

SEL0614 - Microprocessadores e Aplicações

Lista 1

Murilo Mussatto

11234245

1. Fazer um programa que utilize um timer interno do 8051 para criar um tempo de atraso de 0.05 segundos. Utilizando este programa como uma sub-rotina, escrever um programa que atrase 5 segundos.

Solução:

```
1 ;----- Jumps -----
2     ORG      0
3     SJMP     PROG
4 ;*****
5     ORG      000Bh
6     SJMP     OFT0
7 ;*****
8
9 ;----- Programa -----
10 PROG:
11     SETB     EA           ;Habilita Interrupcoes
12     SETB     ET0          ;Habilita Timer 0
13     MOV      TMOD, #01    ;Timer0: Ativacao por Software, Timer e Modo 1
14
15     MOV      A, #100      ;Contador vai rodar 100 vezes para resultar em 5s
16     CLR      20h.0        ;Controle do Timer
17
18 LOOP:
19     JB       20h.0, $      ;Espera enquanto 20h.0 = 1
20     JNZ      LOOP         ;Loop enquanto ACC nao for zero
21     ACALL    TIMER_0      ;Chama timer de 0,05s
22
23
24     SJMP     $            ;Fim Logico do Progrma
25
26
27
28 TIMER_0:
29     MOV      TH0, #03Ch
30     MOV      TL0, #0B0h
31     SETB     TR0          ;Inicia Timer
32     SETB     20h.0        ;Deixa o programa em Loop
33     RET
34
35
36 ;----- Contador -----
37 OFT0:
38     CLR      20h.0
39     DEC      A
40     RETI
41
42 ;----- END -----
43     END
```

2. Um sistema baseado no 8051 utiliza as duas interrupções externas disponíveis e ainda a interrupção gerada por 1 dos Temporizadores/Contadores. As condições em que se pretende que o sistema funcione são as seguintes:

- a interrupção externa 0 deve ser sempre atendida imediatamente e deve copiar o que está na posição de RAM externa 4000H para a posição 4200H;
- a interrupção externa 1 deve escrever o que está em 4200H na porta P1;
- a interrupção gerada pelo timer deve executar uma rotina que copie o que está na porta P2 para a posição 4000H da RAM externa;
- No caso de duas interrupções acontecerem simultaneamente, deve ser atendida a interrupção externa.

Solução:

```

1  ;----- Jumps -----
2      ORG      0
3      SJMP     PROG
4  ;*****
5      ORG      0003h
6      SJMP     EXT0
7  ;*****
8      ORG      000Bh
9      SJMP     OFT0
10 ;*****
11 ORG      0013h
12 SJMP     EXT1
13 ;*****
14
15 ;----- Programa -----
16 PROG:
17     SETB     EA      ;Habilita Interrupcoes
18     SETB     EX0     ;Habilita Interrupcao Externa 0
19     SETB     EX1     ;Habilita Interrupcao Externa 1
20
21     SETB     IT0     ;Habilita Descida de Borda para EX0
22     SETB     IT1     ;Habilita Descida de Borda para EX1
23
24     SETB     PX0     ;Alta prioridade para Interrupcao Externa 0
25     SETB     PX1     ;Alta prioridade para Interrupcao Externa 1
26
27     SETB     ET0     ;Habilita Timer 0
28     MOV      TMOD,#2 ;Timer0: Ativacao por Software, Timer e Modo 2
29
30     MOV      TH0,#0FCh
31     MOV      TLO,#018h
32     SETB     TR0     ;Inicia Timer
33
34     ;Valores Aleatorios Iniciais
35     MOV      P2,#0AAh
36
37     MOV      A,#0Ah
38     MOV      DPTR, #4000H
39     MOVX     @DPTR, A
40
41     MOV      A, #0FFh
42     MOV      DPTR, #4200H
43     MOVX     @DPTR, A
44
45     SJMP     $      ;Fim Logico do Programa
46
47
48 ;----- Interrupcoes -----
49 ;**** Externa 0 ****
50 EXT0:
51     CLR      EA
52
53     MOV      DPTR, #4000h
54     MOVX     A, @DPTR
55     MOV      DPTR, #4200h
56     MOVX     @DPTR, A
57
58     SETB     EA
59     RETI

