

第10章 与外界的沟通

——串行口通信

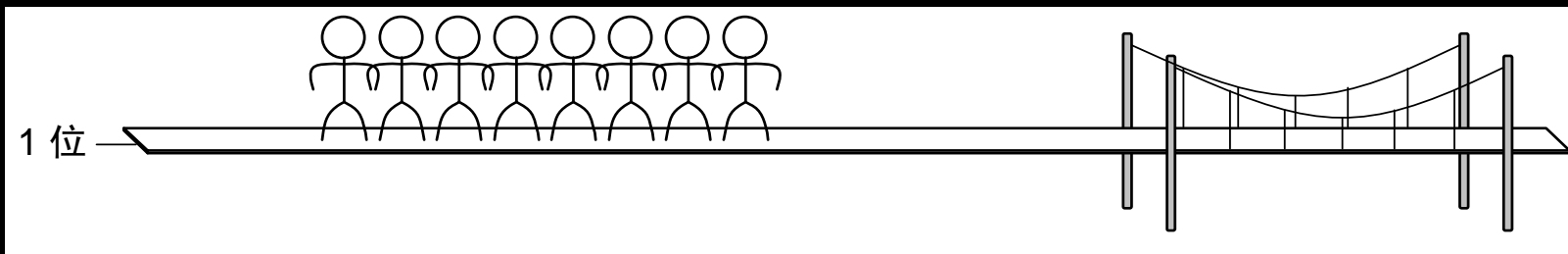
欢迎访问 电路飞翔网

<http://www.circuitfly.com> 获取更多信息

- 10.1 串行与并行通信
- 10.2 单片机串行口如何发送与接收数据
- 10.3 单片机串行口的控制者
- 10.4 串行口工作模式及波特率
- 10.5 单片机之间的通信
- 10.6 计算机的串行口与控制
- 10.7 单片机与计算机之间的通信
- 10.8 实例点拨——智能交通灯

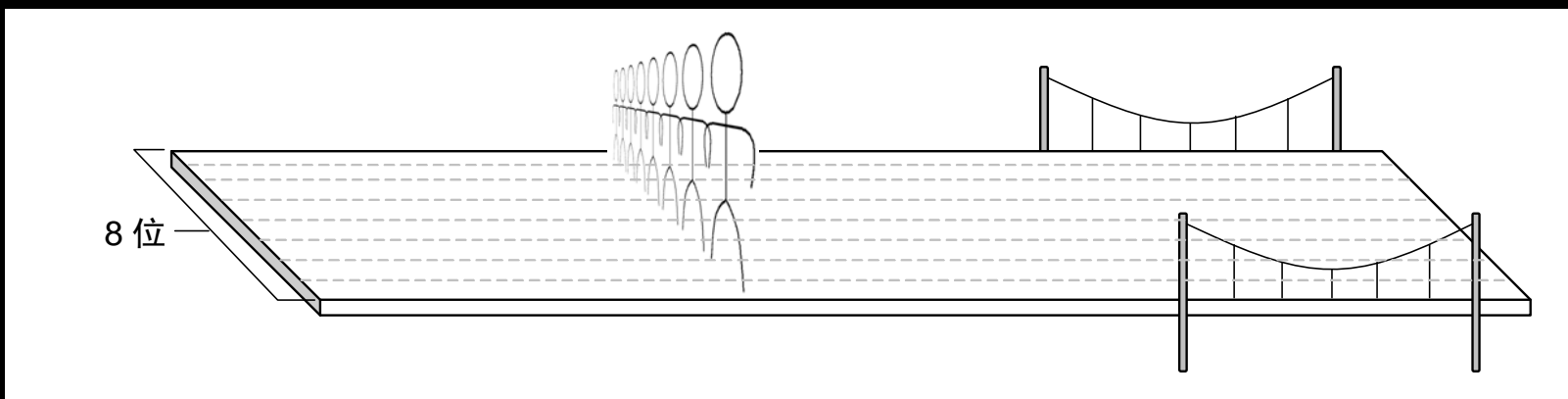
10.1 串行与并行通信

• 10.1.1 串行与并行通信的比喻



依次通过--串行通信--尽量少的数据线

串行方式：
一切USB
设备

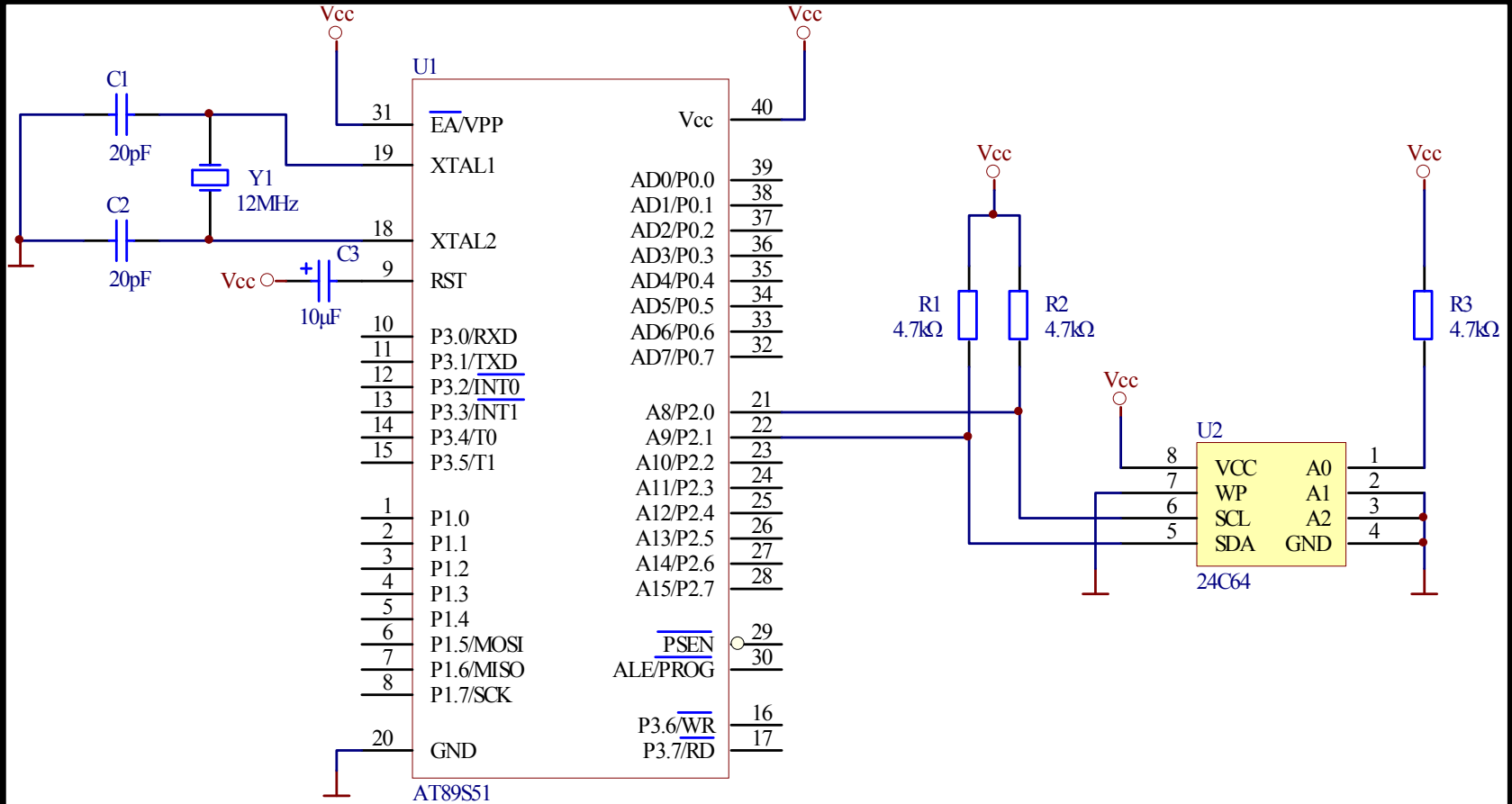


并排通过 --并行通信--更快的传输速度

并行方式：硬
盘；

10.1 串行与并行通信

• 10.1.2 串行与并行存储器



串行存储器的接口

串行存储器24C64只占用单片机的两个I/O口作为通信线

• 10.1.2 串行与并行存储器



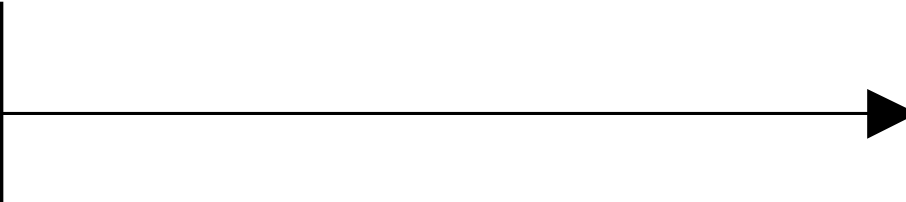
器作为地址锁存器

10.1 串行与并行通信

- 10.1.3 单工、半双工和全双工

在单工（**simplex**）下，发送方与接收方之间只有一条通信链路，数据只能从发送方向接收方传送，如广播电台和电视台发射的节目信号，只需要由接收方（收音机、电视机）接收。

发送方

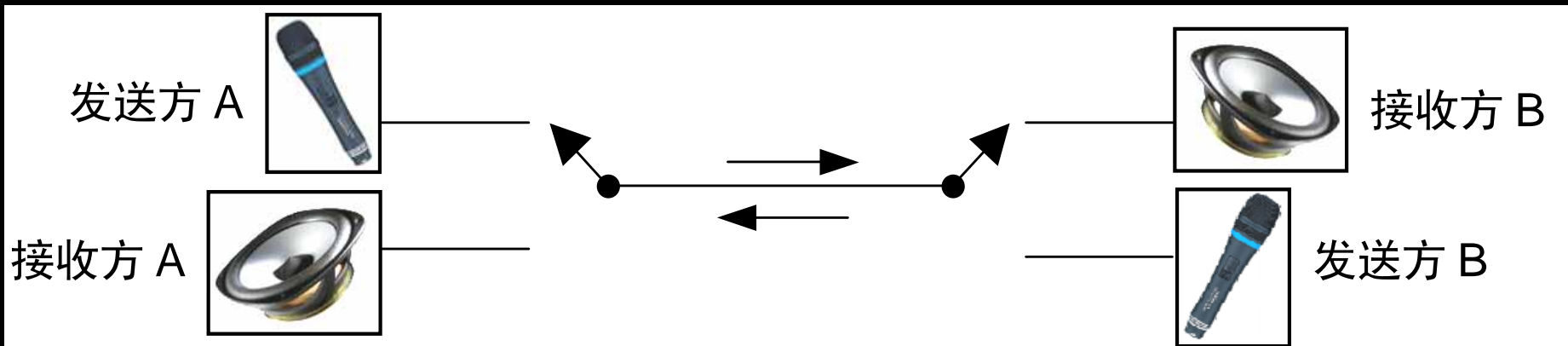


接收方

10.1 串行与并行通信

• 10.1.3 单工、半双工和全双工

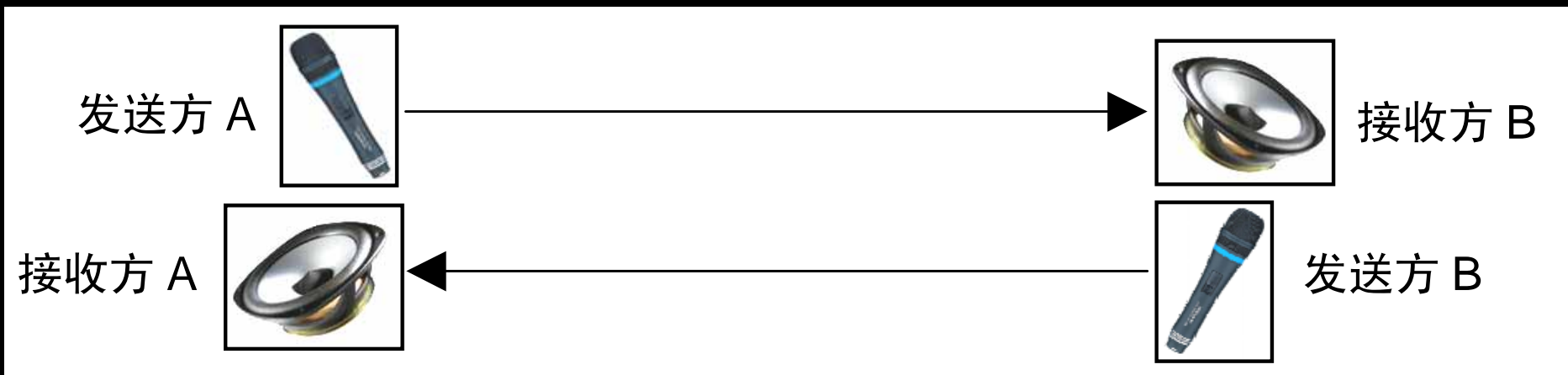
在半双工（**half duplex**）下，双方也只有一条通信链路，通过两个开关控制由哪一方发送，哪一方接收，A、B双方不能同时进行发送与接收。



