

Exercício Prático

Aula Prática 6

SISTEMAS EMBEBIDOS E DE TEMPO REAL

Licenciatura em Engenharia de Sistemas Informáticos

1º semestre 2021/22

Docente:

Pedro Cunha

Alunos:

João Apresentação 21152

Gonçalo Cunha 21145

Pedro Simões 21140

Barcelos, Portugal

22 de novembro de 2022

Resumo

Neste documento serão demonstrados os exercícios práticos desenvolvidos recorrendo ao uso do Arduino IDE e do Arduino, na aula prática 6 da disciplina de Sistemas Embebidos e de Tempo Real no dia 15/11/2022.

Conteúdo

Resumo	2
Exercício 1 - LED RGB (Ajustar a cor)	5
Resumo	5
Material.....	5
Montagem do Arduino	5
Código fonte.....	6
Exercício 2 - Display de 7 segmentos.....	7
Resumo	7
Material.....	7
Montagem do Arduino	7
Código Fonte	8
Conclusão.....	9

Índice de Imagens

Figura 1 - Exercício Color Wheel (Montagem Arduino)	5
Figura 2 - Exercício Color Wheel (Código Fonte)	6
Figura 3 - Exercício 7 Segments Display (Montagem Arduino)	7
Figura 4 - Exercício 7 Segments Display (Código Fonte)	8

Exercício 1 - LED RGB (Ajustar a cor)

Neste tópico será documentado a realização do primeiro exercício da aula prática recorrendo a uma explicação do código presente no IDE do Arduino e demonstração da montagem do próprio Arduino. Contém um vídeo da demonstração na pasta de trabalho.

Resumo

Neste exercício é realizado um teste de ajuste de cor em um led RGB.

Material

- 1 Led RGB;
- 3 Resistências (220 Ohms);
- 1 Arduino UNO R3;
- 1 Breadboard;
- 5 Fios Breadboard;
- 1 Cabo USB.

Montagem do Arduino

Para o Arduino foi realizada a seguinte montagem.

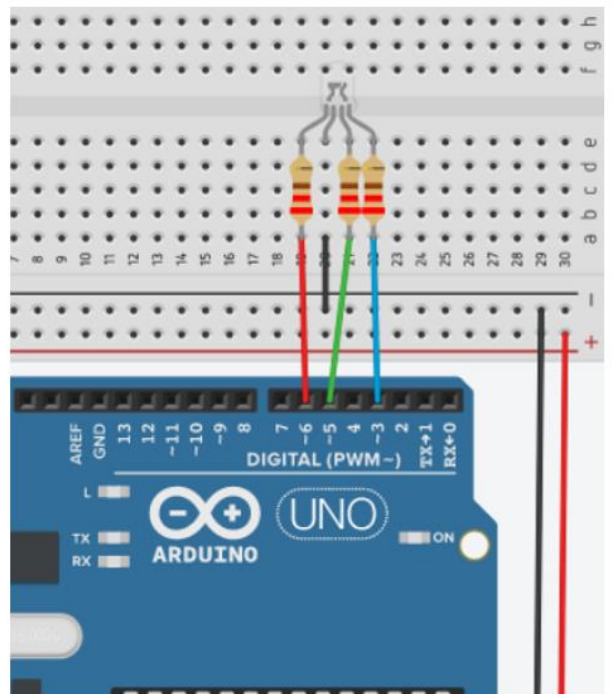
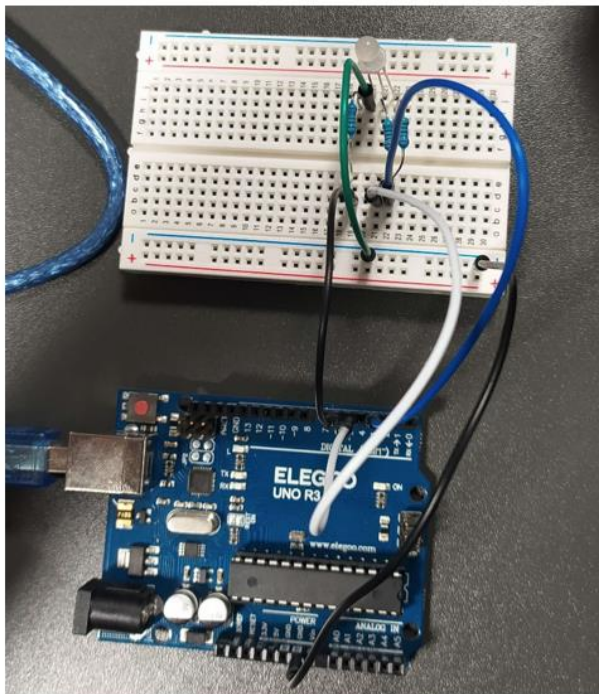