```

```

60
61 ;**** Externa 1 ****
62 EXT1:
63     MOV     DPTR,#4200h
64     MOVX    A,@DPTR
65     MOV     P1,A
66
67     RETI
68
69 ;**** Timer 0 ****
70 OFT0:
71     MOV     A,P2
72     MOV     DPTR,#4000h
73     MOVX    @DPTR,A
74
75     RETI
76 ;----- END -----
77     END
78

```

3. Considere o Controlador de Nível da figura operando da seguinte maneira:

- Dois sensores S1 e S2 emitem nível lógico zero se estiverem fora do líquido e nível lógico 1 se estiverem imersos no líquido.
- Uma válvula V1, acionada pelo bit P1.0 de um microcontrolador 8051 drena o reservatório e uma válvula V2 acionada pelo bit P1.1 enche-o com líquido.
- Inicialmente o reservatório está vazio, ou seja, com os dois sensores em nível lógico zero.

Escrever um programa em Assembly do 8051 que mantenha o nível do líquido próximo à metade do reservatório (1/2 H) automaticamente, utilizando as entradas de Interrupções assinaladas, observando que as mesmas sentem a mudança de borda (descida) quando o líquido passa pelos sensores (S1 → Drenando e S2 → enchendo)

Solução:

```
1
2 ;----- Jumps -----
3     ORG      0
4     SJMP     PROG
5 ;*****
6     ORG      0003h
7     SJMP     EXT0
8 ;*****
9     ORG      0013h
10    SJMP     EXT1
11 ;*****
12
13 ;----- Programa -----
14 PROG:
15     SETB     EA      ;Habilita Interrupcoes
16
17     SETB     EX0      ;Habilita Interrupcao Externa 0
18     SETB     EX1      ;Habilita Interrupcao Externa 1
19
20     SETB     IT0      ;Habilita Descida de Borda para EX0
21     SETB     IT1      ;Habilita Descida de Borda para EX1
22
23     SETB     P1.1      ;Abre valvula V2 para encher o tanque
24     CLR      P1.0      ;Fecha a valvula V1 de dreno
25
26     SJMP     $         ;Fim Logico do Programa
27
28 ;----- Interrupcoes -----
29 ;**** Externa 0 ****
30 EXT0:
31     SETB     P1.1      ;Abre valvula V2 para encher o tanque
32     CLR      P1.0      ;Fecha valvula V1
33     RETI
34
35 ;**** Externa 1 ****
36 EXT1:
37     SETB     P1.0      ;Abre valvula V1 para drenar o tanque
38     CLR      P1.1      ;Fecha valvula V2
39     RETI
40
41 ;----- END -----
42     END
43
```

4. Escrever um programa em Assembly do 8051 que controle o dispositivo de teste térmico de materiais, mostrado na figura. Um recipiente, com determinada substância sob teste, deve ser baixado (Dir = P2.7 = 1) através de um Motor (Ligar = P2.6 = 1), dentro de um forno. O Sensor1 detecta a presença do recipiente e envia uma descida de borda ao pino Int0 do microprocessador. O micro para o Motor (Ligar = P2.6 = 0) e aciona o aquecimento do forno (P1.2 = 1) durante aproximadamente 500 ms. Desliga o aquecimento, inverte o sentido do Motor (Ligar = P2.6 = 1) (Dir = P2.7 = 0), erguendo o recipiente até a posição do Sensor2, que opera da mesma forma que o Sensor1, mas usando a Interrupção Int1. Quando Int1 receber uma descida de borda, o micro deve parar o Motor (Ligar = P2.6 = 0) e acionar o resfriamento (P1.0 = 1) durante aproximadamente 500 ms. O ciclo deve ser repetido 3 vezes e parar. Considerar o Cristal da CPU de 12 MHz.

- Fornecer o programa em Assembly
- Fornecer os parâmetros de programação dos tempos envolvidos.

