

Universidade Federal de Goiás
Regional Catalão
Curso de Bacharelado em Ciências da Computação

Arquitetura de Computadores

Maquina de Lavar com Arduino

**Marco Túlio Macedo Rodrigues, Vitor do Vale Bernardo, Pablo
da Silva Vinicius**

Catalão – GO
2017

**Marco Túlio Macedo Rodrigues, Vitor do Vale Bernardo, Pablo da
Silva Vinicius**

Arquitetura de Computadores

Maquina de Lavar com Arduino

Trabalho apresentado ao Curso de Bacharelado em
Ciências da Computação da Universidade Federal
de Goiás – Regional Catalão, como parte dos
requisitos para obtenção de nota para aprovação
na disciplina.

Orientador: Prof. Dr. Tercio Alberto dos
Santos Filho

Catalão – GO
2017

**Marco Túlio Macedo Rodrigues, Vitor do Vale Bernardo, Pablo da
Silva Vinicius**

Arquitetura de Computadores

Maquina de Lavar com Arduino

Trabalho apresentado ao curso de
Bacharelado em Ciências da Computação da
Universidade Federal de Goiás – Regional
Catalão.

Trabalho aprovado em 29 de Março de 2017.

Tercio Alberto dos Santos Filho
Orientador

Catalão – GO
2017

*Este trabalho é dedicado a todas as pessoas que,
Sonham em aprender mais sobre Arduino.
Em especial, aos que são apaixonados por tecnologia*

AGRADECIMENTOS

Os agradecimentos principais são direcionados à Deus que até aqui nós sustentou e nos ajudou a prevalecer.

Agradecimentos especiais são direcionados ao professor Dr. Tércio Alberto dos Santos Filho que sempre se dispôs à ajudar os alunos da disciplina e nunca mediu esforços para sanar as dúvidas de seus alunos, seja fora ou dentro de sala. Deixamos aqui nossos mais sinceros agradecimentos.

*“As invenções são, sobretudo,
o resultado de um trabalho de teimoso.”
(Santos Dumont)*

RESUMO

A proposta desse projeto é simular o funcionamento de uma Máquina de Lavar roupas através de periféricos e um Kit Arduino Uno. Será retratado todos os aspectos de uma máquina real, tais como: Motor de rotação, Bomba d'água, Painel de informações através de LED's para diferenciar os diferentes de modos de operação e níveis de água. O usuário escolhe os diferentes modos de operação através de um botão e níveis de água também. O sistema é de fato iniciado, após um botão principal ser pressionado, quando o nível de água e modo desejado já selecionado.

Palavras-chave: Arquitetura de computadores, arduino, maquina de lavar, MPL.

ABSTRACT

One project proposal is to simulate or operate a Peripheral Washer machine and an Arduino Uno Kit. Be pictured all of a real machine such as: Rotating motor, Water pump, LED information panel to differentiate the different operating modes and water levels. And the different modes of operation through a key and water levels as well. The system is actually started, after a button, when the desired water level and mode has already been selected.

Keywords: Computer Architecture, Arduino, Washing Machine, PWM.

LISTA DE ILUSTRAÇÕES

Figura 1 – Circuito na ProtoBoard	27
Figura 2 – Esquemático do Circuito	28

SUMÁRIO

1	INTRODUÇÃO	17
2	DEFINIÇÃO DO PROBLEMA	19
2.1	Construção do Problema	19
2.2	Modo Rápido	20
2.3	Modo Normal	21
2.4	Modo Pesado	21
3	DISPOSITIVOS UTILIZADOS	23
3.1	Arduino Uno	24
3.2	Modulação por Largura de Banda	24
4	ESQUEMÁTICO E MONTAGEM	27
4.1	Montagem do Circuito	27
4.2	Esquemático da Montagem	27
5	CÓDIGO EM C	29
6	CONCLUSÃO	39
6.1	Conclusão	39
	REFERÊNCIAS	41

INTRODUÇÃO

Este trabalho descreve um projeto que foi desenvolvido na disciplina de Arquitetura de Computadores, feito sobre a linguagem C, onde utilizamos uma placa controladora Arduino UNO.

