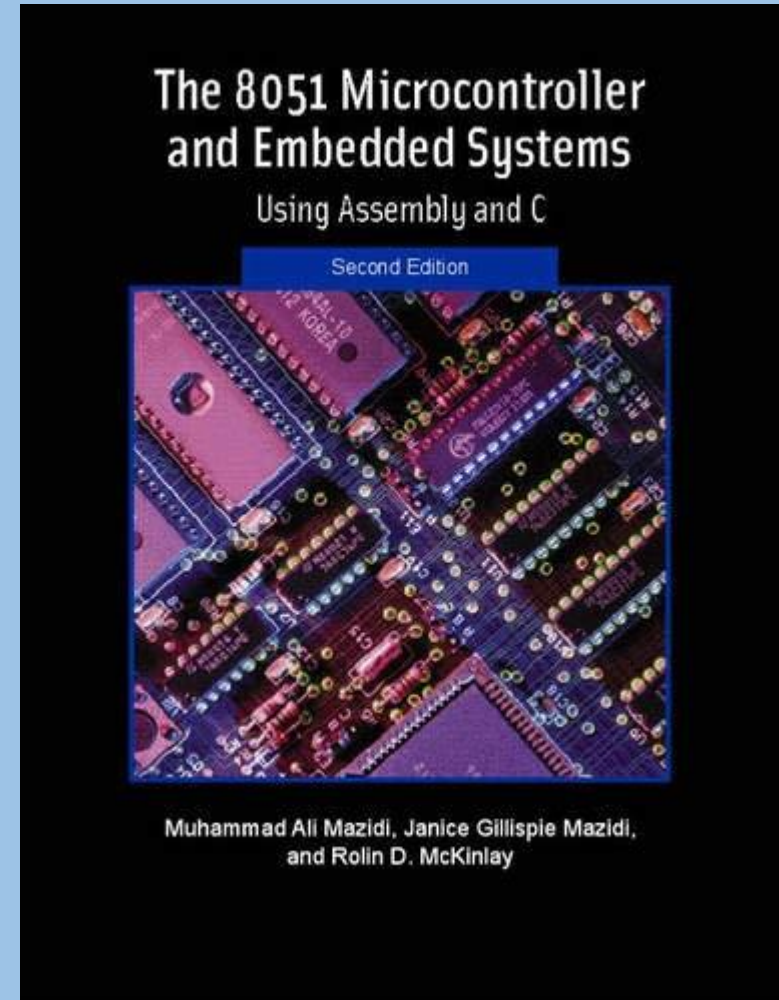


THE 8051 MICROCONTROLLER & Embedded Systems

Muhammad Ali Mazidi, Janice Mazidi
& Rolin McKinlay



8051 SERIAL PORT PROGRAMMING IN ASSEMBLY AND C

Chapter 10

Objectives

Upon completion of this chapter, you will be able to:

- >> Contrast and compare serial versus parallel communication**
- >> List the advantages of serial communication over parallel**
- >> Explain serial communication protocol**
- >> Contrast synchronous versus asynchronous communication**
- >> Contrast half- versus full-duplex transmission**
- >> Explain the process of data framing**
- >> Describe data transfer rate and bps rate**
- >> Define the RS232 standard**
- >> Explain the use of the MAX232 and MAX233 chips**
- >> Interface the 8051 with an RS232 connector**
- >> Discuss the baud rate of the 8051**
- >> Describe serial communication features of the 8051**
- >> Program the 8051 serial port in Assembly and C**
- >> Program the second serial port of DS89C4x0 in Assembly and C**

Objectives

- **Section 10.1, the basics of serial communication,**
- **Section 10.2, 8051 interfacing to RS232 connectors via MAX232 line drivers,**
- **Section 10.3, programming the second serial port of DS89C4x0**

Figure 10-1. Serial versus Parallel Data Transfer

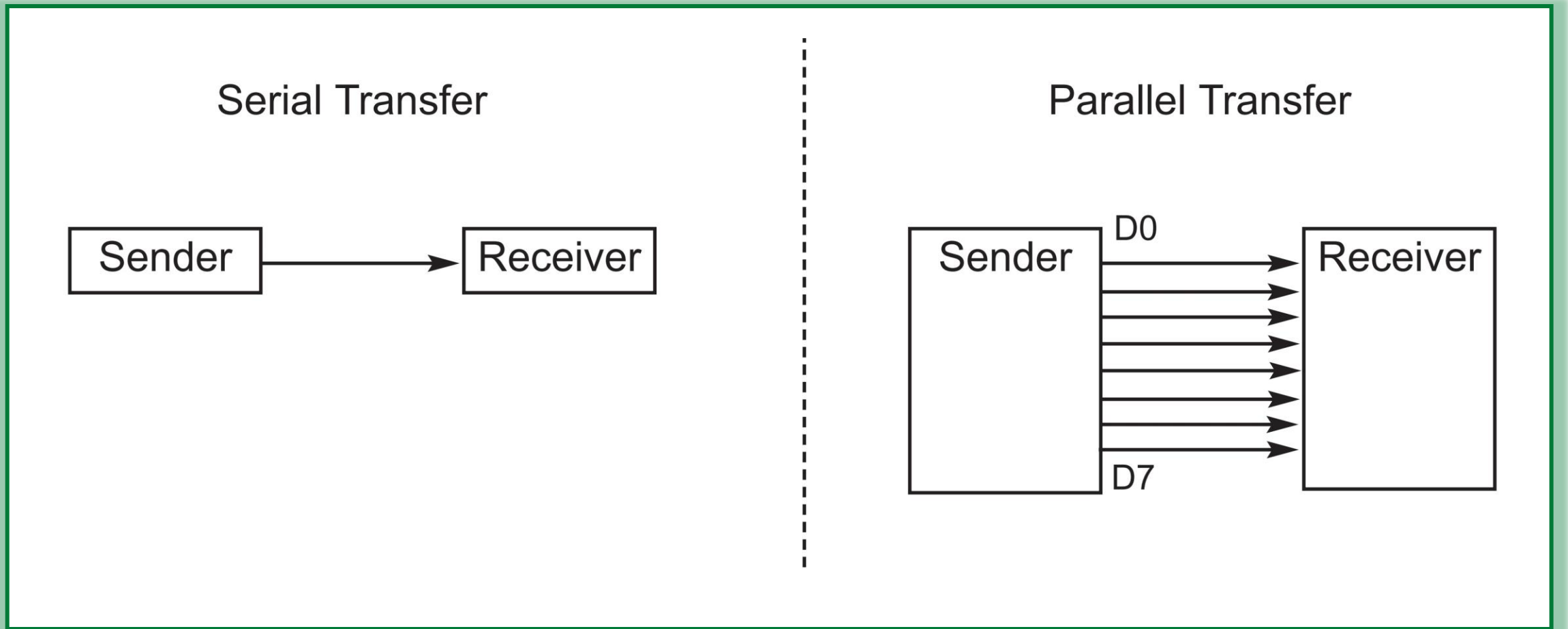
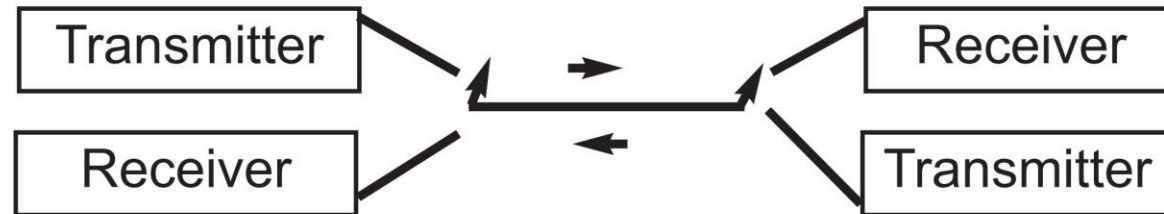


