

Universidade Federal de São Carlos

Laboratório de Microcontroladores e Aplicações
Professor Dr. Edilson Kato

Relatório 2 - Turma A

Bruna Zamith (RA: 628093)
Matheus Vrech (RA: 727349)

10/2018
São Carlos - SP, Brasil

Conteúdo

1	Introdução	1
2	Descrição	1
3	Materiais Utilizados	1
3.1	Arduino	2
3.2	Motor de Passo	2
4	Desenvolvimento	3
4.1	Versionamento	3
4.2	Página Web	3
4.3	Implementação	5
5	Resultados	10

1 Introdução

O presente relatório visa detalhar as atividades desenvolvidas ao longo das aulas de Laboratório de Microcontroladores e Aplicações ministradas nos dias 21/09 e 05/10 pelo professor Dr. Edilson Kato no Departamento de Computação da Universidade Federal de São Carlos.

O documento está organizado da seguinte forma: A Seção 2 descreve o projeto proposto e seus objetivos principais; a Seção 3 expõe os materiais utilizados para a implementação do projeto; a Seção 4 detalha o seu desenvolvimento; por fim, a Seção 5 expõe e discute os resultados obtidos.

2 Descrição

O projeto a ser detalhado neste relatório objetiva o aprofundamento do conhecimento em motores, Arduino e demais tecnologias.

A proposta era utilizar uma página Web e um Arduino para controlar um motor de passo. Assim, através de:

1. Um botão “Reverse” na página Web, muda-se o sentido de rotação do motor;
2. Um botão “Fast” na página Web, faz com que o motor gire na velocidade máxima;
3. Um botão “Medium” na página Web, faz com que o motor gire na velocidade média;
4. Um botão “Slow” na página Web, faz com que o motor gire na velocidade mínima;
5. Um botão “Stop” na página Web, faz com que o motor desacelere e pare;
6. Um botão “Rotate” na página Web, faz com que o motor comece a girar caso ele esteja parado;
7. Um botão “90 degrees” na página Web, faz com que o motor gire a um ângulo igual a 90;
8. Um botão “180 degrees” na página Web, faz com que o motor gire a um ângulo igual a 180.

3 Materiais Utilizados

Para a implementação do projeto descrito na seção anterior, foram utilizados:

- 1 Arduino Mega;
- 1 *Shield* Ethernet;
- 1 Cabo Ethernet;
- 1 Motor de passo;
- 1 Arduino *Integrated Development Environment* (IDE).

3.1 Arduino

O Arduino é uma plataforma *open-source* de *hardware* e *software*, capaz de ler entradas (como sensores e botões) e transformá-las em saídas (como motor e LED). Ele permite a prototipagem eletrônica de *hardware* livre e é projetado com um microcontrolador Atmel AVR¹. É possível programá-lo através de sua IDE e de sua linguagem de programação própria, sendo esta última semelhante à linguagem C. A Figura 1 exibe um Arduino e a Figura 2, a IDE.



Figura 1: Arduino

```

sketch_jan04a | Arduino 1.8.5
File Edit Sketch Tools Help
sketch_jan04a.ino
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}

```

1 Arduino/Genuino Uno on COM1

Figura 2: IDE do Arduino

3.2 Motor de Passo

Um motor de passo é um dispositivo que converte pulsos elétricos em movimentos mecânicos, gerando variações angulares discretas. O eixo de um motor de passo é rotacionado de

¹<https://www.arduino.cc/en/Guide/Introduction>

acordo com pequenos incrementos angulares, os “passos”². A Figura 3 exibe um Motor de Passo.



Figura 3: Motor de passo

4 Desenvolvimento

4.1 Versionamento

O controle de versões do nosso código está sendo feito a partir do GitHub, repositório público “Zavech”³. O projeto detalhado neste relatório e todos os outros projetos da disciplina serão incluídos nesse mesmo repositório.

4.2 Página Web

A página Web do projeto foi desenvolvida fazendo uso de HTML e Javascript, e das tecnologias Ajax e Bootstrap. É composta de duas páginas principais: “Home”, contendo informações sobre o projeto, lista de tarefas e instruções de como conectar a placa; “Activities”, a qual possui todas as *features* implementadas.

