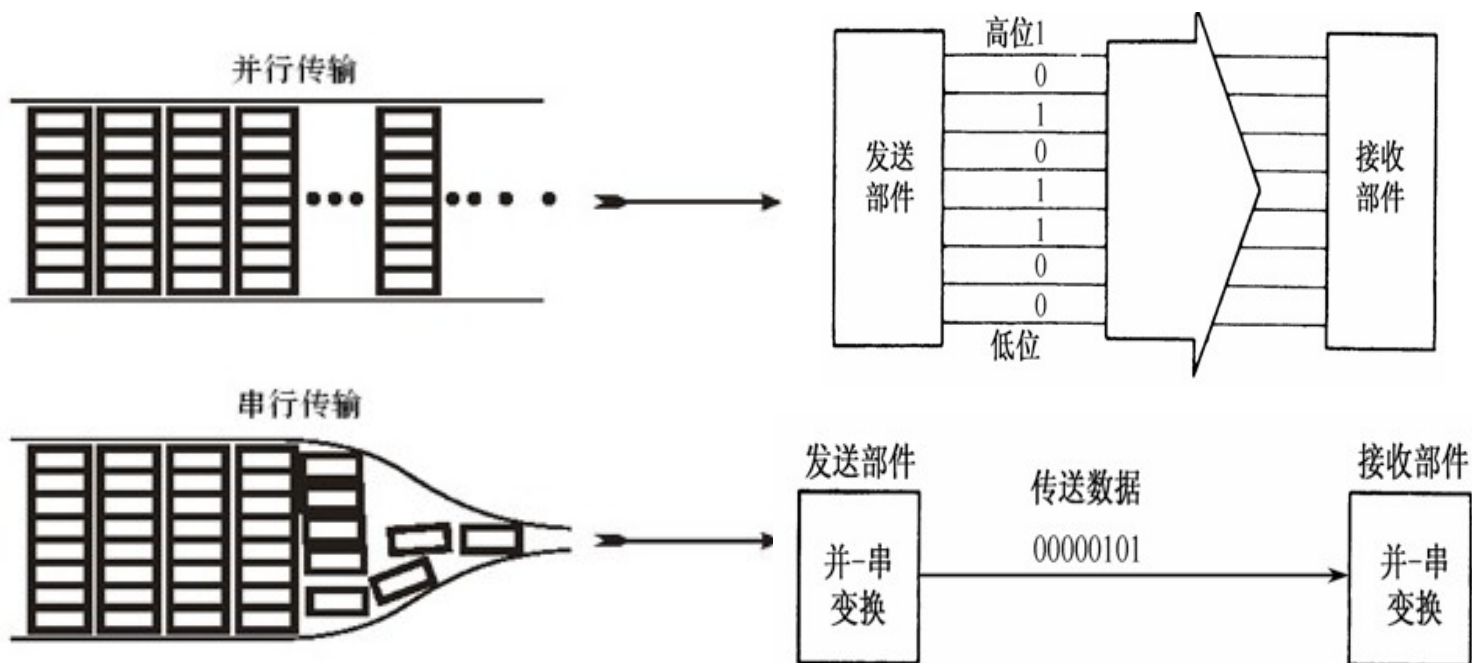


图 10-1 并行、串行通信示意图

【任务 1】 项目知识点学习

❖ 二、串行通讯制式

- 根据信息的传送方向，串行通讯可以进一步分为：**单工**、**半双工**和**全双工**三种。

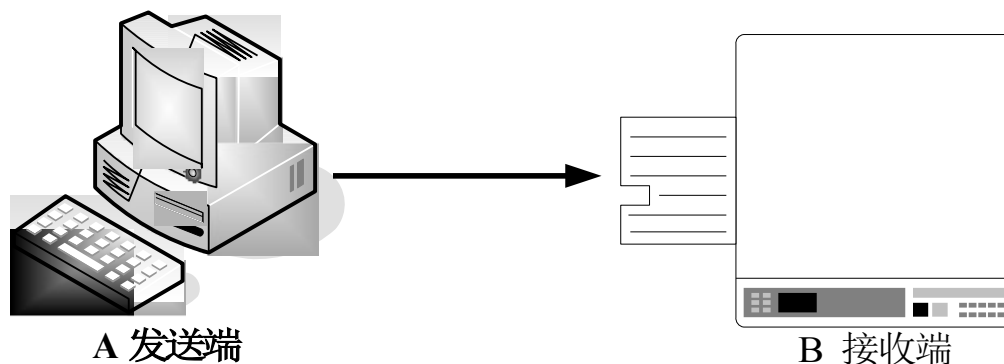


图 10-2 单工方式

【任务 1】 项目知识点学习

❖ 二、串行通讯制式

- 根据信息的传送方向，串行通讯可以进一步分为：**单工**、**半双工**和**全双工**三种。

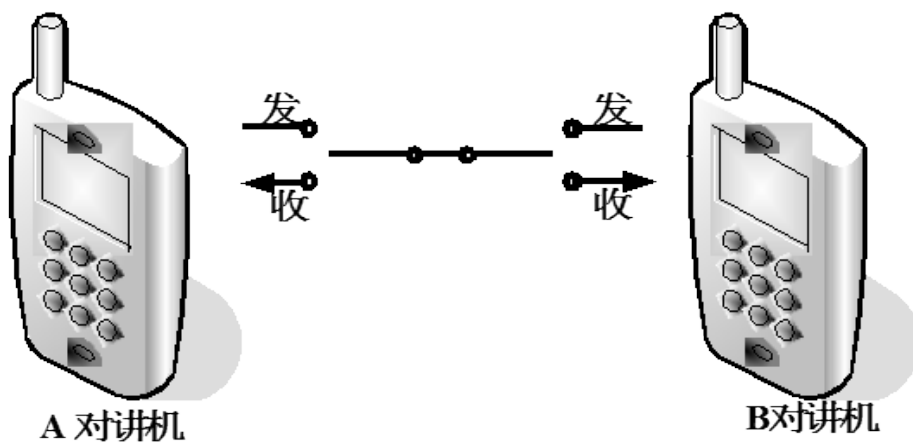


图 10-3 半双工方式

【任务 1】 项目知识点学习

❖ 二、串行通讯制式

- 根据信息的传送方向，串行通讯可以进一步分为：**单工**、**半双工**和**全双工**三种。

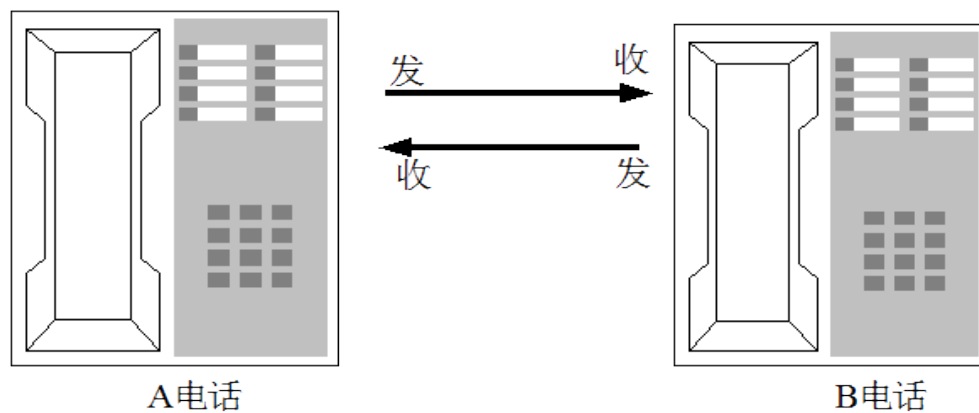


图 10-4 全双工方式

【任务 1】 项目知识点学习

❖ 三、串行数据传输的分类

- 按照串行数据的时钟控制方式，串行通信又可分为**同步通信**和**异步通信**两种。

同步通信 ---Synchronous Communication：发送器和接收器由同一个时钟源控制。



图 10-5 同步通信的字符帧格式

【任务 1】 项目知识点学习

❖ 三、串行数据传输的分类

- 按照串行数据的时钟控制方式，串行通信又可分为**同步通信**和**异步通信**两种。

异步通信 ---Asynchronous Communication。

1) 字符帧 (Character Frame)

