Arquitetura de Computadores

Relatório do Trabalho Prático 3

Docentes:

Dino Vasconcelos

Sofia Inácio

Dionísio Barros

Pedro Camacho

Alunos:

André Caires nº 2076620

Afonso Nunes nº 2078821

1 Índice

2 Introdução

Para este trabalho prático foi estabelecido como objetivo desenvolver um programa em linguagem assembly e C para o microcontrolador 8051 que registasse o tempo de resposta, para uma plataforma de concursos de perguntas de escolha múltipla

3 Descrição da solução e análise de resultados

É dado ao utilizador um tempo máximo de 5 segundos para responder e o cronómetro só começa a contar quando o utilizador carregar no botão B1(P3.2). Caso o utilizador não responda dentro do tempo, é exibido no display o tempo 0.0 e -.- alternadamente a cada segundo. No caso de o utilizador responder o cronómetro para de contar e o display exibe o tempo restante que o utilizador e a resposta que o mesmo deu alternadamente a cada segundo. Para voltar ao início basta o utilizador voltar a clicar no botão B1.

3.1 Descrição da solução

Para os valores a serem exibidos no display foram criadas matrizes com as combinações corretas de forma a que o display exibe o valor correto.

3.1.1 Inicializações

Começamos por criar a função Init responsável pela inicialização dos registos e onde definimos também algumas variáveis consideradas necessárias para o sucesso do programa. Primeiramente definimos os valores default das variáveis de controlo como estado (responsável por controlar se o cronómetro está a contar), estadoOp (responsável por controlar os estados de resposta), contaDS (responsável por contar as decimas de segundo), contaCS (responsável por contar as centésimas de segundo), contaDSop (responsável por contar as décimas de segundo para as respostas), contas (responsável para contar os segundos) e resposta (responsável por indicar a linha da matriz dos valores de resposta a apresentar no display). De seguida ativamos as interrupções globais, a interrupção do Timer 0 e a interrupção externa 0. Depois configuramos os registos TMOD (TMOD = 0x01) e TCON (TR0 = 0 e IT0 = 0)

Sendo que o Timer 0 do microcontrolador consegue contar 65536µs no modo 1, foi definido que para contar os 5s de tempo desejado vamos contar 0,01s dez vezes para conseguir 0,1s. Chegando eventualmente aos 5s. Para tal os valores de TH0 e TL0 foram definidos para 0xD8 e 0xF0 respetivamente (65536-10000 = 55536 ->D8F0).

3.1.2 Interrupção Externa 0

Esta interrupção é ativada sempre que o utilizador clicar no botão B1, sendo que sempre que esta interrupção é ativada colocamos o TR0 = 1, ativando a contagem do Timer 0, EX0 = 0, desativando a interrupção externa 0, EX1 = 1, ativando a interrupção externa 1, o estado = 0 e contaCS = 0.

3.1.3 Interrupção Timer 0

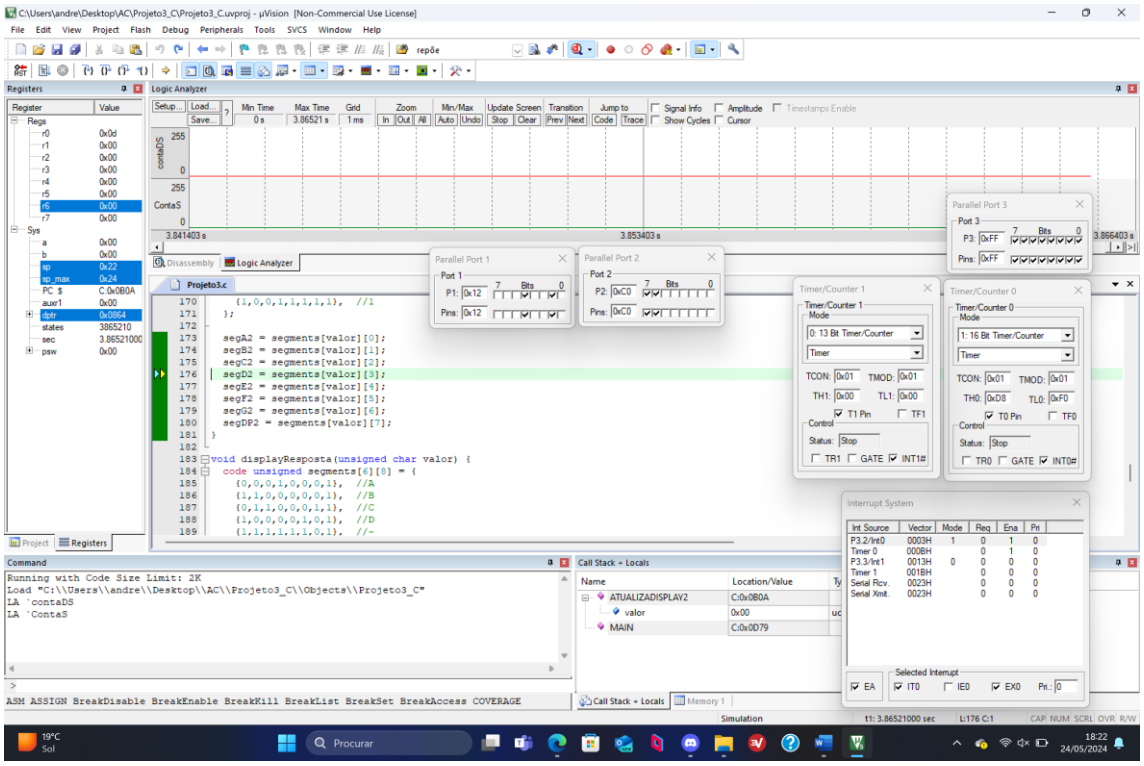
Na interrupção Timer 0 limpamos a flag IE0, repomos os valores de TH0 e TL0 e incrementamos o valor de contaCS. Depois verificamos se caso o valor de contaCS for igual ao valor da constante maxContams (10) então contaCS = 0. Ainda no caso de contaCS ser igual a maxContams, se estado é 0, incrementamos contaDS e se contaDS for igual a 1, incrementamos contaS, e se contaDS for igual a maxContams, então contaDS = 0 e se contas for igual a maxSegundos, então estado passa para 1, EX0 = 1 (ativa a interrupção externa 0) e EX1 = 0 (desativa a interrupção externa 1). Ainda no caso de contaCS ser igual a maxContams, se estado não for 0 então incrementa contaDSop e se contaDSop for igual a maxContams, então contaDSop = 0 e inverte o valor do estadoOp

