



Experiência No. 5 – Relógio Digital Baseado no Timer 0

1- Monte o circuito da Figura 1 no simulador.

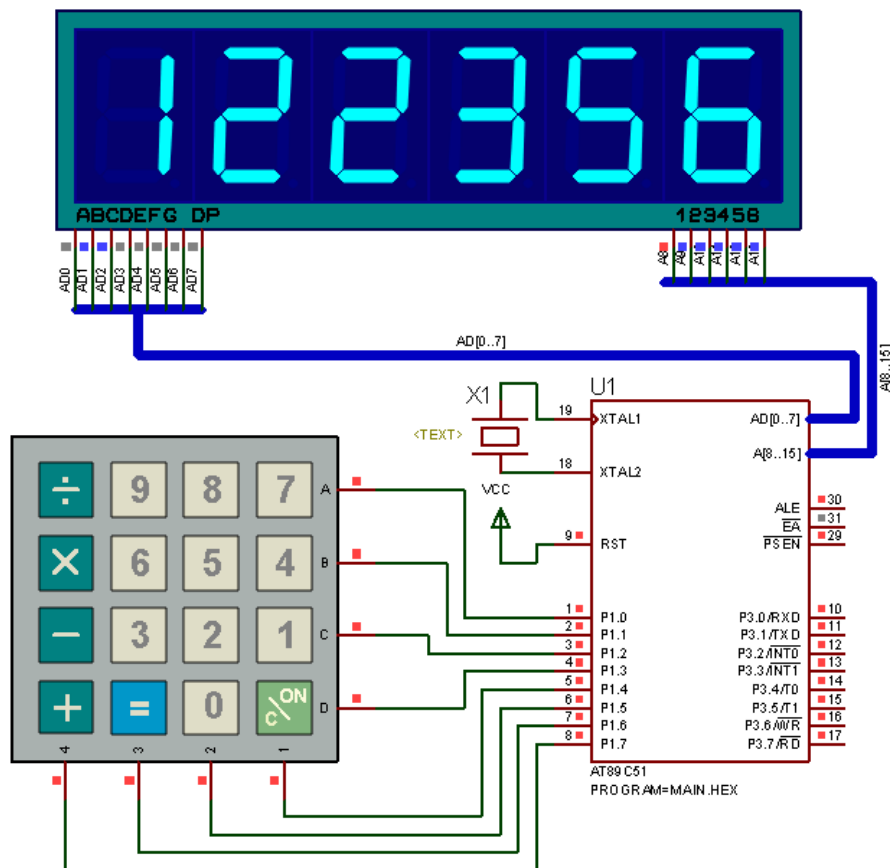


Figura 1

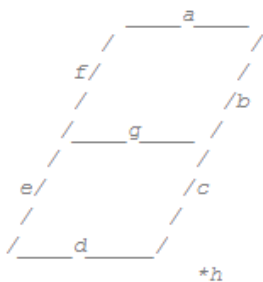
- 2- Digite o código fonte da pagina seguinte, compile e rode no simulador;
- 3- Altere os valores no código para que seja mostrado 23:15:17;



```
10      .chip 8051
11      org    0
12      sjmp    init
13      org    0Bh
14      ;      sjmp    clock
15 init:    mov    sp,#(128-15)
16      MOV     IE,#10000000b      ;Habilita interrupção Timer 0
17      mov     TL0,#<CLOCK_COUNTER      ;Inicializa TIMER1 (byte baixo)
18      mov     TH0,#>CLOCK_COUNTER      ;Inicializa TIMER1 (byte alto)
19      mov     TMOD,#01h      ;Configura timer 0 e 1
20      setb    TR0      ;Liga o temporizador
21      ;-----
22      mov     milisegundos,#CLOCK_ENTRYS
23      mov     horas,#12h      ;Inicializa os registros do relógio (Hora)
24      mov     minutos,#23h      ;Inicializa Minutos
25      mov     segundos,#56h      ;Inicializa Segundos
26      ;-----
27 inicio: call    AT_DISPLAY
28      sjmp    inicio
29      ;-----
30 AT_DISPLAY:
31      mov     p2,#00
32      CALL    AT_DPY_SEC      ;Chama rotina de atualiza o display de segundos
33      CALL    AT_DPY_MIN      ;Chama rotina de atualiza o display de minutos
34      CALL    AT_DPY_HORA      ;Chama rotina de atualiza o display de horas
35      mov     p2,#00
36      RET
37      ;-----
38 AT_DPY_SEC:
39      mov     a,segundos
40      call    convert
41      setb    p2.5
42      mov     p0,a
43      call    delay
44      mov     a,segundos
45      swap    a
46      call    convert
47      clr     p2.5
48      mov     p0,a
49      setb    p2.4
50      call    delay
51      ret
52      ;-----
53 AT_DPY_MIN:
54      mov     a,minutos
55      call    convert
56      clr     p2.4
57      mov     p0,a
58      setb    p2.3
59      call    delay
60      mov     a,minutos
61      swap    a
62      call    convert
63      clr     p2.3
64      mov     p0,a
65      setb    p2.2
66      call    delay
67      ret
68      ;-----
69 AT_DPY_HORA:
70      mov     a,horas
71      call    convert
72      clr     p2.2
73      mov     p0,a
74      setb    p2.1
75      call    delay
76      mov     a,horas
77      swap    a
78      call    convert
79      clr     p2.1
80      mov     p0,a
81      setb    p2.0
82      call    delay
83      ret
84      ;-----
85 convert: anl     a,#0Fh
86      mov     dptr,#table