字符帧也叫数据帧，由起始位、数据位、奇偶校验位和停止位等 4 部分组成，如图 1.6 所示。

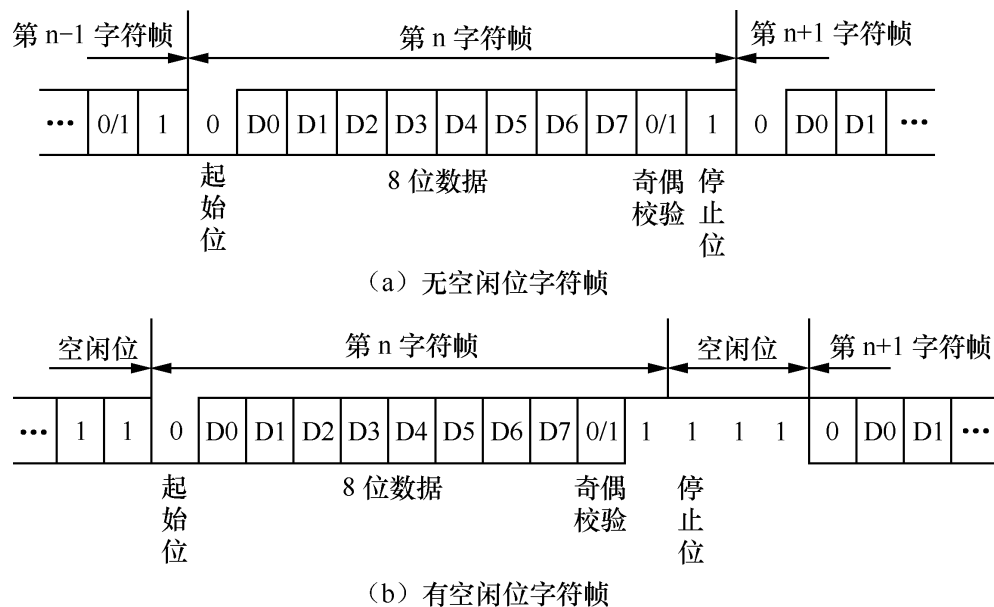


图 10-6 异步通信的字符帧格式

【任务 1】 项目知识点学习

❖ 三、串行数据传输的分类

- 按照串行数据的时钟控制方式，串行通信又可分为**同步通信**和**异步通信**两种。

异步通信 ---Asynchronous Communication 。

1) 字符帧 (Character Frame)

字符帧也叫数据帧，
由起始位、数据位、
奇偶校验位和停止位
等 4 部分组成，如图
1.6 所示。

图 10-6 异步通信的字符帧格式

【任务 1】 项目知识点学习

❖ 三、串行数据传输的分类

- 按照串行数据的时钟控制方式，串行通信又可分为**同步通信**和**异步通信**两种。

异步通信 ---Asynchronous Communication 。

(2) 波特率 (Baud Rate)

异步通信的另一个重要指标为波特率，它是一个衡量符号传输速率的参数，表示每秒钟传送的符号的个数。假如波特率参数设定为 9600bps，即每秒传输 9600bit 数据。

异步通信的优点是不需要传送同步时钟，字符帧长度不受限制，故设备简单；缺点是字符帧中因包含始位和停止位而降低了有效数据的传输速率。

【任务 1】 项目知识点学习

❖ 三、串行数据传输的分类

- 按照串行数据的时钟控制方式，串行通信又可分为**同步通信**和**异步通信**两种。

异步通信 ---Asynchronous Communication 。

(2) 波特率 (Baud Rate)

异步通信的另一个重要指标为波特率，它是一个衡量符号传输速率的参数，表示每秒钟传送的符号的个数。假如波特率参数设定为 9600bps，即每秒传输 9600bit 数据。

