



Experiência No. 10 – Relógio Digital com Mostrador LCD

1- Monte o circuito da Figura 1 no simulador.

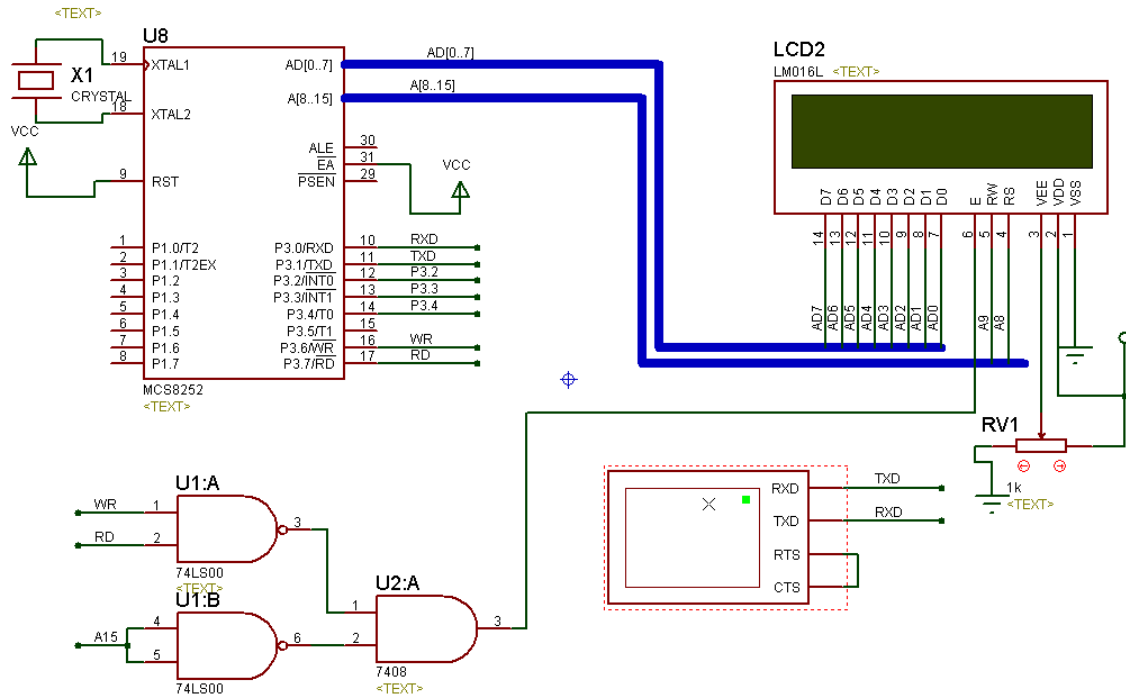


Figura 1

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

3- Escreve o código abaixo:

DECLARAÇÃO DE VARIÁVEIS:

```
1 horas           equ    08h
2 minutos         equ    09h
3 segundos        equ    0Ah
4 milisegundos    equ    0Bh
5
6 LCD_CMD_WR      equ    0000h
7 LCD_DATA_WR     equ    0100h
8 LCD_BUSY_RD     equ    0200h
9 LCD_DATA_RD     equ    0300h
10
11 CLOCK_ENTRYS   EQU     50
12 CLOCK_COUNTER  EQU     65536 - (11059200/12/CLOCK_ENTRYS)
```