O site pode ser acessado a partir do *link*: <https://whoismath.github.io/zavech/>.

As Figuras 4 e 5 exibem as interfaces da página Web. Para correto funcionamento, é preciso apenas que o servidor e o arduino estejam conectados na mesma rede.

²[https://www.telecom.uff.br/pet/petws/downloads/tutoriais/steppermotor/steppermotor2k81119.pdf](https://www.telecom.uff.br/pet/petws/downloads/tutoriais/stepmotor/steppermotor2k81119.pdf)

³<https://github.com/whoismath/zavech>

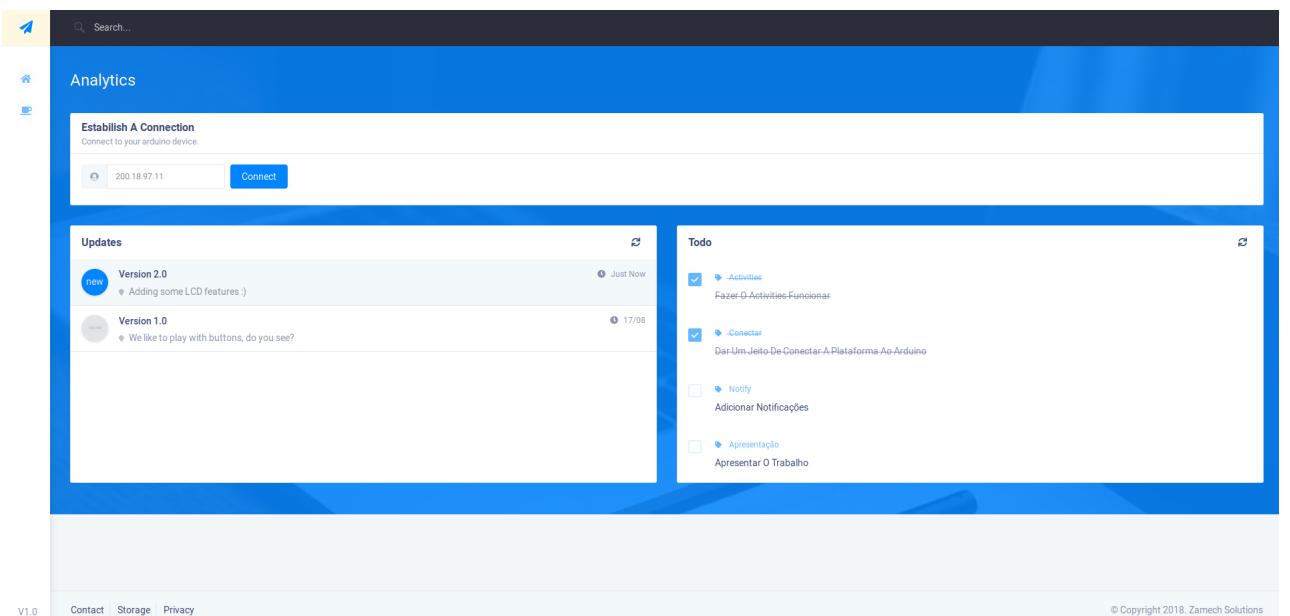


Figura 4: Página “Home”