异步通信的优点是不需要传送同步时钟，字符帧长度不受限制，故设备简单；缺点是字符帧中因包含始位和停止位而降低了有效数据的传输速率。

【任务 1】 项目知识点学习

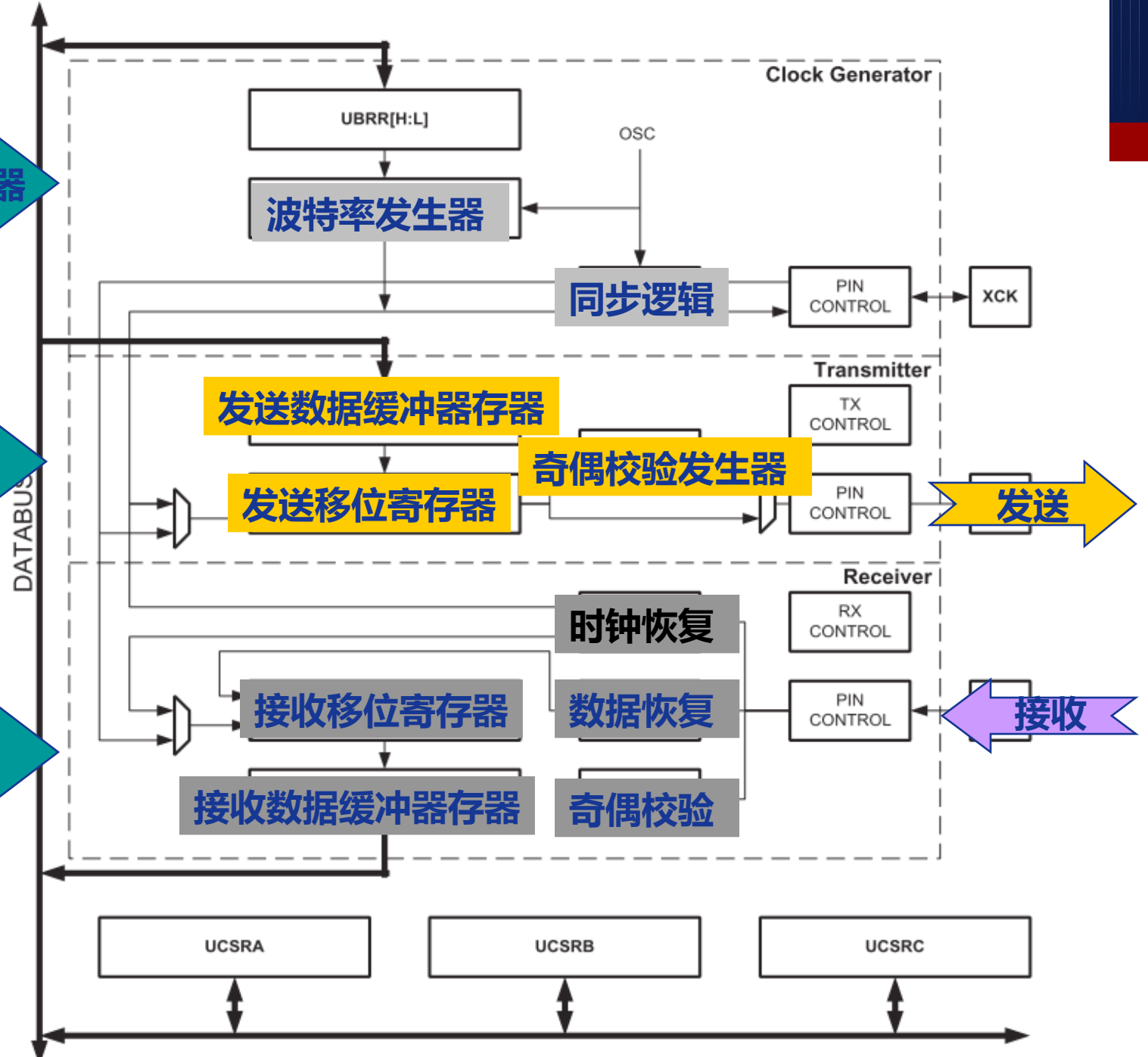
❖ 四、 ATmega16 单片机的串行口及相关寄存器

与 ATmega16 单片机串行口通信有关的特殊功能寄存器有数据缓冲器 UDR，控制和状态寄存器 UCSRA、UCSRB、UCSRC，波特率寄存器 UBRRL、UBRRH。下面对它们分别作简单的介绍。

时钟发生器

发送器

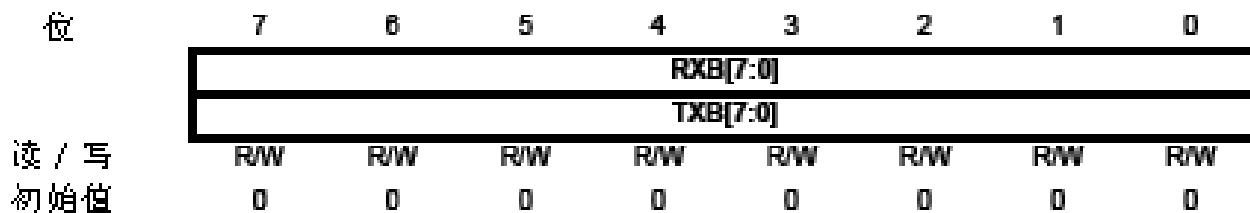
接收器



【任务 1】 项目知识点学习

1) 数据缓冲器 UDR

UDR 数据缓冲器的格式图如图 1.7 所示：



ATmega16 单片机 USART 发送数据缓冲寄存器和 USART 接收数据缓冲寄存器**共享相同的 I/O 地址**，称为 USART 数据寄存器或 UDR。将数据写入 UDR 时实际操作的是发送数据缓冲寄存器 (TXB)，读 UDR 时实际返回的是接收数据缓冲寄存器 (RXB) 的内容。

只有当 UCSRA 寄存器的 **UDRE 标志置位**后才可以对发送缓冲器进行写操作。如果 UDRE 没有置位，那么写入 UDR 的数据会被 USART 发送器忽略。当数据写入发送缓冲器后，若移位寄存器为空，发送器将把数据加载到发送移位寄存器。然后数据串行地从 TxD 引脚输出。

【任务 1】 项目知识点学习

2) 控制状态寄存器 UCSRA、UCSRB、UCSRC

(1) 控制状态寄存器 UCSRA

位	7	6	5	4	3	2	1	0	
	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	UCSRA
读 / 写	R	R/W	R	R	R	R	R/W	R/W	
初始值	0	0	1	0	0	0	0	0	

RXC: USART 接收结束

接收缓冲器中有未读出的数据时 RXC 置位，否则清零。接收器禁止时，接收缓冲器被刷新，导致 RXC 清零。RXC 标志可用来产生接收结束中断（见对 RXCIE 位的描述）。

TXC: USART 发送结束

发送移位缓冲器中的数据被送出，且当发送缓冲器(UDR) 为空时 TXC 置位。执行发送结束中断时 TXC 标志自动清零，也可以通过写 1 进行清除操作。TXC 标志可用来产生发送结束中断（见对 TXCIE 位的描述）。

【任务 1】 项目知识点学习

位	7	6	5	4	3	2	1	0	
	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	UCSRA
读 / 写	R	R/W	R	R	R	R	R/W	R/W	
初始值	0	0	1	0	0	0	0	0	

FE: 帧错误

如果接收缓冲器接收到的下一个字符有帧错误，**即接收缓冲器中的下一个字符的第一个停止位为 0**，那么 FE 置位。这一位一直有效直到接收缓冲器 (UDR) 被读取。当接收到的停止位为 1 时，FE 标志为 0。对 UCSRA 进行写入时，这一位要写 0。

DOR: 数据溢出

数据溢出时 DOR 置位。**当接收缓冲器满（包含了两个数据），接收移位寄存器又有数据，若此时检测到一个新的起始位，数据溢出就产生了。**这一位一直有效直到接收缓冲器 (UDR) 被读取。对 UCSRA 进行写入时，这一位要写 0。

【任务 1】 项目知识点学习

位	7	6	5	4	3	2	1	0	
读 / 写	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	UCSRA
初始值	0	0	1	0	0	0	0	0	

PE: 奇偶校验错误

当奇偶校验使能 ($UPM1 = 1$)，且接收缓冲器中所接收到的下一个字符有奇偶校验错误时 UPE 置位。这一位一直有效直到接收缓冲器 (UDR) 被读取。对 UCSRA 进行写入时，这一位要写 0。

U2X: 倍速发送

这一位仅对异步操作有影响。使用同步操作时将此位清零。此位置 1 可将波特率分频因子从 16 降到 8，从而有效的将异步通信模式的传输速率加倍。

MPCM: 多处理器通信模式

设置此位将启动多处理器通信模式。MPCM 置位后，USART 接收器接收到的那些不包含地址信息的输入帧都将被忽略。发送器不受 MPCM 设置的影响。

【任务 1】 项目知识点学习

(2) 控制状态寄存器 UCSRB 格式

位	7	6	5	4	3	2	1	0	
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	UCSRB
读 / 写	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
初始值	0	0	0	0	0	0	0	0	

RXCIE: 接收结束中断使能 *** 向量号 :

置位后使能 RXC 中断。当 RXCIE 为 1，全局中断标志位 SREG 置位，UCSRA 寄存器的 RXC 亦为 1 时可以产生 USART 接收结束中断。

TXCIE: 发送结束中断使能 *** 向量号 :

置位后使能 TXC 中断。当 TXCIE 为 1，全局中断标志位 SREG 置位，UCSRA 寄存器的 TXC 亦为 1 时可以产生 USART 发送结束中断。

【任务 1】 项目知识点学习

位	7	6	5	4	3	2	1	0	
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	UCSRB
读 / 写	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
初始值	0	0	0	0	0	0	0	0	