Figure 10-2. Simplex, Half-, and Full-Duplex Transfers

Simplex



Half Duplex



Full Duplex



Figure 10-3. Framing ASCII "A" (41H)

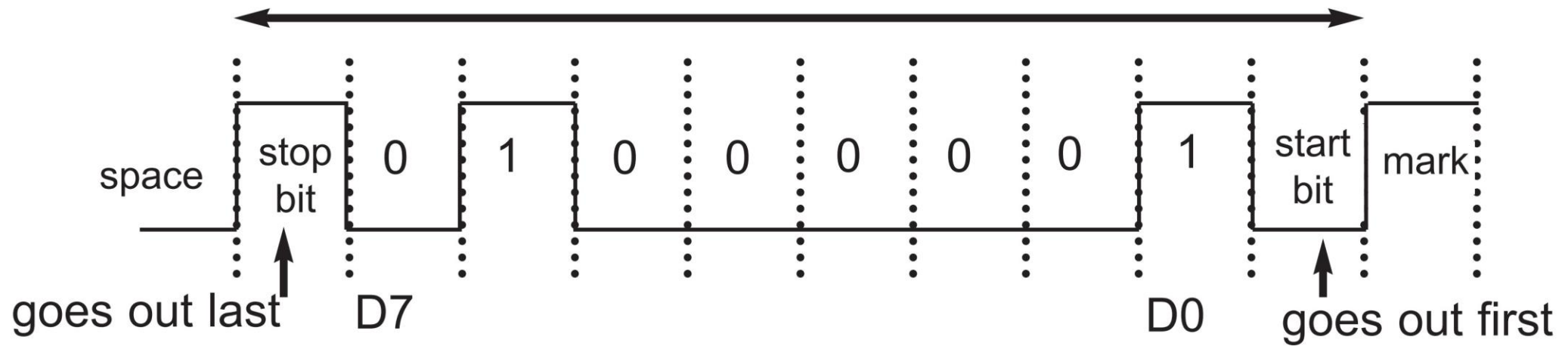


Figure 10-4. The original RS232 Connector DB-25 (No longer in use)