3.1.4 Interrupção Externa 1

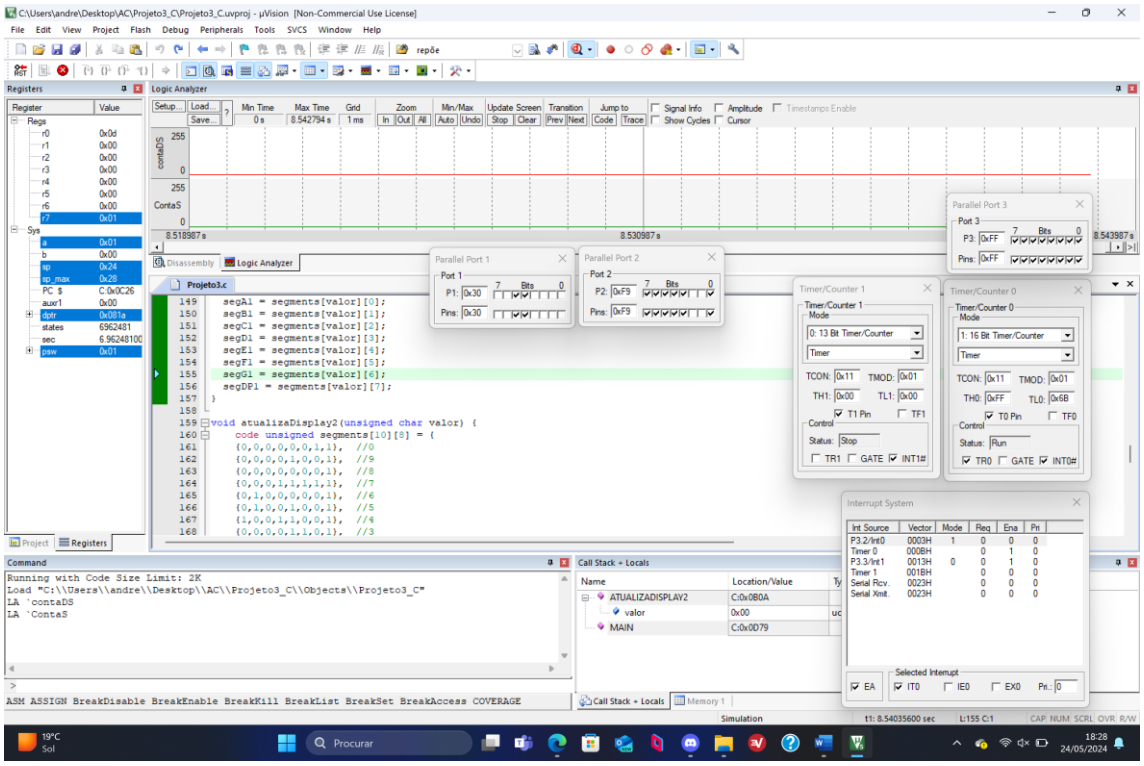
Na interrupção externa 1, contaCS é reposto para 0, estado passa para 1, EX1 é desativada e EX0 é ativada. Para além disso ainda se verifica qual botão de resposta foi pressionado sendo que, se foi pressionado o botão A (P3.4), vai ser exibido o A na resposta, se foi pressionado o botão B (P3.5), vai ser exibido um b na resposta, se foi pressionado o botão C (P3.6), vai ser exibido um C na resposta e se foi pressionado o botão D (P3.7), vai ser exibido um d na resposta.

3.2 Análise de resultados

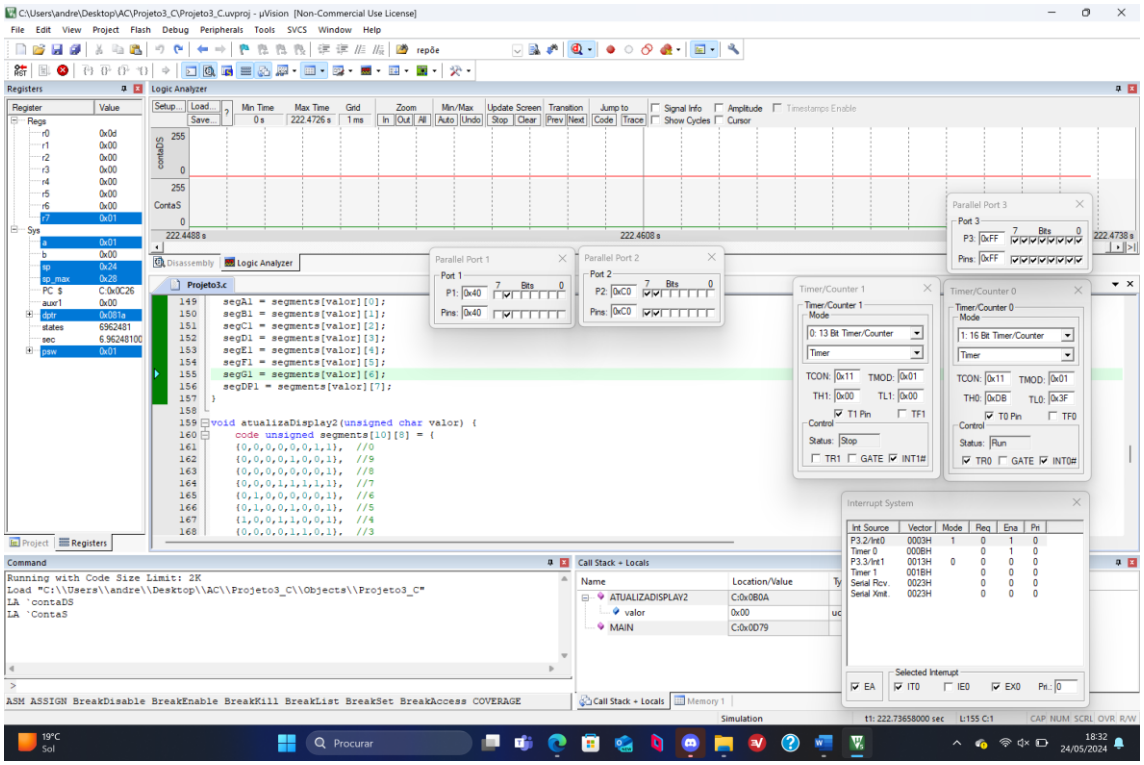
Ao correr a simulação do programa podemos ver que os displays começam por apresentar o tempo de 5.0s.



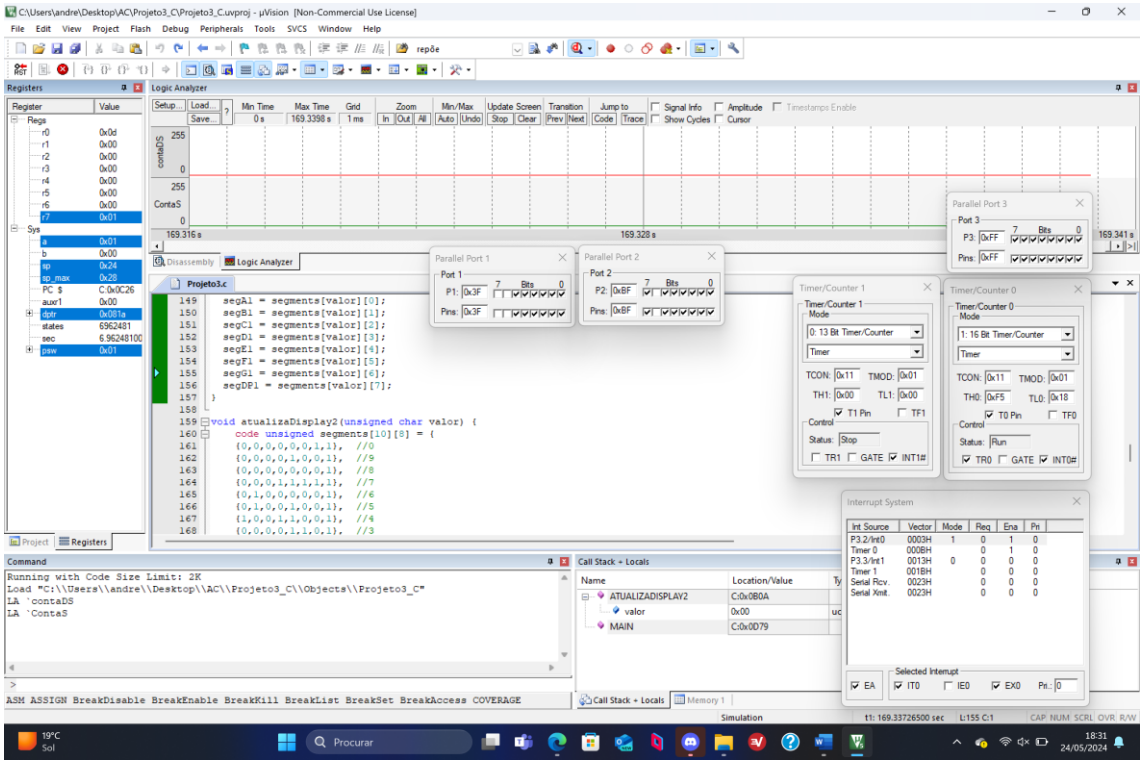
Depois ao clicar no botão B1 o cronómetro começa a contar.