O projeto conta com o intuito para simular o funcionamento das funções de uma máquina de lavar automática. São retratados todos os componentes de uma máquina real, onde estão presentes três modos diferentes de operação.

No decorrer do projeto, são abordados os diferentes técnicas utilizadas para retratar o funcionamento das diferentes funções da máquina, como por exemplo:

1. Uso da Modulação por largura de pulso (MLP) para controlar a frequência/intensidade de rotação de um motor;
2. Uso de um potenciômetro para simular o nível de água;
3. Uso da memória interna do Arduino (memória EEPROM) para o armazenamento do histórico de operação da máquina;

Também são descritos detalhadamente todos os componentes que foram utilizados para a construção do projeto, o esquemático de montagem dos circuitos projetados, os códigos utilizados, além de todos os outros aspectos da execução do projeto.

DEFINIÇÃO DO PROBLEMA

Para realizar a simulação de uma máquina de lavar, é necessário primeiramente entender o funcionamento de uma máquina real. De forma geral, as máquinas de lavar atuais oferecem ao usuário o mesmo leque de opções em comum: o modo de lavagem e o nível de água desejados.

Após todas as opções iniciais serem devidamente selecionadas, a máquina inicia seu trabalho da forma como foi programada, seguindo as instruções de cada etapa, previamente descritas uma a uma em sua memória interna. As máquinas contam também com um pequeno painel com informações sobre seu estado atual, qual modo e nível foram selecionados, em qual etapa do trabalho estão, para dar uma estimativa do tempo restante até a finalização para o usuário.

O projeto então foi feito em etapas. Primeiramente foi decidido como seria o funcionamento da máquina, quais seriam seus modos de operação. Cada modo conta com instruções diferentes e precisas sobre como irão trabalhar todos os diferentes componentes da máquina. As instruções dos modos de operação são, por exemplo, quanto tempo o motor permanece ligado e desligado, quantas repetições são feitas no processo de lavagem, enxágue, centrifugação, quantas vezes a água é retirada e colocada, etc.

Logo após, foi realizada a construção do modelo inicial da máquina, onde existem três LEDs para sinalizar a escolha do modo de operação, três LEDs indicando o nível de água selecionado, dois motores simbolizando o motor principal e a bomba d'água, botões para alternar entre os modos e níveis de água e um botão para iniciar/interromper o processo de lavagem da máquina.

2.1 Construção do Problema

Inicialmente foi decidido que a máquina contaria com três modos de operação: modo rápido, para roupas leves e pouco sujas; modo normal, para roupas com uma necessidade maior

de limpeza; e o modo pesado, para peças como tapetes, roupas de cama extremamente sujas.

As opções de níveis de água disponíveis também são três: nível 1 (baixo, 35% do volume total do tanque); nível 2 (médio, 50% do volume total); e nível 3 (alto, 95% do nível total).

Os modos de operação e os níveis de água são selecionados através de seus respectivos botões, e existem três LEDs indicando o modo e três LEDs indicando o nível selecionados atualmente. Quando ligada, a máquina já vem com os modos 1 e nível 1 pré-selecionados.

Após essas duas instruções básicas serem dadas para a máquina, é possível então iniciar seu funcionamento através do botão de liga/desliga. A máquina então inicia seu processo de lavagem, enxágue e centrifugação das roupas, seguindo as instruções de cada modo que estão em sua memória. Para fins de observação, todas essas etapas do processo de lavagem são exibidas em mensagens pela porta serial do Arduino.

O nível de água atual da máquina normalmente é medido através da quantidade de tempo que a entrada de água é deixada aberta, ou através de um sensor. Neste projeto, o nível da água é simulado através de um potenciômetro, onde então é possível simular com fidelidade como uma máquina real irá reagir durante todo o processo de lavagem.

Assim que a máquina passa por todas as etapas de lavagem selecionadas e o modo de operação é finalizado, a máquina retorna ao seu estado inicial, onde fica no aguardo de instruções para iniciar um novo ciclo de lavagem.