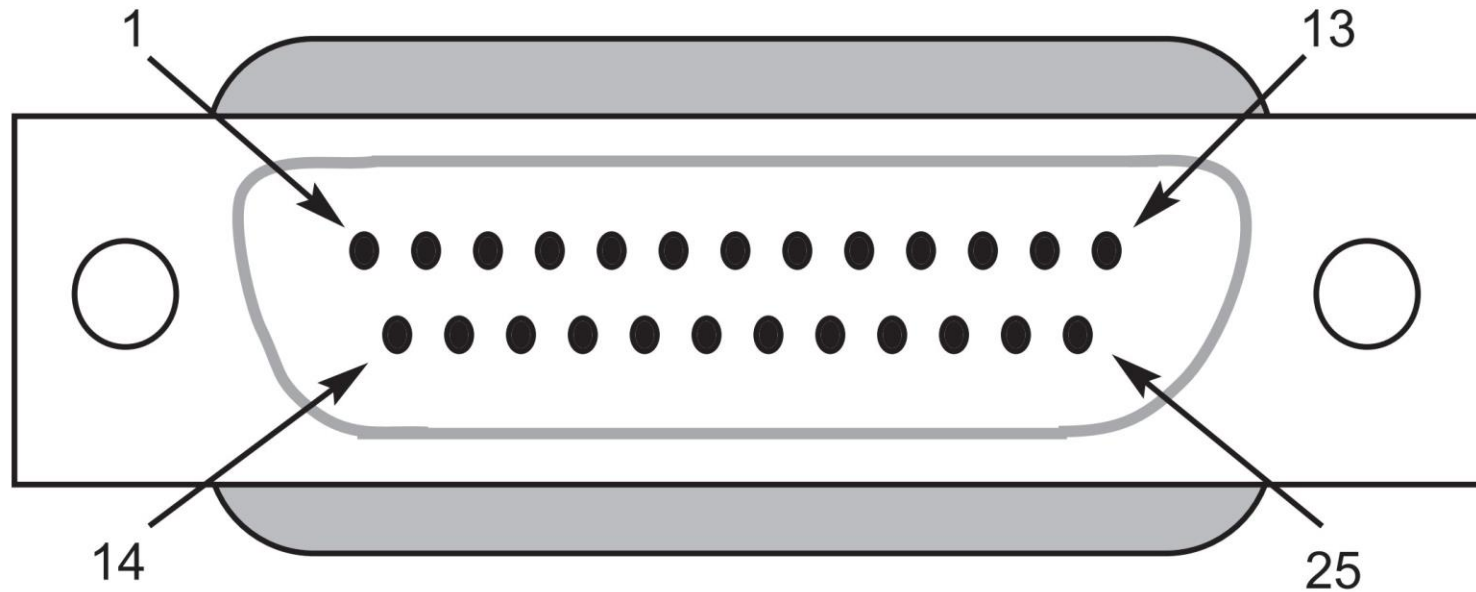


Table 10-1: RS232 Pins (DB-25)

Pin	Description
1	Protective ground
2	Transmitted data (TxD)
3	Received data (RxD)
4	Request to send ($\overline{\text{RTS}}$)
5	Clear to send ($\overline{\text{CTS}}$)
6	Data set ready ($\overline{\text{DSR}}$)
7	Signal ground (GND)
8	Data carrier detect ($\overline{\text{DCD}}$)
9/10	Reserved for data testing
11	Unassigned
12	Secondary data carrier detect
13	Secondary clear to send

Pin	Description
14	Secondary transmitted data
15	Transmit signal element timing
16	Secondary received data
17	Receive signal element timing
18	Unassigned
19	Secondary request to send
20	Data terminal ready ($\overline{\text{DTR}}$)
21	Signal quality detector
22	Ring indicator
23	Data signal rate select
24	Transmit signal element timing
25	Unassigned

Figure 10-5. DB-9 9-Pin Connector

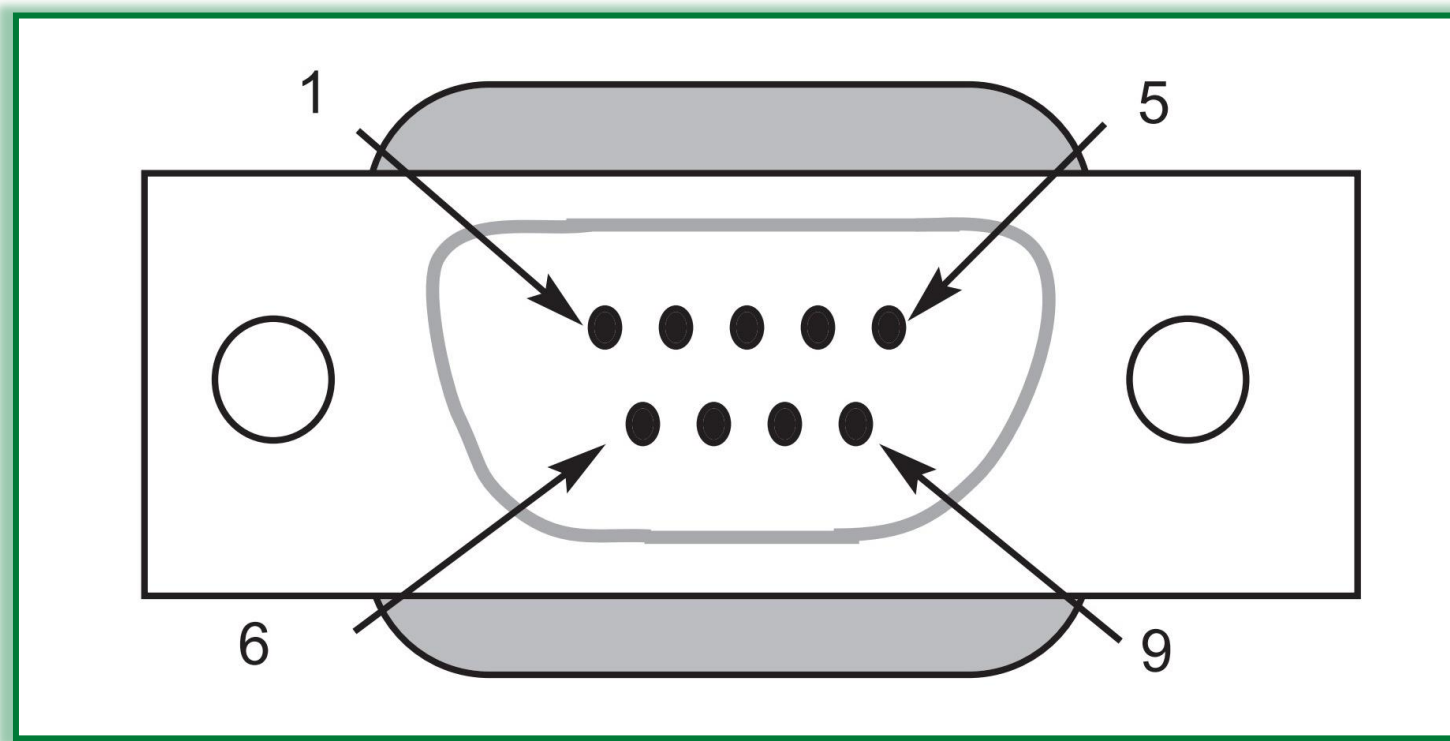


Table 10-2: IBM PC DB-9 Signals

Pin	Description
1	Data carrier detect ($\overline{\text{DCD}}$)
2	Received data (RxD)
3	Transmitted data (TxD)
4	Data terminal ready (DTR)
5	Signal ground (GND)
6	Data set ready ($\overline{\text{DSR}}$)
7	Request to send ($\overline{\text{RTS}}$)
8	Clear to send ($\overline{\text{CTS}}$)
9	Ring indicator (RI)