10.1 串行与并行通信

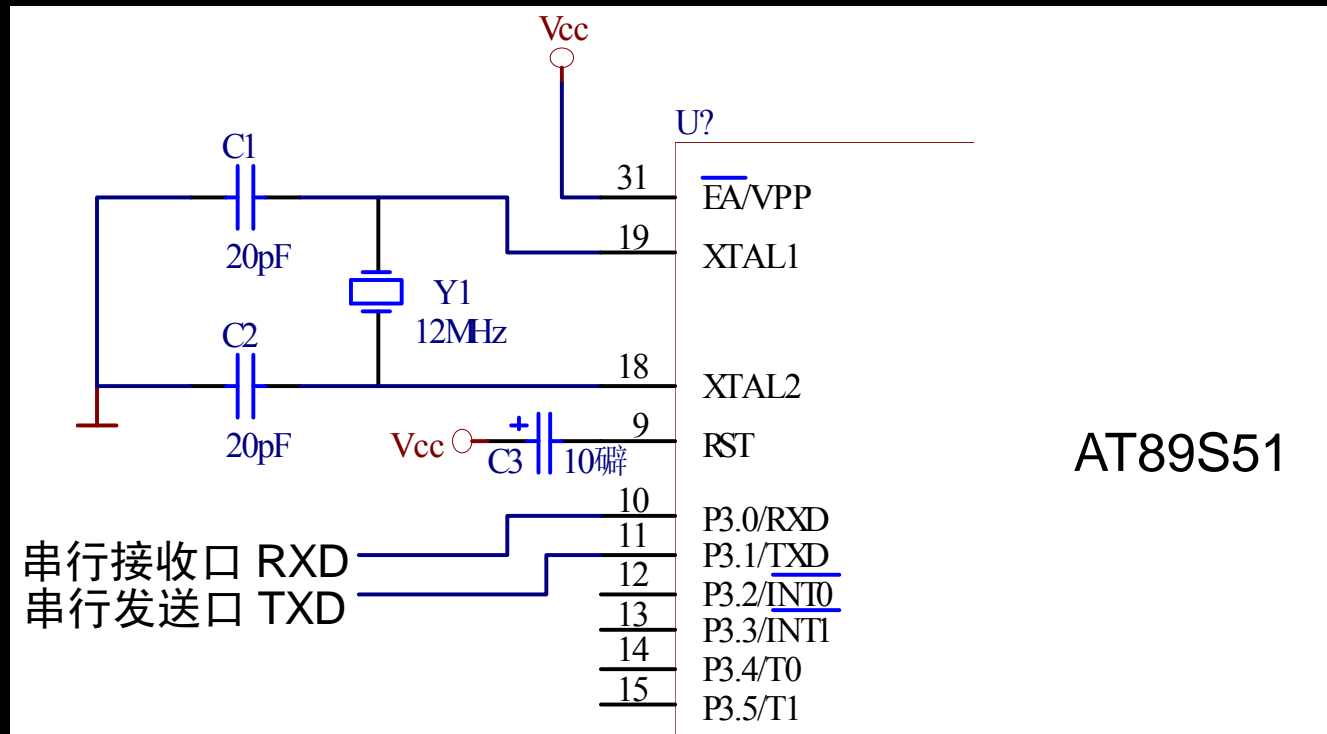
- 10.1.3 单工、半双工和全双工

在全双工（**duplex**）下，双方可以同时进行发送和接收，但它需要两条通信链路。



10.2 单片机串行口如何发送与接收数据

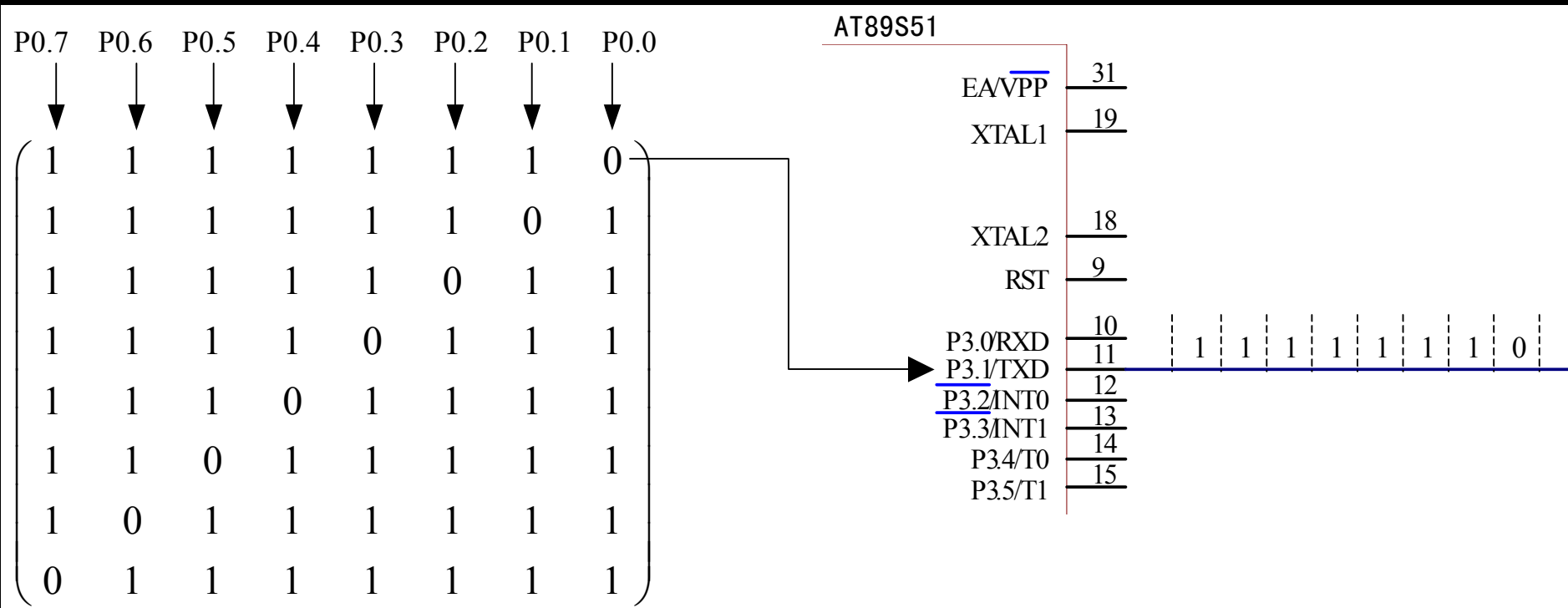
- AT89S51单片机的P3.0和P3.1除作为一般I/O口外，还分别在串行通信中充当接收口（RXD）和发送口（TXD）。



AT89S51单片机的串行口

10.2 单片机串行口如何发送与接收数据

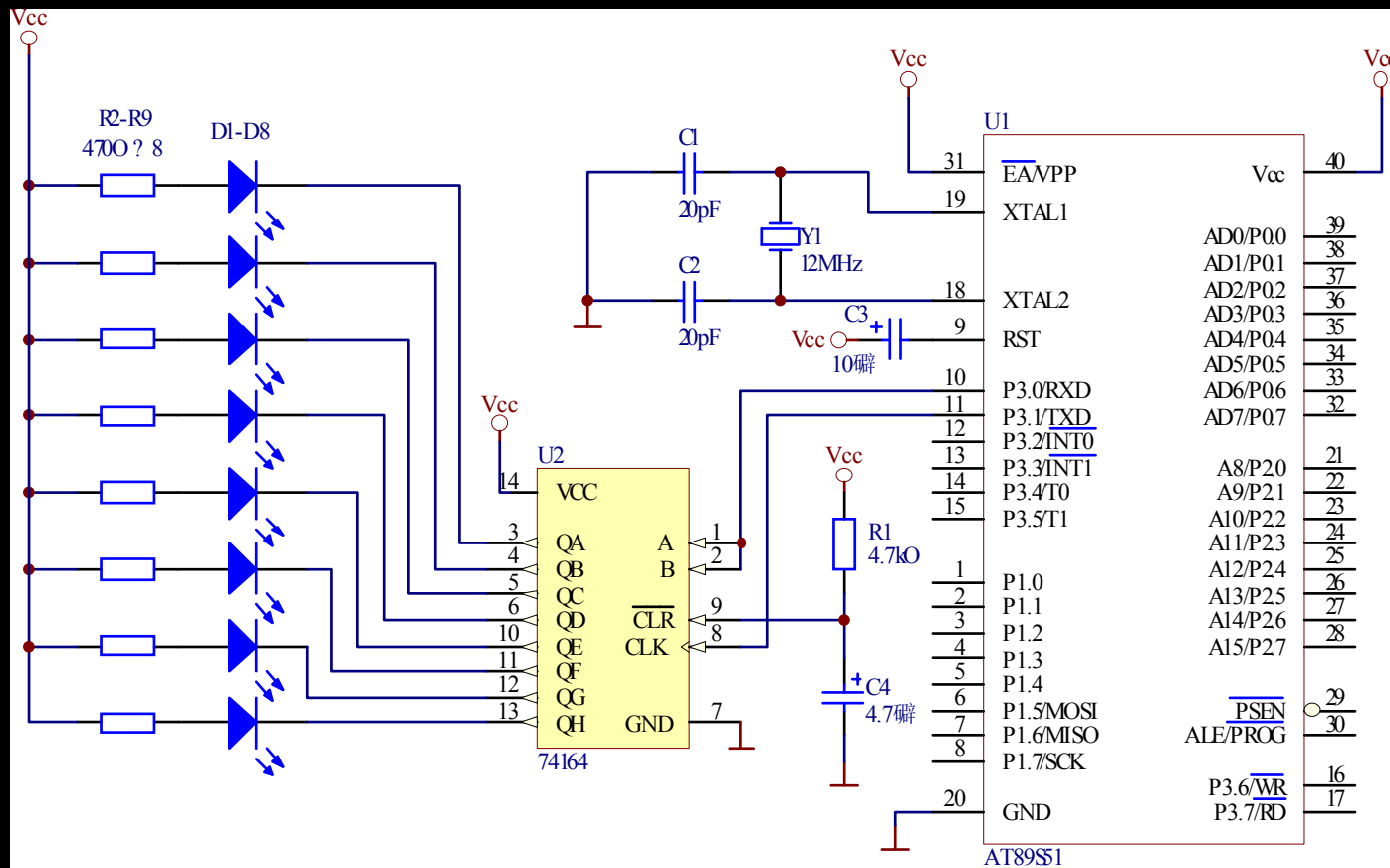
• 10.2.1 串行发送数据



从并行数据到串行数据

10.2 单片机串行口如何发送与接收数据

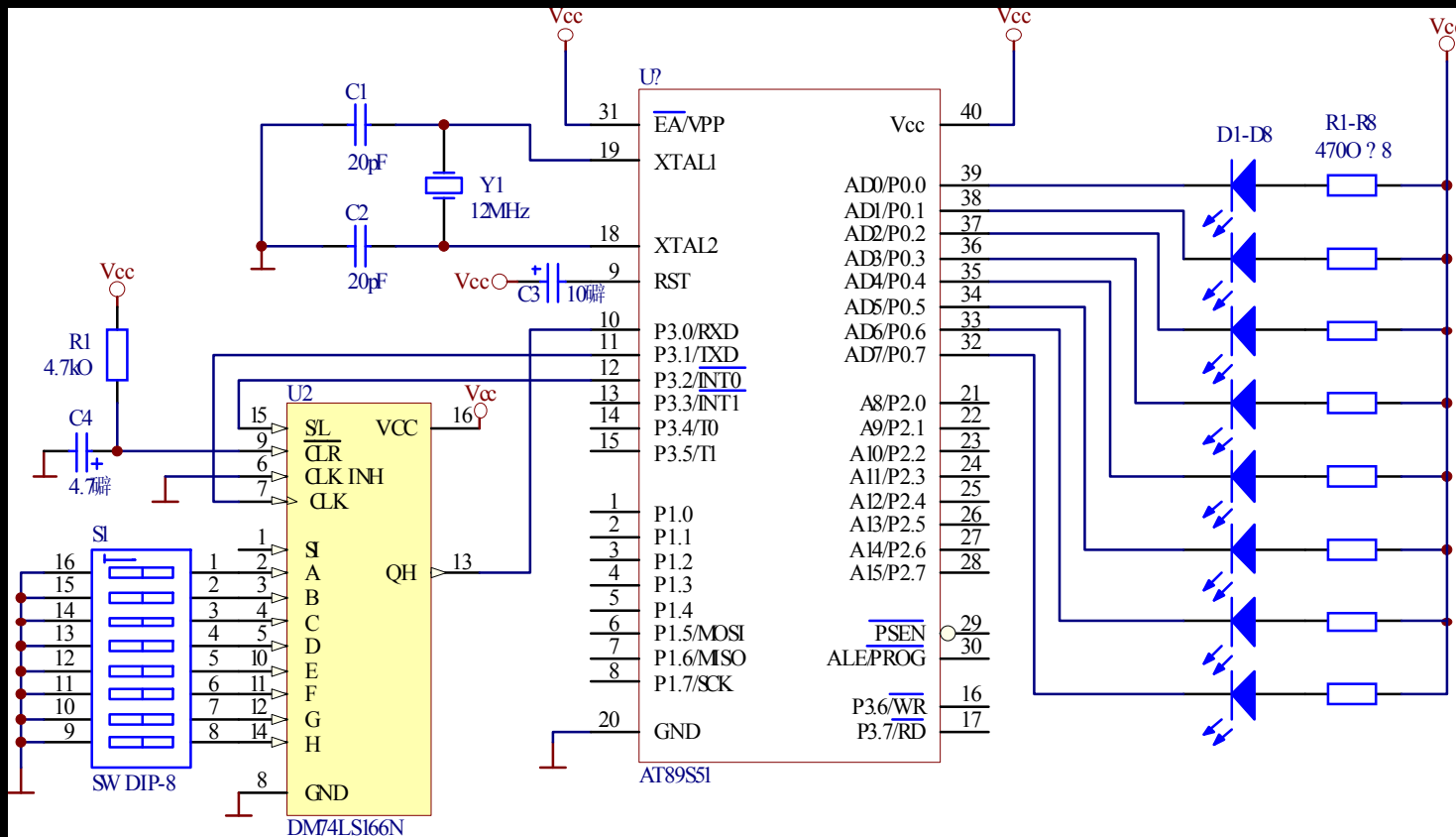
• 10.2.1 串行发送数据



单片机串行口构成的流水灯系统

10.2 单片机串行口如何发送与接收数据

• 10.2.2 串行接收数据



单片机串行口用于接收并行数据系统

10.3 单片机串行口的控制者

• 10.3.1 UART与波特率

- ✓ 在串行数据传输时，分为异步（asynchronous）和同步（synchronous）两种方式。AT89S51单片机的串行口采用的是异步传输方式，每次串行口通信时只发送或接收1个字节的数据。
- ✓ AT89S51单片机集成了一个叫做“通用异步发送-接收器”的功能结构，英文缩写是UART（universal asynchronous receiver-transmitter），我们使用SCON寄存器控制对象就是单片机的UART。
- ✓ 描述数据传输率：“字节/秒”（Bytes per second），“位/秒”（bits per second），波特（baud）。

10.3 单片机串行口的控制者

• 10.3.2 串行口缓冲区SBUF

- ✓ 串行口缓冲区**SBUF**，是一个1个字节长度的寄存器，位于特殊功能寄存器区的**99H**上。
- ✓ 在发送数据时，只要把数据载入**SBUF**中，**UART**就会自动地将数据从串行口发送出去。在接收数据时，**SBUF**寄存器保持着从单片机串行口接收的数据，以供程序读取。

