

Universidade Federal de São Carlos

Laboratório de Microcontroladores e Aplicações
Professor Dr. Edilson Kato

Relatório 4 - 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	Circuito Ponte H	3
3.3	PWM	3
4	Desenvolvimento	3
4.1	Versionamento	3
4.2	Página Web	4
4.3	Implementação	6
5	Resultados	11

1 Introdução

O presente relatório visa detalhar as atividades desenvolvidas ao longo das aulas de Laboratório de Microcontroladores e Aplicações ministradas nos dias 12/10, 19/10 e 26/10 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 corrente contínua. Para tal, tivemos que montar um circuito da ponte H para controlar a corrente e a tensão no motor. Ainda, a velocidade e o sentido de rotação do motor foi feita através de sinais PWM.

Os controles foram feitos 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;

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 CC;
- 4 Transistores TIP122;
- 4 Resistores;
- Jumpers;
- Estanho;
- Ferro de solda;
- Placa para realizar as ligações;
- 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



Figura 2: IDE do Arduino

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

3.2 Circuito Ponte H

O circuito de ponte H é responsável por determinar o sentido de corrente e o valor de tensão no controle de um motor CC. Esse circuito é largamente utilizado, sendo montado com transistores e/ou relés. Ainda, existem circuitos integrados ponte H, com todo o circuito interno². A Figura 3 exibe um circuito Ponte H.

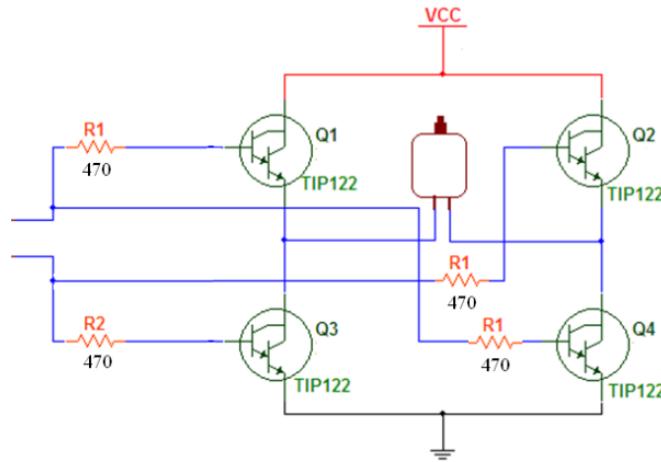


Figura 3: Ponte H. Retirado do material de apoio da disciplina.

3.3 PWM

O sinal PWM (do inglês, *Pulse-Width Modulation*) é um sinal digital que permite controlar a tensão de determinado dispositivo de acordo com o *duty cycle* do sinal. Uma vantagem do PWM é que o sinal permanece digital em todo o percurso (do processador até o sistema controlado) e não é necessária nenhuma conversão de digital para analógico. A Figura 4 apresenta o funcionamento de um PWM.

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.

²<http://engenheirocaicara.com/circuito-ponte-h/>

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

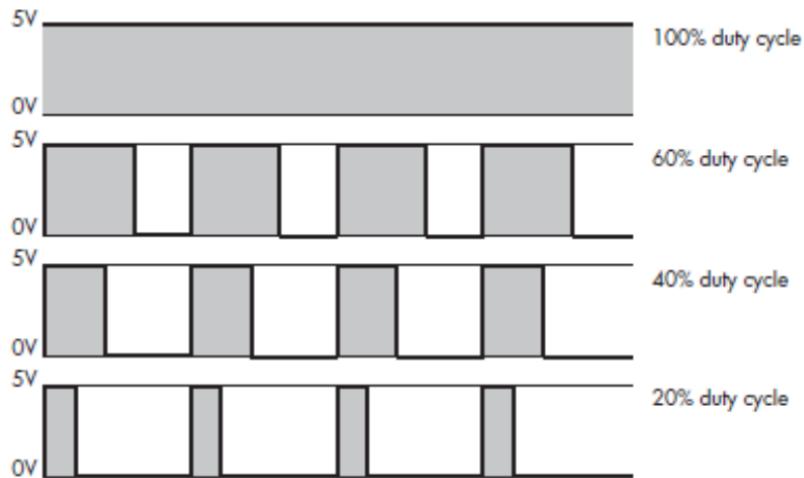


Figura 4: PWM. Retirado do material de apoio da disciplina.

