

Atividade 1- Desafio 2 11/04/2023

Microcontroladores TT516A

Profa. Dra. Talia Simões dos Santos Ximenes

Gustavo Melo Scheidt Paulino - 250358

```
1      T EQU (65535-50000)           ; T = 15535
2      TEMPO EQU 200                 ; Variável TEMPO = 200
3
4      ORG 0000h                     ; Endereço em 0000h
5      LJMPL main                     ; Long Jump para "main", ou seja, endereço "main"
6
7      ORG 0003h                     ; Endereço em 0003h
8      LCALL acende                   ; Long Jump (vai para) "acende"
9      JMP ligatimer                  ; Jump para "ligatimer"
10     RETI                           ; Retorno da interrupção
11
12     ORG 000Bh                     ; Endereço em 000Bh
13     NOP                           ; No operation (não ocorre nada)
14     RETI                           ; Retorno da interrupção
15
16     ORG 0013h                     ; Endereço em 0013h
17     NOP                           ; No operation (não ocorre nada)
18     RETI                           ; Retorno da subrotina
19
20     ORG 001Bh                     ; Endereço em 001Bh
21     JMP repete                     ; Jump para subrotina repete
22
23     conta: LCALL liga               ; Long Call para subrotina "Liga"
24     RETI                           ; Retorno da interrupção
25     ORG 0023h                     ; Endereço em 0023h
26     NOP                           ; No operation (não ocorre nada)
27     RETI                           ; Retorno da interrupção
28     ORG 0040h                     ; Endereço em 0040h
29
30     main: MOV IE,#10001001b        ; Atribui o valor binário "10001001" ao INTERRUPT ENABLE (EX0, ET1 E EA = 1)
31           MOV TMOD,#0001000b       ; Atribui o valor binário "0001000" ao Timer Mode
32           MOV TCON,#00000000b      ; Atribui o valor binário "00000000" ao Timer Control
33           MOV IP,#00000001b        ; Atribui o valor binário "00000001" ao INTERRUPT PRIORITY (PX0 = 1)
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
34     LCALL apaga                   ; Chamada longa para label "apaga"
35
36     ini: MOV A,#00                 ; Atribui ao acumulador A o valor de "00"
37
38     loop: INC A                    ; Incrementa o acumulador A
39           CJNE A,#0FFh,loop        ; Compare JUMP Not EQUAL, em outras palavras, enquanto A não for 0FFh ele dá um jump para o label loop, ou seja, fica nesse loop de incremento
40
41
42     continua: SJMP ini             ; Short jump (vai para) label "ini"
43
44     ligatimer: LCALL interruptimer  ; Chamada longa para label "interruptimer", ou seja, vai para "interruptimer" por meio de um jump, "resolve" o que tem dentro e depois volta para a linha de baixo
45     RETI                           ; Retorno da interrupção
46
47     repete: CLR TR1                ; De um clear em TR1 (zera TR1)
48           CLR TF1                  ; De um clear em TF1 (zera TF1)
49           DJNZ R7, conta           ; Decrementa R7 e JUMPA para "conta" caso R7 NÃO seja 0
50     LCALL apaga                   ; Long call para "apaga"
51     RETI                           ; Retorno da interrupção
52
53     interruptimer: MOV R7,#TEMPO   ; Atribui ao registrador R7 o valor de "TEMPO"
54
55     liga: MOV TH1,#HIGH(T)         ; Atribui ao TH1 o valor alto de T
56           MOV TL1,#LOW(T)         ; Atribui ao TL1 o valor baixo de T
57           SETB TR1                ; Set bit em TR1 (TR1 = 1)
58     RETI                           ; Retorno da subrotina
59
60     acende: MOV P1,#0FFh           ; Atribui à porta P1 o valor de 0FFh
61     RETI                           ; Retorno da subrotina
62
63     apaga: MOV P1,#0               ; Atribui o valor de 0 para a porta P1
64     RETI                           ; Retorno da subrotina
65     END                           ; Finalização do programa
66
```

Exemplo.asm

```
34 para label "apaga"
35
36 acumulador A o valor de "00"
37
38 acumulador A
39 Not EQUAL, em outras palavras, enquanto A não for 0FFh ele dá um jump para o label loop, ou seja, fica nesse loop de incrementação de A
40
41
42 vai para) label "ini"
43
44 para label "interruptimer", ou seja, vai para "interruptimer" por meio de um jump, "resolve" o que tem dentro e depois volta para a linha de baixo
45 interrupção
46
47 m TR1 ( zera TR1)
48 m TF1 ( zera TF1)
49 e JUMPA para "conta" caso R7 NÃO seja 0
50 a "apaga"
51 interrupção
52
53 gistrador R7 o valor de "TEMPO"
54
55 li o valor alto de T
56 li o valor baixo de T
57 li ( TR1 = 1)
58 brotina
59
60 ta P1 o valor de 0FFh
61 brotina
62
63 or de 0 para a porta P1
64 brotina
65 io programa
66
```

Nesse programa, o objetivo é acender e apagar periodicamente os LEDs conectados à porta P1 do microcontrolador 8051. Esse processo é controlado por meio das interrupções de INT0, INT1, TIMER0, TIMER1 e SERIAL. Cada uma dessas interrupções tem um papel específico no controle dos LEDs e, quando ocorrem, o programa salta para um endereço de memória correspondente. Isso permite que o microcontrolador execute a rotina correspondente à interrupção e, em seguida, retorne ao fluxo normal de execução do programa.

A variável "TEMPO" determina o tempo em que os LEDs permanecem acesos. Isso significa que, quando uma interrupção ocorre, o programa salta para o endereço correspondente, os LEDs permanecem acesos por um determinado período de tempo, que é definido pela variável "TEMPO". Esse tempo é contado pelo registrador R7, que é decrementado a cada ciclo de clock. Quando o valor de R7 chega a zero, os LEDs são apagados e o programa retorna ao fluxo normal de execução. O controle de tempo é feito por meio dos timers TIMER0 e TIMER1, que são zerados durante as rotinas de interrupção e iniciam a contagem novamente após a conclusão da rotina. Caso não haja interrupções, o programa permanecerá em loop infinito.