往**SBUF**中载入数据就自动发送；
接收的数据保存在**SBUF**中供读取。

10.3 单片机串行口的控制者

• 10.3.3 串行口控制寄存器SCON

串行口控制寄存器SCON是一个1个字节长度的寄存器，位于特殊功能寄存器区的98H，可对其进行位寻址。

7	6	5	4	3	2	1	0
FE/SM0	SM1	SM2	REN	TB8	RB8	TI	RI

FE/SM0 误帧位 / 串行口模式0位。

SM1 串行口模式1位。

SM2 多单片机通信使能位。

REN 接收使能位。

TB8 不常用

RB8 不常用

TI 发送中断标志位。

RI 接收中断标志位。

10.3 单片机串行口的控制者

• 10.3.3 串行口控制寄存器SCON

- ✓ **FE/SM0位与SM1位**——设置串行口工作模式(见后)。
- ✓ **SM2**——多单片机通信使能位。该位可以控制使能多个单片机之间的通信，。
- ✓ **REN**——接收使能位。当**REN**位置1时，单片机可以从串行口接收数据，如果**REN**清0，则接收功能被关闭。
- ✓ **TI**——发送中断标志位。**TI**与**RI**一样都是串行通信中重要的标志位。当单片机完成**SBUF**中的数据发送后，该位由硬件置1。
- ✓ **RI**——接收中断标志位。粗略的描述是当单片机从串行口接收完数据后该位被硬件置1。

10.4 串行口工作模式及波特率

- 由SCON寄存器中的SM0和SM1位设置单片机串行口工作在4种不同模式下，如表：

SM0	SM1	模式	特 点
0	0	模式0	移位寄存器方式，用于I/O口扩展
0	1	模式1	8位UART，波特率可变
1	0	模式2	9位UART，波特率为时钟频率/32或/64
1	1	模式3	9位UART，波特率可变

10.4 串行口工作模式及波特率

• 10.4.1 模式0

UART实际上是一个同步移位寄存器，该模式下只发送或接收8位数据。注意，此时数据由单片机的RXD管脚发送或接收，而TXD作为发送或接收的移位脉冲。发送或接收数据时，低位数据在前，高位数据在后。

- ① 模式0下发送。将要发送的8位数据载入SBUF，UART自动将SBUF中的数据转换成8位串行数据，并以晶振频率/12作为波特率从RXD发送出去。当数据发送完成后，TI会被置1。
- ② 模式0下接收。在UART接收串行数据之前需要设SCON中的RI=0、REN=1。UART就会启动接收过程，此时RXD为数据接收端，TXD仍然为移位脉冲输出。当1个字节的数据接收完毕后，UART将其载入SBUF，同时RI被置1。

$$\text{波特率} = \frac{\text{晶振频率}}{12} \quad \text{例如晶振频率}=12\text{MHz, 则波特率} = \frac{12 \times 10^6}{12} = 1 \text{ (MHz)}$$

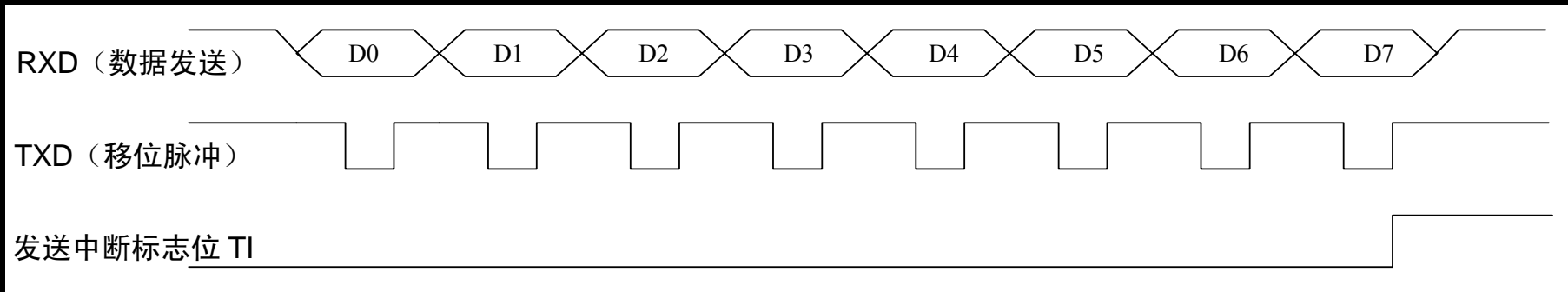
10.4 串行口工作模式及波特率

- 10.4.2 模式1

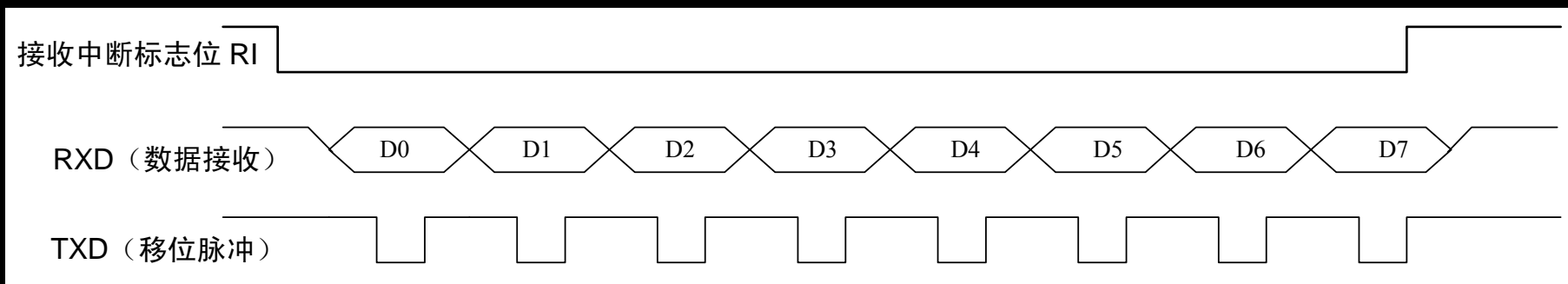
该模式下，UART作为异步通信口，每一帧发送或接收10位数据，这10个位分别是1个起始位“0”、8个数据位和1个停止位“1”。单片机的TXD为发送管脚，RXD为接收管脚。

10.4 串行口工作模式及波特率

• 10.4.2 模式1（模式1下收发时序）



模式1下发送时序



模式1下接收时序

10.4 串行口工作模式及波特率

- 10.4.2 模式1（模式1下的波特率）

该模式下通信的波特率是可变的，一般由Timer 1工作在模式2下，通过载入TH1和TL1的计数初始值来设置波特率。Timer工作在模式2下，是一个8位自动重新装载的定时器，装载时需要向TH1、TL1同时装载相同的计数初始值。单片机会自动根据Timer 1的设置情况使UART工作在特定的波特率下。

10.4 串行口工作模式及波特率

• 10.4.2 模式1（模式1下的波特率）

模式1下波特率和TH1（TL1）中载入计数初始值之间的关系如公式所示：

$$\text{波特率} = \frac{2^{\text{SMOD1}}}{32} \times \frac{\text{晶振频率}}{12 \times [256 - (\text{TH1})]}$$

SMOD1为PCON.7 {
 =0为单倍波特率
 =1为双倍波特率

TH1是Timer1寄存器。

7	6	5	4	3	2	1	0
SMOD1	SMOD0	保留	POF	GF1	GF0	PD	IDL

SMOD1在PCON中的位

10.4 串行口工作模式及波特率

• 10.4.2 模式1（模式1下的波特率实例）

例：使用单倍波特率，SMOD1=0，晶振频率=11.0592MHz，向TH1中载入F3H，即TH1=243，则波特率为：

$$\text{波特率} = \frac{2^{\text{SMOD1}}}{32} \times \frac{\text{晶振频率}}{12 \times [256 - (\text{TH1})]} = \frac{2^0}{32} \times \frac{11.0592 \times 10^6}{12 \times (256 - 243)} = 2400$$

① UART以模式1发送时，数据由TXD端送出。向SBUF载入数据后即自动发送。发送完一帧数据后，发送中断标志位TI=1，可用“JBC TI, CHECK”来检测TI位并将其清除为0。

② UART接收串行数据之前需将SCON中的REN位置1，UART就会启动接收过程。RXD为数据接收端。接收完1帧数据后，RI位被置1，可用“JB RI, CHECK”来检测RI位以判断接收完成情况。

10.4 串行口工作模式及波特率

- 10.4.2 模式1（模式1和模式3下波特率与TH1（=TL1）的计数初始值）

波特率	晶振频率 (MHz)	SMOD1 位	Timer 1工作在模式2下载入 TH1（=TL1）的计数初始值
600	12	0	0CCH
1200	12	1	0F6H
2400	12	0	0F3H
4800	12	1	0F3H
1200	11.0592	0	0E8H
2400	11.0592	0	0F4H
4800	11.0592	0	0FAH
9600	11.0592	0	0FDH
19200	11.0592	1	0FDH

10.4 串行口工作模式及波特率

- 10.4.3 模式2

该模式下UART为一个9位异步通信口，每一帧共发送或接收11位数据。这11位数据由1个起始位“0”、8个数据、第9位数据（TB8位，位于SCON内）和1个停止位“1”组成。模式2下的波特率为晶振频率的1/32或1/64，这取决于PCON中的SMOD1的设置，计算公式为：

$$\text{波特率} = \frac{2^{\text{SMOD1}}}{64} \times \text{晶振频率}$$

10.4 串行口工作模式及波特率

• 10.4.3 模式2

① UART以模式2发送时，数据由TXD端送出，数据中的第9位数据从SCON中的TB8位取得（可用指令“MOV TB8, bit”向TB8位载入数据）。向SBUF载入数据后即自动发送。发送完一帧数据后，发送中断标志位TI=1，可用指令“JBC TI, CHECK”来检测TI并将其清除为0。

② 在UART接收串行数据之前需要设SCON中的REN=1。UART就会启动接收过程。RXD为数据接收端，数据中的第9位载入SCON中的RB8位上。接收完1帧数据后，RI位被置1，可用指令“JB RI, CHECK”来检测RI位以判断接收完成情况。

10.4 串行口工作模式及波特率

• 10.4.4 模式3

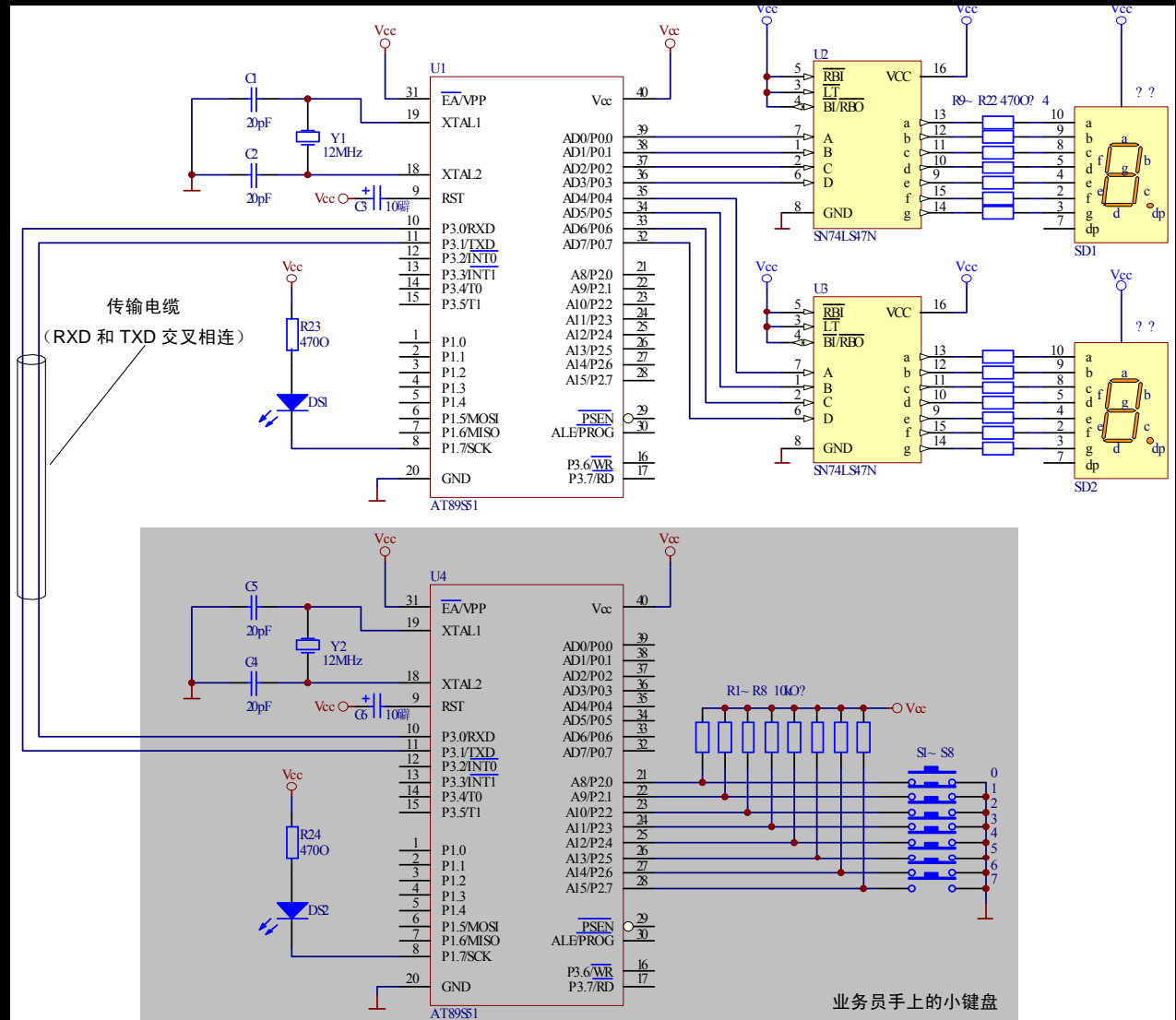
该模式与模式2的发送/接收过程和UART功能几乎完全一样，只是模式3的波特率可变，与模式1具有相同的计算公式，在实际应用中，可以参考下表来选择晶振和设置计数初始值。

波特率	晶振频率 (MHz)	SMOD1位	Timer 1工作在模式2下载入TH1 (=TL1) 的计数 初始值
600	12	0	0CCH
1200	12	1	0F6H
2400	12	0	0F3H
4800	12	1	0F3H
1200	11.0592	0	0E8H
2400	11.0592	0	0F4H
4800	11.0592	0	0FAH
9600	11.0592	0	0FDH
19200	11.0592	1	0FDH

10.5 单片机之间的通信

• 10.5.1 排队 呼叫器 ——两个单片机 之间的通信

排队呼叫器 系统电路



10.5 单片机之间的通信

• 10.5.1 排队呼叫器——两个单片机之间的通信

- ✓ 两个单片机的RXD和TXD是交叉连接的。
- ✓ 两个单片机的串行口直接相连时有一个问题，就是它们之间的传输电缆不能太长（应当在几米范围内），否则在传输过程中会因干扰和信号的损耗而产生误码。
- ✓ 确定通信协议：

单片机U1和U4之间以2400的波特率进行串行口通信。

U4首先向U1发起通信请求，发送呼叫号99H。

U1接收到呼叫号后，向U4回送应答号01H。

通信建立，开始数据传输。

10.5 单片机之间的通信

- 两个单片机之间的通信程序

U4向U1发送显示数据 (1/2)

ORG 00H

START:

MOV SCON,#50H; ①设置串行口工作在模式1下, 接收使能

MOV TMOD, #20H ; ②使用Timer 1工作在模式2下

MOV TH1, #0F3H ; ③波特率2400

MOV TL1, #0F3H

SETB TR1 ; ④启动Timer 1

DIAL:

MOV SBUF, #99H ; ⑤发送呼叫号99H

CHECK_99:

JBC TI, WAIT_RESPONSE ; ⑥判断是否发送完毕, 完毕跳
转到接收应答

JMP CHECK_99

10.5 单片机之间的通信

- 两个单片机之间的通信程序

U4向U1发送显示数据 (2/2)

WAIT_RESPONSE:

JBC RI, CHECK_01 ; 串行口接收到数据则跳转到判断应答号

JMP WAIT_RESPONSE

CHECK_01:

MOV A, SBUF ; 将接收到的数据从SBUF载入ACC

CJNE A, #01H, DIAL ; 判断接收到的数据是否为应答号01H,
; 如果不是正确应答号, 就跳到DIAL重呼

SEND_NUM:

MOV A, 30H ; 将30H中的键盘按键值载入ACC

MOV SBUF, A ; 发送键盘按键值

CHECK_SEND:

JBC TI, FINISH ; 判断是否发送完毕

JMP CHECK_SEND

FINISH:.....