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 possuí todas as *features* implementadas.

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

As Figuras 5 e 6 exibem as interfaces da página Web. Para correto funcionamento, é preciso apenas que o servidor e o arduino estejam conectados na mesma rede.

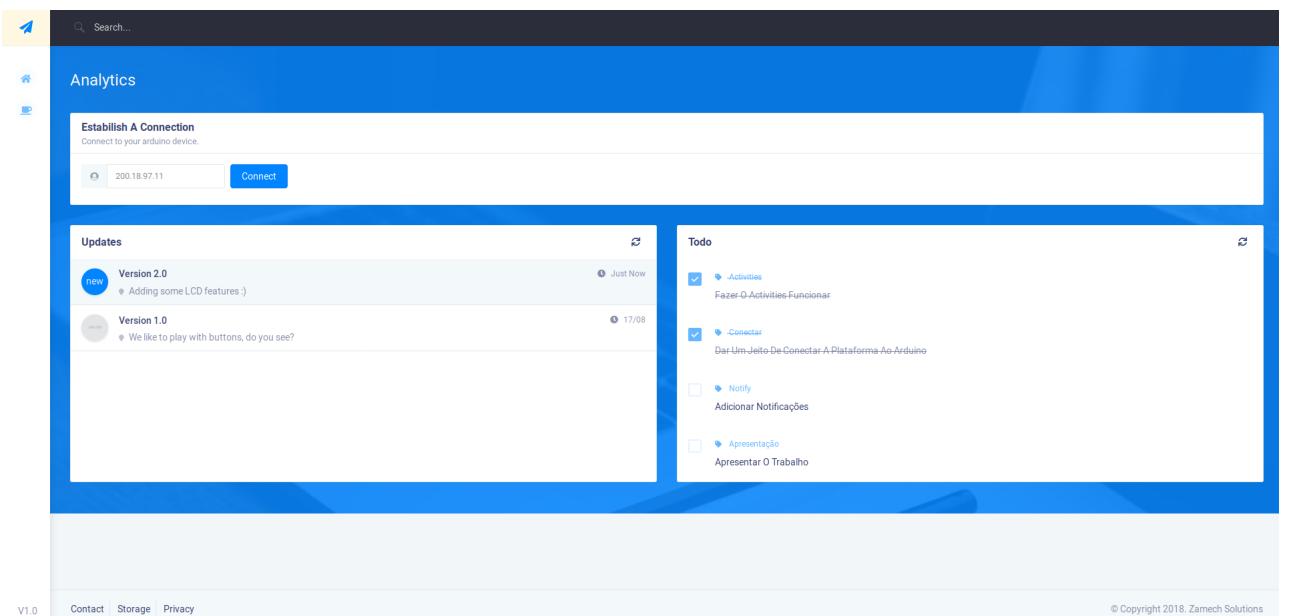


Figura 5: Página “Home”