UDRIE: USART 数据寄存器空中断使能. 置位后使能 UDRE 中断。当 UDRIE 为 1，全局中断标志位 SREG 置位，UCSRA 寄存器的 UDRE 亦为 1 时可以产生 USART 数据寄存器空中断。 ***** 向量号：**

RXEN: 接收使能. 置位后将启动 USART 接收器。RxD 引脚的通用端口功能被 USART 功能所取代。禁止接收器将刷新接收缓冲器，并使 FE、DOR 及 PE 标志无效。

TXEN: 发送使能置. 位后将启动将启动 USART 发送器。TxD 引脚的通用端口功能被 USART 功能所取代。TXEN 清零后，只有等到所有的数据发送完成后发送器才能够真正禁止，即发送移位寄存器与发送缓冲寄存器中没有要传送的数据。发送器禁止后，TxD 引脚恢复其通用 I/O 功能。

【任务 1】 项目知识点学习

位	7	6	5	4	3	2	1	0	
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	UCSRB
读 / 写	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
初始值	0	0	0	0	0	0	0	0	

UCSZ2: 字符长度

UCSZ2 与 UCSRC 寄存器的 UCSZ1:0 结合在一起可以设置数据帧所包含的数据位数 (字符长度)。

RXB8: 接收数据位 8

对 9 位串行帧进行操作时，RXB8 是第 9 个数据位。读取 UDR 包含的低位数据之前首先要读取 RXB8。

TXB8: 发送数据位 8

对 9 位串行帧进行操作时，TXB8 是第 9 个数据位。写 UDR 之前首先要对它进行写操作。

【任务 1】 项目知识点学习

(3) 控制状态寄存器 UCSRC

位	7	6	5	4	3	2	1	0	
	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	UCSRC
读 / 写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初始值	1	0	0	0	0	1	1	0	

在 ATmega16 单片机中，UCSRC 寄存器与 UBRRH 寄存器共用相同的 I/O 地址。对控制寄存器 UCSRC 的各位介绍如下：

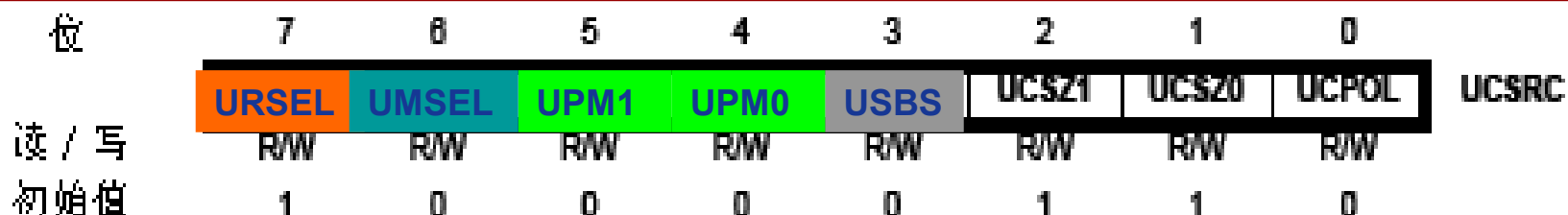
URSEL: 寄存器选择

通过该位选择访问 UCSRC 寄存器或 UBRRH 寄存器。当读 UCSRC 时，该位为 1；当写 UCSRC 时，URSEL 为 1。

UMSEL: USART 模式选择

当 UMSEL 位为 0 时，串行口工作于异步操作模式；当 UMSEL 位为 1 时，串行口工作于同步操作模式。

【任务 1】 项目知识点学习

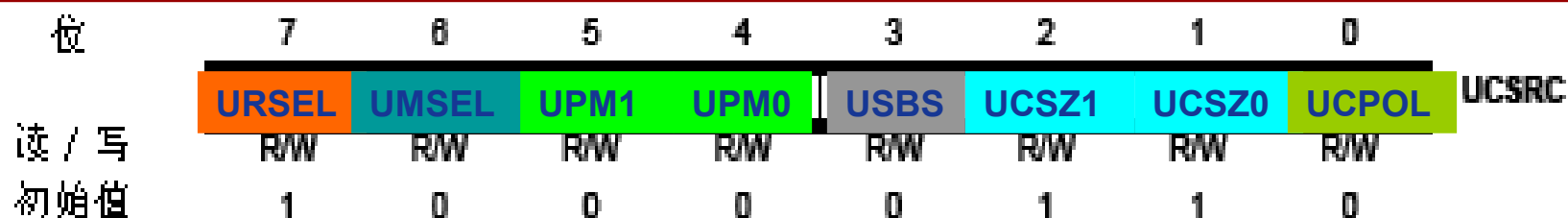


- ❖ **UPM1:0: 奇偶校验模式** . 这两位设置奇偶校验的模式并使能奇偶校验。如果使能了奇偶校验，那么在发送数据，发送器都会自动产生并发送奇偶校验位。对每一个接收到的数据，接收器都会产生一奇偶值，并与接收到的数据进行比较。如果不匹配，那么就产生一个错误标志。ATmega16 单片机串行接口的奇偶校验模式设置如表 1.1 所示：

UPM1	UPM0	奇偶模式
0	0	禁止
0	1	保留
1	0	校验
1	1	奇校验

- ❖ **USBS: 停止位选择**通过这一位可以设置停止位的位数。接收器忽略这一位的设置。当 USBS 位为 0 时，停止位位数为 1；当 USBS 位为 1 时，停止位位数为 2。

【任务 1】 项目知识点学习



- ❖ UCSZ1:0: 字符长度 .UCSZ1:0 与 UCSRB 寄存器的 UCSZ2 结合在一起可以设置数据帧包含的数据位数 (字符长度)。其具体设置如表所示：

UCSZ2	UCSZ1	UCSZ0	字符长度
0	0	0	5 位
0	0	1	6 位
0	1	0	7 位
0	1	1	8 位
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	9 位

- ❖ UCPOL: 时钟极性 . 这一位仅用于同步工作模式。使用异步模式时，将这一位清零。UCPOL 设置了输出数据的改变和输入数据采样，以及同步时钟 XCK 之间的关系。