Figure 10-6. Null Modem Connection

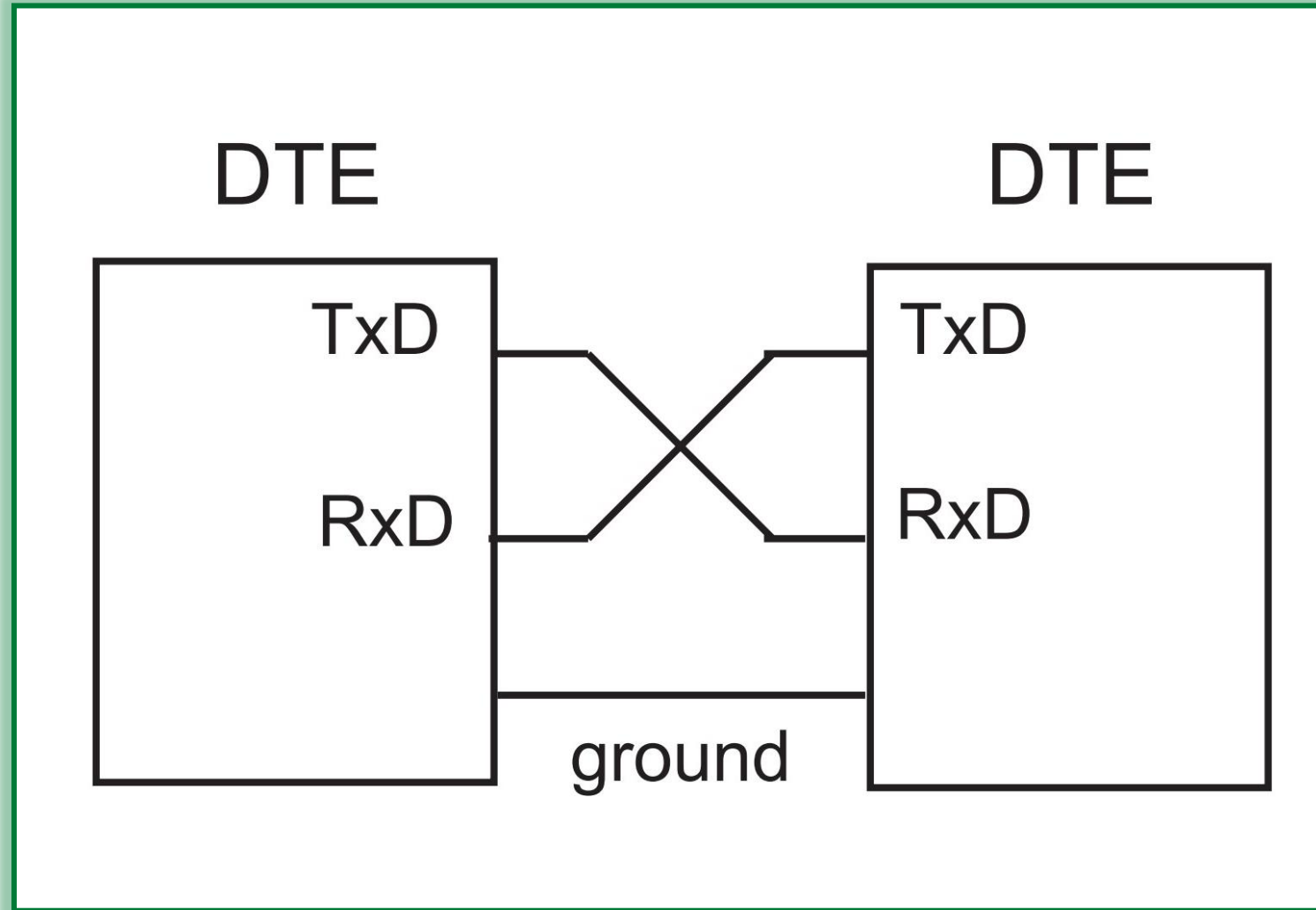


Figure 10-7. (a) Inside MAX232 and (b) its Connection to the 8051 (Null Modem)