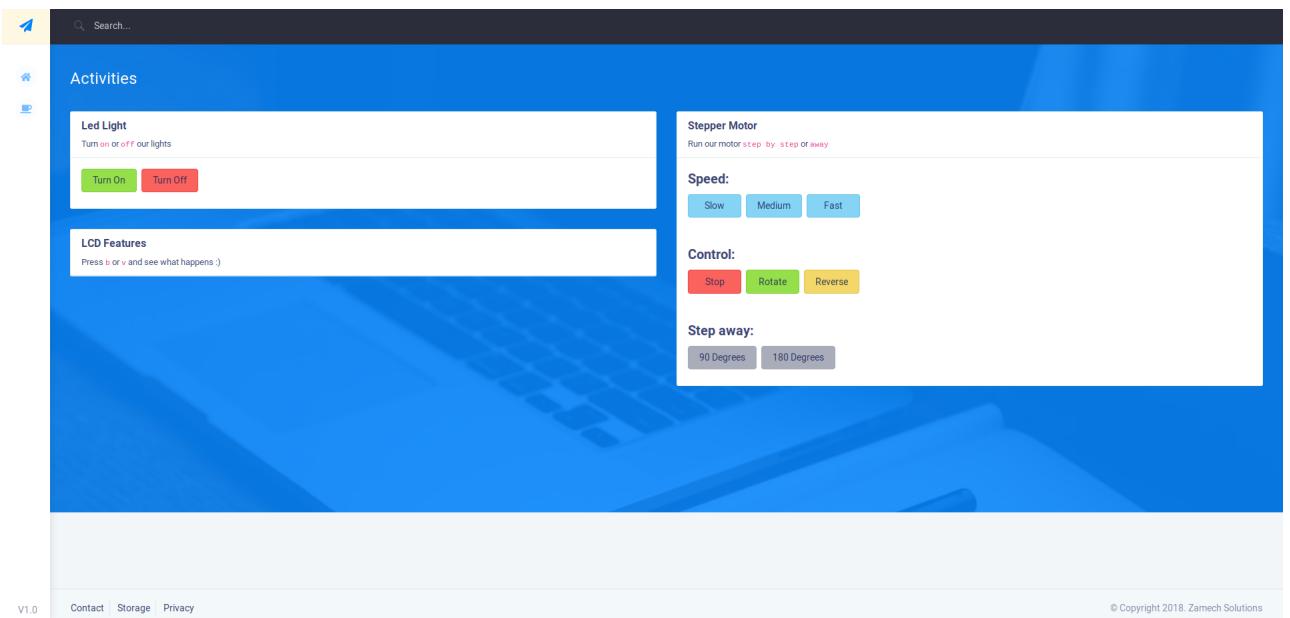


Figura 5: Página “Activities”

4.3 Implementação

As conexões podem ser visualizadas na Figura 6.