【任务 1】 项目知识点学习

3) 波特率设置寄存器

波特率寄存器 UBRRL、UBRRH 格式

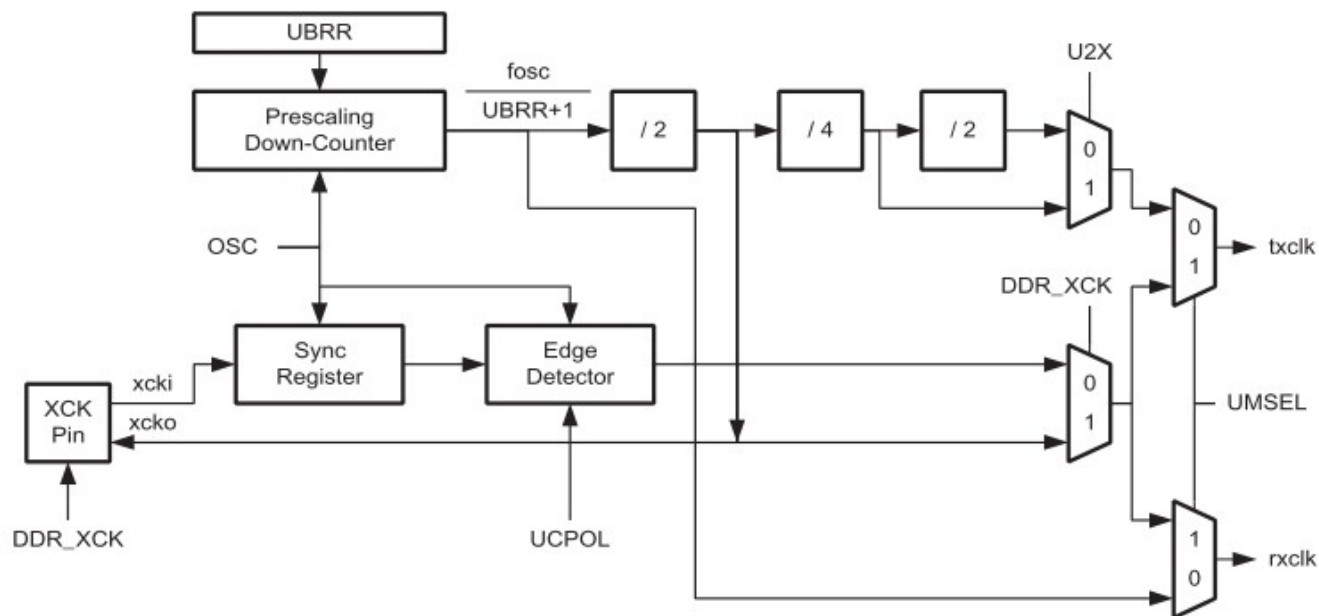
位	15	14	13	12	11	10	9	8				
	URSEL	—	—	—	UBRR 11:8				UBRRH			
	UBRR 7:0											UBRRL
读 / 写	R/W	R	R	R	R/W	R/W	R/W	R/W				
初始值	0	0	0	0	0	0	0	0				

URSEL: 寄存器选择通过该位选择访问 UCSRC 寄存器或 UBRRH 寄存器。当读 UBRRH 时, 该位为 0 ; 当写 UBRRH 时, URSEL 为 0。

- ❖ Bit 14:12 – 保留位 这些位是为以后的使用而保留的。为了与以后的器件兼容, 写 UBRRH 时将这些位清零。
- ❖ Bit 11:0 – UBRR11:0: USART 波特率寄存器. 这个 12 位的寄存器包含了 USART 的波特率信息。其中 UBRRH 包含了 USART 波特率高 4 位, UBRRL 包含了低 8 位。波特率的改变将造成正在进行的数据传输受到破坏。写 UBRRL 将立即更新波特率分频器。

【任务 1】 项目知识点学习

4) 时钟的产生 - 波特率发生器



❖ 信号说明：

❖ **txclk** 发送器时钟（内部信号） **rxclk** 接收器基础时钟（内部信号）

❖ **xcki** XCK 引脚输入（内部信号），用于同步从机操作；

❖ **xcko** 输出到 XCK 引脚的时钟（内部信号），用于同步主机操作；

❖ **fosc** XTAL 频率（系统时钟）

【任务 1】 项目知识点学习

- ❖ 图 1.13 给出了计算波特率 (位 / 秒) 以及计算每一种使用内部时钟源工作模式的 UBRR 值的公式

使用模式	波特率的计算公式 ⁽¹⁾	UBRR 值的计算公式
异步正常模式 (U2X = 0)	$BAUD = \frac{f_{OSC}}{16(UBRR + 1)}$	$UBRR = \frac{f_{OSC}}{16BAUD} - 1$
异步倍速模式 (U2X = 1)	$BAUD = \frac{f_{OSC}}{8(UBRR + 1)}$	$UBRR = \frac{f_{OSC}}{8BAUD} - 1$
同步主机模式	$BAUD = \frac{f_{OSC}}{2(UBRR + 1)}$	$UBRR = \frac{f_{OSC}}{2BAUD} - 1$

说明

BAUD **波特率 (使用单位 bit/s)**
fOSC **系统时钟频率**
UBRR **UBRRH 与 UBRL 的数值 (0-4095)**

【任务 2】 双单片机间通信

❖ 一、任务要求

- 利用 ATmega16 单片机内部的 USART 串行通信功能，实现双单片机系统之间的数据通信。通过系统 A 可以控制系统 B 的秒表启停，如果系统 B 的秒表超过 60s，系统 B 发送给系统报警指示信号，系统 A 发出灯光指示。

【任务 2】 双单片机间通信

二、硬件设计

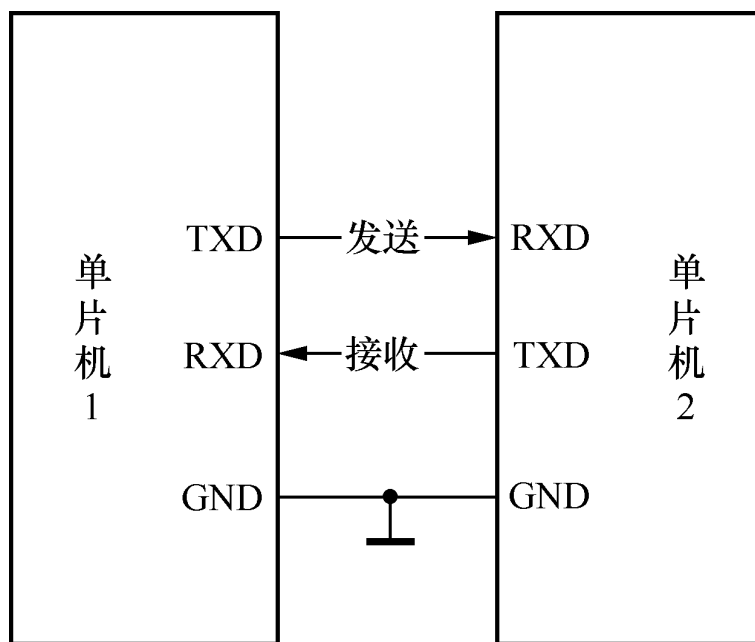


图 10-8 硬件连线示意图

【任务 2】 双单片机间通信

三、程序设计

系统 A、B 通信程序的设计思路都是采用中断方式进行程序设计的，首先对系统 A、B 串行通信模块 USART 相应寄存器进行设置，通过寄存器 UBRRL、UBRRH 设置波特率为 9600。

在系统 A 中定义全局变量 start（0x00 表示停止，0x01 表示启动），通过外部中断 0 进行启动、停止的切换，进入一次外部中断 0 切换一次，并通过把 start 赋值给数据缓冲器 UDR，实现系统 A 给系统 B 自动发送启动 / 停止指令。

在系统 B 中，设置串口数据接收中断函数功能，系统 B 接收完数据后自动进入中断程序，读取 UDR 的数值并判断是启动还是停止指令。

【任务 2】 双单片机间通信

四、项目实施

1. 根据元器件清单选择合适的元器件。
2. 根据硬件设计原理图，在万能电路板进行元器件布局，并进行系统 A、B 双系统的焊接工作。
3. 焊接完成后，重复进行线路检查，防止短路、虚接现象。
4. 用 AVR Studio 软件分别创建项目 A、B，输入源代码并生成各自的 *.hex 文件。
5. 用线缆将焊接好的系统 A、B 双系统的串行接口进行硬件连接（TXD、RXD 交叉连接）。
6. 在确认硬件电路正确的前提下，通过 JTAG 仿真器进行程序的下载与硬件在线调试。