Abaixo estão descritas todas as instruções de cada componente da máquina, em cada um dos três modos de operação:

2.2 Modo Rápido

1. Enche a máquina de água/sabão de acordo com o nível selecionado;
2. Deixa as roupas em molho por 5 minutos;
3. Bate as roupas por outros 5 minutos;
4. Retira a água suja da máquina;
5. Enche a máquina de água limpa de acordo com o nível selecionado;
6. Bate as roupas por 5 minutos para o enxágue;
7. Retira a água suja da máquina;
8. Centrifuga as roupas por 5 minutos;

2.3 Modo Normal

1. Enche a máquina de água/sabão de acordo com o nível selecionado;
2. Bate as roupas (70
3. Deixa as roupas em molho por 5 minutos;
4. Bate as roupas por outros 10 minutos;
5. Deixa as roupas em molho por 5 minutos;
6. Retira a água suja da máquina;
7. Enche a máquina de água limpa de acordo com o nível selecionado;
8. Bate as roupas por 5 minutos para o enxágue;
9. Retira a água suja da máquina;
10. Centrifuga as roupas por 5 minutos;

2.4 Modo Pesado

1. Enche a máquina de água/sabão de acordo com o nível selecionado;
2. Bate as roupas (100
3. Deixa as roupas em molho por 10 minutos;
4. Bate as roupas por outros 15 minutos;
5. Deixa as roupas em molho por 10 minutos;
6. Retira a água suja da máquina;
7. Enche a máquina de água limpa de acordo com o nível selecionado;
8. Bate as roupas por 5 minutos para enxágue;
9. Retira a água suja da máquina;
10. Enche a máquina de água limpa de acordo com o nível selecionado;
11. Bate as roupas por 5 minutos para enxágue;
12. Retira a água suja da máquina;
13. Centrifuga as roupas por 5 minutos;

DISPOSITIVOS UTILIZADOS

Para a construção do projeto, utilizamos um kit completo Arduino UNO e iniciamos a programação com os modos de operação da máquina. Toda a programação é feita em uma linguagem baseada em C/C++.

1. 1x Arduino Uno REV3 Original
2. 1x Protobord 840P MSB-100
3. 3x Chave Tactil 6X6X4,3mm (botões)
4. 6x LED 3MM Branco DIFUSO
5. 1x LED 3MM Vermelho DIFUSO
6. 11x Resistor 10K
7. 1x Transistor BC337-25 TO-92
8. 1x Potenciômetro linear 10k
9. Jumper MM (fios)
10. 1x Cabo USB-A M X USB-B M 1,1/2,0 1,8M PT

Além destes componentes do kit Arduino, utilizamos dois motores à parte para realizar a simulação do funcionamento do motor principal da máquina e do motor da bomba d'água, enquanto os LEDs e alguns botões ficaram responsáveis por simular o painel eletrônico da máquina, onde selecionamos modo de operação, nível da água, além de um botão liga/desliga.

3.1 Arduino Uno

O Arduino/Genuino Uno é uma placa de microcontrolador baseada no modelo ATmega328P. Conta com 14 pinos de entrada e saída digitais, dos quais 6 podem ser usados como saídas PWM, 6 entradas analógicas, um cristal de quartzo de 16 MHz, uma conexão USB, uma entrada de força, um header ICSP e um botão de reinicialização.

O Uno difere de todas as placas antecessoras no sentido de não utilizar o chip FTDI para conversão do sinal serial. Utiliza no seu lugar um Atmega8U2 programado como conversor de USB para serial. O ATmega328 tem 32KB (dos quais 0,5 são utilizados pelo bootloader). Também tem 2KB de SRAM e 1KB de EEPROM (que pode ser lido ou gravado com a biblioteca EEPROM).

Cada um dos 14 pinos digitais do Uno podem ser utilizados como uma entrada ou uma saída utilizando-se as funções `pinMode()`, `digitalWrite()`, e `digitalRead()`. Eles operam a 5V. Cada pino pode fornecer ou receber um máximo de 40mA e tem um resistor pull-up interno (desconectado por padrão) de 20-50k.

Ele contém todo o necessário para suportar um microcontrolador, bastando apenas conectá-lo a um computador via cabo USB ou alimentá-lo com um adaptador AC-DC ou bateria. A palavra "Uno" significa UM em italiano e foi escolhida para marcar o lançamento do Arduino Software (IDE).