No caso de o utilizador deixar o tempo acabar, o display vai ficar a alternar entre 0.0 e – a cada segundo

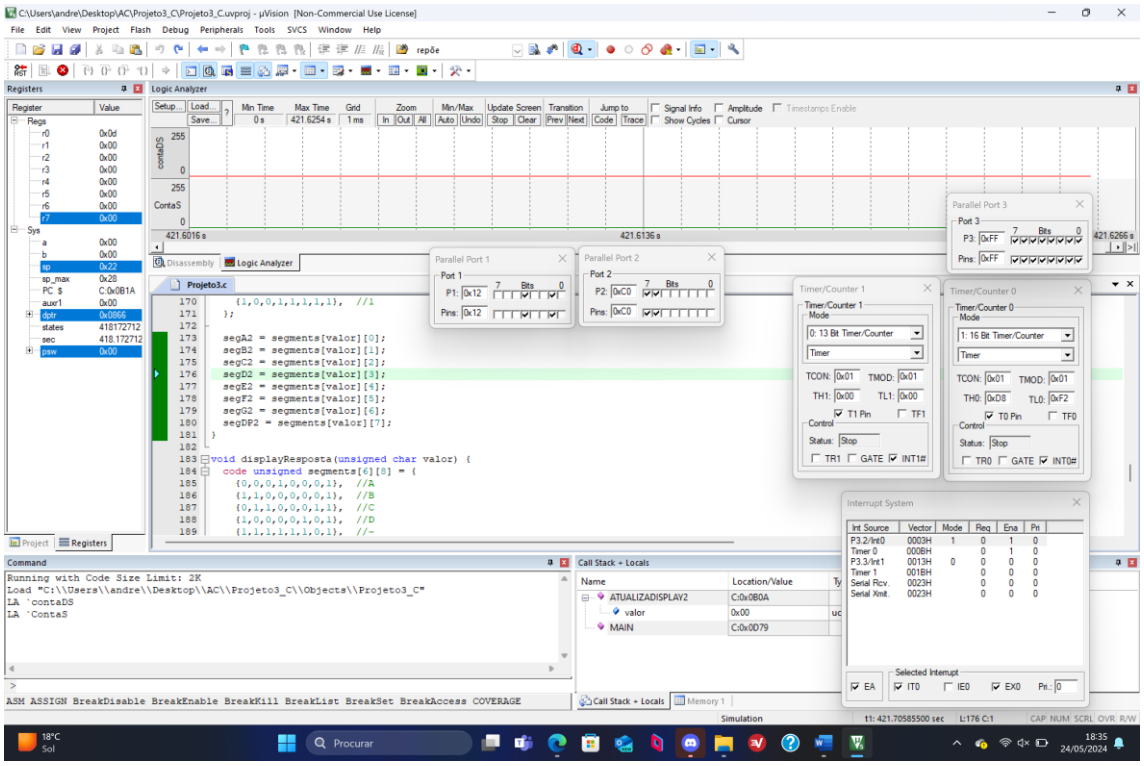


0.0

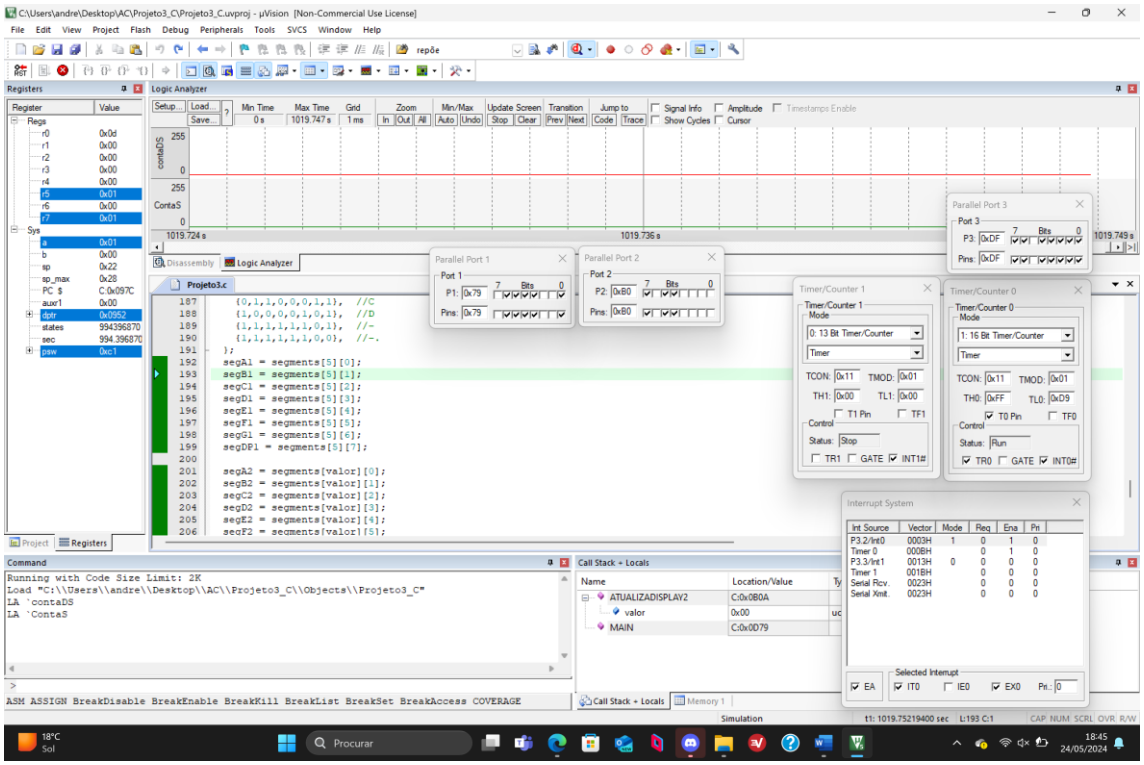


-.-

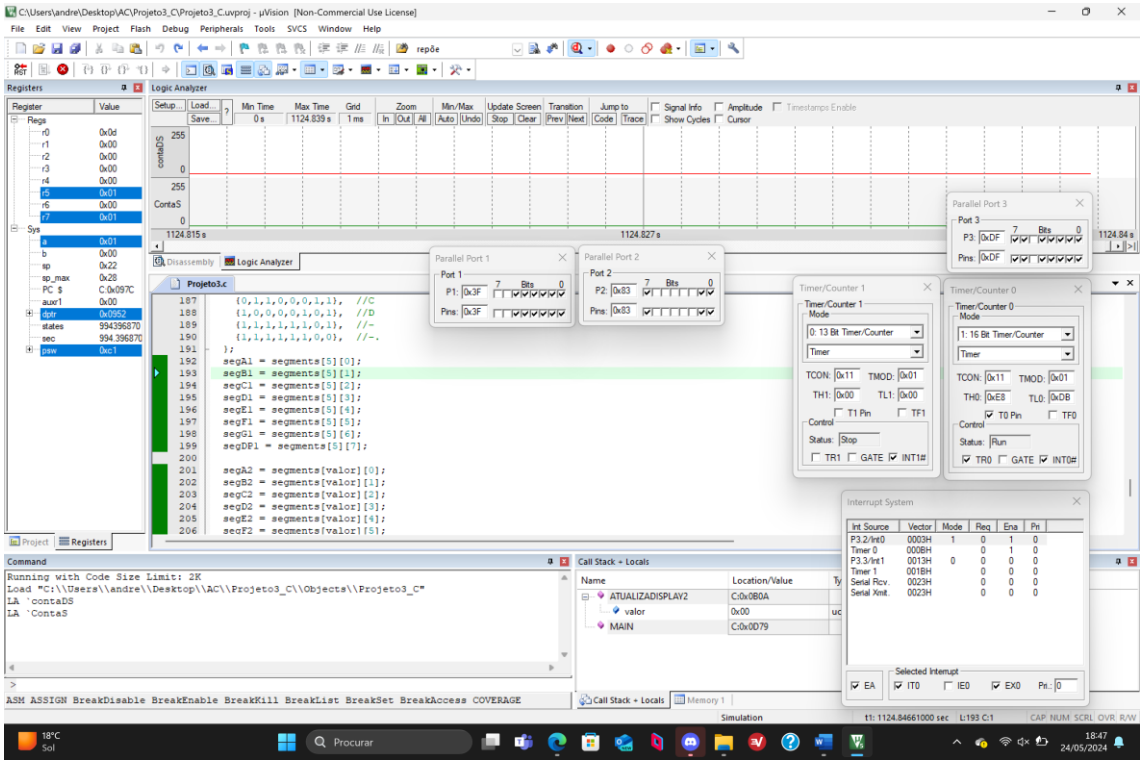
Depois se o utilizador voltar a clicar no botão B1 o cronómetro reinicia.



E caso o utilizador responda antes do tempo acabar o display vai alternar entre o tempo restante e a resposta do utilizador a cada segundo.



