```
 1: /*****************************************************************
 2: **                          Exercício 05                        **
 3: **                                                              **
 4: **       Exercício realizado durante aula prática 05            **
 5: **       Temporizador Digital usando display 7 segmentos        **
 6: **                                                              **
 7: ** Arquivo: exercicio5.c                                        **
 8: ** Compilador: MikroC PRO PIC v.7.2.0                          **
 9: **                                                              **
10: ** Aluno: Carlos Magno do Nascimento Junior                     **
11: **                                                              **
12: ** UFLA - Lavras/MG - 27/11/2024                               **
13: *****************************************************************/
14:
15: // Define o tempo de acendimento do display em ms.
16: #define tempo 5
17:
18: //Declaração de variáveis globais
19: int mil, cen, dez, uni, cont=0;
20:
21: // Converte valor numerico decimal para codigo 7 segmentos
22: unsigned short mask(unsigned short num)
23: {
24:   switch (num)
25:   {
26:     case 0 : return 0x3F;
27:     case 1 : return 0x06;
28:     case 2 : return 0x5B;
29:     case 3 : return 0x4F;
30:     case 4 : return 0x66;
31:     case 5 : return 0x6D;
32:     case 6 : return 0x7D;
33:     case 7 : return 0x07;
34:     case 8 : return 0x7F;
35:     case 9 : return 0x6F;
36:   }
37: }
38:
39: //função para mostrar valores no display
40: void mostrar(unsigned short num1, unsigned short num2, unsigned short num3, unsign
    ned short num4){
41:     // Escreve valor 1 no display 1 em codigo 7 segmentos.
42:     PORTD = mask(num1);
43:     porta.f2 = 1;     // Ativa display 1.
44:     Delay_ms(tempo);
45:     porta.f2 = 0;     // Desativa display 1.
46:
47:     // Escreve valor 2 no display 2 em codigo 7 segmentos.
48:     PORTD = mask(num2);
49:     porta.f3 = 1;     // Ativa display 2.
50:     Delay_ms(tempo);
51:     porta.f3 = 0;     // Desativa display 2.
52:
53:     // Escreve valor 3 no display 3 em codigo 7 segmentos
54:     PORTD = mask(num3);
55:     porta.f4 = 1;     // Ativa display 3.
56:     Delay_ms(tempo);
57:     porta.f4 = 0;     // Desativa display 3.
58:
59:     // Escreve valor 4 no display 4 em codigo 7 segmentos.
60:     PORTD = mask(num4);
61:     porta.f5 = 1;     // Ativa display 4.
```

```
 62:     Delay_ms(tempo);
 63:     porta.f5 = 0;     // desativa display 4.
 64: }
 65:
 66: //Decomponhe número inteiro em 4 números inteiros
 67: void separarMil(unsigned num){
 68:    int resto;
 69:
 70:    mil = num/1000;
 71:    resto = num%1000;
 72:
 73:    cen = resto/100;
 74:    resto = resto%100;
 75:
 76:    dez = resto/10;
 77:
 78:    uni = resto%10;
 79: }
 80:
 81: //Decomponhe número inteiro em 4 números inteiros
 82: void separarCen(unsigned num){
 83:    int resto;
 84:
 85:    mil = 0;
 86:
 87:    cen = num/100;
 88:    resto = num%100;
 89:
 90:    dez = resto/10;
 91:
 92:    uni = resto%10;
 93: }
 94:
 95: //Decomponhe número inteiro em 4 números inteiros
 96: void separarDez(unsigned num){
 97:    int resto;
 98:
 99:    mil = 0;
100:
101:    cen = 0;
102:
103:    dez = num/10;
104:
105:    uni = num%10;
106: }
107:
108: //Decomponhe número inteiro em 4 números inteiros
109: void separarUni(unsigned num){
110:    int resto;
111:
112:    mil = 0;
113:
114:    cen = 0;
115:
116:    dez = 0;
117:
118:    uni = num;
119: }
120:
121: //atualiza valor e mostra no display
122: void atualizaValor(unsigned valorTempo){
123:    if (valorTempo >= 1000){
```

```
124:        separarMil(valorTempo);
125:     }
126:     else if (valorTempo >= 100 && valorTempo < 1000){
127:        separarCen(valorTempo);
128:     }
129:     else if (valorTempo >= 10 && valorTempo < 100){
130:        separarDez(valorTempo);
131:     }
132:     else{
133:        separarUni(valorTempo);
134:     }
135:     mostrar(mil, cen, dez, uni);
136: }
137:
138: void main(void)
139: {
140:   unsigned valorTempo = 0;
141:   ADCON0 = 0X00;
142:   ADCON1 = 0X06;          // desabilita conversor A/D.
143:   INTCON = 0;            // desabilita interrupcoes.
144:   TRISA  = 0;            // configura portA como saida.
145:   PORTA  = 0;
146:   TRISD  = 0;            // configura portD como saida.
147:   PORTD  = 0;
148:   TRISB  = 7;
149:   TRISC  = 0xFD; // 0b11111101
150:
151:    // Inicia com buzzer desligado.
152:    portc.rc1 = 0;
153:   while(1)                        // inicio do loop infinito.
154:   {
155:     atualizaValor(valorTempo);
156:     if (portb.rb0 == 0){  //se botão RB0 for pressionado, incrementa valor do temp
      po
157:        valorTempo += 1;
158:        delay_ms(tempo);
159:     }
160:     if (portb.rb1 == 0){ //se botão RB1 for pressionado, decrementa valor do tempo
      o
161:        valorTempo -= 1;
162:        delay_ms(tempo);
163:     }
164:
165:     if (portb.rb2 == 0){ //se botão RB1 for pressionado, incia a contagem regressi
      iva até 0
166:        while (valorTempo > 0){
167:           while (cont < 45){  //atraso de 1s garantindo o display sempre aceso
168:               atualizaValor(valorTempo);
169:               cont+=1;
170:            }
171:           valorTempo -=1;
172:           cont = 0;
173:        }
174:        if (valorTempo == 0){ //se o valor for 0
175:           atualizaValor(valorTempo);
176:           portc.rc1 = 1;    //toca buzzer
177:           while (cont < 25){  //atraso de 1s garantindo o display sempre aceso
178:               atualizaValor(valorTempo);
179:               cont+=1;
180:            }
181:           portc.rc1 = 0;    //para buzzer
182:        }
```

```
183:      }
184:    } // Fim do loop infinito
185: }    // Fim do programa principal.
```