



## Experiência No. 9 – Comunicação Serial Atualização do Relógio via Serial

1- Monte o circuito da Figura 1 no simulador.

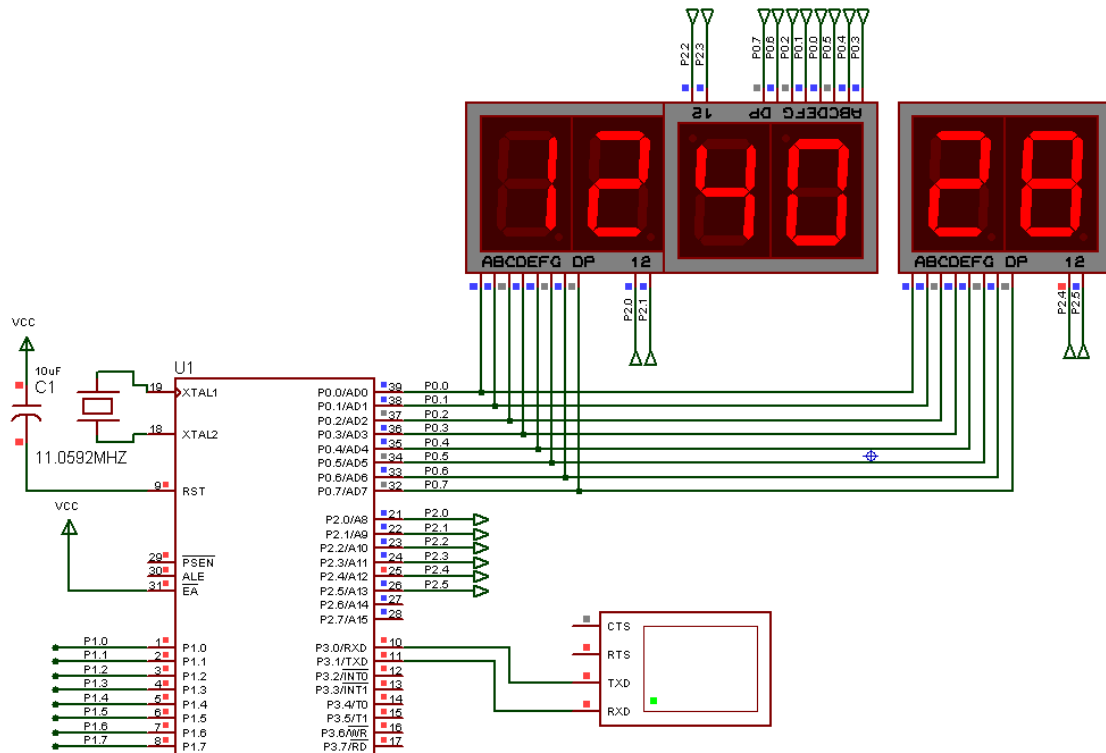


Figura 1

2- Configure o terminal serial para uma taxa de 19200bps, 8 bits de dados, sem paridade, 1 stop bit.

3- Escreva o código abaixo:

### DECLARAÇÃO DE VARIÁVEIS:

```
1 horas          equ    08h
2 minutos        equ    09h
3 segundos       equ    0Ah
4 milisegundos   equ    0Bh
5
6 CLOCK_ENTRYS   EQU     50
7 CLOCK_COUNTER  EQU     65536 - (11059200/12/CLOCK_ENTRYS)
```



### PROGRAMA PRINCIPAL:

```
9      .chip 8051
10     org      0
11     sjmp     init
12     org      0Bh
13     ljmp     clock
14 init: mov     sp, # (128-15)
15     MOV      IE, #10000010b      ;Habilita interrupção Timer 0
16     mov      TLO, #<CLOCK_COUNTER ;Inicializa TIMER1 (byte baixo)
17     mov      TH0, #>CLOCK_COUNTER ;Inicializa TIMER1 (byte alto)
18     mov      TMOD, #21h          ;Configura timer 0 e 1
19     setb     TR0                 ;Liga o temporizador
20 ;-----
21     mov      milisegundos, #CLOCK_ENTRYS
22     mov      horas, #00h         ;Inicializa os registros do relógio (Hora)
23     mov      minutos, #00h       ;Inicializa Minutos
24     mov      segundos, #00h      ;Inicializa Segundos
25
26     call     InitSerial
27
28     MOV      DPTR, #MENSAGEM_1
29     CALL     TEXT_OUT
30 ;-----
31 inicio: call   AT_DISPLAY
32     jnb      RI, inicio
33     clr      RI
34     mov      A, SBUF
35     CJNE     A, #'0', inicio
36     CALL     AT_RELOGIO_SERIAL
37     SJMP     inicio
38 ;-----
```