【任务 3】 多单片机间通信

❖ 一、任务要求

- 利用 ATmega16 单片机内部的 USART 串行通信功能，实现对 3 个单片机系统之间的数据通信。通过主控制系统 A 可以控制从机系统 B、C 秒表的启停，并在主控制系统 A 上显示从机系统是否启动成功。

【任务 3】 多单片机间通信

二、硬件设计

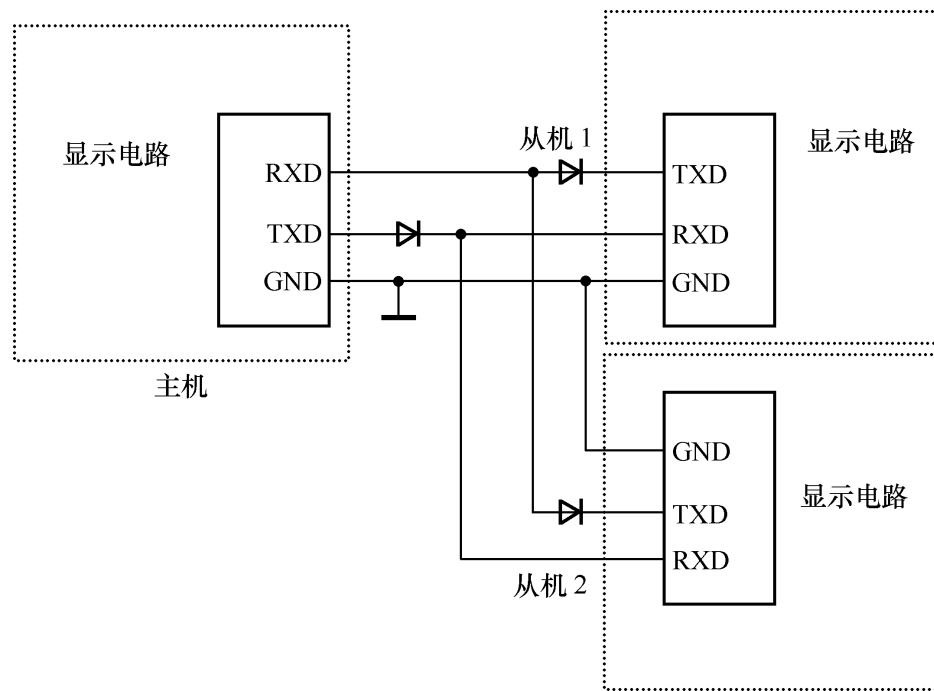


图 10-9 硬件连线示意图

【任务 3】 多单片机间通信

三、程序设计

系统 A、B、C 通信程序设计思路都是采用中断方式进行程序设计的，首先对系统 A、B、C 串行通信模块 USART 相应寄存器进行设置，通过寄存器 UBRRL、UBRRH 设置波特率为 9600。同时，在多机通信过程中需要特别注意多处理器通信模式的设定。

置位 UCSRA 的多处理器通信模式位（MPCM）可以对 USART 接收器接收到的数据帧进行过滤。那些没有地址信息的数据帧将被忽略，也不会存入接收缓冲器。如果接收器所接收的数据帧长度为 9 位，那么由第 9 位（RXB8）来确定是数据还是地址信息。如果确定帧类型的位（第 1 个停止位或第 9 个数据位）为“1”，那么这是地址帧，否则为数据帧。

【任务 3】 多单片机间通信

三、程序设计

下面即为在多处理器通信模式下进行数据交换的步骤。

- ① 所有从处理器都工作在多处理器通信模式（**UCSRA** 寄存器的 **MPCM** 置位）。
- ② 主处理器发送地址帧后，所有从处理器都会接收并读取此帧。从处理器 **UCSRA** 寄存器的 **RXC** 正常置位。
- ③ 每一个从处理器都会读取 **UDR** 寄存器的内容，以确定自己是否被选中。如果选中，就清零 **UCSRA** 的 **MPCM** 位，否则它将等待下一个地址字节的到来，并保持 **MPCM** 为“1”。
- ④ 被寻址的从处理器将接收所有的数据帧，直到收到一个新的地址帧。而那些保持 **MPCM** 位为“1”的从处理器将忽略这些数据。
- ⑤ 被寻址的处理器接收到最后一个数据帧后，它将置位 **MPCM**，并等待主处理器发送下一个地址帧。然后重复进行第 2 步之后的步骤。

【任务 3】 多单片机间通信

四、项目实施

1. 根据元器件清单选择合适的元器件。
2. 根据硬件设计原理图，在万能电路板进行元器件布局，并进行系统 A、B、C 三系统的焊接工作。
3. 焊接完成后，重复进行线路检查，防止短路、虚接现象。
4. 用 AVR Studio 软件分别创建项目 A、B、C，输入源代码并生成各自的 *.hex 文件。
5. 用线缆将焊接好的系统 A、B、C 系统的串行接口进行硬件连接。
6. 在确认硬件电路正确的前提下，通过 JTAG 仿真器进行程序的下载与硬件在线调试。

【任务 4】 PC 机与单片机串口通信

❖ 一、任务要求

- 利用 ATmega16 单片机内部的 USART 串行通信功能，实现对单片机系统与 PC 机之间的数据通信。单片机接收 PC 机传输过来的数据，将接收到的数据通过小灯显示出来。