1.9

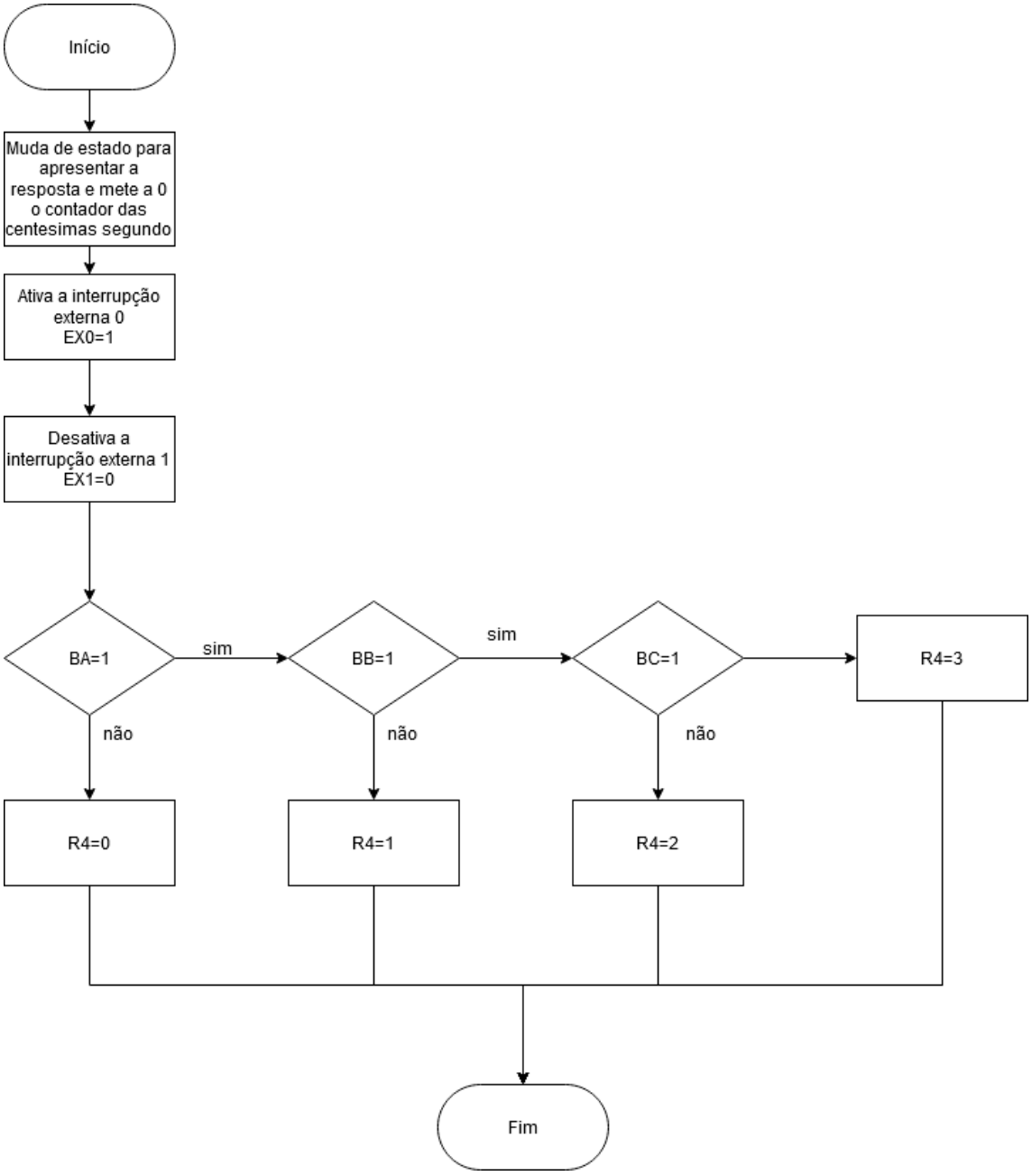


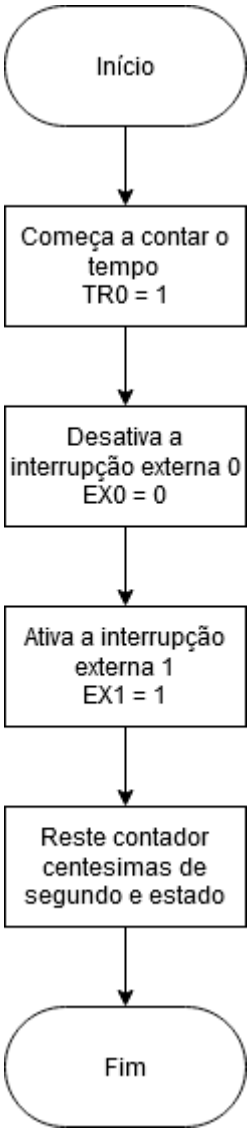
-b

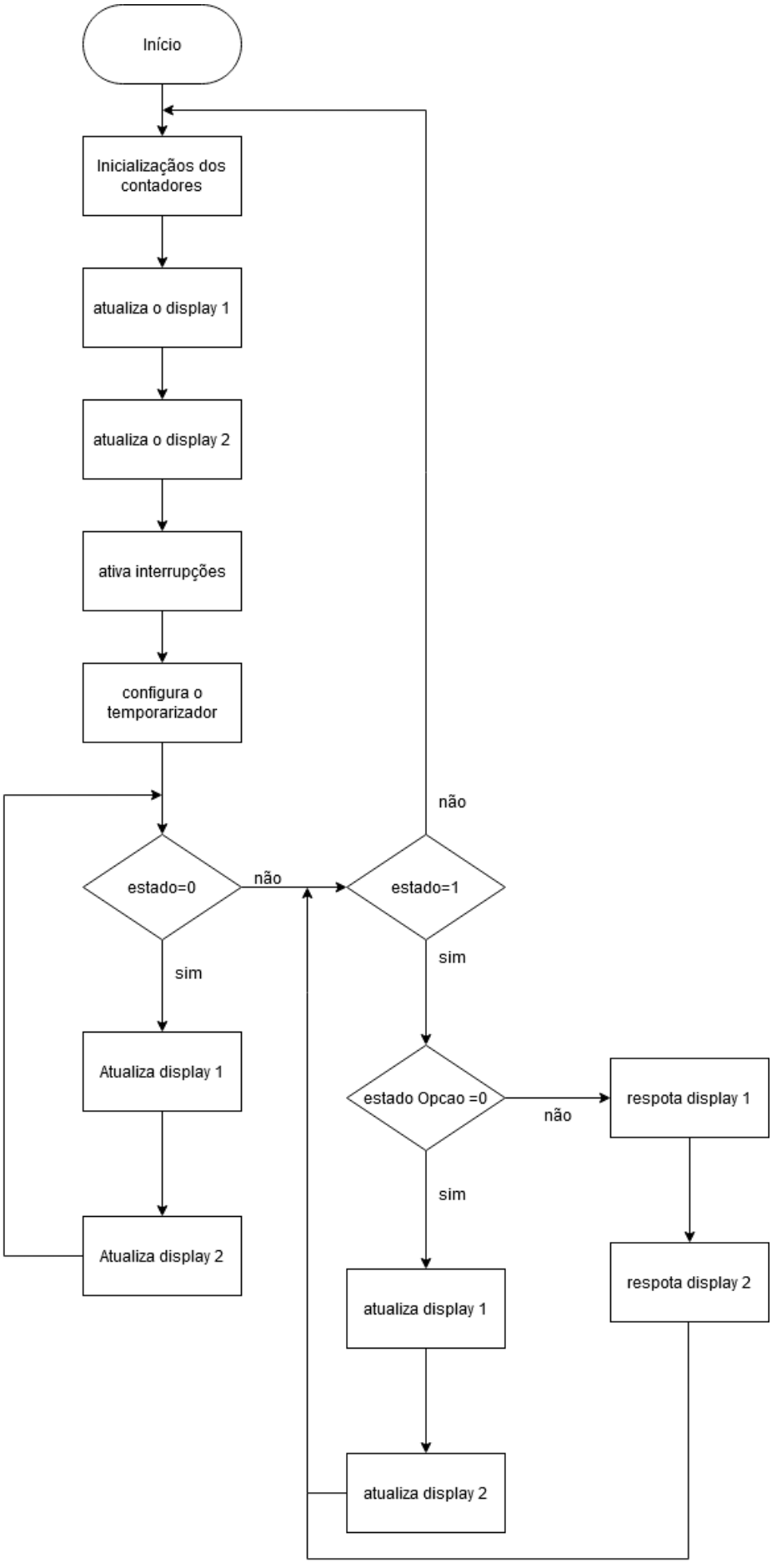
4 Conclusão

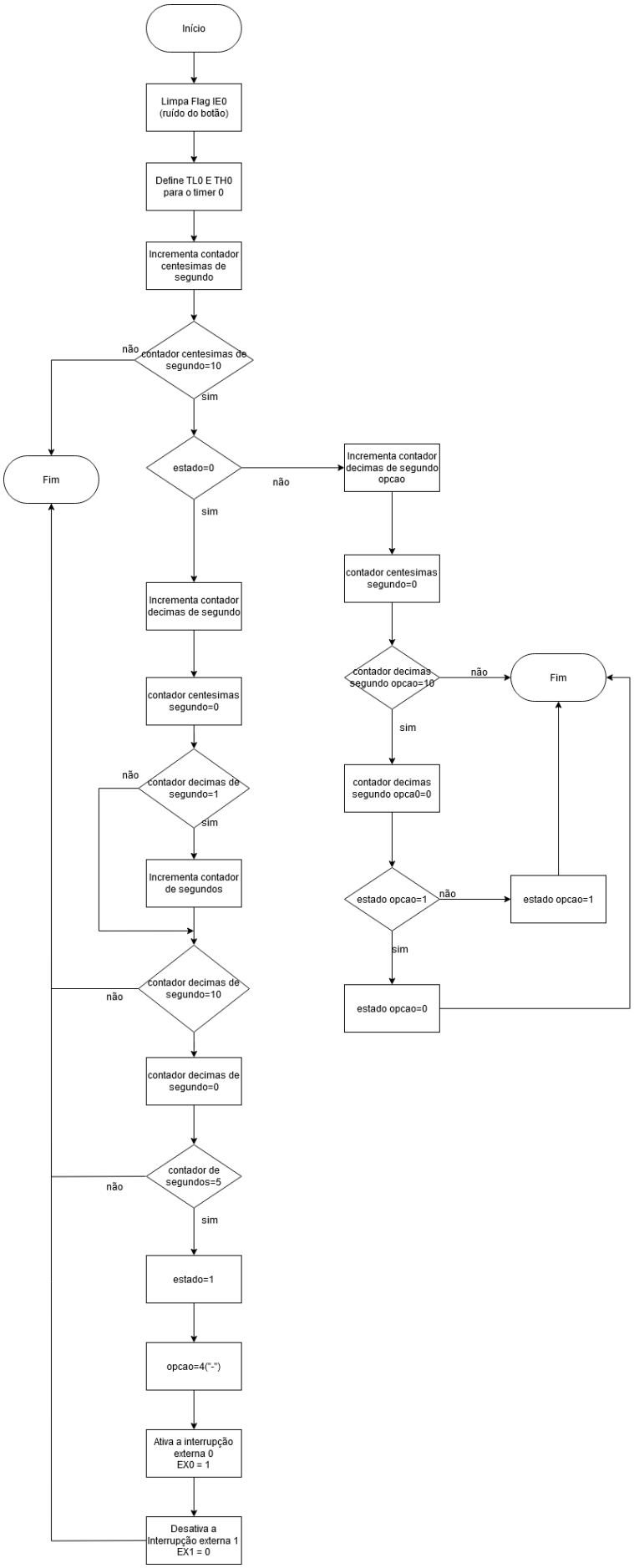
Em suma, conseguimos alcançar todos os objetivos estabelecidos para este trabalho, sendo que o programa funciona de forma correta tanto em linguagem assembly como em C.