INICIALIZAÇÃO:

```
14      .chip 8051
15      org    0
16      sjmp   init
17      org    0Bh
18      ljmp   clock
19 init:  mov    sp, #(128-15)
20      MOV    IE, #10000010b      ;Habilita interrupção Timer 0
21      mov    TLO, #<CLOCK_COUNTER ;Inicializa TIMER1 (byte baixo)
22      mov    TH0, #>CLOCK_COUNTER ;Inicializa TIMER1 (byte alto)
23      mov    TMOD, #21h          ;Configura timer 0 e 1
24      setb   TR0                 ;Liga o temporizador
25 ;-----
26      mov    milisegundos, #CLOCK_ENTRYS
27      mov    horas, #00h         ;Inicializa os registros do relógio (Hora)
28      mov    minutos, #00h       ;Inicializa Minutos
29      mov    segundos, #00h      ;Inicializa Segundos
30
31      call    InitSerial
32      lcall   init_lcd           ; Chama rotina de inicializacao do LCD
33      lcall   home              ; Chama rotina de HOME do LCD
34      lcall   limpa_lcd
35
36      mov    dptr, #mes_2 ; DPTR aponta para mes_2
37      call    DISPLAY_OUT ; Escreve mes_2 no LCD
38
39      MOV    DPTR, #MENSAGEM_1
40      CALL    TEXT_OUT
41 ;-----
```

PROGRAMA PRINCIPAL:

```
41 ;-----
42 inicio: call    AT_DISPLAY
43          jnb     RI, inicio
44          clr     RI
45          mov     A, SBUF
46          CJNE    A, #'0', inicio
47          CALL    AT_RELOGIO_SERIAL
48          lcall   limpa_lcd
49          lcall   home
50          mov     dptr, #mes_2      ; DPTR aponta para mes_2
51          call    DISPLAY_OUT      ; Escreve mes_2 no LCD
52          SJMP    inicio
53 ;-----
```

SUBROTINA DE ATUALIZAÇÃO DO RELÓGIO PELA SERIAL:

```
53 ;-----
54 AT_RELOGIO_SERIAL:
55     CLR     TR0
56     MOV     DPTR, #ENTER_HORA
57     CALL    TEXT_OUT
58     lcall   limpa_lcd
59     lcall   home
60     CALL    DISPLAY_OUT
61     CALL    ReceiveByte
62     anl     a, #0Fh
63     swap    a
64     anl     horas, #0Fh
65     orl     horas, A
66     CALL    ReceiveByte
67     anl     a, #0Fh
68     anl     horas, #F0h
69     orl     horas, A
70
71     MOV     DPTR, #ENTER_MIN
72     CALL    TEXT_OUT
73     lcall   limpa_lcd
74     lcall   home
75     CALL    DISPLAY_OUT
76     CALL    ReceiveByte
77     anl     a, #0Fh
78     swap    a
79     anl     minutos, #0Fh
80     orl     minutos, A
81     CALL    ReceiveByte
82     anl     a, #0Fh
83     anl     minutos, #F0h
84     orl     minutos, A
85
86     MOV     DPTR, #ENTER_SEC
87     CALL    TEXT_OUT
88     lcall   limpa_lcd
89     lcall   home
90     CALL    DISPLAY_OUT
91     CALL    ReceiveByte
92     anl     a, #0Fh
93     swap    a
94     anl     segundos, #0Fh
95     orl     segundos, A
96     CALL    ReceiveByte
97     anl     a, #0Fh
98     anl     segundos, #F0h
99     orl     segundos, A
100    SETB    TR0
101    ret
```



SUBROTINA ATUALIZAR DISPLAY LCD

```
102 ;-----
103 AT_DISPLAY:
104     lcall    cursor_linha2
105     CALL     AT_DPY_HORA
106     mov      a,#':'
107     call     lcd_data
108     CALL     AT_DPY_MIN
109     mov      a,#':'
110     call     lcd_data
111     CALL     AT_DPY_SEC
112     RET
114 AT_DPY_SEC:
115     mov      a,segundos
116     swap     a
117     anl      a,#0Fh
118     orl      a,#30h
119     call     lcd_data
120     mov      a,segundos
121     anl      a,#0Fh
122     orl      a,#30h
123     call     lcd_data
124     ret
125 ;-----

126 AT_DPY_MIN:
127     mov      a,minutos
128     swap     a
129     anl      a,#0Fh
130     orl      a,#30h
131     call     lcd_data
132     mov      a,minutos
133     anl      a,#0Fh
134     orl      a,#30h
135     call     lcd_data
136     ret
137 ;-----
138 AT_DPY_HORA:
139     mov      a,horas
140     swap     a
141     anl      a,#0Fh
142     orl      a,#30h
143     call     lcd_data
144     mov      a,horas
145     anl      a,#0Fh
146     orl      a,#30h
147     call     lcd_data
148     ret
```