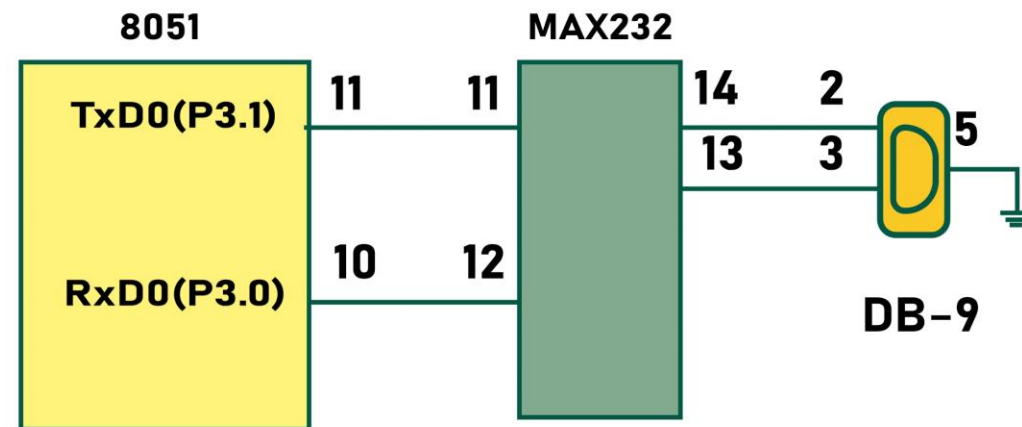
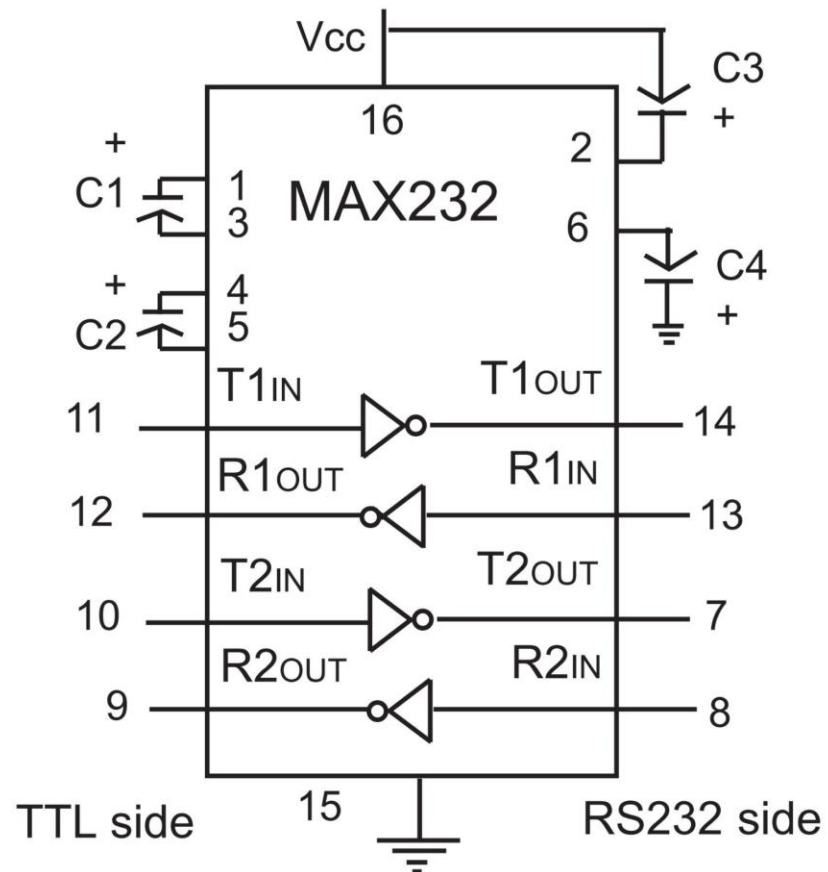
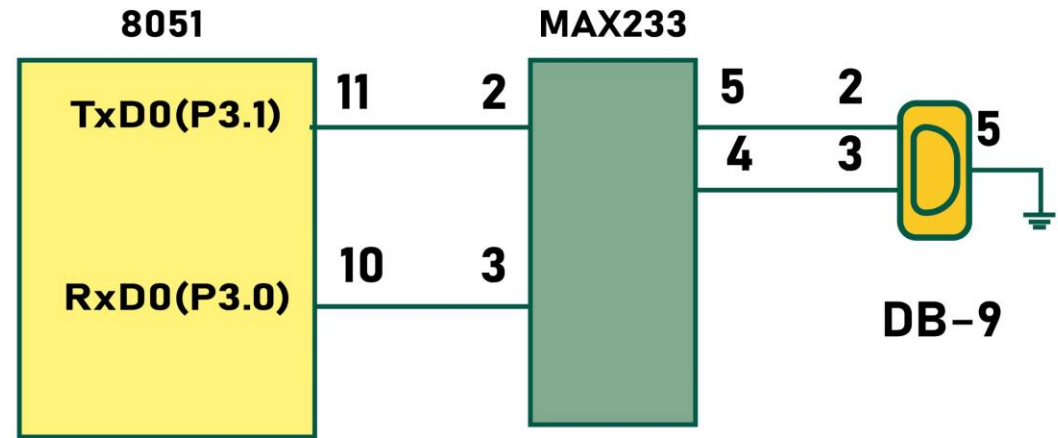
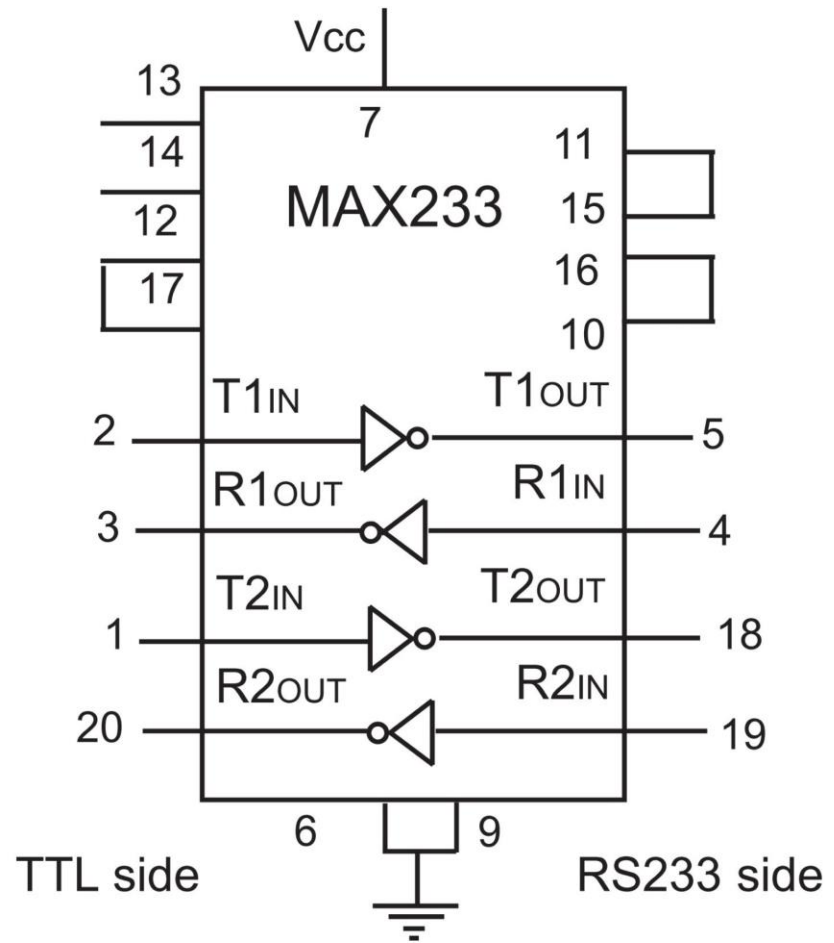


Figure 10-8. (a) Inside MAX233 and (b) Its Connection to the 8051 (Null Modem)



110
150
300
600
1200
2400
4800
9600
19200

Note: Some of the Baud rates supported by x86 PC BIOS.