Solução:

```

1
2 ;----- Jumps -----
3     ORG          0
4     SJMP         PROG
5 ;*****
6     ORG          0003h
7     SJMP         EXT0
8 ;*****
9     ORG          0013h
10    SJMP         EXT1
11 ;*****
12
13 ;----- Programa -----
14 PROG:
15     SETB         EA           ;Habilita Interrupcoes
16
17     SETB         EX0          ;Habilita Interrupcao Externa 0
18     SETB         EX1          ;Habilita Interrupcao Externa 1
19
20     SETB         IT0          ;Habilita Descidada de Borda para EX0
21     SETB         IT1          ;Habilita Descidada de Borda para EX1
22
23     MOV P2,#0
24     MOV P1,#0
25
26     MOV A,#03                ;Ciclo ocorrera 3 vezes
27
28 LOOP:
29     SETB         20h.0        ;Controle do Ciclo
30
31     SETB         P2.7         ;Direcao p/ baixo
32     SETB         P2.6         ;Liga o motor
33
34
35     JB          20h.0, $       ;Aguarda o Ciclo Acabar
36     DJNZ        ACC,LOOP      ;Se A diferente de 0, ainda restam ciclos
37
38     SJMP        $             ;Fim Logico do Programa
39
40 ;----- Delay -----
41 DELAY:
42 ; START: Wait loop, time: 500 ms
43 ; Clock: 12000.0 kHz (12 / MC)
44 ; Used registers: R0, R1, R2
45     MOV R2, #004h
46     MOV R1, #0FAh
47     MOV R0, #0F8h
48     NOP
49     DJNZ        R0, $
50     DJNZ        R1, $-5
51     DJNZ        R2, $-9
52 ; Rest: -13.0 us
53 ; END: Wait loop
54     RET
55
56 ;----- Interrupcoes -----

```

```

57 ;**** Externa 0 ****
58 EXT0:
59     CLR      P2.6      ;Desliga Motor
60     SETB     P1.2      ;liga Forno
61     ACALL    DELAY     ;Delay de 500ms
62     CLR      P1.2      ;Deliga Forno
63     CLR      P2.7      ;Direcao p/ cim
64     SETB     P2.6      ;Liga Motor
65     RETI
66
67 ;**** Externa 1 ****
68 EXT1:
69     CLR      P2.6      ;Desliga Motor
70     SETB     P1.0      ;Liga Resfriamento
71     ACALL    DELAY     ;Delay de 500ms
72     CLR      P1.0      ;Desliga Resfriamento
73     CLR      20h.0     ;Controle do Ciclo
74     RETI
75
76 ;----- END -----
77     END
78

```

5. Dois microcontroladores 8051 estão se comunicando através de uma interface padrão RS232 com handshaking via RTS e CTS. No esquema abaixo conectar os fios do cabo de comunicação corretamente e responder aos itens:

- Qual o valor de TH1 em ambos os micros se a taxa de comunicação é de 19200 bauds?
- Qual o valor de tensão na linha de comunicação quando esta está em repouso?
- Qual o tamanho, em microssegundos, do Start Bit?

Solução:

- O valor de TH1 pode ser calculado por meio da fórmula:

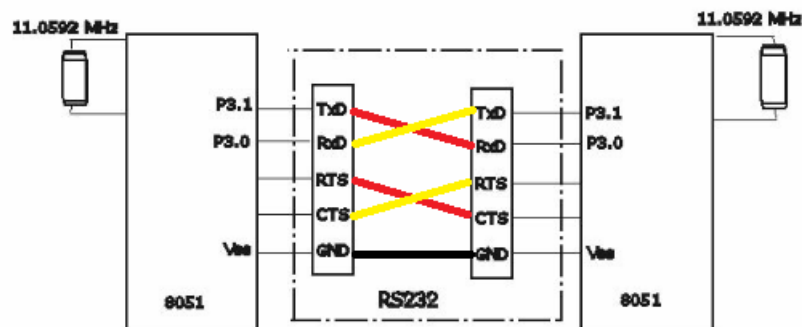
$$TH1 = 256 - \frac{K \cdot 11,0592 \cdot 10^6}{32 \cdot 12 \cdot \text{BaudRate}}$$

Utilizando $K = 2$ e $\text{BaudRate} = 19200$, temos:

$$TH1 = 256 - \frac{2 \cdot 11,0592 \cdot 10^6}{32 \cdot 12 \cdot 19200} = 253$$

- Como é adotado a lógica negativa para o padrão RS232, a tensão está entre -25 e -3 Volts quando a linha está em repouso
- O tamanho do StartBit pode ser calculado como o inverso do BaudRate.

$$\text{tam} = \frac{1}{19200} \approx 52,08 \mu s$$



6. Um sistema baseado no Microcontrolador 8051 utiliza as duas interrupções externas e as interrupções geradas pelos Timers/Counters. Escrever um programa em Assembly tal que:

- A interrupção externa 0, atendida prioritariamente, deve trocar o que está na Porta P1 com o que está contido na posição de RAM externa 5000h.
- A interrupção externa 1, com baixa prioridade, deve transferir o que está armazenado na RAM externa do endereço 5000h para a RAM interna no endereço 7Fh;
- A interrupção gerada pelo T/C 0 (a cada 10 ms) com alta prioridade, deve executar uma rotina que copie o que está no endereço da RAM interna 7Fh para a posição 5200h da RAM externa;
- A cada 60 milissegundos (aproximadamente) e controlado pelo T/C 1 com interrupção de baixa prioridade, o dado armazenado na RAM externa no endereço 5200h deve ser enviado para a posição de memória externa 5000h.
- O programa deve ficar em Loop infinito sempre executando o algoritmo acima.
- Considerar o cristal do oscilador de 12 MHz.

Solução:

```

1  ;----- Jumps -----
2  ORG          0
3  SJMP         PROG
4
5  ;*****
6  ORG          0003h
7  SJMP         EXT0
8  ;*****
9  ORG          000Bh
10 SJMP         OFT0
11 ;*****
12 ORG          0013h
13 SJMP         EXT1
14 ;*****
15 ORG          001Bh
16 SJMP         OFT1
17 ;*****
18
19 ;----- Programa -----
20 PROG:
21     SETB      EA          ;Habilita Interrupcoes
22     SETB      EX0         ;Habilita Interrupcao Externa 0
23     SETB      EX1         ;Habilita Interrupcao Externa 1
24
25     SETB      IT0         ;Habilita Descida de Borda para EX0
26     SETB      IT1         ;Habilita Descida de Borda para EX1
27
28     SETB      ET0         ;Habilita Timer 0
29     SETB      ET1         ;Habilita Timer 1
30     MOV       TMOD, #11h   ;Timer1: Software, Timer e Modo 1; Timer0: Software,
31                               Timer e Modo 1
32
33     MOV       TH0, #0D8h   ;Timer 0: 10ms
34     MOV       TL0, #0F0h
35     SETB      TR0         ;Inicia Timer 0
36
37     MOV       TH1, #015h   ;Timer 1: 60ms
38     MOV       TL1, #0A0h
39     SETB      TR1         ;Inicia Timer 1
40
41     SETB      PX0         ;Alta prioridade para Interrupcao Externa 0
42     CLR       PX1         ;Baixa prioridade para Interrupcao Externa 1
43     SETB      PT0         ;Alta prioridade para o Timer 0
44     CLR       PT1         ;Baixa prioridade para o Timer 1
45
46     ;Valores Aleatorios Iniciais
47     MOV       P1, #0AAh
48
49     MOV       7Fh, #0Fh
50
51     MOV       A, #0F0h
52     MOV       DPTR, #5000H

```



```

53      MOVX    @DPTR, A
54
55      SJMP    $           ;Fim Logico do Programa
56
57
58      ;----- Interrupcoes -----
59      ;**** Externa 0 ****
60      EXT0:
61          PUSH    ACC
62          PUSH    DPL
63          PUSH    DPH
64
65          MOV     DPTR,#5000h ;Salva o que esta em 500h Externo em B
66          MOVX    A,@DPTR
67          MOV     B,A
68
69          MOV     A,P1       ;Salva P1 em 5000h Externo
70          MOVX    @DPTR,A
71
72          MOV     P1,B       ;Transfere B para P1
73
74          POP     DPH
75          POP     DPL
76          POP     ACC
77
78          RETI
79
80      ;**** Externa 1 ****
81      EXT1:
82          MOV     DPTR,#5000h
83          MOVX    A,@DPTR
84          MOV     7Fh,A
85
86          RETI
87
88      ;**** Timer 0 ****
89      OFT0:
90          PUSH    ACC
91          PUSH    DPL
92          PUSH    DPH
93
94          MOV     A,7Fh      ;Transfere o que estava em 7Fh
95          MOV     DPTR,#5200h ;para 5200h na memoria externa
96          MOVX    @DPTR,A
97
98          MOV     TH0,#0D8h  ;Timer 0: 10ms
99          MOV     TLO,#0F0h
100
101          POP     DPH
102          POP     DPL
103          POP     ACC
104
105          RETI
106
107      ;**** Timer 1 ****
108      OFT1:
109
110          MOV     DPTR,#5200h
111          MOVX    A,@DPTR
112          MOV     DPTR,#5000h
113          MOVX    @DPTR,A
114
115          MOV     TH1,#015h  ;Timer 1: 60ms
116          MOV     TL1,#0A0h
117
118          RETI
119      ;----- END -----
120      END
121

```

7. Fazer um programa em Assembly do 8051 que calcule o valor da frequência de uma onda quadrada entrando pelo pino da Interrupção Externa 0 e envie-o para a interface serial RS232 a uma taxa de 4800,N,8,1. Considerar o cristal da CPU de 11,0592 MHz. Utilizar a interrupção Int0 sensível à descida de borda. O valor da frequência a ser enviada para a interface serial é um número hexadecimal de 16 Bits.