END

10.5 单片机之间的通信

- 两个单片机之间的通信程序

U1接收U4的显示数据 (1/2)

ORG 00H

START:

MOV SCON, #50H ; ①设置串行口工作在模式1下, 接收使能

MOV TMOD, #20H ; ②使用Timer 1工作在模式2下

MOV TH1, #0F3H ; ③波特率2400

MOV TL1, #0F3H

SETB TR1 ; ④启动Timer 1

WAIT_DIAL:

JBC RI, CHECK_99 ; ⑤判断是否接收到呼叫

JMP WAIT_DIAL

CHECK_99:

MOV A, SBUF ; ⑥接收到的数据载入ACC

CJNE A, #99H, WAIT_DIAL ; 判断是否为呼叫号99H

10.5 单片机之间的通信

- 两个单片机之间的通信程序

U1接收U4的显示数据 (2/2)

SEND_RESPONSE:

MOV SBUF, #01H ; 发送应答号01H

CHECK_01:

JBC TI, RECEIVE_NUM ; 判断应答号是否发送完毕

JMP CHECK_01

RECEIVE_NUM:

JBC RI, DISPLAY ; 等待接收显示数据

JMP RECEIVE_NUM

DISPLAY:

MOV A, SBUF ; 将接收到的显示数据载入ACC中

MOV P0, A ; 输出至P0口显示

CALL DELAY ; 延时

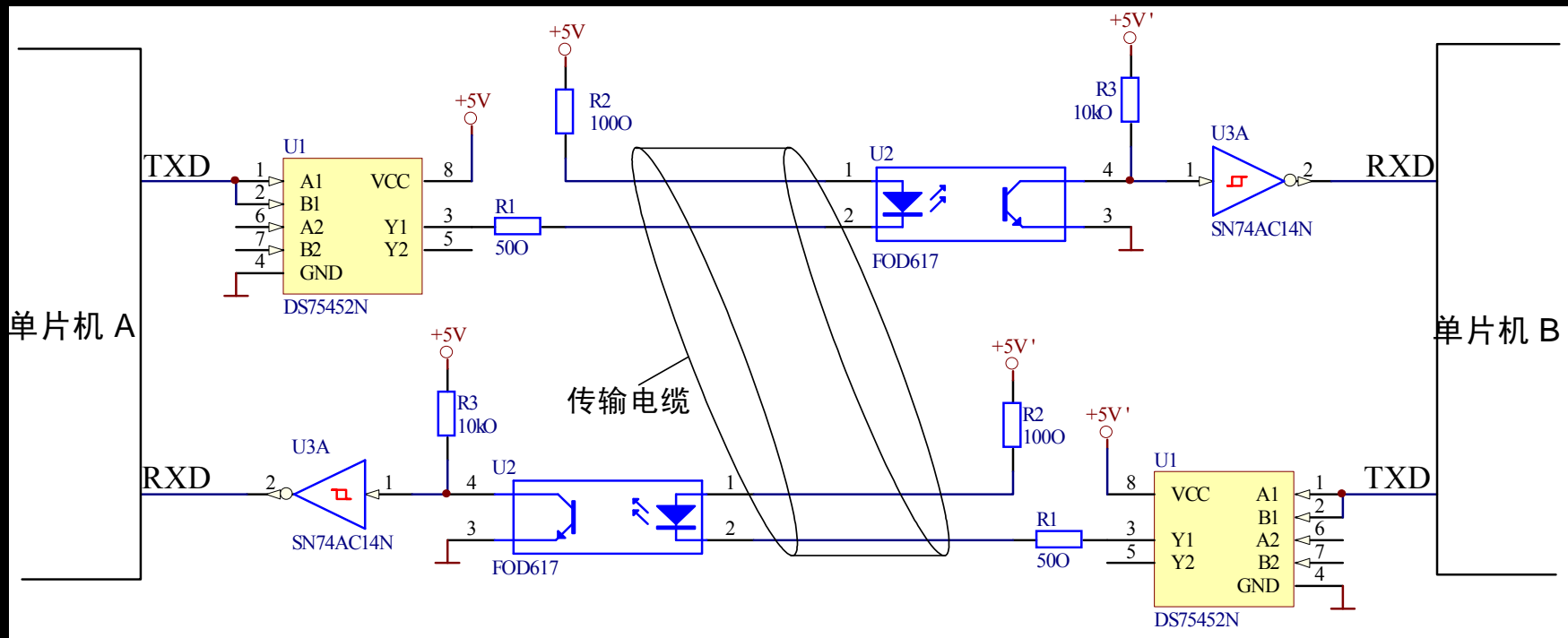
JMP START ; 循环

END

10.5 单片机之间的通信

• 10.5.2 远距离通信的解决方案

单片机串行口的电平比较低（5V左右），两个单片机的TXD和RXD对绞相连不适合直接进行远距离通信。解决的办法是提高信号的驱动能力后再由电缆进行传送。

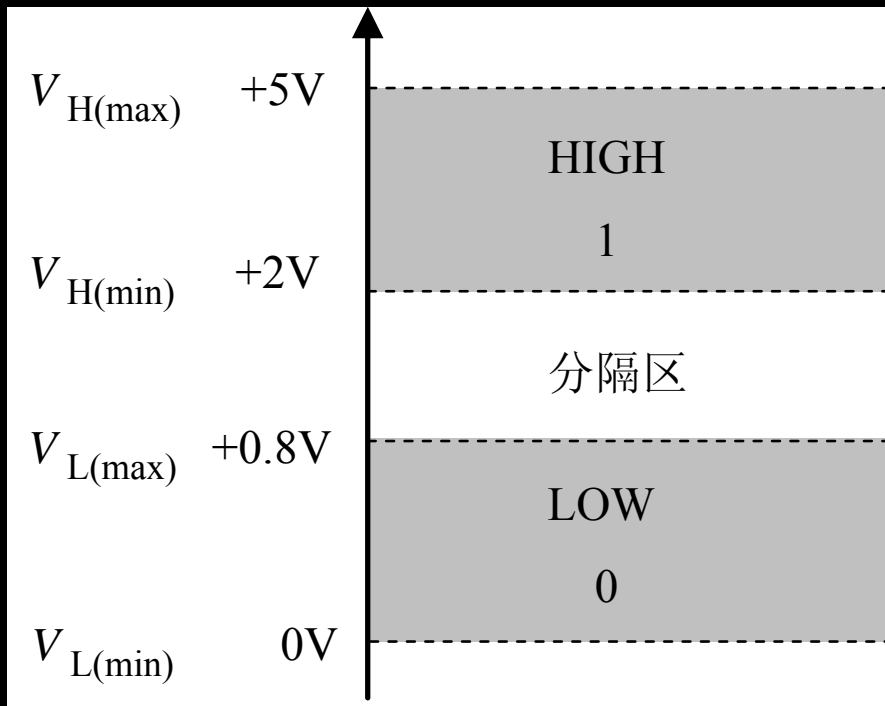


10.6 计算机的串行口与控制

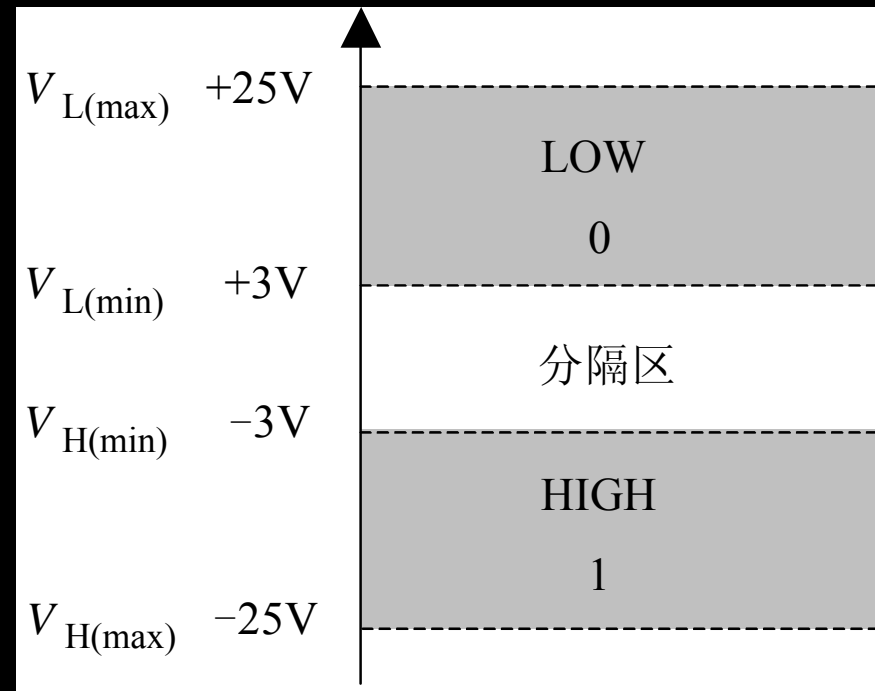
- 10.6.1 从RS-232标准到计算机串行口
 - ✓ RS-232中RS是“RETMA Standard”的首字母缩写，RETMA（Radio-Electronics and Television Manufactures Association，无线电、电子、电视设备制造商协会）是RS-232标准的发布者。现在RETMA已经发展成为EIA（Electronic Industries Alliance，电子工业联合会）。所以RS-232标准与EIA-232标准是相同的。
 - ✓ 由于RS-232标准早在1960年就已经被制订，而那时TTL逻辑电路还没有降生，所以它规定的逻辑电平比较奇怪，与今天的TTL电平不兼容。

10.6 计算机的串行口与控制

• 10.6.1 从RS-232标准到计算机串行口



(a) TTL逻辑电平

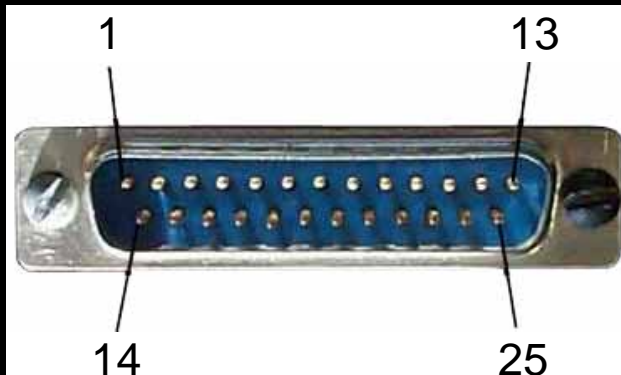


(b) RS-232逻辑电平

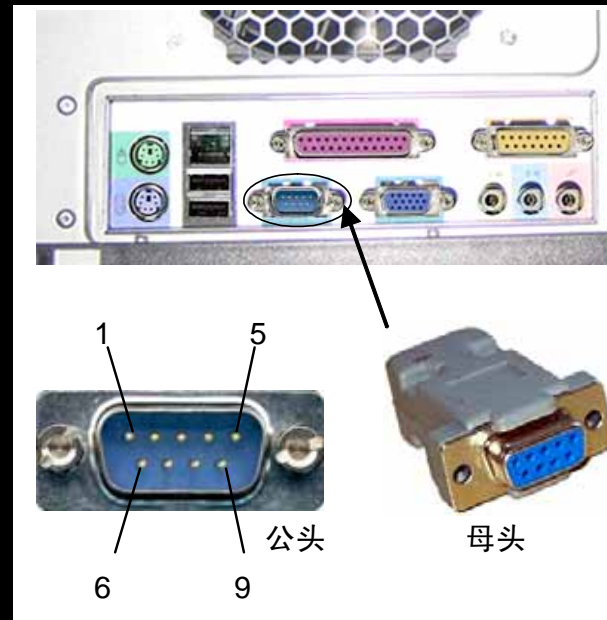
10.6 计算机的串行口与控制

• 10.6.1 从RS-232标准到计算机串行口

RS-232确定的是一种在数据终端设备（DTE）与数据运载设备（DCE）之间进行串行二进制数据交换的标准，它一定涉及接口器件的尺寸、规格等问题。



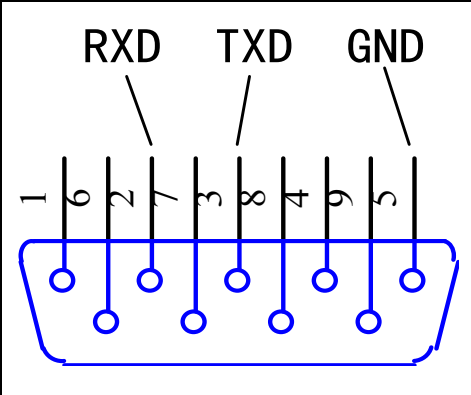
标准25芯DB-25接头



DB-9接头

10.6 计算机的串行口与控制

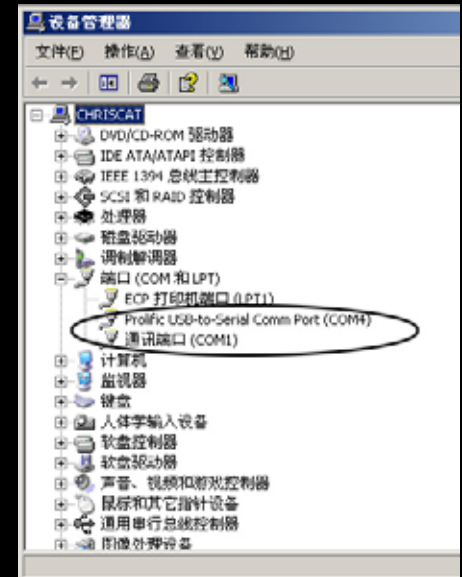
• 10.6.1 从RS-232标准到计算机串行口

串口示意图	序 号	名称及功能说明
	1	DCD: 载波检测位
	2	RXD: 接收数据位
	3	TXD: 发送数据位
	4	DTR: 数据终端准备信号位
	5	GND: 接地位
	6	DSR: 数据发送准备信号位
	7	RTS: 请求发送位
	8	CTS: 等待发送位
	9	RI: 响铃位