【任务 3】 多单片机间通信

二、硬件设计

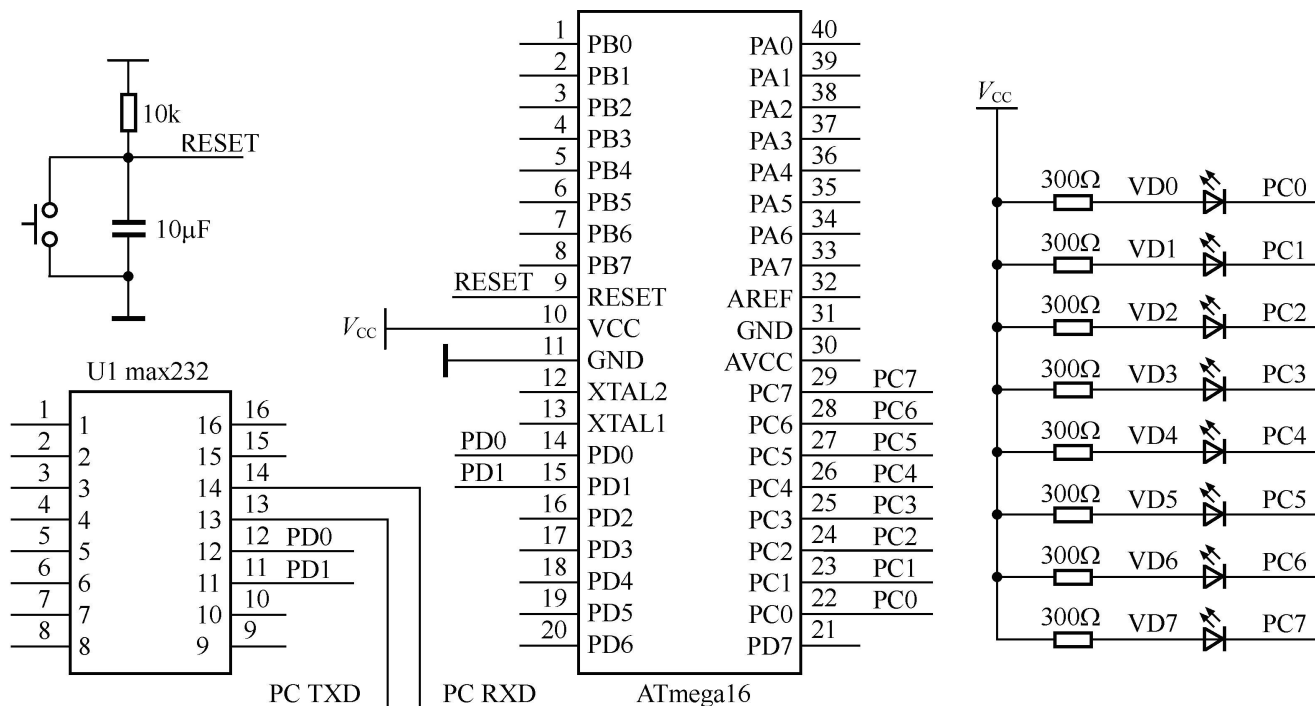


图 10-10 单片机与 PC 机之间的异步串行通信电路原理图

【任务 3】 多单片机间通信

三、程序设计

1. RS-232C 异步串行通信接口的概述

RS-232C 是使用最早、应用最多的一种异步串行通信总线标准。RS-232 主要用来定义计算机系统的一些数据终端设备（DTE）和数据电路终端设备（DCE）之间的电器性能。

RS-232C 串行接口总线适用于：设备之间的通信距离不大于 15m，传输速率最大为 20kbit/s。

【任务 3】 多单片机间通信

三、程序设计

2. RS-232C 异步串行通信信息格式标准

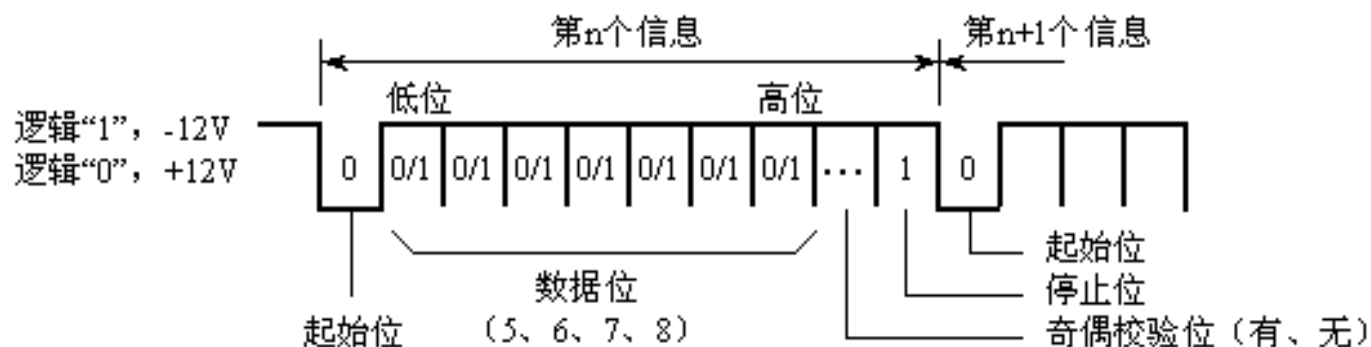


图 10-11 RS-232C 信息格式

【任务 3】 多单片机间通信

三、程序设计

2. RS-232C 异步串行通信信息格式标准

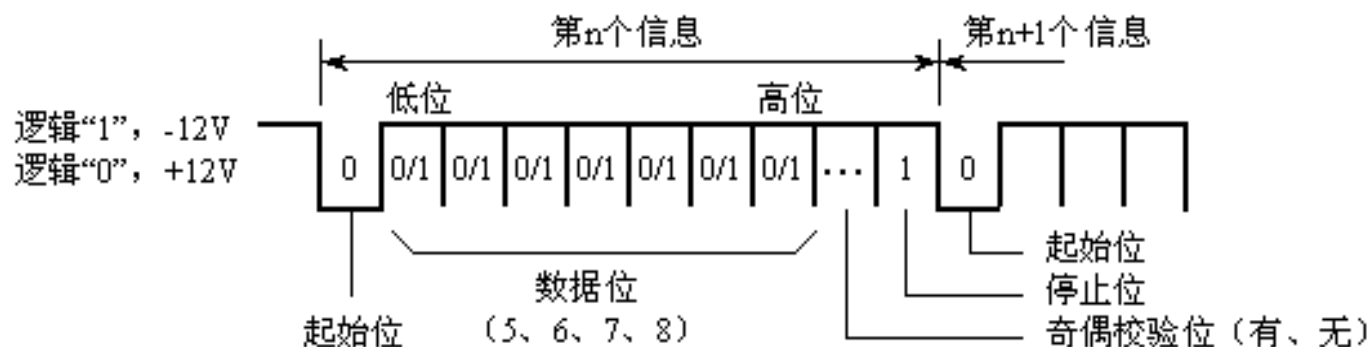


图 10-11 RS-232C 信息格式

【任务 3】 多单片机间通信

三、程序设计

3. RS-232C 总线规定

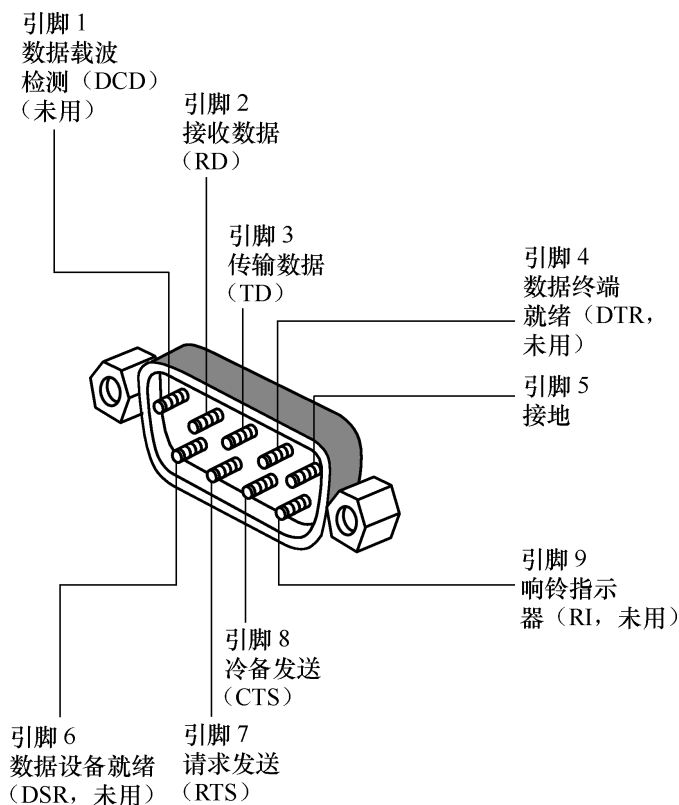
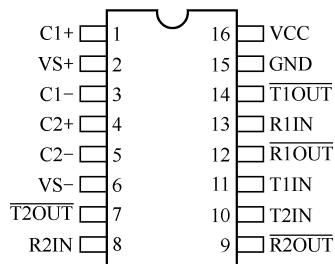