Solução:

```

1  ;----- Jumps -----
2      ORG          0
3      SJMP         PROG
4  ;*****
5      ORG          0003h
6      SJMP         EXT0
7  ;*****
8  ;----- Programa -----
9  PROG:
10     SETB         EA           ;Habilita Interrupcoes
11     SETB         EX0          ;Habilita Interrupcao Externa 0
12     SETB         IT0          ;Habilita Descida de Borda para EX0
13
14     MOV          SCON,#40h    ;Modo 1 (Variavel 10bits)
15
16     MOV          TMOD,#20h    ;Timer1: Software, Timer e Modo 2;
17
18     MOV          PCON,#00h    ;K = 1 --> Baud = 4800
19
20     MOV          TH1,#250     ;Timer 1: recarregavel com 250 decimal
21     MOV          TL1,#250
22     SETB         TR1         ;Inicia Timer 1
23
24
25  LOOP:
26     MOV          DPTR,#0      ;Reset DPTR
27
28     ACALL        DELAY
29
30     MOV          SBUF,DPL     ;Enviar Menos Significativo
31     JNB          TI,$
32     CLR          TI
33
34     MOV          SBUF,DPH     ;Enviar Mais Significativo
35     JNB          TI,$
36     CLR          TI
37
38     SJMP         LOOP
39
40 ;----- DELAY -----
41 DELAY:
42 ; START: Wait loop, time: 1 s
43 ; Clock: 12000.0 kHz (12 / MC)
44 ; Used registers: R0, R1, R2, R3
45     MOV          R3, #003h
46     MOV          R2, #0D2h
47     MOV          R1, #024h
48     MOV          R0, #014h
49     NOP
50     DJNZ         R0, $
51     DJNZ         R1, $-5
52     DJNZ         R2, $-9
53     DJNZ         R3, $-13
54     MOV          R0, #059h
55     DJNZ         R0, $
56     NOP
57 ; Rest: 0
58 ; END: Wait loop
59     RET
60 ;----- Interrupcoes -----
61 ;*** Externa 0 ***
62 EXT0:
63     INC          DPTR
64     RETI
65
66 ;----- END -----
67     END

```

8. Escrever um programa em Assembly do 8051 que armazene uma tabela com 10 números na memória de programa. Ler cada um dos números e se o número for par, enviar para a porta P1 e para a interface serial RS232 a uma velocidade de 9600,N,8,1. Se o número for ímpar enviar para a porta P2 e para a interface serial RS232, a uma taxa de 4800,N,8,1. Contar o número de números pares e ímpares encontrados e armazenar o valor na RAM externa, nos seguintes endereços: (2030h) = quantidade de números pares e (2031h) = quantidade de números ímpares. O cristal é de 11.0592 MHz.