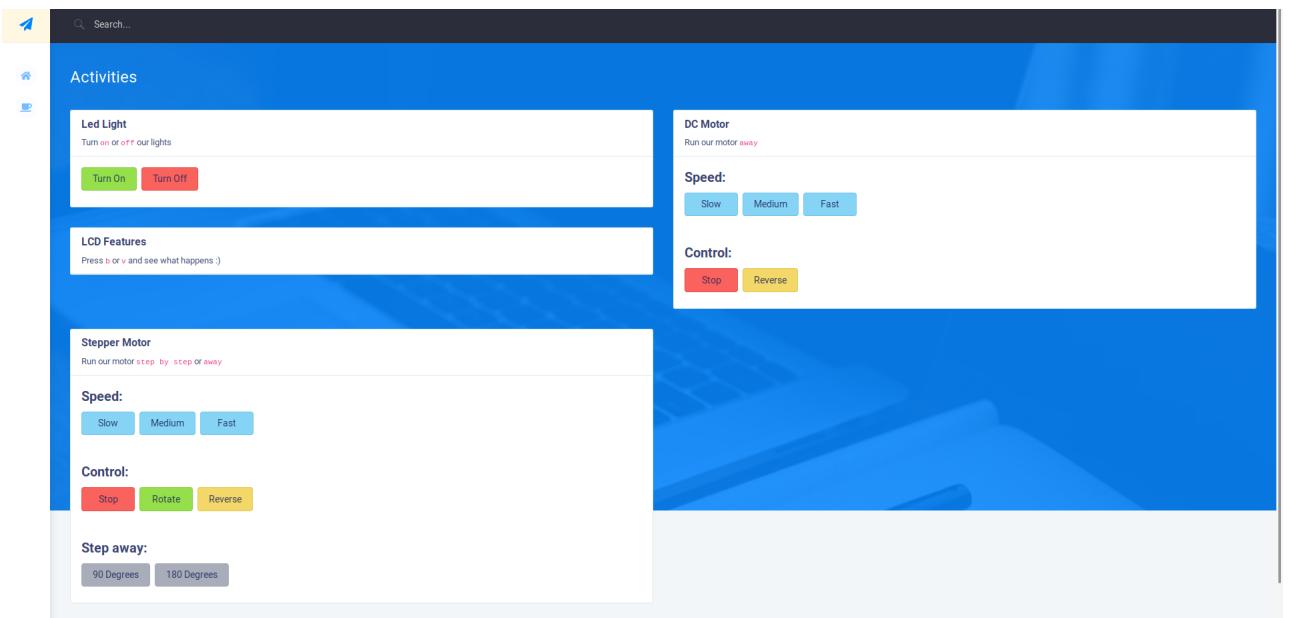


Figura 6: Página “Activities”

4.3 Implementação

As conexões podem ser visualizadas nas Figuras 7 e 8.

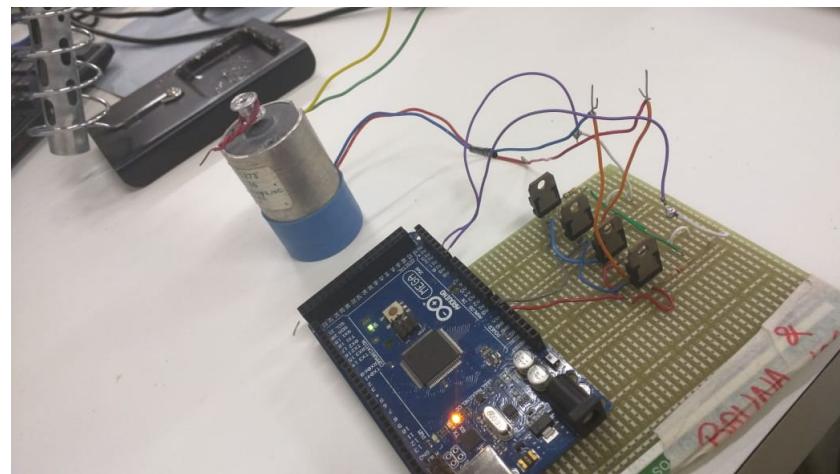


Figura 7: Circuito final

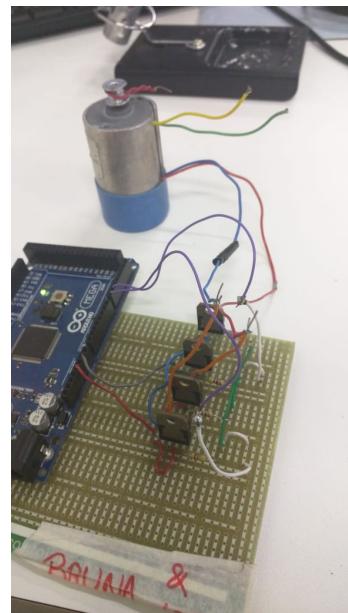


Figura 8: Circuito final

Para a programação, foram primeiramente importadas 3 diferentes bibliotecas:

- <**SPI**>: *Serial Peripheral Interface*, permite a conexão com periféricos através do protocolo SPI⁴;
- <**Wire**>: Permite a comunicação com dispositivos I2C/TWI, como é o caso do LCD⁵;
- <**Stepper**>: Permite o controle de motores de passo unipolares e bipolares⁶;