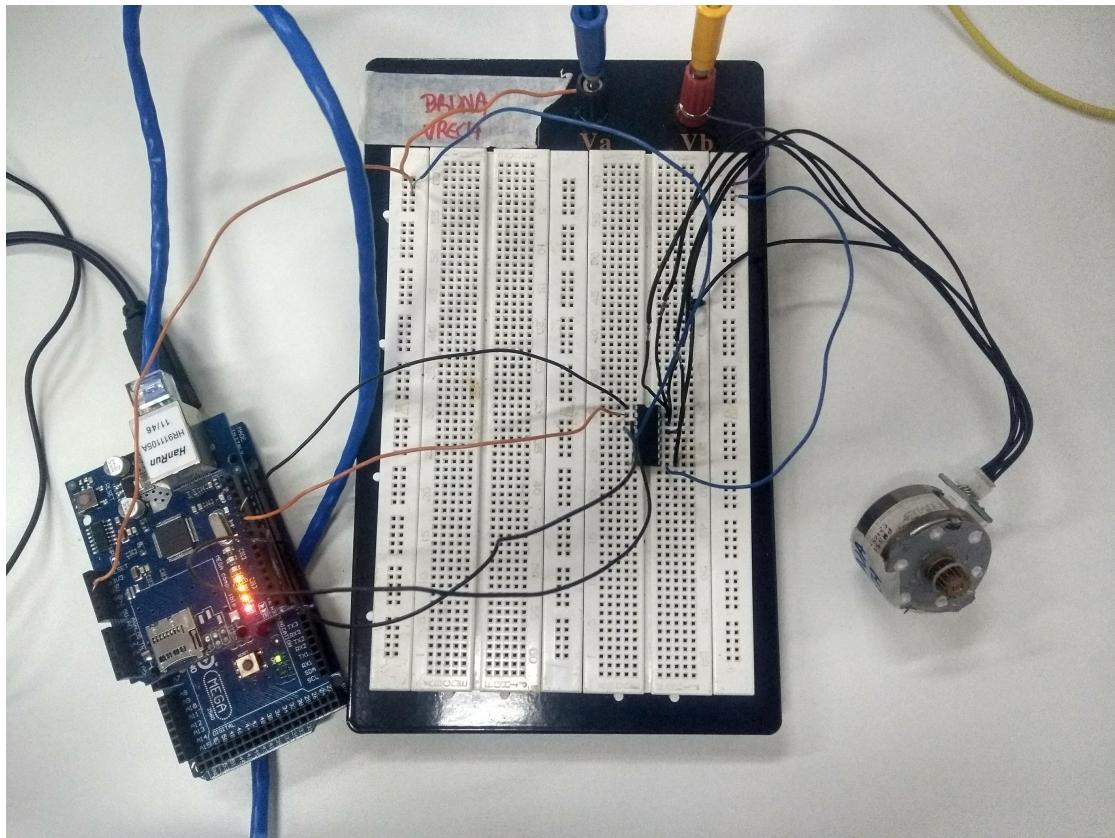


Figura 6: Circuito final

Para a programação, foram primeiramente importadas 5 diferentes bibliotecas:

- <**SPI**>: *Serial Peripheral Interface*, permite a conexão com periféricos através do protocolo SPI⁴;
- <**Ethernet**>: Desenvolvida para o *Shield Ethernet*, permitindo que o Arduino conecte-se à internet⁵;
- <**Wire**>: Permite a comunicação com dispositivos I2C/TWI, como é o caso do LCD⁶;
- <**Stepper**>: Permite o controle de motores de passo unipolares e bipolares⁷;

⁴<https://www.arduino.cc/en/Reference/SPI>

⁵<https://www.arduino.cc/en/Reference/Ethernet>

⁶<https://www.arduino.cc/en/Reference/Wire>

⁷<https://www.arduino.cc/en/Reference/Stepper>

- **rgb_lcd.h**: Permite a mudança de cor de fundo do LCD, dentre outras funcionalidades⁸.

O código final `.ino` pode ser encontrado em <https://github.com/whoismath/zavech/blob/master/class3.ino> e também exposto a seguir.

```

1 #include <SPI.h>
2 #include <Stepper.h>
3 #include <Ethernet.h>
4 #include <Wire.h>
5 #include "rgb_lcd.h"

7 rgb_lcd lcd;

9 const int pinButton = 4;
10 int counter = 0;

11 byte mac[] = {
12     0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
13 };
14

15 IPAddress ip(200, 18, 97, 11);
16 EthernetServer server(80);

19 /* Motor definitions */
20 const int stepsPerRevolution = 200;
21 Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);

23 int _direction = 1;
24 boolean _running = 0;

25
26 void setup() {
27     Serial.begin(9600);
28     while (!Serial) {}
29     Ethernet.begin(mac, ip);
30     server.begin();
31     Serial.print("server is at ");
32     Serial.println(Ethernet.localIP());

33     /* set up the LCD's number of columns and rows: */
34     lcd.begin(16, 2);
35     lcd.setRGB(186, 85, 211);

36
37

```

⁸http://wiki.seeedstudio.com/Grove-LCD_RGB_Backlight/

```

/* Print a message to the LCD. */
39 lcd.print("Press B or V!");

41 /* Define a led to use */
42 pinMode(LED_BUILTIN, OUTPUT);
43 pinMode(pinButton, INPUT);
44 }

45

47 void loop() {
48 // myStepper.step(1);
49 if (digitalRead(pinButton)) {
50   counter++;
51 }

53 if (counter > 0){
54   if (counter % 2 == 0) {
55     display_lcd("Bruna Zamith", "628093");
56   } else {
57     display_lcd("Matheus Vrech", "727349");
58   }
59 }

61 if (_running == 1)
62   myStepper.step(_direction * 20);

63

64 EthernetClient client = server.available();
65 /* Get the http packet */
66 String readString;
67 if (client) {
68   boolean currentLineIsBlank = true;
69   while (client.connected()) {
70     if (client.available()) {
71       char c = client.read();

73       Serial.write(c);

75       /* Store the entire packet */
76       if (readString.length() < 100) {
77         readString += c;
78       }
79     }