Solução:

```

1  ;----- Jumps -----
2      ORG          0
3      SJMP         PROG
4  ;*****
5
6  ;----- Programa -----
7  PROG:
8      SETB         EA          ;Habilita Interrupcoes
9
10     MOV          SCON,#40h    ;Modo 1 (Variavel 10bits)
11
12     MOV          TMOD,#20h    ;Timer1: Software, Timer e Modo 2;
13
14     ; Formula para TH1 do Timer: 256 - (K*110592*10^6)/(32*12*BaudRate)
15     ; Com K = 1 e BaudRate = 4800 --> TH1 = 250
16     ; Com K = 1 e BaudRate = 9600 --> TH1 = 253
17     ; Com K = 2 e BaudRate = 9600 --> TH1 = 250
18     ; Logo, eh possivel alterar apenas o valor de K (SMOD) e
19     ; manter o contador com 250
20
21     MOV          TH1,#250     ;Timer 1: recarregavel com 250 decimal
22     MOV          TL1,#250
23     SETB         TR1         ;Inicia Timer 1
24
25
26     MOV          DPTR,#VET    ;Endereco do vetor
27     MOV          R0,#0FFh     ;Indice do Vetor
28     MOV          R1,#0        ;Quantidade de Nros Impares
29     MOV          R2,#0        ;Quantidade de Nros Pares
30
31
32 LOOP:  INC        R0          ;Incrementa Indice
33
34     CJNE         R0,#10,TEST  ;Finaliza se R0 >= 10 (tam vetor)
35 TEST:  JNC        FIM
36
37     MOV          A,R0         ;Le o dado de VET[R0]
38     MOVC         A,@A+DPTR
39     MOV          R3,A         ;Salva dado Lido
40
41     MOV          B,#02
42     DIV          AB
43     MOV          A,B         ;Resto da Divisao por 2
44
45     JNZ          IMPAR       ;JMP se for Impar
46
47 PAR:
48     INC          R2          ;Incrementa Nro Pares
49     MOV          A,R3         ;Recupera o Nro Lido
50     MOV          P1,A         ;Envia para P1
51     MOV          PCON,#80h    ;K = 2 --> Baud = 9600
52     ACALL        ENVIA
53     SJMP         LOOP
54
55 IMPAR:
56     INC          R1          ;Incrementa Nro Impares
57     MOV          A,R3         ;Recupera o Nro Lido
58     MOV          P2,A         ;Envia para P2
59     MOV          PCON,#00h    ;K = 1 --> Baud = 4800
60     ACALL        ENVIA
61     SJMP         LOOP
62
63
64 FIM:
65     MOV          DPTR,#2030h
66     MOV          A,R2

```

```

67      MOVX    @DPTR,A      ;Armazena Nro de Pares
68
69      MOV     DPTR,#2031h
70      MOV     A,R1
71      MOVX    @DPTR,A      ;Armazena Nro de Impares
72
73      SJMP    $             ;Fim Logico do Programa
74
75  ENVIA:
76      MOV     SBUF,A
77      JNB     TI,$
78      CLR     TI
79      RET
80
81
82  ;----- Dados -----
83  VET:      DB  22, 255, 100, 48, 97, 128, 64, 45, 56, 10
84  TAM      EQU 10
85
86  ;----- END -----
87      END
88

```

9. Um robô como mostrado na figura é acionado por dois motores de corrente contínua, um para cada roda, conforme o esquema, e possui um sensor localizado na parte da frente que tem a função de detectar a presença de obstáculos. Desenvolver um programa em Assembly do 8051 que controle o robô fazendo-o navegar por uma sala onde diversos obstáculos podem ser encontrados, de tal forma que ele não colida com nenhum.

O circuito do sensor está ligado na entrada de interrupção Int0 que gera um pulso negativo quando um obstáculo é detectado.

Os motores são acionados da seguinte maneira, conforme mostra o esquema eletrônico:

P1.0 = 1 → liga a alimentação do motor da roda da esquerda (P1.0 = 0 → desliga)

P1.2 = 1 → liga a alimentação do motor da roda da direita (P1.2 = 0 → desliga)

O movimento do robô é dado pela tabela:

P1.1	P1.3	Movimento
0	0	Robô movimenta-se para trás
0	1	Robô vira para a esquerda (roda esquerda pra trás e roda direita à frente)
1	0	Robô vira para a direita (roda esquerda à frente e roda direita para trás)
1	1	Robô movimenta-se à frente

O programa deve:

- Inicialmente movimentar o robô à frente.
- Quando o primeiro obstáculo for detectado o robô deve voltar atrás por 2 segundos e virar à direita por 2 segundos. A frequência do oscilador do microcontrolador é de 12 MHz.
- A cada obstáculo detectado o robô deve movimentar-se para trás por 2 segundos e inverter a última direção durante 2 segundos (direita, 2s, esquerda, 2s, direita, 2s, esquerda, 2s,.....).
- Após cada inversão de direção, o robô deve ser movimentado para frente até que novo obstáculo seja encontrado. Durante o movimento para trás e direita/esquerda a Int0 deve ser desabilitada.