RS-232接口数据线说明

10.6 计算机的串行口与控制

• 10.6.1 从RS-232标准到计算机串行口(串口转换器)

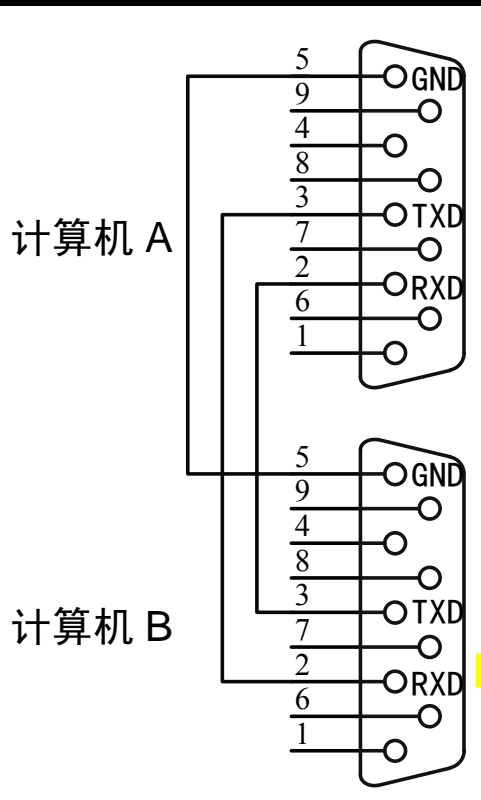


(a) USB转串口型 (b) PCMCIA转串口型

插上串口转换器并安装完成对应的驱动后，会在Windows的设备管理器中找到相应的硬件。“通信端口（COM1）”是计算机本身的串口，而名为“Prolific USB-to-Serial Comm Port（COM4）”才是USB转换器生成的串口。

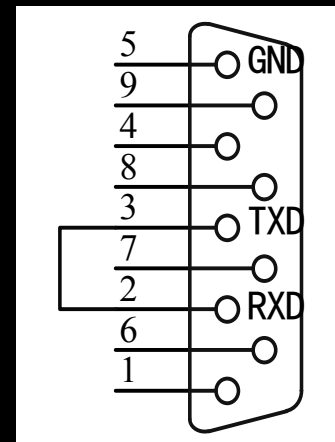
10.6 计算机的串行口与控制

• 10.6.1 从RS-232标准到计算机串行口(硬件连接)



(a) 两机通信

计算机A (GND)与B (GND) 相连, 计算机A (RXD) 与B (TXD) 相连, 计算机A (TXD) 与B (RXD) 相连。



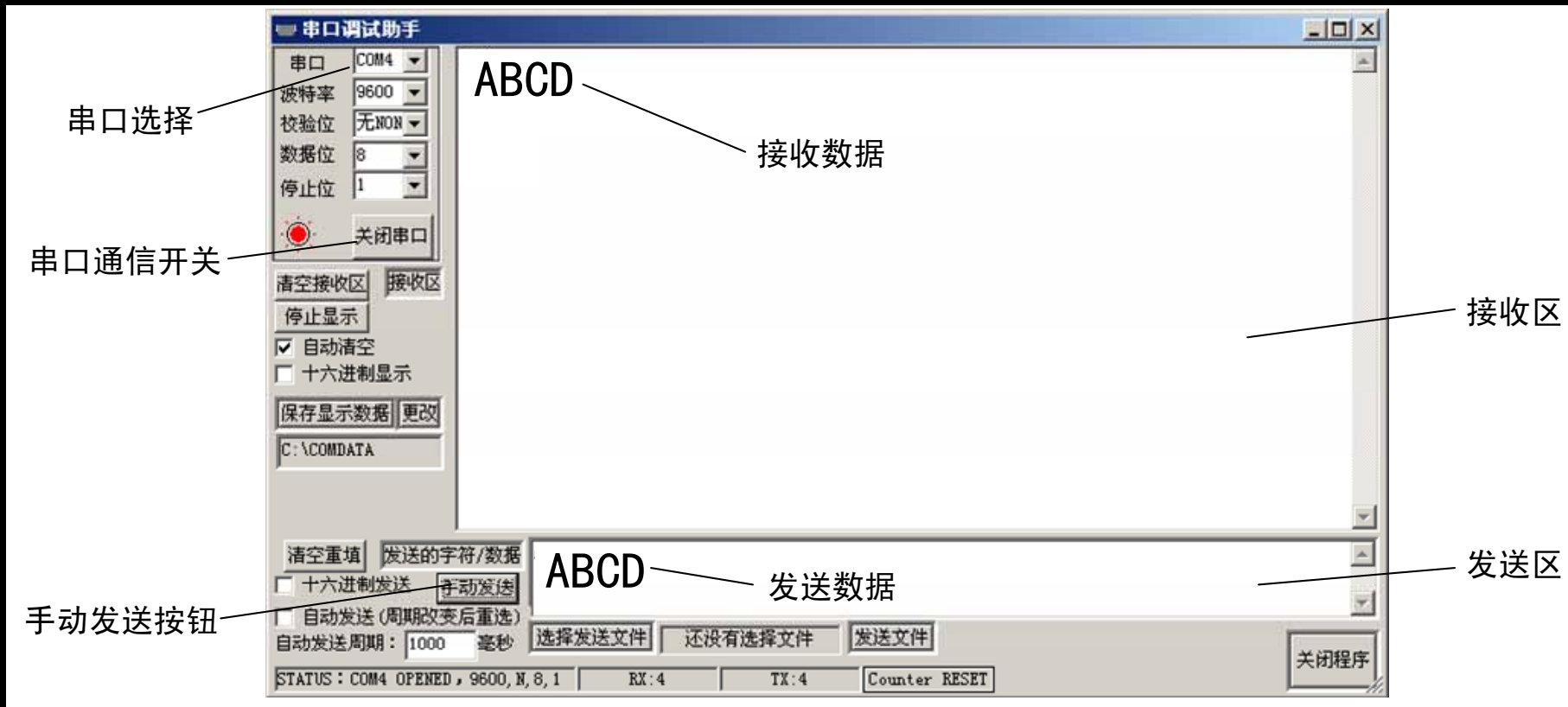
(b) 单机通信

计算机串口 TXD 与该口上 RXD 相连, 实现了本机发送的数据由本机接收的效果。

串口通信实验的硬件连接

10.6 计算机的串行口与控制

• 10.6.3 计算机串口通信实验——软件操作



注:如果接收不到数据或数据与发送的不一样,很有可能是串口号设置不正确,或串口的TXD和RXD没有连接上等问题造成的,或者试着把波特率调低。

10.6 计算机的串行口与控制

• 10.6.4 用Visual Basic编写一个串口通信程序

① “新建工程”对话框

新建一个“标准
EXE”工程

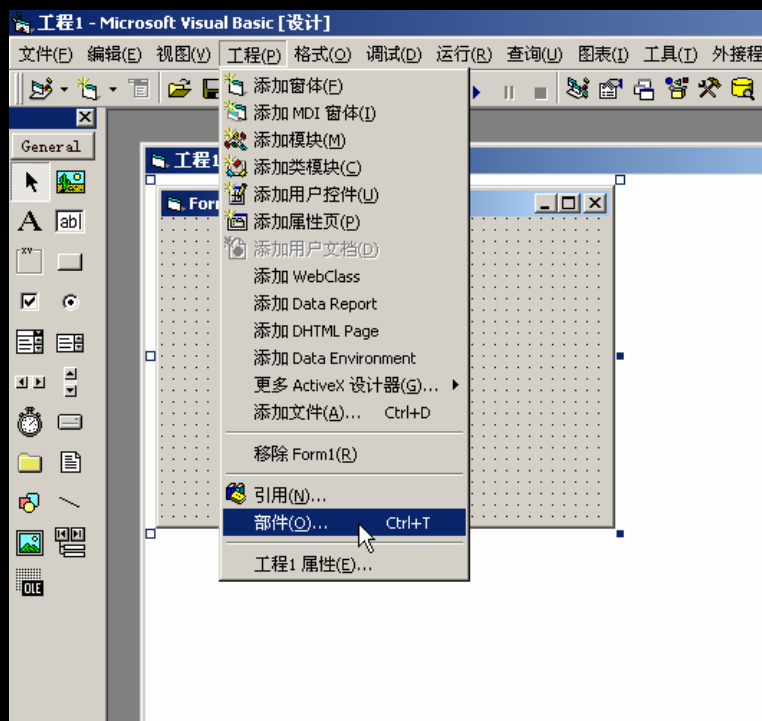


Open（打开）按钮

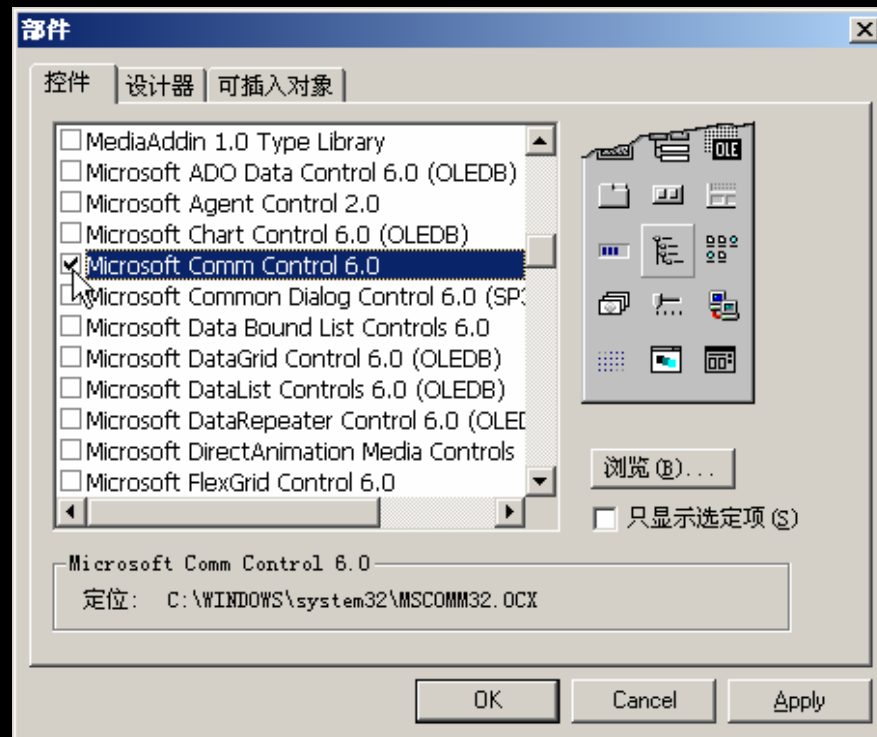
10.6 计算机的串行口与控制

• 10.6.4 用Visual Basic编写一个串口通信程序

② “部件”命令

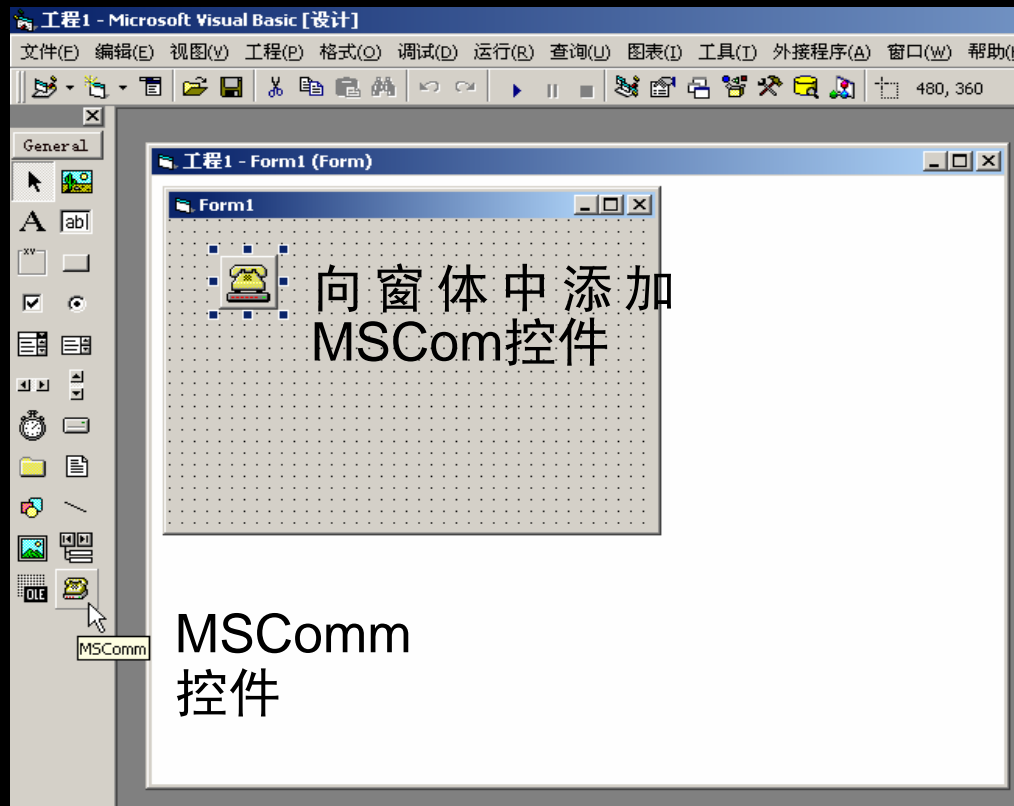


③ “部件”对话框



10.6 计算机的串行口与控制

- 10.6.4 用Visual Basic编写一个串口通信程序
- ④ MSComm控件及向窗体中添加



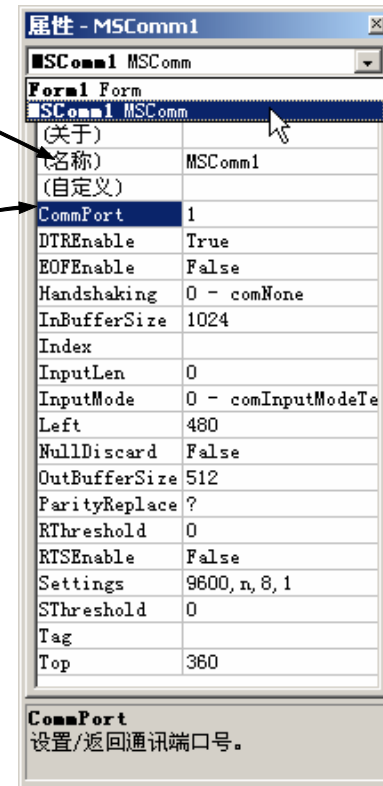
10.6 计算机的串行口与控制

• 10.6.4 用Visual Basic编写一个串口通信程序

⑤ MSComm1属性编辑栏

(名称): 该属性设置窗体中 MSComm 控件的名称, 在窗体代码中将使用这个名称来指向 MSComm 控件。

CommPort (串口号设置): CommPort 属性用于设置或返回通信连接串口号。程序在初始化时必须指定所要使用的串口号, 以便在程序运行期间对特定串口进行读写操作。同时, 程序也可借助此属性返回所使用的连接串口号。注意, 最大的串口号不能超过 16, 确定使用的串口号与图 10-21 中设备管理器中的显示串口号有关。例如使用 COM1, 就在 CommPort 属性中填写“1”。