5 Anexo A


Início
Muda de estado para apresentar a resposta e mete a 0 o contador das centesimas segundo
Ativa a interrupção externa 0 EX0=1
Desativa a interrupção externa 1 EX1=0
BA=1
sim
BB=1
sim
BC=1
R4=3
não
não
não
R4=0
R4=1
R4=2
Fim


Início
Começa a contar o tempo
TR0 = 1
Desativa a interrupção externa 0
EX0 = 0
Ativa a interrupção externa 1
EX1 = 1
Reste contador centesimas de segundo e estado
Fim


Início
Inicializaçãos dos contadores
atualiza o display 1
atualiza o display 2
ativa interrupções
configura o temporarizador
estado=0
não
estado=1
não
sim
sim
Atualiza display 1
Atualiza display 2
estado Opcao =0
não
sim
respota display 1
respota display 2
atualiza display 1
atualiza display 2


Início
Limpa Flag IE0 (ruído do botão)
Define TL0 E TH0 para o timer 0
Incrementa contador centesimas de segundo
contador centesimas de segundo=10
não
sim
Fim
estado=0
não
sim
Incrementa contador decimas de segundo opcao
contador centesimas segundo=0
contador decimas segundo opcao=10
não
Fim
sim
contador decimas segundo opca0=0
estado opcao=1
não
estado opcao=1
sim
estado opcao=0
Incrementa contador decimas de segundo
contador centesimas segundo=0
contador decimas de segundo=1
não
sim
Incrementa contador de segundos
contador decimas de segundo=10
não
contador decimas de segundo=0
contador de segundos=5
não
sim
estado=1
opcao=4("-")
Ativa a interrupção externa 0 EX0 = 1
Desativa a Interrupção externa 1 EX1 = 0