4- Modifique as mensagens para adequação ao LCD:

```
165 ;-----
166 ; MENSAGENS DO LCD
167 mes_1:  db      '      UNIFOR      ',0
168 mes_2:  db      'RELOGIO DIGITAL ',0
169
170 ;-----
171 MENSAGEM_1: DB  0Dh,0Ah,'RELOGIO DIGITAL',0Dh,0Ah,0
172 ENTER_HORA: DB  0Dh,0Ah,'HORAS (00-23): ',0
173 ENTER_MIN:  DB  0Dh,0Ah,'MIN (00-59): ',0
174 ENTER_SEC:  DB  0Dh,0Ah,'SEG (00-59): ',0
175 ;-----
```

5- Modifique as rotinas do SERIAL.ASM :

```
56 ; -----
57 ReceiveByte:
58     JNE      RI,ReceiveByte
59     mov      A,SBUF
60     clr      RI
61     mov      B,A
62     call     SendByte
63     lcall    lcd_data
64     ret
```



6- Digite as rotinas do LCD abaixo em um arquivo nomeado LCD.asm

```
6 init_lcd:
7     mov A,#30h ; Inicio do reset
8     lcall lcd_inst ; do
9     lcall lcd_inst ; LCD
10    lcall lcd_inst
11    mov A,#38h ; Seleciona DL=1:8-bit
12    lcall lcd_inst ; 2 linhas para o display
13    lcall lcd_inst ; F=0:5x7dots
14    mov A,#0000110b ; Incrementa Cursor e NAO Desloca o Display
15    lcall lcd_inst ;
16    mov A,#00001100b ; Liga o Display, Liga o Cursor, NAO Blink
17    lcall lcd_inst
18    mov A,#00011000b ; Liga o Deslocamento do Cursor a Direita
19    lcall lcd_inst
20    ret

21 ;*****
22 ;* Rotina usada para executar uma instrucao no LCD. *
23 ;*****
24 lcd_inst:
25     push    dph
26     push    dpl
27     mov     dptr,#LCD_CMD_WR
28     movx    @dptr,A
29     lcall   wtbusy
30     pop     dpl
31     pop     dph
32     ret
33 ;*****
34 ;* Subroutine to wait for busy clear
35 ;*****
36 wtbusy:  PUSH    A
37 wtb:     mov     DPTR,#LCD_BUSY_RD
38           movx    A,@DPTR
39           jnb     ACC.7,wtb
40           POP     A
41           ret
42 ;*****
43 ;* Escreve dados no LCD
44 ;*****
45 lcd_data:
46     push    dph
47     push    dpl
48     mov     dptr,#LCD_DATA_WR
49     movx    @dptr,a
50     lcall   wtbusy
51     pop     dpl
52     pop     dph
53     ret
54 ;*****

57 DISPLAY_OUT:
58     PUSH    ACC
59 WTDISPLAY:
60     CLR     A
61     MOVC    A,@A+DPTR
62     CJNE    A,#0DH,test_OA
63     SJMP    nBYTE
64 test_OA:  CJNE    A,#0AH,sLCD
65     SJMP    nBYTE
66 sLCD:     JZ      WTLCD
67           LCALL   lcd_data
68 nBYTE:     INC     DPTR
69           SJMP    WTDISPLAY
70 WTLCD:
71           POP     ACC
72           RET
73 ;-----
74 limpa_lcd:
75     mov     A,#01b
76     call    lcd_inst
77     ret
78 ;-----
79 home:
80     MOV     A,#10b
81     CALL    lcd_inst
82     RET
83 ;-----
```

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

```
include clock.asm
include serial.asm
include lcd.asm
```

Exercício

1. Implemente uma solução para fazer piscar o blink quando da entrada dos valores do relógio.