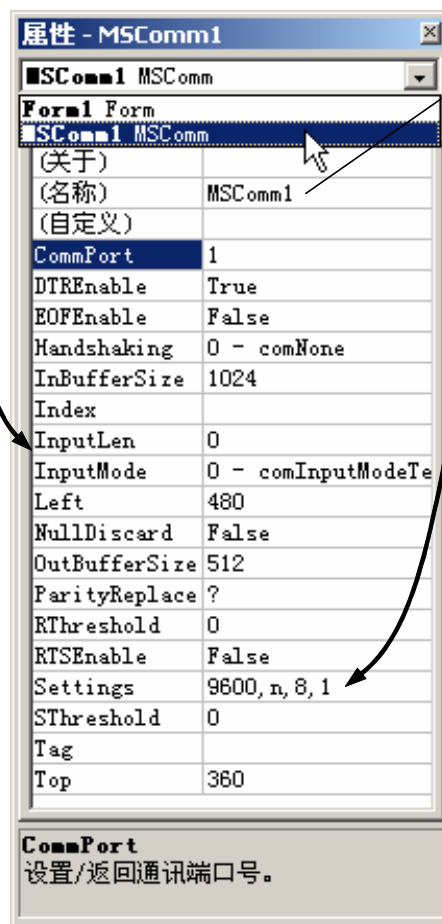
10.6 计算机的串行口与控制

• 10.6.4 用Visual Basic编写一个串口通信程序

⑤ MSComm1属性编辑栏

InputLen（接收字符数）：

InputLen 属性用于指定由串口读入的字符串长度或字节数，在需要指定读入输入缓冲区的字符长度时设置该属性。例如：InputLen=10，则当串口从RXD接收数据时，只会读取10个字符。如果在输入缓冲区中有55个字符，而设置InputLen=10，由于每一次的Input命令只能读10个字符，所以需要执行6次Input才能把缓冲区的数据读入完。

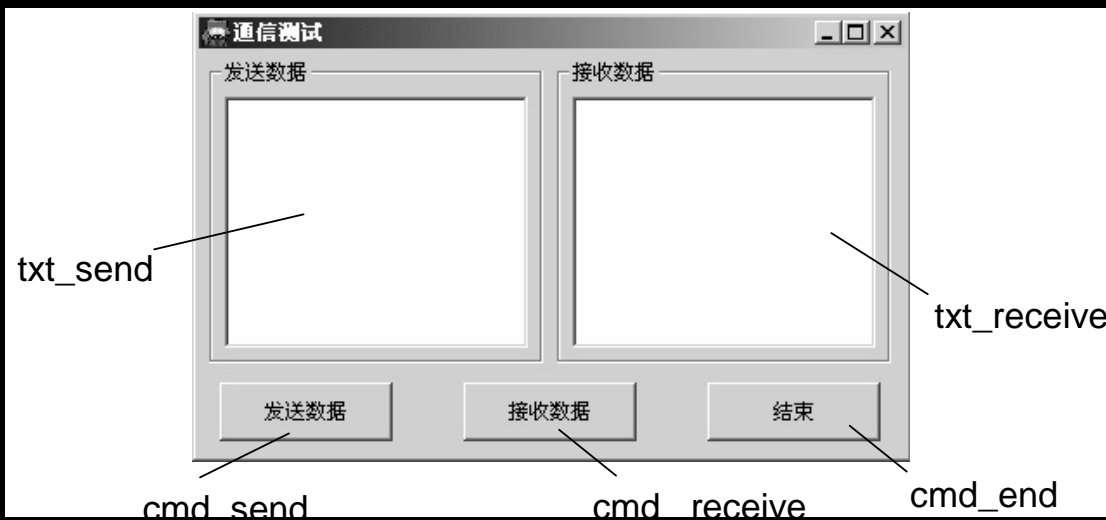


MSComm1 控件

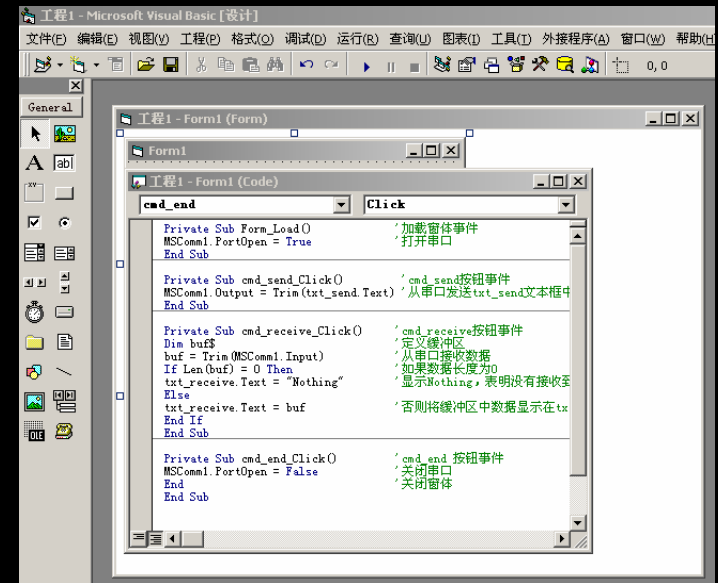
Settings（串口控制设置）：Settings 属性用于设置串口的初始化参数。默认值是“9600, n, 8, 1”。这个由逗号分隔的参数分别代表了波特率、奇偶校验位检查方式、数据位数和停止位数。波特率可选值包括 110、300、600、1200、2400、9600、14400、19200、28800 及 38400 等几种。在设置时需要注意与单片机 UART 设置的波特率一致（表 10-2 所示），否则无法正常接收和发送数据。Settings 默认值中的“n”是奇偶校验位，可以根据表 10-4 进行设置。默认值的“8”代表数据位数，可选择的值有 4、5、6、7 和 8。最后一位“1”是停止位数，可选择的值有 1、1.5 和 2。一般来说，这些值只需要修改波特率与单片机 UART 波特率相同即可正常工作。

10.6 计算机的串行口与控制

- 10.6.4 用Visual Basic编写一个串口通信程序
- ⑥ 串口通信实验窗体编辑与窗体代码编辑器



串口通信实验窗体编辑



窗体代码编辑器

10.6 计算机的串行口与控制

• 10.6.4 用Visual Basic编写一个串口通信程序

⑦ 控制串口的Visual Basic程序解析:

```
Private Sub Form_Load()      ;加载窗体事件
MSComm1.PortOpen=True      ;打开串口
End Sub
```

```
Private Sub cmd_send_Click() ;cmd_send按钮事件
MSComm1.Output=Trim(txt_send.Text)
                        ;从串口发送txt_send文本框中的数据
End Sub
```

```
Private Sub cmd_end_Click() ;cmd_end按钮事件
MSComm1.PortOpen=False    ;关闭串口
End                        ;关闭窗体
End Sub
```

10.6 计算机的串行口与控制

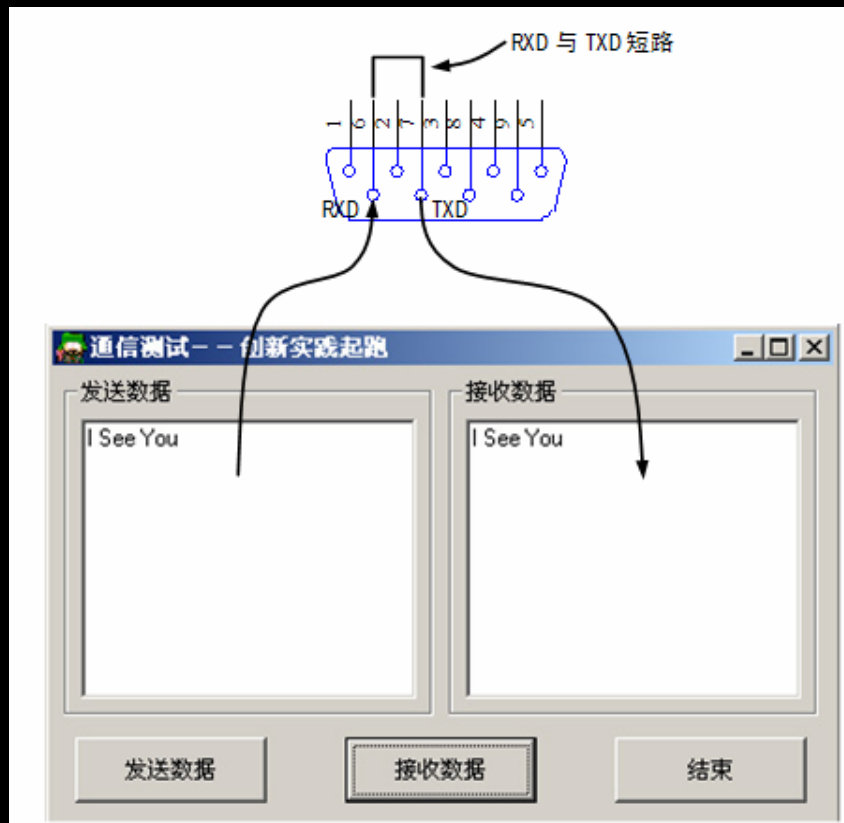
- 10.6.4 用Visual Basic编写一个串口通信程序

⑦ 控制串口的Visual Basic程序解析:

```
Private Sub cmd_receive_Click()      ;cmd_receive按钮事件
Dim buf$                            ;定义缓冲区
buf=Trim(MSComm1.Input)             ;从串口接收数据
If Len(buf)=0 Then                  ;如果数据长度为0
txt_receive.Text="Nothing"          ;显示Nothing，表明没有接收到任何数据
Else
txt_receive.Text=buf;否则将缓冲区中数据显示在txt_receive文本框中
End If
End Sub
```


10.6 计算机的串行口与控制

- 10.6.4 用Visual Basic编写一个串口通信程序
- ⑧ 单击按钮编译代码并运行

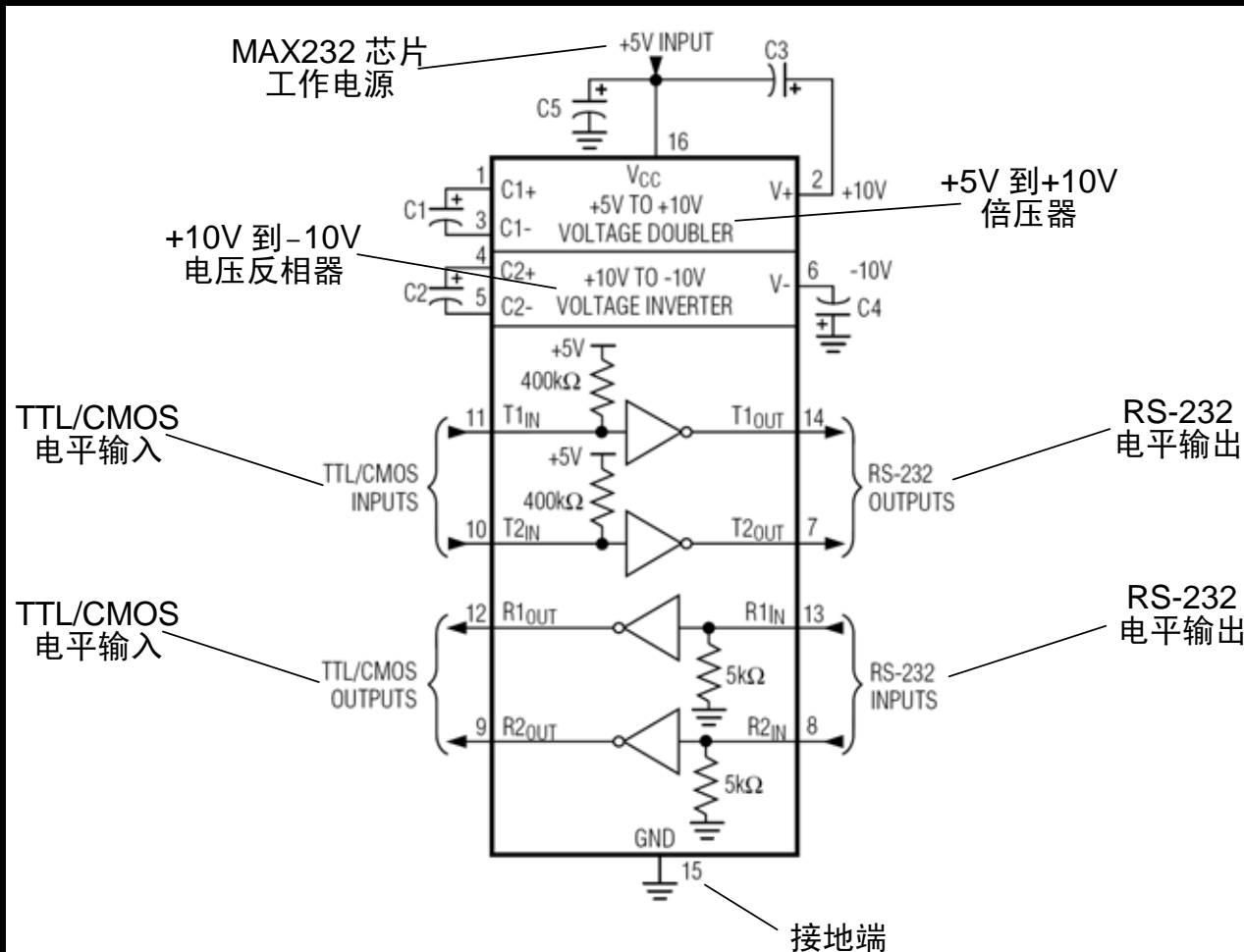


10.7 单片机与计算机之间的通信

- 10.7.1 单片机与计算机通信中的接口芯片
 - ✓ 计算机串口执行的是RS-232标准，高、低电平分别为-3~-25V和+3~+25V，与单片机的TTL逻辑中的+2~+5V和0~+0.8V不兼容。
 - ✓ 较为常用的接口芯片是MAX232。它能很好地完成TTL与RS-232的逻辑电平转换，工作电压与AT89S51单片机相同均为+5V，可以使用同一个电源。