Table 10-3: PC Baud Rates

Baud Rate	TH1 (Decimal)	TH1 (Hex)
9600	-3	FD
4800	-6	FA
2400	-12	F4
1200	-24	E8

Note: XTAL = 11.0592 MHz.

Table 10-4: Timer 1 TH1 Register Values for Various Baud Rates

Example 10-1

With XTAL = 11.0592 MHz, find the TH1 value needed to have the following baud rates.

(a) 9600

(b) 2400

(c) 1200

Solution:

With XTAL = 11.0592 MHz, we have:

The machine cycle frequency of the 8051 = $11.0592 \text{ MHz} / 12 = 921.6 \text{ kHz}$, and
 $921.6 \text{ kHz} / 32 = 28,800 \text{ Hz}$ is the frequency provided by UART to Timer 1 to set baud rate.

(a) $28,800 / 3 = 9600$

where -3 = FD (hex) is loaded into TH1

(b) $28,800 / 12 = 2400$

where -12 = F4 (hex) is loaded into TH1

(c) $28,800 / 24 = 1200$

where -24 = E8 (hex) is loaded into TH1

Notice that 1/12th of the crystal frequency divided by 32 is the default value upon activation of the 8051 RESET pin. We can change this default setting. This is explained at the end of this chapter.

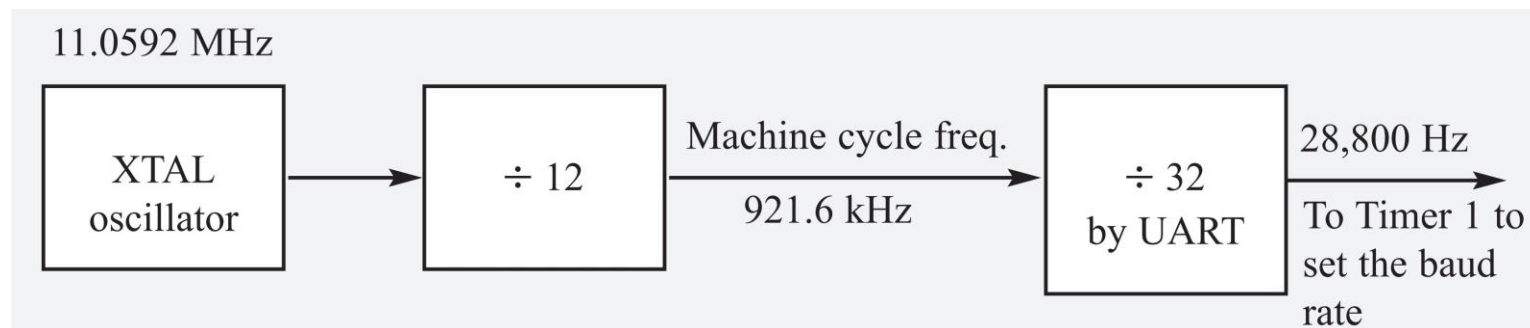


Figure 10-9. SCON Serial Port Control Register (Bit-Addressable)

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
-----	-----	-----	-----	-----	-----	----	----

SM0	SCON.7	Serial port mode specifier
SM1	SCON.6	Serial port mode specifier
SM2	SCON.5	Used for multiprocessor communication. (Make it 0.)
REN	SCON.4	Set/cleared by software to enable/disable reception.
TB8	SCON.3	Not widely used.
RB8	SCON.2	Not widely used.
TI	SCON.1	Transmit interrupt flag. Set by hardware at the beginning of the stop bit in mode 1. Must be cleared by software.
RI	SCON.0	Receive interrupt flag. Set by hardware halfway through the stop bit time in mode 1. Must be cleared by software.

Note: Make SM2, TB8, and RB8 = 0.

Example 10-2

Write a program for the 8051 to transfer letter "A" serially at 4800 baud, continuously.

Solution:

```
                MOV    TMOD,#20H    ;Timer 1, mode 2(auto-reload)
                MOV    TH1,#-6      ;4800 baud rate
                MOV    SCON,#50H    ;8-bit, 1 stop, REN enabled
                SETB   TR1          ;start Timer 1
AGAIN:          MOV    SBUF,"A"     ;letter "A" to be transferred
HERE:          JNB    TI,HERE       ;wait for the last bit
                CLR    TI           ;clear TI for next char
                SJMP   AGAIN        ;keep sending A
```

Example 10-3

Write a program to transfer the message “YES” serially at 9600 baud, 8-bit data, 1 stop bit. Do this continuously.

Solution:

```

                                MOV     TMOD,#20H      ;Timer 1, mode 2
                                MOV     TH1,#-3        ;9600 baud
                                MOV     SCON,#50H      ;8-bit, 1 stop bit, REN enabled
                                SETB    TR1            ;start Timer 1
AGAIN:                          MOV     A,"Y"          ;transfer "Y"
                                ACALL   TRANS
                                MOV     A,"E"          ;transfer "E"
                                ACALL   TRANS
                                MOV     A,"S"          ;transfer "S"
                                ACALL   TRANS
                                SJMP    AGAIN          ;keep doing it
;-----serial data transfer subroutine
TRANS:                          MOV     SBUF,A;load SBUF
HERE:                           JNB    TI,HERE        ;wait for last bit to transfer
                                CLR     TI            ;get ready for next byte
                                RET
```

Example 10-4

Program the 8051 to receive bytes of data serially, and put them in P1. Set the baud rate at 4800, 8-bit data, and 1 stop bit.