```

```

81      /* Receive GET commands */
82      if (readString.indexOf("lighton") > 0) {
83          digitalWrite(LED_BUILTIN, HIGH);
84          client.println("HTTP/1.1 200 OK");
85          break;
86      }
87      else if (readString.indexOf("lightoff") > 0) {
88          digitalWrite(LED_BUILTIN, LOW);
89          client.println("HTTP/1.1 200 OK");
90          break;
91      }
92      else if (readString.indexOf("bruna") > 0) {
93          display_lcd("Bruna Zamith", "628093");
94          client.println("HTTP/1.1 200 OK");
95          break;
96      }
97      else if (readString.indexOf("vrechson") > 0) {
98          display_lcd("Matheus Vrech", "727349");
99          client.println("HTTP/1.1 200 OK");
100         break;
101     }
102     /* working with our motor right now */
103     else if (readString.indexOf("slow") > 0) {
104         myStepper.setSpeed(10);
105         client.println("HTTP/1.1 200 OK");
106         break;
107     }
108     else if (readString.indexOf("medium") > 0) {
109         myStepper.setSpeed(40);
110         client.println("HTTP/1.1 200 OK");
111         break;
112     }
113     else if (readString.indexOf("fast") > 0) {
114         myStepper.setSpeed(75);
115         client.println("HTTP/1.1 200 OK");
116         break;
117     }
118     else if (readString.indexOf("stop") > 0) {
119         _running = 0;
120         client.println("HTTP/1.1 200 OK");
121         break;
122     }

```

```

    else if (readString.indexOf("90") > 0) {
123      myStepper.step(_direction * 27);
      _running = 0;
125      client.println("HTTP/1.1 200 OK");
      break;
127    }
    else if (readString.indexOf("180") > 0) {
129      myStepper.step(_direction * 55);
      _running = 0;
131      client.println("HTTP/1.1 200 OK");
      break;
133    }
    else if (readString.indexOf("rotate") > 0) {
135      _running = 1;
      client.println("HTTP/1.1 200 OK");
      break;
137    }
139    else if (readString.indexOf("reverse") > 0) {
      _direction = _direction ? -1 : 1;
141      client.println("HTTP/1.1 200 OK");
      break;
143    }
/* end of motor functions */
145    if (c == '\n' && currentLineIsBlank) {
      client.println("HTTP/1.1 200 OK");
147      client.println("Content-Type: text/html");
      client.println("Connection: keep-alive");
149      //client.println("Refresh: 5"); // refresh the page
      automatically every 5 sec
      client.println();
      client.println("<!DOCTYPE HTML>");
      client.println("<html>");
153      client.println("<head></head>");
      /* easter egg for ninjas */
155      client.println("<body>i'm a happy arduino, and you found my
      easter egg!<br>curiosity will kill you anyway.</body>");
      client.println("</html>");
157    }
    if (c == '\n') {
159      currentLineIsBlank = true;
    } else if (c != '\r') {
      currentLineIsBlank = false;
161

```

```

        }
    }

165    /* wait for something: nothing at all*/
167    delay(1);
168    // client.stop();
169    // Serial.println("client disconnected");
170}
171}

173 void display_lcd(String nome, String ra){
174     lcd.clear();
175     if(nome.indexOf("Zamith") > 0){
176         lcd.setRGB(255, 127, 80);
177     }
178     else if(nome.indexOf("Vrech") > 0){
179         lcd.setRGB(135, 206, 235);
180     }
181     lcd.setCursor(0, 0);
182     lcd.print(nome);
183     lcd.setCursor(0,1);
184     lcd.print(ra);
185     delay(300);
186}

```

5 Resultados

Todos os resultados obtidos se mostraram satisfatórios, sendo possível implementar e testar as funcionalidades propostas, incrementais às implementações previamente desenvolvidas.