([ARDUINO, 2016a](#))

3.2 Modulação por Largura de Banda

A tecnologia Pulse Width Modulation (PWM), no português Modulação por Largura de Pulso (MLP), permite que os microcontroladores atenuem as luzes, controlem a velocidade de operação de motores e gerem tensões analógicas. Isso é feito alterando o comprimento do pulso, permitindo assim que a saída seja controlada.

Um pulso ocorre em uma frequência regular, neste caso na frequência de modulação. Chamamos então de Ciclo de trabalho (Duty Cycle) a razão entre o tamanho do pulso pelo período de tempo de operação. Logo, quanto maior é o ciclo de trabalho, maior também será a saída. Portanto, quando alteramos essa saída que alimenta um determinado componente, alteramos a intensidade de operação do mesmo.

Neste projeto utilizamos as saídas PWM do Arduino para o controle da potência de rotação do motor principal da máquina de lavar. Os diferentes modos são direcionados para diferentes tipos de roupas, e algumas necessitam de cuidados especiais, como por exemplo a intensidade da lavagem.

O modo leve, por exemplo, trabalha a uma intensidade menor que o modo pesado. A

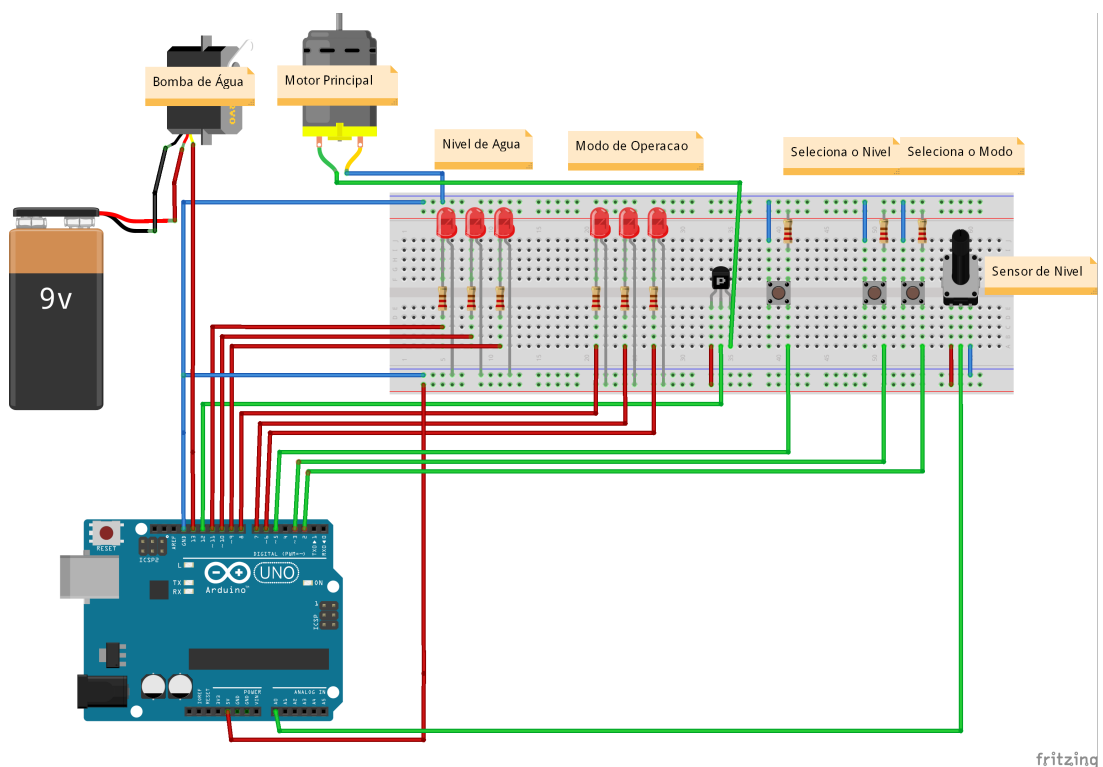
intensidade de giro do motor da máquina então é menor no modo leve (35% da potência total do motor), enquanto no modo pesado o motor trabalha com sua potência máxima.