Solution:

```

MOV    TMOD,#20H    ;Timer 1, mode 2(auto-reload)
MOV    TH1,#-6      ;4800 baud
MOV    SCON,#50H    ;8-bit, 1 stop, REN enabled
SETB   TR1          ;start Timer 1
HERE:   JNB    RI,HERE    ;wait for char to come in
MOV    A,SBUF        ;save incoming byte in A
MOV    P1,A          ;send to port 1
CLR    RI            ;get ready to receive next byte
SJMP   HERE          ;keep getting data

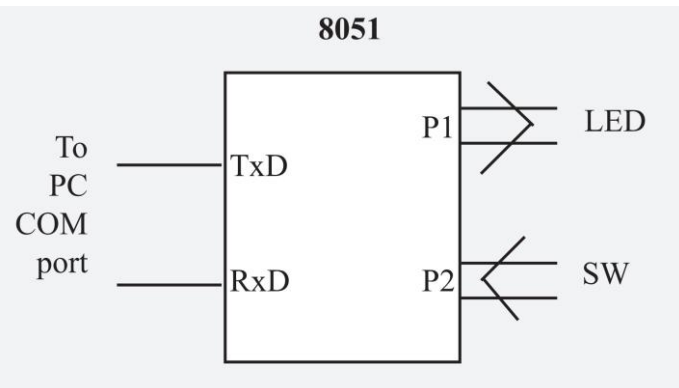
```

Example 10-5

Assume that the 8051 serial port is connected to the COM port of the IBM PC, and on the PC we are using the HyperTerminal program to send and receive data serially. P1 and P2 of the 8051 are connected to LEDs and switches, respectively. Write an 8051 program to (a) send to the PC the message “We Are Ready”, (b) receive any data sent by the PC and put it on LEDs connected to P1, and (c) get data on switches connected to P2 and send it to the PC serially. The program should perform part (a) once, but parts (b) and (c) continuously. Use the 4800 baud rate.

Example 10-5

Solution:



```
ORG 0
MOV P2,#0FFH ;make P2 an input port
MOV TMOD,#20H ;Timer 1, mode 2(auto-reload)
MOV TH1,#0FAH ;4800 baud rate
MOV SCON,#50H ;8-bit,1 stop, REN enabled
SETB TR1 ;start Timer 1
MOV DPTR,#MYDATA ;load pointer for message
H_1: CLR A
      MOVC A,@A+DPTR ;get the character
      JZ B_1 ;if last character get out
      ACALL SEND ;otherwise call transfer
      INC DPTR ;next one
      SJMP H_1 ;stay in loop
B_1: MOV A,P2 ;read data on P2
      ACALL SEND ;transfer it serially
      ACALL RECV ;get the serial data
      MOV P1,A ;display it on LEDs
      SJMP B_1 ;stay in loop indefinitely
;-----serial data transfer. ACC has the data
SEND: MOV SBUF,A ;load the data
H_2: JNB TI,H_2 ;stay here until last bit gone
      CLR TI ;get ready for next char
      RET ;return to caller
;-----receive data serially in ACC
RECV: JNB RI,RECV ;wait here for char
      MOV A,SBUF ;save it in ACC
      CLR RI ;get ready for next char
      RET ;return to caller
;-----The message
MYDATA: DB "We Are Ready",0
END
```

Doubling the Baud Rate in the 8051

1. Use a higher-frequency crystal.
2. Change a bit in the PCON register, shown below.

D7

D0

SMOD	--	--	--	GF1	GF0	PD	IDL
------	----	----	----	-----	-----	----	-----

Table 10-5: Baud Rate Comparison for SMOD = 0 and SMOD = 1

TH1	(Decimal)	(Hex)	SMOD = 0	SMOD = 1
	-3	FD	9,600	19,200
	-6	FA	4,800	9,600
	-12	F4	2,400	4,800
	-24	E8	1,200	2,400

Note: XTAL = 11.0592 MHz.

Example 10-6

Assuming that XTAL = 11.0592 MHz for the following program, state (a) what this program does, (b) compute the frequency used by Timer 1 to set the baud rate, and (c) find the baud rate of the data transfer.

```

        MOV     A,PCON           ;A = PCON
        SETB    ACC.7           ;make D7 = 1
        MOV     PCON,A          ;SMOD = 1, double baud rate
                                   ;with same XTAL freq.
        MOV     TMOD,#20H       ;Timer 1, mode 2(auto-reload)
        MOV     TH1,-3          ;19200 (57,600 / 3 = 19200 baud rate
                                   ;since SMOD=1)
        MOV     SCON,#50H       ;8-bit data,1 stop bit, RI enabled
        SETB    TR1             ;start Timer 1
        MOV     A,#"B"          ;transfer letter B
A_1:    CLR     TI              ;make sure TI=0
        MOV     SBUF,A          ;transfer it
H_1:    JNB     TI,H_1          ;stay here until the last bit is gone
        SJMP    A_1             ;keep sending "B" again and again
```

Example 10-6

Solution:

- (a) This program transfers ASCII letter B (01000010 binary) continuously.
- (b) With XTAL = 11.0592 MHz and SMOD = 1 in the above program, we have:

$11.0592 \text{ MHz} / 12 = 921.6 \text{ kHz}$ machine cycle frequency

$921.6 \text{ kHz} / 16 = 57,600 \text{ Hz}$ frequency used by Timer 1 to set the baud rate