Figura 1 - Exercício Color Wheel (Montagem Arduino)

Código fonte

```
1 const int redPin = 3; //Pino do RED
2 const int greenPin = 5; //Pino do GREEN
3 const int bluePin = 6; //Pino do BLUE
4 const bool invert = true;
5 int color = 0; // valor HUE
6 int R, G, B; // variáveis RGB
7
8 void setup() {
9     // pins driven by analogWrite do not need to be declared as outputs
10 }
11 void loop() {
12     // Valor do brilho (0 - 255)
13     int brightness = 255;
14
15     hueToRGB(color, brightness);
16
17     // Fornecer valores às variáveis
18     analogWrite(redPin, R);
19     analogWrite(greenPin, G);
20     analogWrite(bluePin, B);
21
22     color++;
23
24     if (color > 255)
25         color = 0;
26
27     delay(25);
28 }
29
30 //Função de conversão da cor para os componentes RGB
31 void hueToRGB(int hue, int brightness) {
32     //Hue à escala
33     unsigned int scaledHue = (hue * 6);
```

```
34
35     //Segmento de 0 a 5 (ciclo de cores)
36     unsigned int segment = scaledHue / 256;
37
38     //Posição no segmento
39     unsigned int segmentOffset = scaledHue - (segment * 256);
40
41     unsigned int complement = 0;
42     unsigned int prev = (brightness * (255 - segmentOffset)) / 256;
43     unsigned int next = (brightness * segmentOffset) / 256;
44
45     if (invert) {
46         brightness = 255 - brightness;
47         complement = 255;
48         prev = 255 - prev;
49         next = 255 - next;
50     }
51
52     switch (segment) {
53         case 0: // red
54             R = brightness;
55             G = next;
56             B = complement;
57             break;
58         case 1: // yellow
59             R = prev;
60             G = brightness;
61             B = complement;
62             break;
63         case 2: // green
64             R = complement;
65             G = brightness;
66             B = next;
67             break;
68         case 3: // cyan
69             R = complement;
70             G = prev;
71             B = brightness;
72             break;
73         case 4: // blue
74             R = next;
75             G = complement;
76             B = brightness;
77             break;
78         case 5: // magenta
79             default:
80                 R = brightness;
81                 G = complement;
82                 B = prev;
83                 break;
84     }
85 }
```

Figura 2 - Exercício Color Wheel (Código Fonte)

Exercício 2 - Display de 7 segmentos

Neste tópico será documentado a realização do segundo exercício da aula prática recorrendo a uma explicação do código presente do IDE do Arduino e demonstração da montagem do próprio Arduino. Contém um vídeo da demonstração na pasta de trabalho.

Resumo

Neste exercício é realizado um teste em um display de 7 segmentos, no qual será ligado e desligado partes específicas e desta forma criando padrões para exibição do número desejado. Estes tipos de displays são utilizados em relógios digitais, calculadoras, entre outros.

Material

- 1 Painel digital de 7 segmentos;
- 1 Resistências (220 Ohms);
- 1 Arduino UNO R3;
- 1 Breadboard;
- 9 Fios Breadboard;
- 1 Cabo USB.

Montagem do Arduino

Para o Arduino foi realizada a seguinte montagem.