6 Anexo B

```
#include <reg51.h>

// Constantes
#define maxContams 10
#define maxSegundos 5

#define A 0
#define B 1
#define C 2
#define D 3

// 65536 - 10000 = 55536 -> D8F0
//interrupção de 0.01 em 0.01 ssegundos
#define TempH 0xD8
#define TempL 0xF0

// Pinos dos botões
sbit BA = P3^4;
sbit BB = P3^5;
sbit BC = P3^6;
sbit BD = P3^7;

// Pinos dos displays de 7 segmentos
sbit segA1 = P1^0;
sbit segB1 = P1^1;
sbit segC1 = P1^2;
sbit segD1 = P1^3;
sbit segE1 = P1^4;
sbit segF1 = P1^5;
```

```
sbit segG1 = P1^6;
sbit segDP1 = P1^7;

sbit segA2 = P2^0;
sbit segB2 = P2^1;
sbit segC2 = P2^2;
sbit segD2 = P2^3;
sbit segE2 = P2^4;
sbit segF2 = P2^5;
sbit segG2 = P2^6;
sbit segDP2 = P2^7;

// Estados do sistema
bit estado = 0;
bit estadoOp = 0;

// Variáveis globais
unsigned char contaCS = 0;              //conta centesimas de segundo
unsigned char contaDS = 0;              //conta decimas de segundo
unsigned char contaDSop = 0;            //conta decimas de segundo no modo para mostrar
resposta
unsigned char contaS = 0;                       //conta segundoS
unsigned char resposta = 4;    // Inicialmente é o traço '-'

// Segmentos dos displays


void Init(void) {

        estado = 0;
        estadoOp = 0;
```

```
	contaDS = 0;
	contaCS = 0;
	contaDSop = 0;
	contaS = 0;
	resposta = 4;

	// Configuração do Registo IE
	EA = 1; // ativa interrupções globais
	ET0 = 1; // ativa interrupção Timer 0
	EX0 = 1; // ativa interrupção externa 0
	EX1 = 0; // desativa interrupção externa 1

	// Configuração Registo TMOD
	TMOD = 0x01; // Timer 0 no modo 1 (16 bits)

	// Configuração Timer 0
	TH0 = TempH;
	TL0 = TempL;

	// Configuração Registo TCON
	TR0 = 0; // Timer 0 começa desligado
	IT0 = 1; // Interrupção externa 0 ativa no falling edge
}

void Externall0_ISR(void) interrupt 0 {
	TR0 = 1;			// Inicia o Timer 0
	EX0 = 0;			// Desativa interrupção externa 0
	EX1 = 1;			// Ativa interrupção externa
	estado = 0;
```

```
        contaCS = 0;
}

void Timer0_ISR(void) interrupt 1 {

        IE0 = 0; // Limpa flag se P3.2 foi pressionado (ruido botao)

        TH0 = TempH;
        TL0 = TempL;

        contaCS++;
                //incrementa o contar de centesimas de segundo
        if (contaCS == maxContams) {                    //contador centesimas de segundo =
10
                contaCS = 0;
        //centesimas de segundo = 0

                if (estado == 0) {                                                      //caso
o estado = 0(Modo escolhe pergunta)
                        contaDS++;
                //incrementa o contador de decimas de segundo
                        if (contaDS == 1){                                              //caso
aumente uma decima de segundo e é igual 1 (5.0 -> 4.9) o contador dos segundos tem de
incrementar
                                contaS++;
                        //incrementa o contador de segundos
                        }
                        if(contaDS == maxContams){          //caso o contador das
decimas de segundo cheguem ao máximo -> 10
                                contaDS = 0;
        //o contador de decimas de segundo vai resetar
                                if(contaS == maxSegundos){  //caso chegue aos 5
segundos e decimas de segundo a 10 ou seja 0.0
                                        estado = ~estado;
        //o estado irá passar de 0 para 1
```

```
                                        EX0 = 1;
                        //a interrupção externa 0 irá ser ativada porque chegou ao final
                                        EX1 = 0;
                        //desabilita a interrupção externa 1 porque não pode escolher
nenhuma opção
                                }
                        }
                }
                else{
                                //caso o estado = 1(modo mostra pergunta)
                        contaDSop++;
        //incrementa o contador de decimas de segundo para o modo de mostrar a opção
                        if(contaDSop == maxContams){        //caso chegue a 10, ou seja,
passe 1 segundo
                                contaDSop = 0;
        //contador de decimas de segundo = 0
                                estadoOp = ~estadoOp;                           //o
estado de op muda para o inverso
                        }
                }
        }
}

void Externall1_ISR(void) interrupt 2 {
        contaCS = 0;
        estado = 1;
        EX1 = 0; // desativa interrupção externa 1
        EX0 = 1; // Ativa interrupção externa 0

        if (BA == 0) resposta = A; // Botão A pressionado
        else if (BB == 0) resposta = B; // Botão B pressionado
        else if (BC == 0) resposta = C; // Botão C pressionado
        else if (BD == 0) resposta = D; // Botão D pressionado
```

```
}

void atualizaDisplay1(unsigned char valor) {
        code unsigned segments[6][8] = {
                {0,1,0,0,1,0,0,0},      //5.
                {1,0,0,1,1,0,0,0},      //4.
                {0,0,0,0,1,1,0,0},      //3.
                {0,0,1,0,0,1,0,0},      //2.
                {1,0,0,1,1,1,1,0},      //1.
                {0,0,0,0,0,0,1,0},      //0.
        };
        segA1 = segments[valor][0];
        segB1 = segments[valor][1];
        segC1 = segments[valor][2];
        segD1 = segments[valor][3];
        segE1 = segments[valor][4];
        segF1 = segments[valor][5];
        segG1 = segments[valor][6];
        segDP1 = segments[valor][7];
}

void atualizaDisplay2(unsigned char valor) {
   code unsigned segments[10][8] = {
                {0,0,0,0,0,0,1,1},      //0
                {0,0,0,0,1,0,0,1},      //9
                {0,0,0,0,0,0,0,1},      //8
                {0,0,0,1,1,1,1,1},      //7
                {0,1,0,0,0,0,0,1},      //6
                {0,1,0,0,1,0,0,1},      //5
```

```
                {1,0,0,1,1,0,0,1},      //4
                {0,0,0,0,1,1,0,1},      //3
                {0,0,1,0,0,1,0,1},      //2
                {1,0,0,1,1,1,1,1},      //1
        };

        segA2 = segments[valor][0];
        segB2 = segments[valor][1];
        segC2 = segments[valor][2];
        segD2 = segments[valor][3];
        segE2 = segments[valor][4];
        segF2 = segments[valor][5];
        segG2 = segments[valor][6];
        segDP2 = segments[valor][7];
}

void displayResposta(unsigned char valor) {
        code unsigned segments[6][8] = {
                {0,0,0,1,0,0,0,1},      //A
                {1,1,0,0,0,0,0,1},      //B
                {0,1,1,0,0,0,1,1},      //C
                {1,0,0,0,0,1,0,1},      //D