$57,600 \text{ Hz} / 3 = 19,200$ baud rate

Example 10-7

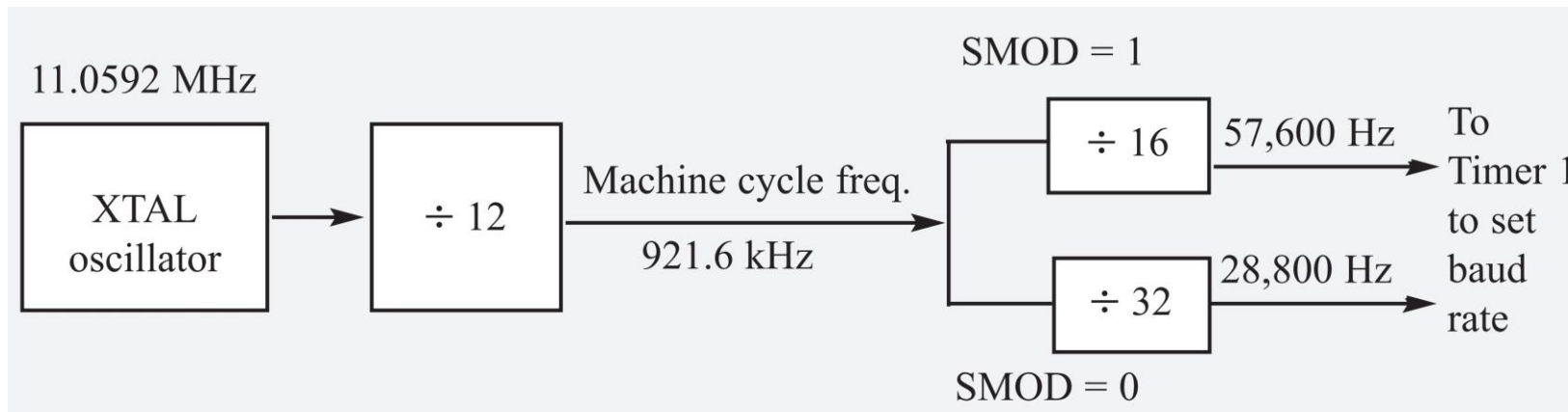
Find the TH1 value (in both decimal and hex) to set the baud rate to each of the following. (a) 9600 (b) 4800 if SMOD = 1 Assume that XTAL = 11.0592 MHz.

Solution:

With XTAL = 11.0592 MHz and SMOD = 1, we have Timer 1 frequency = 57,600 Hz.

(a) $57,600 / 9600 = 6$; therefore, TH1 = -6 or TH1 = FAH.

(b) $57,600 / 4800 = 12$; therefore, TH1 = -12 or TH1 = F4H.



Example 10-8

Find the baud rate if TH1 = -2, SMOD = 1, and XTAL = 11.0592 MHz. Is this baud rate supported by IBM/compatible PCs?

Solution:

With XTAL = 11.0592 MHz and SMOD = 1, we have Timer 1 frequency = 57,600 Hz. The baud rate is $57,600 / 2 = 28,800$. This baud rate is not supported by the BIOS of the PCs; however, the PC can be programmed to do data transfer at such a speed. Also, HyperTerminal in Windows supports this and other baud rates.

Example 10-9

Assume a switch is connected to pin P1.7. Write a program to monitor its status and send two messages to serial port continuously as follows: SW=0 send "NO" SW=1 send "YES"
Assume XTAL = 11.0592 MHz, 9600 baud, 8-bit data, and 1 stop bit.

Solution:

```

                                SW1      EQU P1.7
                                ORG       0H                      ;starting position
MAIN:                          MOV       TMOD,#20H
                                MOV       TH1,#-3                ;9600 baud rate
                                MOV       SCON,#50H
                                SETB      TR1                    ;start timer
                                SETB      SW1                    ;make SW an input
S1:                            JB        SW1,NEXT                ;check SW status
FN:                            MOV       DPTR,#MESS1             ;if SW=0 display "NO"
                                CLR       A
                                MOVC      A,@A+DPTR              ;read the value
                                JZ        S1                      ;check for end of line
                                ACALL     SENDCOM                  ;send value to serial port
                                INC       DPTR                    ;move to next value
                                SJMP      FN                      ;repeat
NEXT:                          MOV       DPTR,#MESS2             ;if SW=1 display "YES"
LN:                            CLR       A
                                MOVC      A,@A+DPTR              ;read the value
                                JZ        S1                      ;check for end of line
                                ACALL     SENDCOM                  ;send value to serial port
                                INC       DPTR                    ;move to next value
                                SJMP      LN                      ;repeat

;-----
SENDCOM: MOV       SBUF,A                      ;place value in buffer
HERE:    JNB       TI,HERE                    ;wait until transmitted
                                CLR       TI
                                RET                          ;clear
                                ;return
;-----
MESS1:   DB        "NO",0
MESS2:   DB        "YES",0
                                END

```

Example 10-10

Write a program to send the message "The Earth is but One Country" to serial port. Assume a SW is connected to pin P1.2. Monitor its status and set the baud rate as follows: SW = 0, 4800 baud rate
SW = 1, 9600 baud rate Assume XTAL = 11.0592 MHz, 8-bit data, and 1 stop bit.

Solution:

```

                                SW      BIT P1.2
                                ORG      0H                ;Starting position
MAIN:                          MOV      TMOD,#20H
                                MOV      TH1,#-6           ;4800 baud rate (default)
                                MOV      SCON,#50H
                                SETB     TR1
                                SETB     SW                ;make SW an input
S1:                            JNB      SW,SLOWSP          ;check SW status
                                MOV      A,PCON            ;read PCON
                                SETB     ACC.7             ;set SMOD High for 9600
                                MOV      PCON,A           ;write PCON
                                SJMP     OVER              ;send message
SLOWSP:                        MOV      A,PCON            ;read PCON
                                CLR      ACC.7            ;make SMOD Low for 4800
                                MOV      PCON,A           ;write PCON
OVER:                          MOV      DPTR,#MESS1       ;load address to message
FN:                            CLR      A
                                MOVC     A,@A+DPTR        ;read value
                                JZ       S1               ;check for end of line
                                ACALL    SENDCOM          ;send value to the serial port
                                INC      DPTR              ;move to next value
                                SJMP     FN               ;repeat
;-----
SENDCOM:
                                MOV      SBUF,A           ;place value in buffer
HERE:                          JNB      TI,HERE          ;wait until transmitted
                                CLR      TI               ;clear
                                RET                      ;return
;-----
MESS1:  DB      "The Earth is but One Country",0
                                END
```