10.7 单片机与计算机之间的通信

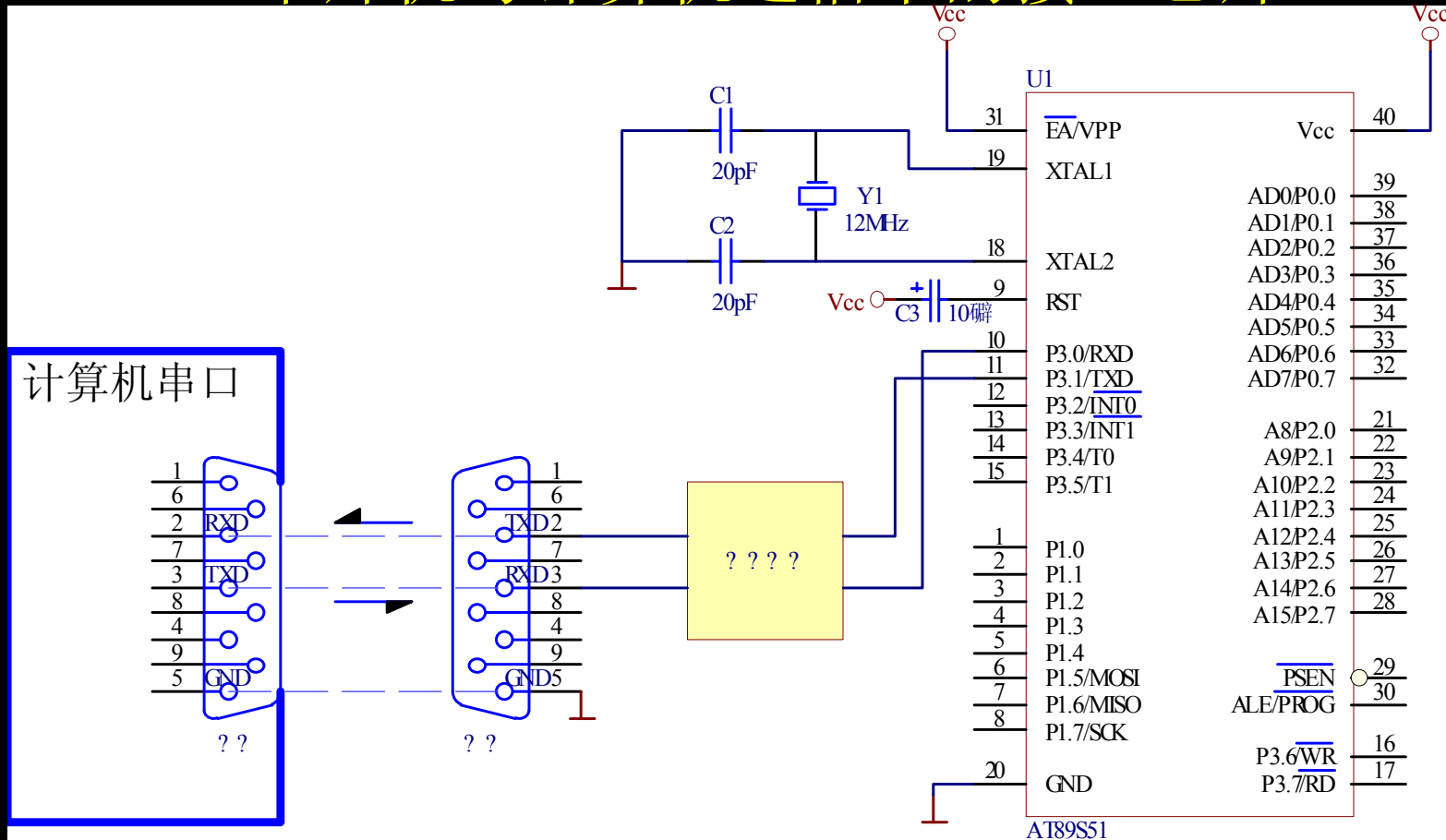
• 10.7.1 单片机与计算机通信中的接口芯片



MAX232集
成了两组电
平转换器。

10.7 单片机与计算机之间的通信

• 10.7.1 单片机与计算机通信中的接口芯片

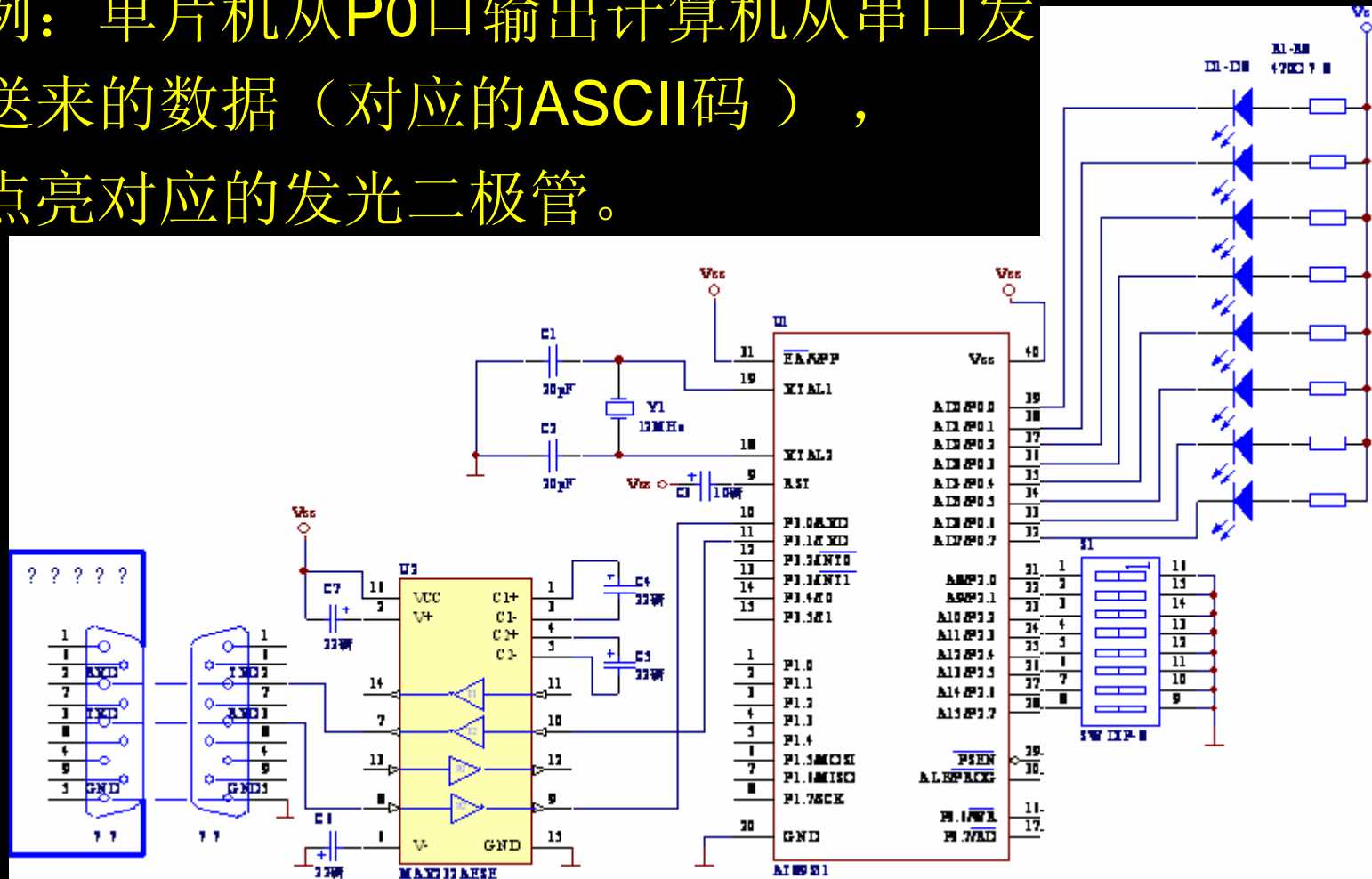


计算机与单片机串行口的接口

10.7 单片机与计算机之间的通信

• 10.7.2 单片机与计算机的串行通信

例：单片机从P0口输出计算机从串口发送来的数据（对应的ASCII码），
点亮对应的发光二极管。



10.7 单片机与计算机之间的通信

• 10.7.2 单片机与计算机的串行通信（软件）

单片机接收计算机串口发送的数据：

```
                ORG          00H
START:
    MOV  SCON,#50H ; ①设置串行口工作在模式1下，接收使能
    MOV  TMOD,#20H ; ②使用Timer 1工作在模式2下
    MOV          TH1, #0F3H ; ③波特率2400
    MOV          TL1, #0F3H
    SETB  TR1      ; ④启动Timer 1
WAIT:
    JBC  RI, DISPLAY ; ⑤判断是否接收到数据
    JMP          WAIT
DISPLAY:
    MOV          A, SBUF ; ⑥接收到的数据载入ACC
    MOV          P0, A   ; 将接收到的数据输出显示
    JMP          WAIT   ; 循环
END
```

10.7 单片机与计算机之间的通信

• 10.7.2 单片机与计算机的串行通信（软件）

单片机通过串口向计算机发送数据：

ORG 00H

START:

MOV SCON, #50H; ①设置串行口工作在模式1下，接收使

MOV TMOD, #20H; ②使用Timer 1工作在模式2下

MOV TH1, #0F3H ; ③波特率2400

MOV TL1, #0F3H

SETB TR1 ; ④启动Timer 1

SEND:

MOV A, P2; 把DIP开关的状态载入ACC中

MOV P0, A ; 显示DIP开关的状态

MOV SBUF, A ; ⑤发送数据

WAIT:

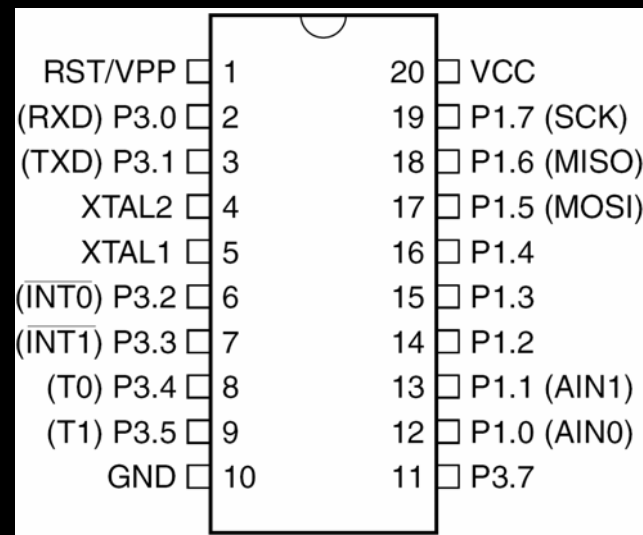
JBC TI, SEND ; ⑥判断是否发送完毕

JMP WAIT

END

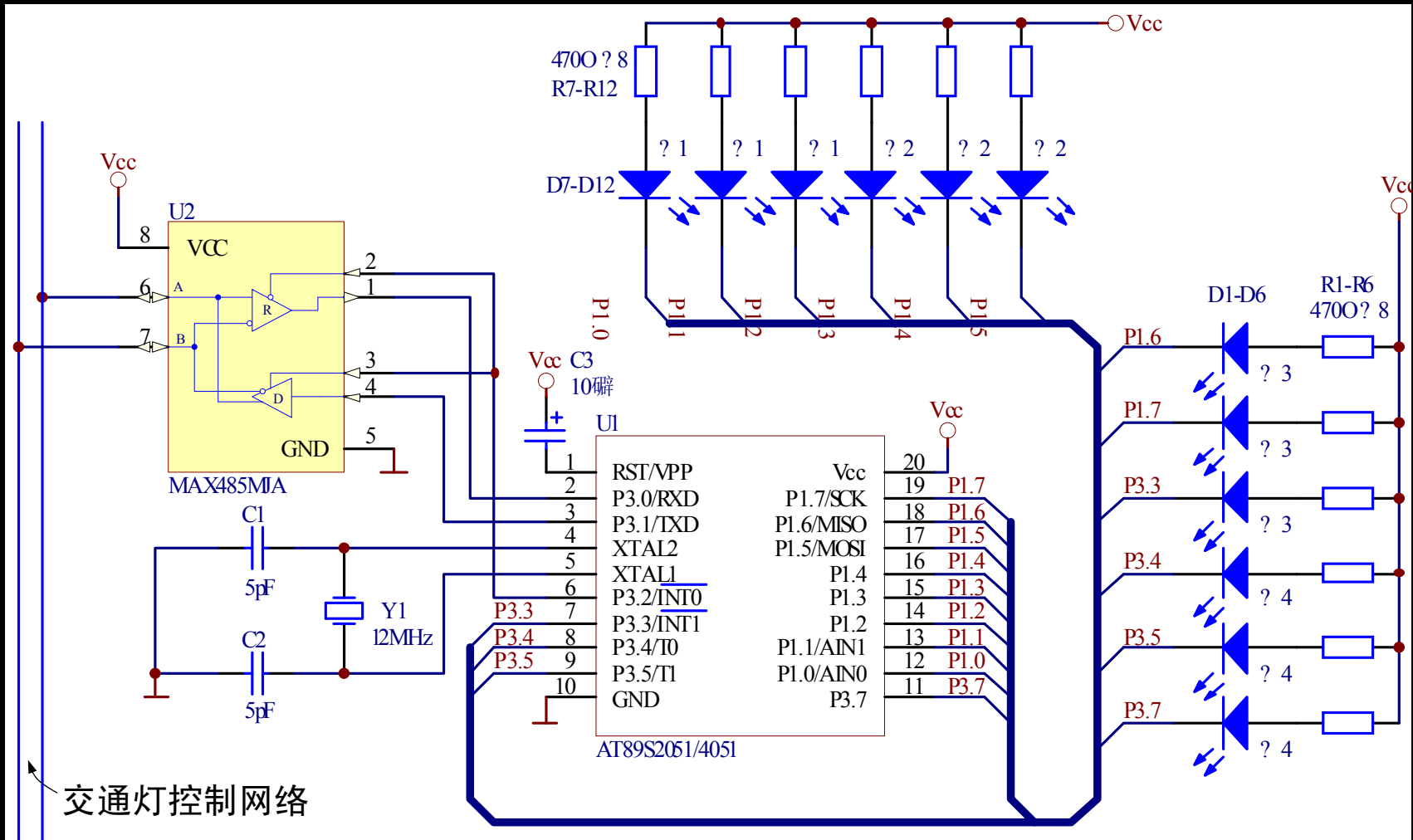
10.8 实例点拨——智能交通灯

- AT89S2051单片机简介：
 - ✓ 与51单片机指令、逻辑电平兼容。
 - ✓ 2K Bytes的可在线编程（ISP）Flash存储器（AT89S4051为4K字节）。
 - ✓ 256×8位片内RAM。
 - ✓ *可以在+2.7~+5.5V范围内工作。
 - ✓ 15个可编程的I/O口。
 - ✓ 两个16位定时/计数器。
 - ✓ 6个中断源。
 - ✓ 可编程串行口UART。
 - ✓ *具有一个可产生中断的模拟信号比较器。
 - ✓ *8位脉冲宽度调制（PWM）。
 - ✓ 低功耗的待机模式和休眠。
 - ✓ *电源电压走低时复位（brown out reset）。
 - ✓ *增强了串行口的错误帧检测及自动地址识别。