                {1,1,1,1,1,1,0,1},      //-
                {1,1,1,1,1,1,0,0},      //-.
        };
        segA1 = segments[5][0];
        segB1 = segments[5][1];
        segC1 = segments[5][2];
        segD1 = segments[5][3];
        segE1 = segments[5][4];
```

```
        segF1 = segments[5][5];
        segG1 = segments[5][6];
        segDP1 = segments[5][7];

        segA2 = segments[valor][0];
        segB2 = segments[valor][1];
        segC2 = segments[valor][2];
        segD2 = segments[valor][3];
        segE2 = segments[valor][4];
        segF2 = segments[valor][5];
        segG2 = segments[valor][6];
        segDP2 = segments[valor][7];
}

void main(void) {

        while (1) {
         Init();
                atualizaDisplay1(contaS);                                    //inicio
comecar a 5.0
                atualizaDisplay2(contaDS);
                while (estado==0){
        //estado->0(Modo escolhe pergunta)/1(modo mostra pergunta)
                        atualizaDisplay1(contaS);
        //atualiza ao longo da execução o tempo
                        atualizaDisplay2(contaDS);
                }
                while (estado==1){
                        if(estadoOp==0){
                //estadoOp->0(Mostra segundos restantes)/1(mostra resposta)
                                atualizaDisplay1(contaS);
        //atualiza os displays com o tempo guardo anteriormente
```

```
                    atualizaDisplay2(contaDS);
                }
                else{
                    displayResposta(resposta);          //mostra a
resposta
                }
            }
        }
}
```

```
;Nota: Timer 0 - modo 1(16bits)
;       1 contagem = 1 microseg, 10000 contagens = 10000 microseg = 10 ms
;       65536 - 10000 = 55536(D8F0H)

;R0-> contador para centesimas de segundo(0.01s)
;R1-> contador para decimas de segundo(0.1s)
;R2-> segundos(1s)
;R3->0(Modo escolhe pergunta)/1(modo mostra pergunta)
;R4->Resposta
;R5->0(Mostra segundos restantes)/1(mostra resposta)
;R6->Contar 1 segundo para o modo mostra pergunta

;Definição de constantes
TempoH0         EQU 0xD8    ;65536 - 10000 = 55536(D8F0H)
TempoL0         EQU 0xF0
contaMsegundos  EQU 0x0A    ;Numero de contagens de 10 ms para contar 0,1s
(10 contagens)
maximoSegundos  EQU 0X05    ;Número máximo de segundos a serem contados (5
segundos)
```

```
displayD1               EQU     P1
displayD2               EQU P2

BA                              EQU P3.4
BB                              EQU P3.5
BC                              EQU P3.6
BD                              EQU P3.7

tracoponto                      EQU 0x3F ; -.


CSEG AT 0300H
segmentosD1:                    DB      0x12, 0x19, 0x30, 0x24, 0x79, 0x40
                        ; 5., 4., 3., 2., 1., 0.

CSEG AT 0310H
segmentosD2Segundos:    DB      0xC0, 0x90, 0x80, 0xF8, 0x82, 0x92, 0x99, 0xB0,
0xA4, 0xF9   ; 0, 9, 8, 7, 6, 5, 4, 3, 2, 1

CSEG AT 0320H
segmentosD2Opcao:               DB 0x88, 0x83, 0xC6, 0xA1, 0xBF
                                        ; A, b, C, d, -


CSEG AT 0000H
        JMP Inicio

;interrupcao externa 0(P3.2 -> B1)
CSEG AT 0003H
        JMP InterrupcaoExt0
```

```
;interrupcao timer 0 (de 0,01s a 0,01s)
CSEG AT 000BH
        JMP InterrupcaoTemp0

;interrupcao externa 0(P3.3 -> BA & BB & BC & BD)
CSEG AT 0013H
        JMP InterrupcaoExt1

CSEG AT 0050H
Inicio:
        CALL inicializacoes                     ;chama a rotina de
inicializacoes
        CALL atualizadisplay1                   ;mostra inicialmente no display1 o
valor 5.
        CALL atualizadisplay2                   ;mostra inicialmente no display2 o
valor 0
        CALL ativaInterrupcoes                  ;rotina para configurar as
interrupcoes Ext0 e Timer0
        CALL configuraTemporizador              ;rotina para ativar o temporarizador

;Programa principal
cicloTemporarizador:
        CJNE R3, #0, cicloMostraResposta    ;caso R3 seja diferente de 0 irá para o modo
de mostrar resposta
        CALL atualizadisplay1                   ;atualiza o display1 consoante o
tempo decorrido
        CALL atualizadisplay2                   ;atualiza o display2 consoante o
tempo decorrido
        JMP cicloTemporarizador

cicloMostraResposta:
        CJNE R3, #1, Inicio                     ;caso R3 seja diferente de 1
quer dizer que B1 foi pressionada e é para voltar ao início
```

```
        CJNE R5, #0, mostraRespostaDisplay ;caso R5 seja diferente de 0 salta para o
mostraRespostaDisplay

        CALL atualizadisplay1                   ;mostra no display1 os segundos que
faltavam até chegar a 0

        CALL atualizadisplay2                   ;mostra no display2 os milésimos de
segundo que faltavam até chegar a 0

        JMP     cicloMostraResposta

mostraRespostaDisplay:

        CALL respostadisplay1                   ;mostra no display1 um -.

        CALL respostadisplay2                   ;mostra no display2 a opção
que o utilizador escolheu

        JMP cicloMostraResposta


;rotina inicializacoes

inicializacoes:

        MOV R0, #0                              ;contador das
centesimas de segundo = 0

        MOV R1, #0                              ;contador das
decimas de segundo = 0

        MOV R2, #0                              ;contador dos
segundos = 0

        MOV R3, #0                              ;modo => 0(escolhe
pergunta)

        MOV R4, #0                              ;opção => 0

        MOV R5, #0                              ;mostra resposta/
pergunta =>0

        MOV R6, #0                              ;contador dos
segundos para o modo de ver a resposta = 0

        RET


;rotina ativaInterrupcoes

ativaInterrupcoes:

        MOV IE, #10000011b                      ;EA=1, ET1=0, EX1=0, ET0=1 e
EX0=1 -> IE=10000011
```

```
        RET


configuraTemporizador:
        MOV TMOD, #0x01
        MOV TL0, #TempoL0                   ;valor do byte menos
significativo
        MOV TH0, #TempoH0                   ;valor do byte mais
significativo
        CLR TR0                             ;garante que o Timer 0 esteja
desligado inicialmente
        SETB IT0                            ;gonfigura a interrupção
externa 0 para ser ativada na descida (falling edge)
        RET                                 ;retorna da sub-rotina


atualizadisplay1:
        MOV DPTR, #segmentosD1              ;move para o dptr o endereço
da tabela dos segmentosD1
        MOV A, R2                           ;move o conteúdo R2
para o A. R2 contém o índice dos segundos do segmento a ser exibido
        MOVC A, @A+DPTR                     ;usa o conteúdo de A