Solução:

```

1  ;----- Jumps -----
2  ORG      0
3  SJMP     PROG
4  ;*****
5  ORG      0003h
6  SJMP     EXT0
7  ;*****
8
9  ;----- Programa -----
10 PROG:
11   SETB    EA      ;Habilita Interrupcoes
12   SETB    EX0     ;Habilita Interrupcao Externa 0
13   SETB    IT0     ;Habilita Descida de Borda para EX0
14
15   MOV     P1,#0    ;Reset no pino P1
16
17   SETB    P1.0     ;Roda esquerda ligada
18   SETB    P1.2     ;Roda direita ligada
19
20   SETB    20h.0    ;Robo comece virando para a direita
21
22   ACALL   FRENTE   ;Robo anda para frente
23
24   SJMP    $        ;Fim Logico do Programa
25
26
27 ;----- Movimento -----
28 ;**** Para Frente ****
29 FRENTE:
30   SETB    P1.1     ;Roda esquerda p/ frente
31   SETB    P1.3     ;Roda direita p/ frente
32
33   RET

```

```

34 ;**** Para Tras *****
35 TRAS:
36     CLR     P1.1      ;Roda esquerda p/ tras
37     CLR     P1.3      ;Roda direita p/ tras
38
39     RET
40 ;**** Para Direita *****
41 DIREITA:
42     SETB    P1.1      ;Roda esquerda p/ frente
43     CLR     P1.3      ;Roda direita p/ tras
44
45     RET
46 ;**** Para Esquerda *****
47 ESQUERDA:
48     CLR     P1.1      ;Roda esquerda p/ tras
49     SETB    P1.3      ;Roda direita p/ frente
50
51     RET
52 ;----- DELAY -----
53 DELAY:
54 ; START: Wait loop, time: 2 s
55 ; Clock: 12000.0 kHz (12 / MC)
56 ; Used registers: R0, R1, R2, R3
57     MOV     R3, #003h
58     MOV     R2, #0D2h
59     MOV     R1, #00Ch
60     MOV     R0, #082h
61     NOP
62     DJNZ    R0, $
63     DJNZ    R1, $-5
64     DJNZ    R2, $-9
65     DJNZ    R3, $-13
66     MOV     R1, #006h
67     MOV     R0, #0BAh
68     NOP
69     DJNZ    R0, $
70     DJNZ    R1, $-5
71     NOP
72     NOP
73     NOP
74 ; Rest: 0
75 ; END: Wait loop
76     RET
77 ;----- Interrupcoes -----
78 ;**** Externa 0 *****
79 EXT0:
80     CLR     EA         ;Desabilita Interrupcao INTO
81
82     ACALL   TRAS        ; Robo anda para tras por
83     ACALL   DELAY       ;dois segundos
84     JNB     20h.0, VIRAESQ ;Verifica a ultima direcao
85
86     ACALL   DIREITA     ;Robo vira para a direita
87     SJMP    FIM_INT
88
89 VIRAESQ:
90     ACALL   ESQUERD     ;Robo vira para a esquerda
91
92
93
94 FIM_INT:
95     CPL     20h.0      ;Inverte a direcao
96     ACALL   DELAY       ;Delay de 2s
97     ACALL   FRENTE     ;Robo volta a andar para a frente
98
99     SETB    EA         ;Reabilita Interrupcoes
100    RETI
101
102 ;----- END -----
103     END
104

```

10. Automatizar uma Máquina de Doces com o Microcontrolador 89S52. A máquina deve fornecer em cada operação, somente um doce que custa 20 centavos. A cada operação o programa reinicia e espera nova sequência de moedas. As moedas aceitas pela máquina são de 5 centavos, 10 centavos e 20 centavos. Como cada moeda tem um tamanho diferente, um sensor óptico alinhado com o coletor de moedas determina qual moeda foi inserida. Apenas uma moeda pode ser inserida por vez. A inserção de uma moeda é detectada através da Interrupção Int0. O circuito de reconhecimento de moedas é mostrado abaixo e sua operação é de acordo com a Tabela 1.

P1.2 = 20 cents	P1.1 = 10 cents	P1.0 = 5 cents	Moeda inserida
1	1	1	Nenhuma (Int0 = 1)
1	1	0	5 centavos (Int0 = 0)
1	0	0	10 centavos (Int0 = 0)
0	0	0	20 centavos (Int0 = 0)

Tabela 1: Sinal nos pinos do 89S52 quando as moedas são inseridas.

A Máquina de Doces tem três controles:

P2.0 = 1 → Fornecer o Doce

P2.1 = 1 → Devolver 5 centavos de troco

P2.2 = 1 → Devolver 10 centavos de troco

A operação da Máquina de Doce pode ser vista na Tabela 2 com todas as sequencias possíveis (S1 a S11) de inserção de moedas pelo usuário e com as ações a serem tomadas pelo microcontrolador.

