

```
1: /* ****
2: **                                     Exercício 02                                **
3: **                                     ****                                **
4: ** Ao apertar e segurar o pushbutton RB0, é gerado dois números          **
5: ** aleatórios e são correspondentes a acender os LED's PORTB e          **
6: ** LED's PORTD, quando esses valores são ambos iguais a 6,              **
7: ** o buzzer emite um sinal sonoro de meio segundo.                      **
8: ** ****                                **
9: ** Aluno: Carlos Magno do Nascimento Junior                             **
10: ** ****                                **
11: ** Arquivo: exercicio2.c                                                **
12: ** Compilador : MikroC PRO PIC v7.2.0                                  **
13: ** ****                                **
14: ** UFLA - Lavras /MG - 30/10/2024                                       **
15: *****/
16:
17: // Biblioteca para incluir a função rand()
18: #include <stdlib.h>
19:
20: // variáveis de controle do valor aleatório
21: int ledB=1, ledD=1;
22:
23: // função para gerar valor aleatório
24: int aleatorio(){
25:     return (rand()%6)+1;
26: }
27:
28: // função para acender leds do grupo D, de acordo com o valor de entrada
29: void acendeLedD(int n){
30:     if(n == 1){
31:         portd.rd1 = 1;
32:     }
33:     else if(n == 2){
34:         portd.rd2 = 1;
35:     }
36:     else if(n == 3){
37:         portd.rd3 = 1;
38:     }
39:     else if(n == 4){
40:         portd.rd4 = 1;
41:     }
42:     else if(n == 5){
43:         portd.rd5 = 1;
44:     }
45:     else if(n == 6){
46:         portd.rd6 = 1;
47:     }
48: }
49:
50: // função para acender leds do grupo B, de acordo com o valor de entrada
51: void acendeLedB(int n){
52:     if(n == 1){
53:         portb.rb1 = 1;
54:     }
55:     else if(n == 2){
56:         portb.rb2 = 1;
57:     }
58:     else if(n == 3){
59:         portb.rb3 = 1;
60:     }
61:     else if(n == 4){
62:         portb.rb4 = 1;
```

```
63:     }
64:     else if (n == 5) {
65:         portb.rb5 = 1;
66:     }
67:     else if (n == 6) {
68:         portb.rb6 = 1;
69:     }
70: }
71:
72: void main ()
73: {
74:
75:     // Ativa somente o pino RB0 como entrada da PORTB.
76:     // Os demais pinos são configurados como saída.
77:     // Seta macro "trisb" com valor 0b00000001 (bin) = 0d1 (dec)
78:     trisb = 1;
79:
80:     // Todos os pinos da PORTD (RD) são configurados como saída.
81:     // Seta macro "trisd" com valor 0b00000000 (bin) = 0d0 (dec)
82:     trisd = 0;
83:
84:     // trisc é uma posição (registrador) na memória de dados
85:     // do microcontrolador. Configura o pino RC1 como saída para
86:     // comunicação com o buzzer.
87:     trisc = 0xFD; // 0b11111101
88:
89:     // Inicia com buzzer desligado.
90:     portc.rc1 = 1;
91:
92:     // Desliga todos os LEDs - 0b00000000 (bin) ou 0d0 (dec)
93:     portb = 0;
94:     portd = 0;
95:
96:     // Loop infinito
97:     while(1)
98:     {
99:         // Lê o estado do push button conectado ao pino RB0.
100:        // Caso esteja pressionado (valor = 0), executa laço.
101:        while (portb.rb0 == 0)
102:        {
103:            // Desliga todos os LEDs - 0b00000000 (bin) ou 0d0 (dec)
104:            portb = 0;
105:            portd = 0;
106:
107:            // Verifica o valor gerado aleatoriamente
108:            // se ambos forem iguais a 6, emite o sinal sonoro e aciona os leds
109:            if ((ledB == 6) && (ledD == 6)) {
110:                portc.rc1 = 0;
111:                acendeLedB(ledB);
112:                acendeLedD(ledD);
113:                delay_ms(500);
114:                portc.rc1 = 1;
115:            }
116:            // se não, apenas acende os leds correspondentes ao valor gerado
117:            else {
118:                acendeLedB(ledB);
119:                acendeLedD(ledD);
120:                delay_ms(500);
121:            }
122:            // gera novos valores aleatórios
123:            ledB = aleatorio();
124:            ledD = aleatorio();
```

```
125:     }
126:
127:     // Desliga todos os LEDs - 0b00000000 (bin) ou 0d0 (dec) ao sair do laço
128:     portb = 0;
129:     portd = 0;
130: }
131: }
```