### SUBROTINA DE ATUALIZAÇÃO DO RELÓGIO PELA SERIAL:

```
39 AT_RELOGIO_SERIAL:
40     CLR      TR0
41     MOV      DPTR, #ENTER_HORA
42     CALL     TEXT_OUT
43     CALL     ReceiveByte
44     anl      a, #0Fh
45     swap     a
46     anl      horas, #0Fh
47     orl      horas, A
48     CALL     ReceiveByte
49     anl      a, #0Fh
50     anl      horas, #F0h
51     orl      horas, A
52
53     MOV      DPTR, #ENTER_MIN
54     CALL     TEXT_OUT
55     CALL     ReceiveByte
56     anl      a, #0Fh
57     swap     a
58     anl      minutos, #0Fh
59     orl      minutos, A
60     CALL     ReceiveByte
61     anl      a, #0Fh
62     anl      minutos, #F0h
63     orl      minutos, A
64
65     MOV      DPTR, #ENTER_SEC
66     CALL     TEXT_OUT
67     CALL     ReceiveByte
68     anl      a, #0Fh
69     swap     a
70     anl      segundos, #0Fh
71     orl      segundos, A
72     CALL     ReceiveByte
73     anl      a, #0Fh
74     anl      segundos, #F0h
75     orl      segundos, A
76     SETB     TR0
77     ret
```



## SUBROTINA DE ENVIAR STRINGS PELA SERIAL

```
143 ;-----
144 TEXT_OUT:
145     PUSH    ACC
146 WT1:     CLR     A
147     MOV     A, @A+DPTR
148     INC     DPTR
149     JZ      WT2
150     MOV     B, A
151     LCALL   SendByte
152     SJMP    WT1
153 WT2:     POP     ACC
154     RET
155 ;-----
156 MENSAGEM_1: DB 0Dh, 0Ah, '*** RELOGIO DIGITAL ***', 0Dh, 0Ah, 0
157 ENTER_HORA: DB 0Dh, 0Ah, 'ENTRE COM AS HORAS (00-23) : ', 0
158 ENTER_MIN:  DB 0Dh, 0Ah, 'ENTRE COM OS MINUTOS (00-59) : ', 0
159 ENTER_SEC:  DB 0Dh, 0Ah, 'ENTRE COM OS SEGUNDOS (00-59) : ', 0
160 ;-----
```

- 4- Adicione as seguintes sub-rotinas já implementadas em experiências anteriores:

AT\_DISPLAY, AT\_DPY\_SEC, AT\_DPY\_MIN, AT\_DPY\_HORA, CONVERT, DELAY, TABLE

- 5- Crie um novo arquivo chamado SERIAL.ASM e digite o código abaixo para as sub-rotinas seriais:

```
20 InitSerial:
21     MOV     SCON, #50h           ;Modo 1
22     MOV     TH1, #0FDh          ;Configura a taxa serial
23     ORL     PCON, #10000000B    ;para 19200bps
24     SETB    TR1                 ;Liga o Timer 1
25     RET

38 ;-----
39 SendByte:
40     mov     SBUF, B
41     jnb     TI, $
42     clr     TI
43     ret
44 ; Fim SendByte -----

54 ; -----
55 ReceiveByte:
56     call    AT_DISPLAY
57     JNB     RI, ReceiveByte
58     mov     A, SBUF
59     clr     RI
60     ret
61 ; Fim ReceiveByte -----
```

- 6- Faça o include no final do programa das rotinas do relógio já implementada [CLOCK.ASM] e [SERIAL.ASM].

```
178 ;-----
179     include clock.asm
180     include serial.asm
181
182     END
```

## Exercício

1. Implemente uma solução para fazer piscar os dois pontos do relógio em uma frequência de meio hertz.
2. Modifique o programa para que os dados de entrada pela serial sejam ecoados para o terminal.