O código final **.ino** pode ser encontrado em <https://github.com/whoismath/zavech/blob/master/class4.ino> e também exposto a seguir.

```
1  /*
2   * Web Server
3   * A simple web server that shows the value of the analog input pins.
4   * using an Arduino Wiznet Ethernet shield.
5   * Circuit:
6   *   * Ethernet shield attached to pins 10, 11, 12, 13
7   *   * Analog inputs attached to pins A0 through A5 (optional)
8   * created 18 Dec 2009
9   * by David A. Mellis
10  * modified 9 Apr 2012
11  * by Tom Igoe
12  * modified 02 Sept 2015
13  * by Arturo Guadalupi
14  * Besides that, this project was modified by Matheus Vrech and Bruna Zamith
15  * for university classes purpose, enjoy it
16  */
17 /*
18  * Libraries and Definitions
19  ****
20 #include <SPI.h>
21 #include <Ethernet.h>
22 #include <Wire.h>
23
24 /*
25  * Ethernet Configuration
26  ****
27 byte mac[] = {
```

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

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

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

```

    0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
29 }

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

33

35 /* Motor Configuration
***** */
37 int motorPin1 = 5,
38     motorPin2 = 6,
39     _speed = 0,
40     _direction = 1;

41

43 /* Setup
***** */
45 void setup() {
46     Serial.begin(9600);
47     while (!Serial) {}
48     Ethernet.begin(mac, ip);
49     server.begin();
50     Serial.print("server is at ");
51     Serial.println(Ethernet.localIP());

53     pinMode(5, OUTPUT);
54     pinMode(6, OUTPUT);
55 }

57 /*
58 * Loop
59 ***** */
60 void loop() {
61     //myStepper.step(1);
62     if (digitalRead(pinButton)) {
63         counter++;
64     }

65     EthernetClient client = server.available();
66     /* Get the http packet */
67     String readString;
68     if (client) {

```

```

    boolean currentLineIsBlank = true;
71   while (client.connected()) {
72     if (client.available()) {
73       char c = client.read();

75       Serial.write(c);

77       /* Store the entire packet */
78       if (readString.length() < 100) {
79         readString += c;
80       }

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       /* working with our motor right now */
93       else if (readString.indexOf("dc_slow") > 0) {
94         _speed = 160;
95         client.println("HTTP/1.1 200 OK");
96         break;
97       }
98       else if (readString.indexOf("dc_medium") > 0) {
99         _speed = 200;
100        client.println("HTTP/1.1 200 OK");
101        break;
102      }
103      else if (readString.indexOf("dc_fast") > 0) {
104        _speed = 255;
105        client.println("HTTP/1.1 200 OK");
106        break;
107      }
108      else if (readString.indexOf("dc_stop") > 0) {
109        _speed = 0;
110        client.println("HTTP/1.1 200 OK");
111      }

```

```

        break;
113    }
114    else if (readString.indexOf("dc_reverse") > 0) {
115        _direction = _direction ? -1 : 1;
116        client.println("HTTP/1.1 200 OK");
117        break;
118    }
119    /* end of motor functions */
120    if (c == '\n' && currentLineIsBlank) {
121        client.println("HTTP/1.1 200 OK");
122        client.println("Content-Type: text/html");
123        client.println("Connection: keep-alive");
124        // client.println("Refresh: 5"); // refresh the page
125        // automatically every 5 sec
126        client.println();
127        client.println("<!DOCTYPE HTML>");
128        client.println("<html>");
129        client.println("<head></head>");
130        /* easter egg for ninjas */
131        client.println("<body>i'm a happy arduino, and you found my
easter egg!<br>curiosity will kill you anyway.</body>");
132        client.println("</html>");
133    }
134    if (c == '\n') {
135        currentLineIsBlank = true;
136    } else if (c != '\r') {
137        currentLineIsBlank = false;
138    }
139}
140
141 /* wait for something: nothing at all*/
142 delay(1);
143 // client.stop();
144 // Serial.println("client disconnected");
145}

```

5 Resultados

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