como um índice para acder à tabela de segmentos. MOVC lê um byte da memória de
programa no endereço (DPTR + A) e move para o acumulador
        MOV displayD1, A                    ;move o valor do acumulador
para displayD1, atualizando o display 1
        RET


atualizadisplay2:
        MOV DPTR, #segmentosD2Segundos      ;guarda no dptr o endereço da tabela
dos segmentosD2 para os segundos
        MOV A, R1                           ;move o conteúdo R1
para o A. R1 contém o índice dos milésimos de segundo do segmento a ser exibido
        MOVC A, @A+DPTR                     ;usa o conteúdo de A
como um índice para acder à tabela de segmentos. MOVC lê um byte da memória de
programa no endereço (DPTR + A) e move para o acumulador.
```

```
        MOV displayD2, A                    ;move o valor do acumulador
para displayD2, atualizando o display 2

        RET


respostadisplay1:

        MOV A, #tracoponto                  ;move o valor constante
'tracoponto' para o acumulador (A)

        MOV displayD1, A                    ;move o valor do acumulador
para displayD1, atualizando o display 1

        RET


respostadisplay2:

        MOV DPTR, #segmentosD2Opcao         ;guarda no dptr o endereço da
tabela dos segmentosD2 para as opções

        MOV A, R4                           ;move o conteúdo R4
para o A. R4 contém o índice da opção do segmento a ser exibido

        MOVC A, @A+DPTR                     ;usa o conteúdo de A
como um índice para acder à tabela de segmentos. MOVC lê um byte da memória de
programa no endereço (DPTR + A) e move para o acumulador

        MOV displayD2, A                    ;move o valor do acumulador
para displayD2, atualizando o display 2

        RET


InterrupcaoExt0:

        SETB TR0                            ;timer 0 começa a
contar o tempo

        CLR EX0                             ;desativa
interrupcao externa 0

        SETB EX1                            ;ativa interrupcao
externa 1

        MOV R0, #0                          ;reseta o contador dos
microsegundos

        MOV R3, #0                          ;reseta o valor do
modo pra zero(modo de escolha da pergunta onde decrece os 5 segundos)

        RETI
```

```
InterrupcaoExt1:
        MOV R0, #0                      ;reset do contador dos 0,01s
        MOV R3, #1                      ;Mete o registo 3 a 1(modo onde mostra a ressposta do utilizador)
        SETB EX0                        ;ativa interrupcao externa 0
        CLR EX1                         ;desativa a interrupção externa 1
        JB BA, verificaB                ;verifica se o Botão BA não está pressionado, se não estiver salta para verificaB
        MOV R4, #0                      ;caso a opção escolhida seja A guarda o valor 0 em R4 para incrementar depois no indice da tabela de segmento
        JMP fimExt1                     ;sai da interrupção
verificaB:
        JB BB, verificaC                ;verifica se o Botão BB não está pressionado, se não estiver salta para verificaC
        MOV R4, #1                      ;caso a opção escolhida seja B guarda o valor 1 em R4 para incrementar depois no indice da tabela de segmento
        JMP fimExt1                     ;sai da interrupção
verificaC:
        JB BC, verificaD                ;verifica se o Botão BC não está pressionado, se não estiver salta para verificaD
        MOV R4, #2                      ;caso a opção escolhida seja C guarda o valor 2 em R4 para incrementar depois no indice da tabela de segmento
        JMP fimExt1                     ;sai da interrupção
verificaD:
        MOV R4, #3                      ;caso a opção escolhida seja D guarda o valor 3 em R4 para incrementar depois no indice da tabela de segmento
fimExt1:
        RETI
```

```
InterrupcaoTemp0:
        CLR IE0                                 ;limpa flag se
P3.2 foi pressionado (ruido botao)
        MOV TL0, #TempoL0                       ;TL0=0xF0
        MOV TH0, #TempoH0                       ;TH0=0xD8
        INC R0                                  ;incrementa o
contador das centesimas de segundo
        CJNE R0, #contaMsegundos, fimI          ;caso o valor das centesimas de
segundo seja igual a 10 temos que incrementar o contador das decimas de segundo se
não acaba e salta da interrupção
        CJNE R3, #0, tempoResposta              ;caso o valor do modo seja diferente
de 0 queremos que o contador conte de 1 em segundo
verificaMsegundo:
        INC R1                                  ;incrementa o valor
das decimas de segundo
        MOV R0, #0                              ;reseta o valor das
centesimas de segundo
        CJNE R1, #1, verificaResetSeg           ;caso aumente uma decima de segundo e é
igual 1 (5.0 -> 4.9) o contador dos segundos tem de incrementar,
        INC R2                                  ;incrementa o valor
das decimas de segundo
verificaResetSeg:
        CJNE R1, #contaMsegundos, fimI          ;verica se o contador das decimas de
segundo chegou a 10 para voltar ao início se não sái da exceção
        MOV R1, #0                              ;reseta o valor das
decimas de segundo
        CJNE R2, #maximoSegundos, fimI          ;caso ja tenho chegado aos 0
segundos
        MOV R3, #1                              ;r3 fica com o valor 1
para mostrar a resposta
        MOV R4, #4                              ;r4 fica com o valor 4
para mostrar em vez de opção um traço -
        SETB EX0                                ;ativa interrupcao
externa 0
```

```
        CLR EX1                                                 ;desativa
interrupcao externa 1

        JMP fimI


tempoResposta:

        INC R6                                                  ;incrementa o valor do
contador das decimas de segundo

        MOV R0, #0                                              ;reseta o valor do
contador de centesimas

        CJNE R6, #contaMsegundos, fimI          ;caso o valor do contador das
decimas de segundo seja igual a 10 incrementa um segundo se não acaba a interrupção

        MOV R6, #0                                              ;reseta o valor do
contador das decimas de segundo

        CJNE R5, #1, mudaPara1                          ;caso o valor de r5 não seja
igual a 1 salta

        MOV R5, #0                                              ;muda o valor de r5
para 0(mostrar segundos restantes)

        JMP fimI

mudaPara1:

        MOV R5, #1                                              ;muda o valor de r5
para 1(mostrar opção)


fimI:

        RETI


END
```