87      movc    a,@a+dptr
88      cpl     a
89      ret
90      ;-----
91 delay:  mov     r7,#255
92      djnz    r7,$
93      ret
94      ;-----
```



```
95 ; Tabela de conversão dos valores binários (0-A) em 7 segmentos
96 ;-----
97 table: DB      00111111b      ; 0
98         DB      00000110b      ; 1
99         DB      01011011b      ; 2
100        DB      01001111b      ; 3
101        DB      01100110b      ; 4
102        DB      01101101b      ; 5
103        DB      01111101b      ; 6
104        DB      00000111b      ; 7
105        DB      01111111b      ; 8
106        DB      01101111b      ; 9
107        DB      01000000b      ; Ah = (-)
108        DB      01111001b      ; Bh = (E)
109        DB      01010000b      ; Ch = (r)
110        DB      01011100b      ; Dh = (o)
111        DB      00000000b      ; Eh = ( ) Display apagado
112 ;-----
113        END
```



a=d0
b=d1
c=d2
d=d3
e=d4
f=d5
g=d6
h=d7

Exercício:

- 1- Introduza no código a rotina abaixo para geração de uma base de tempo de 20mS (subrotina CLOCK). Habilite a interrupção do Timer 0 e descomente a chamada de interrupção do Timer no código anterior.

```
12 ;-----
13 clock:
14     push    A                      ;Salva acumulador
15     mov     TL0,#<CLOCK_COUNTER    ;Reinicializa TIMER1 (byte baixo)
16     mov     TH0,#>CLOCK_COUNTER    ;Reinicializa TIMER1 (byte alto)
17     djnz    milisegundos,$Exit      ;Se não passou 1 s, finaliza
18     mov     milisegundos,#CLOCK_ENTRYS ;
19
20     mov     a,segundos
21     add     a,#01h                  ;Incrementa segundos
22     da      a
23     mov     segundos,a              ;Move para comparar
24     cjne    A,#60h,$Exit            ;Verifica se chegou 60 segundos
25     mov     segundos,#00h           ;Inicializa contador de segundos
26
27     mov     A,minutos
28     add     a,#01h                  ;Incrementa minuto
29     da      a
30     mov     minutos,a              ;Move para comparar
31     cjne    A,#60h,$Exit            ;Verifica se chegou 60 minutos
32     mov     minutos,#00h           ;Inicializa contador de minutos
33
34     mov     A,horas
35     add     a,#01h                  ;Incrementa hora
36     da      a
37     mov     horas,a                ;Move para comparar
38     cjne    A,#24h,$Exit            ;Verifica se chegou 24 horas
39     mov     horas,#00h             ;Inicializa contador de horas
40 $Exit:
41     pop     A                      ;Restaura acumulador
42     RETI
43 ; Fim Clock -----
```