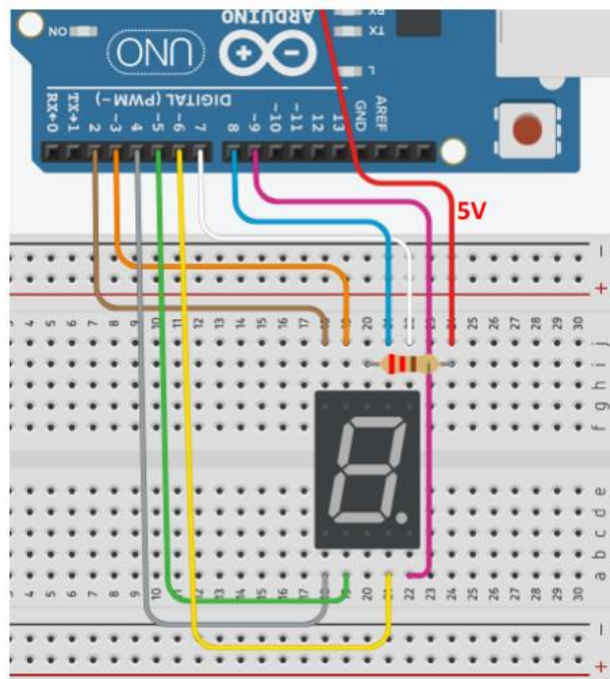
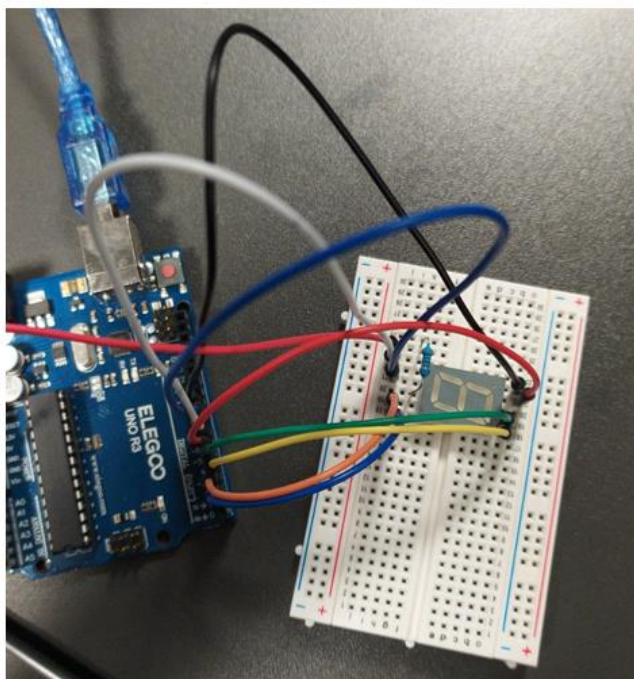


Figura 3 - Exercício 7 Segments Display (Montagem Arduino)