图 10-12 9 芯插头座引脚图

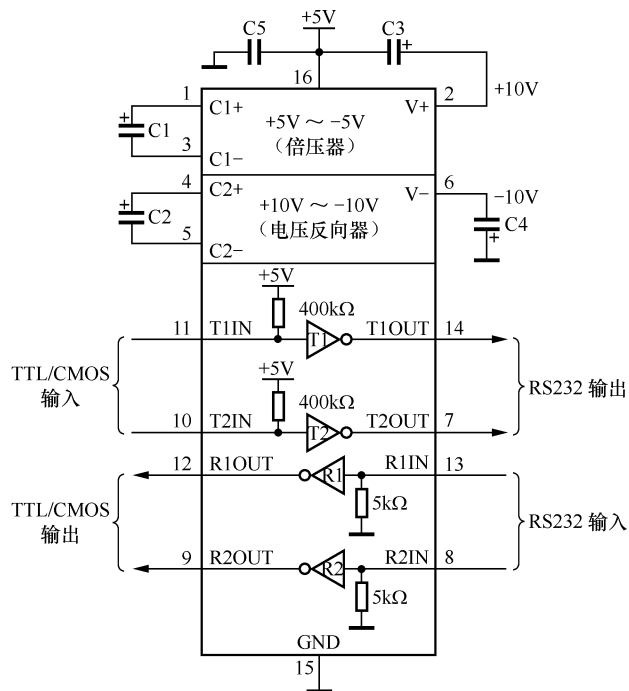
【任务 3】 多单片机间通信

三、程序设计

4. RS-232C 异步串行通信电平转换器



(a) MAX232 引脚图



(b) MAX232 结构原理图

图 10-13 MAX232 引脚图及结构原理图

【任务 3】 多单片机间通信

三、程序设计

5. 串口调试助手软件的功能及简单使用方法

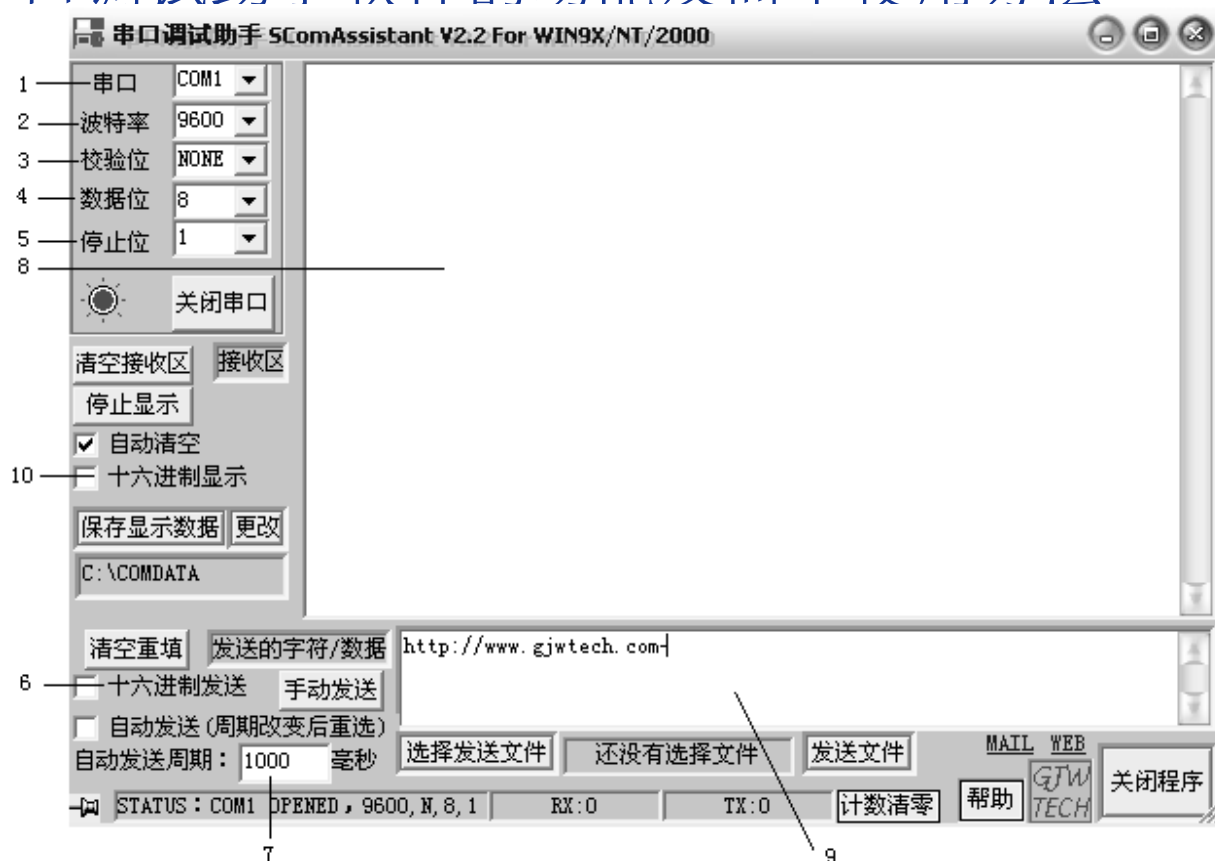


图 10-14 串口调试助手界面

【任务 3】 多单片机间通信

三、程序设计

6. 程序流程图

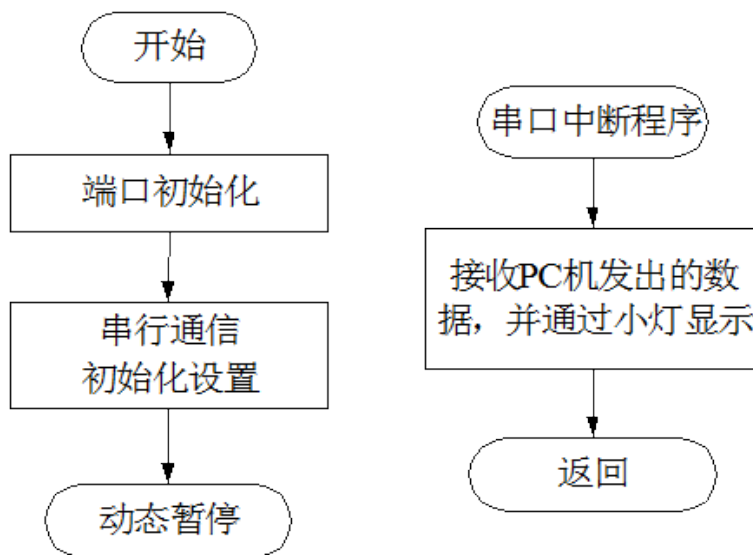


图 10-15 单片机与 PC 机之间的异步串行通信程序流程图

【任务 3】 多单片机间通信

四、项目实施

1. 根据元器件清单选择合适的元器件。
2. 根据硬件设计原理图，在万能电路板进行元器件布局，并进行系统的焊接工作。
3. 焊接完成后，重复进行线路检查，防止短路、虚接现象。
4. 用 **AVR Studio** 软件分别创建项目，输入源代码并生成 *.hex 文件。
5. 用串行通信线将焊接好的系统与 **PC** 机的串行接口进行硬件连接。
6. 在确认硬件电路正确的前提下，通过 **JTAG** 仿真器进行程序的下载与硬件在线调试。
7. 打开串行助手软件，**PC** 与单片机进行串行通信调试。