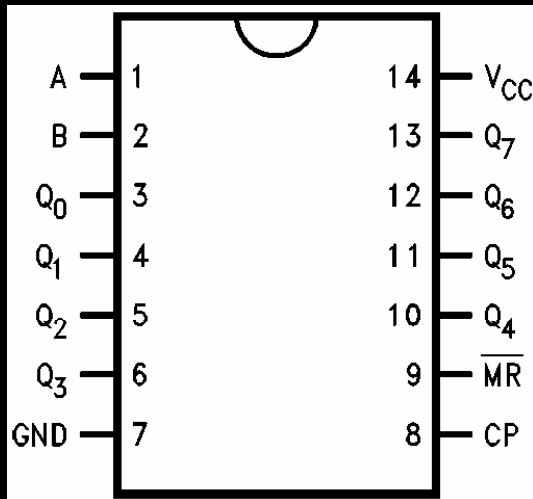
10.8 实例点拨——智能交通灯

• 基于AT89S2051单片机的交通灯

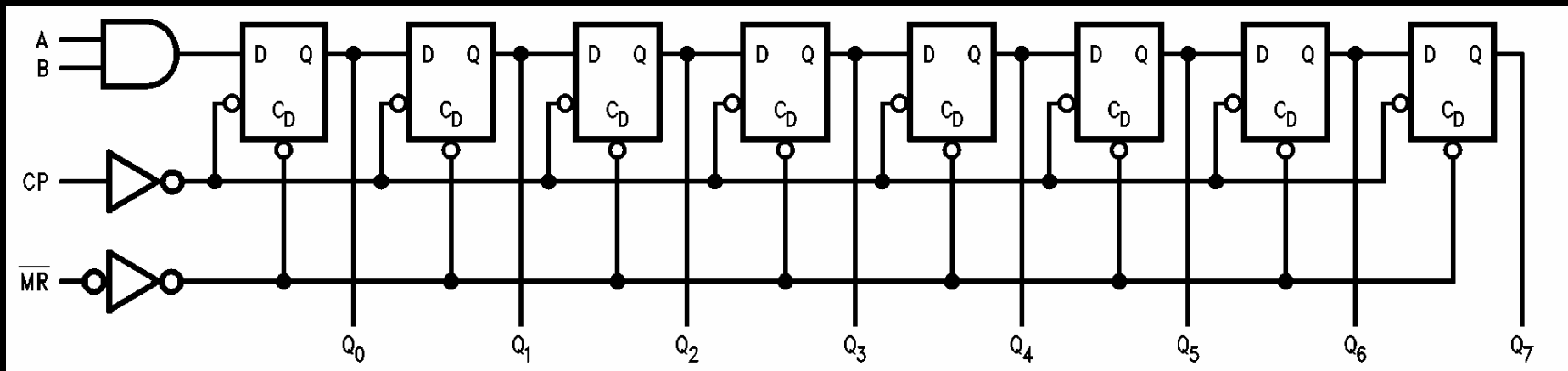


器件介绍

1. 8位串入/并出移位寄存器74164

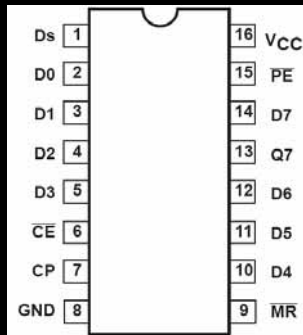


操作模式	输 入			输 出	
	$\overline{\text{MR}}$	A	B	Q_0	Q_1-Q_7
清0	L	X	X	L	L-L
移位	H	L	L	L	Q_0-Q_6
	H	L	H	L	Q_0-Q_6
	H	H	L	L	Q_0-Q_6
	H	H	H	H	Q_0-Q_6

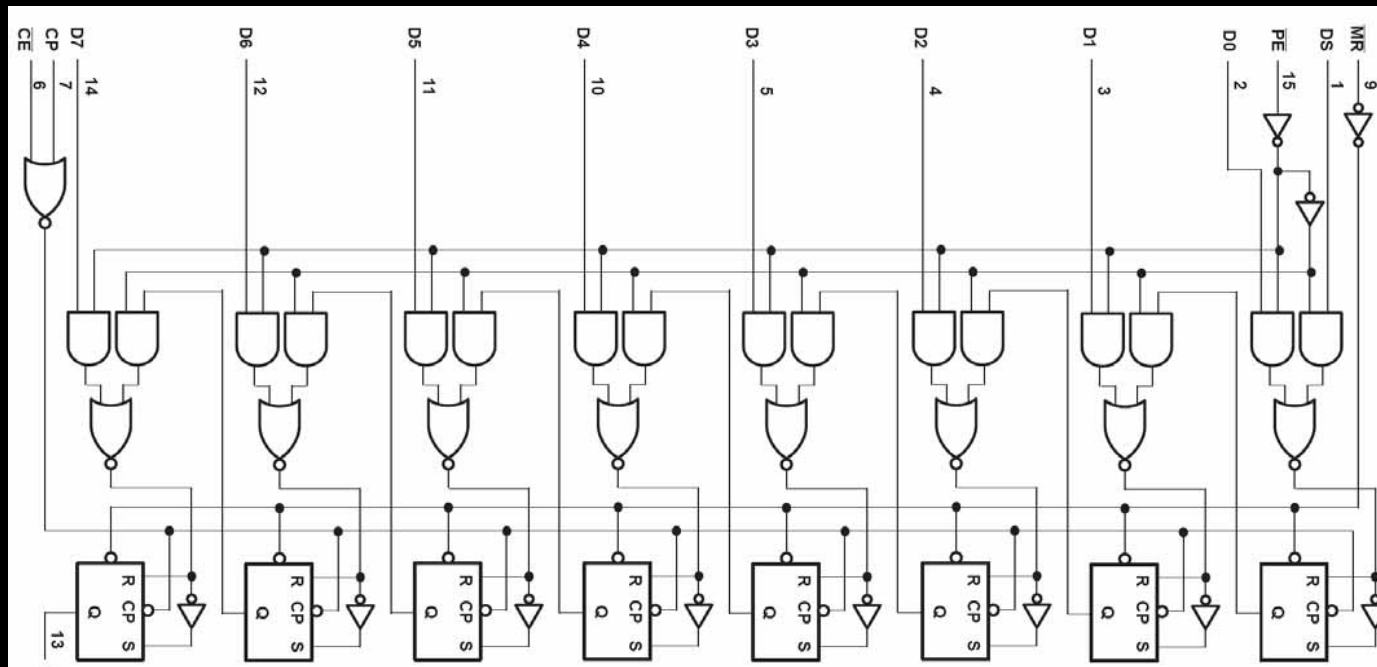


器件介绍

2. 8位并行装载移位寄存器74166



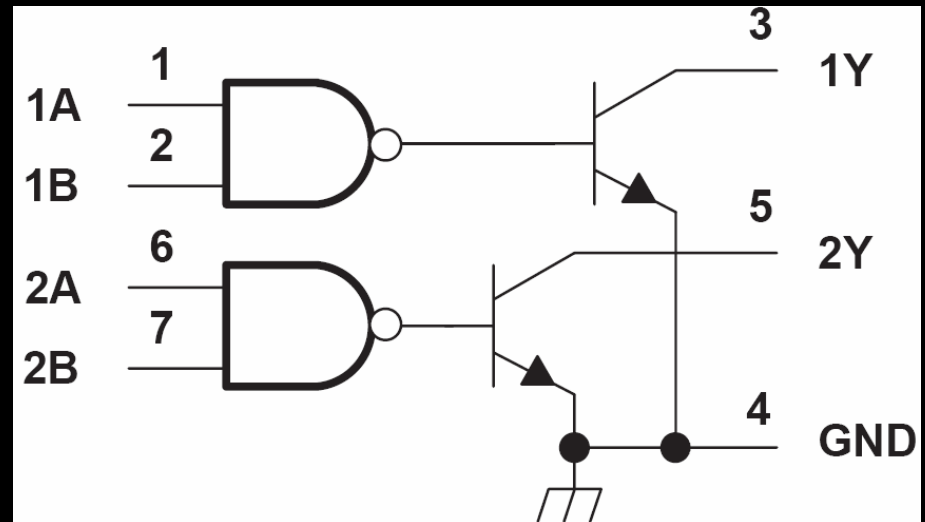
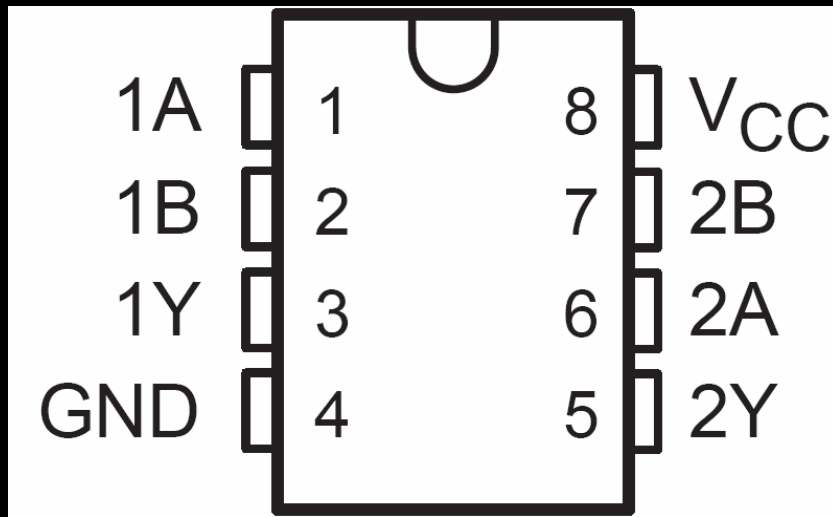
输入					输出			操作模式
PE	CE	CP	DS	D0-D7	Q0	Q1-Q6	Q7	
l	l	↑	X	l-l	L	L-L	L	并行输入
l	l	↑	X	h-h	H	H-H	H	
h	l	↑	l	X-X	L	q0-q5	q6	串行输出
h	l	↑	h	X-X	H	q0-q5	q6	
X	h	X	X	X-X	qn	q1-q6	q7	保持



器件介绍

3. 信号驱动DS75452

DS75452芯片可作为以TTL为逻辑的系统中的信号驱动。常用于高速逻辑缓冲器、功率驱动、继电器驱动、**MOS**驱动、总线驱动和存储器驱动等，



器件介绍

4. 光电耦合器FOD617

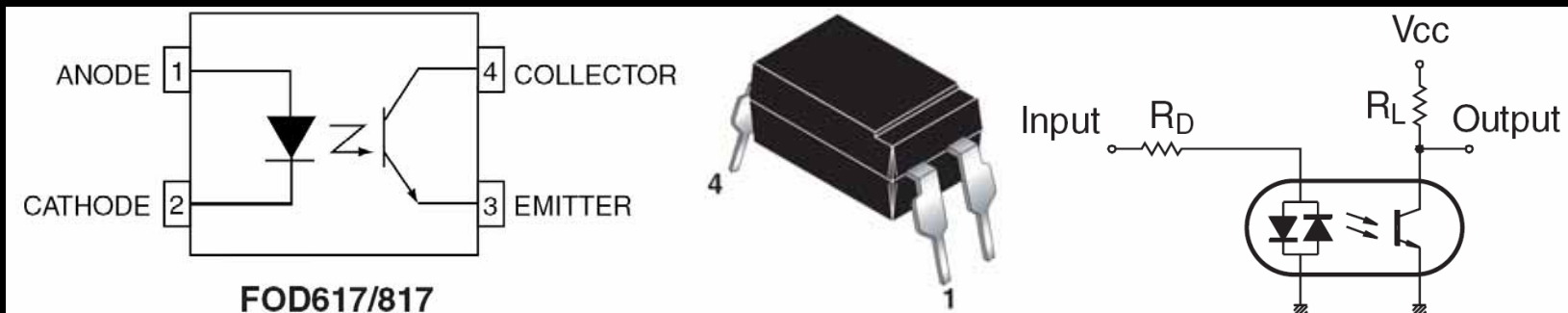
光电耦合器是以光为媒介传输电信号的一种电—光—电转换器件。它由发光源和受光器两部分组成。把发光源和受光器组装在同一密闭的壳体内，彼此间用透明绝缘体隔离。发光源的引脚为输入端，受光器的引脚为输出端。当发光源根据输入信号进行变化时，在受光器端能输出一个相应的信号。

常见的发光源为发光二极管，受光器为光敏二极管、光敏三极管等。

器件介绍

4. 光电耦合器FOD617

光电耦合器的种类较多，常见的有光电二极管型、光电三极管型、光敏电阻型、光控晶闸管型、光电达林顿型和集成电路型等。由于输入的电信号经过了电—光—电的转换，输出的电信号并不是直接的输入信号，而是具有相关性的“新”信号，所以光电耦合器常用于信号的隔离。



光电耦合器FOD617