Código Fonte

```
13 #define A 8
14 #define B 7
15 #define C 6
16 #define D 5
17 #define E 4
18 #define F 3
19 #define G 2
20 #define DP 9 // decimal
21 #define common_cathode 0
22 #define common_anode 1
23 bool segMode = common_anode; // Tipo de Segmento
24 int seg[] { A, B, C, D, E, F, G, DP }; // Pins do Segmento
25 byte chars = 35;
26 byte Chars[35][0] {
27 { '0', 1, 1, 1, 1, 1, 1, 0, 0 }, //0
28 { '1', 0, 1, 1, 0, 0, 0, 0, 0 }, //1
29 { '2', 1, 1, 0, 1, 1, 0, 1, 0 }, //2
30 { '3', 1, 1, 1, 1, 0, 0, 1, 0 }, //3
31 { '4', 0, 1, 1, 0, 0, 1, 1, 0 }, //4
32 { '5', 1, 0, 1, 1, 0, 1, 1, 0 }, //5
33 { '6', 1, 0, 1, 1, 1, 1, 1, 0 }, //6
34 { '7', 1, 1, 1, 1, 0, 0, 0, 0 }, //7
35 { '8', 1, 1, 1, 1, 1, 1, 1, 0 }, //8
36 { '9', 1, 1, 1, 1, 1, 0, 1, 0 }, //9
37 { 'a', 1, 1, 1, 0, 1, 1, 1, 0 }, //A/10
38 { 'b', 0, 0, 1, 1, 1, 1, 1, 0 }, //b/11
39 { 'c', 1, 0, 0, 1, 1, 1, 0, 0 }, //C/12
40 { 'd', 0, 1, 1, 1, 1, 0, 1, 0 }, //d/13
41 { 'e', 1, 0, 0, 1, 1, 1, 1, 0 }, //E/14
42 { 'f', 1, 0, 0, 0, 1, 1, 1, 0 }, //F/15
43 { 'g', 1, 0, 1, 1, 1, 1, 0, 0 }, //G/16
44 { 'h', 0, 1, 1, 0, 1, 1, 1, 0 }, //H/17
45 { 'i', 0, 0, 0, 0, 1, 1, 0, 0 }, //I/18
46 { 'j', 0, 1, 1, 1, 1, 0, 0, 0 }, //J/19
47 { 'l', 0, 0, 0, 1, 1, 1, 0, 0 }, //L/20
48 { 'n', 0, 0, 1, 0, 1, 0, 1, 0 }, //n/21
49 { 'o', 0, 0, 1, 1, 1, 0, 1, 0 }, //o/22
50 { 'p', 1, 1, 0, 0, 1, 1, 1, 0 }, //P/23
51 { 'q', 1, 1, 1, 0, 0, 1, 1, 0 }, //q/24
52 { 'r', 0, 0, 0, 1, 0, 1, 0, 0 }, //r/25
53 { 's', 1, 0, 1, 1, 0, 1, 1, 0 }, //S/26 looks like number 5
54 { 't', 0, 0, 0, 1, 1, 1, 1, 0 }, //t/27
55 { 'u', 0, 1, 1, 1, 1, 1, 0, 0 }, //U/28
56 { 'y', 0, 1, 1, 1, 0, 1, 1, 0 }, //y/29
57 { '-', 0, 0, 0, 0, 0, 0, 1, 0 }, //-/30
58 { '.', 0, 0, 0, 0, 0, 0, 0, 1 }, //./31
59 { ']', 1, 1, 1, 1, 1, 0, 0, 0 }, //]/32
60 { '[', 1, 0, 0, 1, 1, 1, 0, 0 }, //[/33
61 { '_', 0, 0, 0, 1, 0, 0, 0, 0 }, //_/34
62 };
63
64 void setup() {
65 // Define os pins dos segmentos como OUTPUT
66 pinMode(seg[0], OUTPUT);
67 pinMode(seg[1], OUTPUT);
68 pinMode(seg[2], OUTPUT);
69 pinMode(seg[3], OUTPUT);
70 pinMode(seg[4], OUTPUT);
71 pinMode(seg[5], OUTPUT);
72 pinMode(seg[6], OUTPUT);
73 pinMode(seg[7], OUTPUT);
74 }
75 void setState(bool mode)
76 {
77 for (int i = 0; i <= 6; i++) {
78 digitalWrite(seg[i], mode);
79 }
80 }
81 void Print(char Char) // Printa qualquer caracter no segmento (exceto maiusculas)
82 {
83 int charNum = -1;
84
85 //Desativa o segmento
86 setState(segMode);
87
88 for (int i = 0; i < chars; i++) { // Procura o caracter introduzido
89 if (Char == Chars[i][0]) { // Quando caracter for encontrado
90 charNum = i;
91 }
92 }
93
94 if (charNum == -1) //Se caracter nao for encontrado
95 {
96 for (int i = 0; i <= 6; i++) {
97 digitalWrite(seg[i], HIGH);
98 delay(100);
99 digitalWrite(seg[i], LOW);
100 }
101
102 for (int i = 0; i <= 2; i++) {
103 delay(100);
104 setState(HIGH);
105 delay(100);
106 setState(LOW);
107 }
```

```
107 }
108 }
109 else
110 {
111 for (int i = 0; i < 8; i++) {
112 //Printar caracter no caso de o encontrar
113 digitalWrite(seg[i], Chars[charNum][i + 1]);
114 }
115 }
116 }
117 void Print(int num) // Printar qualquer numero no segmento
118 {
119 //Desativa o segmento
120 setState(segMode);
121
122 if (num > chars || num < 0) // Se o numero nao for declarado
123 {
124 for (int i = 0; i <= 6; i++) {
125 digitalWrite(seg[i], HIGH);
126 delay(100);
127 digitalWrite(seg[i], LOW);
128 }
129
130 for (int i = 0; i <= 2; i++) {
131 delay(100);
132 setState(HIGH);
133 delay(100);
134 setState(LOW);
135 }
136 } else // Se for declarado, printa-o
137 {
138 if (segMode == 0) {
139 for (int i = 0; i < 8; i++) {
140 digitalWrite(seg[i], Chars[num][i + 1]);
141 }
142 }
143 else
144 {
145 for (int i = 0; i < 8; i++) {
146 digitalWrite(seg[i], !Chars[num][i + 1]);
147 }
148 }
149 }
150 }
151
152 void loop() {
153 for (int i = 0; i < chars; i++) //print
154 {
155 Print(i);
156 delay(1000);
157 }
158 }
```

Figura 4 - Exercício 7 Segments Display (Código Fonte)

Conclusão

Esta aula prática desenvolveu as nossas capacidades para trabalhar com Arduino e o seu IDE com recurso a lâmpadas led RGB e displays com segmentos. Estes proporcionaram além de novas capacidades, ideias para futuros projetos não só da unidade curricular em questão como extracurriculares.