	Sequência de moedas inseridas	Ações de controle	
S1	5 + 5 + 5 + 5 = 20	Doce → P2.0 = 1	Troco = 00
S2	5 + 5 + 5 + 10 = 25	Doce → P2.0 = 1	Troco = 05
S3	5 + 5 + 5 + 20 = 35	Doce → P2.0 = 1	Troco = 15
S4	5 + 5 + 10 = 20	Doce → P2.0 = 1	Troco = 00
S5	5 + 5 + 20 = 30	Doce → P2.0 = 1	Troco = 10
S6	5 + 10 + 10 = 25	Doce → P2.0 = 1	Troco = 05
S7	5 + 10 + 20 = 35	Doce → P2.0 = 1	Troco = 15
S8	5 + 20 = 25	Doce → P2.0 = 1	Troco = 05
S9	10 + 10 = 20	Doce → P2.0 = 1	Troco = 00
S10	10 + 20 = 30	Doce → P2.0 = 1	Troco = 10
S11	20 = 20	Doce → P2.0 = 1	Troco = 00

Tabela 2: Operação da Máquina de Doces

Solução:

```

1  ;----- Jumps -----
2      ORG      0
3      SJMP     PROG
4  ;*****
5      ORG      0003h
6      SJMP     EXT0
7  ;*****
8
9  ;----- Programa -----
10 PROG:
11      SETB     EA      ;Habilita Interrupcoes
12      SETB     EX0     ;Habilita Interrupcao Externa 0
13      SETB     IT0     ;Habilita Descida de Borda para EX0
14
15      ACALL    RESET
16
17 LOOP:    CJNE    A,#20,TEST ;Espera soma dar >= 20
18 TEST:    JC     LOOP
19
20      CJNE     A,#20,R_25
21
22      ;Resultado eh 20

```

```

23     ACALL    DOCE
24     SJMP     FIM
25
26 R_25:    CJNE     A,#25,R_30
27
28     ;Resultado eh 25
29     ACALL    DOCE
30     ACALL    T_5
31     SJMP     FIM
32
33 R_30:    CJNE     A,#30,R_35
34
35     ;Resultado eh 30
36     ACALL    DOCE
37     ACALL    T_10
38     SJMP     FIM
39
40 R_35:    ;Resultado eh 35
41     ACALL    DOCE
42     ACALL    T_5
43     ACALL    T_10
44
45 FIM:
46     ACALL    DELAY
47     ACALL    RESET
48     SJMP     LOOP
49
50
51 ;**** Doce ****
52 DOCE:    SETB    P2.0        ;Fornece Doce
53     RET
54
55 ;**** TROCO 5 cents ****
56 T_5:    SETB    P2.1        ;Retorna moeda de 5 cents
57     RET
58
59 ;**** TROCO 10 cents ****
60 T_10:    SETB    P2.2        ;Retorna moeda de 10 cents
61     RET
62
63 ;**** Reset ****
64 RESET:
65     MOV     P1,#0FFh        ;Reset no pino P1
66     MOV     P2,#0           ;Reset no pino P2
67     MOV     A,#0            ;Reset no ACC
68     RET
69 ;----- Interrupcoes -----
70 ;**** Externa 0 ****
71 EXT0:
72     CLR     EA              ;Desabilita Interrupcao INTO
73     PUSH    PSW             ;Salva o Carry
74
75     JNB     P1.2,M_20        ;Moeda de 20 Cents
76
77     JNB     P1.1,M_10        ;Moeda de 10 Cents
78
79     ADD     A,#05            ;Moeda de 5 Cents
80     SJMP    FIM_INT
81
82 M_20:    ADD     A,#20
83     SJMP     FIM_INT
84
85 M_10:    ADD     A,#10
86
87 FIM_INT:
88     SETB    EA              ;Reabilita Interrupcoes
89     POP     PSW             ;Recupera o Carry
90     RETI
91
92 ;----- DELAY -----
93 DELAY:
94 ; START: Wait loop, time: 0.5 ms
95 ; Clock: 12000.0 kHz (12 / MC)
96 ; Used registers: R0
97     MOV     R0, #0F9h
98     NOP

```



```
99      DJNZ      RO, $
100 ; Rest: 0
101 ; END: Wait loop
102 RET
103
104 ;----- END -----
105      END
106
```