([ARDUINO, 2016b](#))

ESQUEMÁTICO E MONTAGEM

4.1 Montagem do Circuito

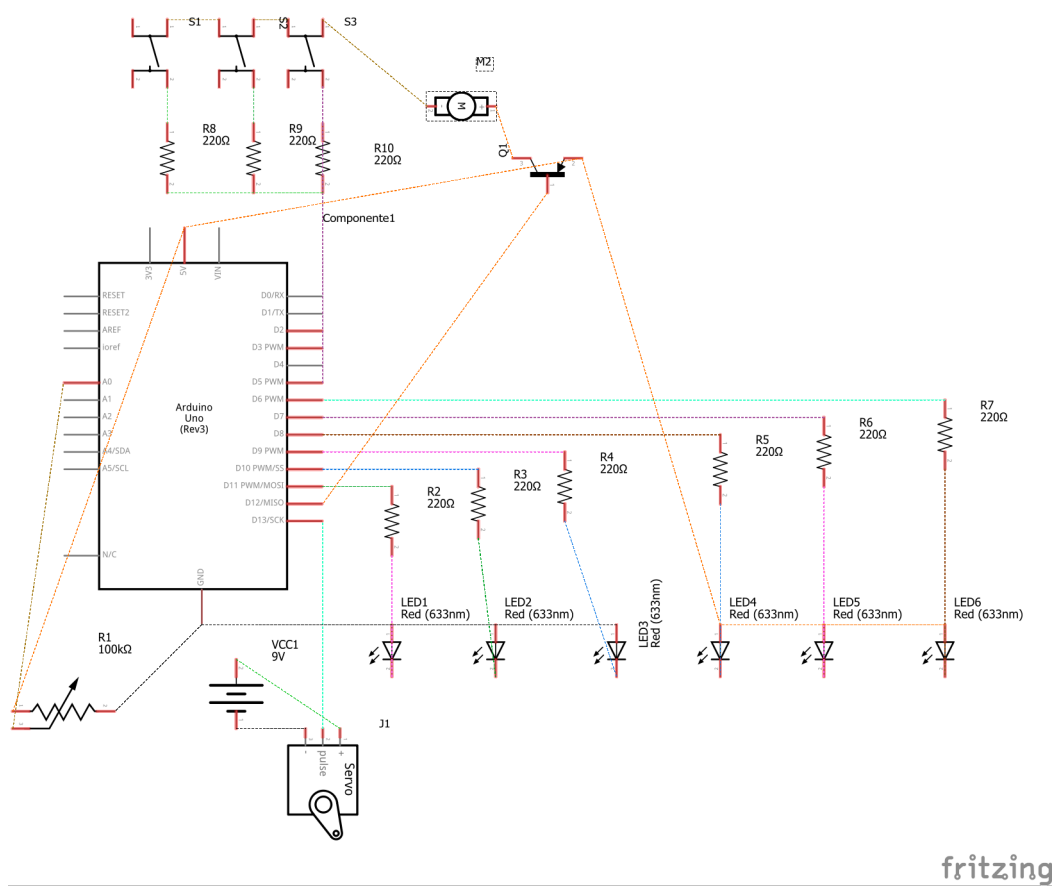
Figura 1 – Circuito na ProtoBoard



Fonte: Elaborada pelo autor.

4.2 Esquemático da Montagem

Figura 2 – Esquemático do Circuito



Fonte: Elaborada pelo autor.

CÓDIGO EM C

Código-fonte 1 – Máquina de Lavar

```
1: /*
2:  * Universidade Federal de Goiás
3:  * Disciplina: Arquitetura de Computadores
4:  * Projeto: Máquina de Lavar
5:  * Desenvolvido pelos alunos: Marco Túlio, Vitor do Vale, Pablo
   Vinicius
6: */
7: #include <EEPROM.h>
8: #define bomba 13 // Pino que ativa a bomba
9: #define motor 12 // Pino que ativa o motor
10:
11: #define led_nivel_1 11 // Pino que o led1 (Nivel 1)(vermelho)
12: #define led_nivel_2 10 //Pino que o led2 (Nivel 2)(vermelho)
13: #define led_nivel_3 9 //Pino que o led3 (Nivel 3)(vermelho)
14:
15: #define led_mod0_1 8 // Pino que o led1 (Nivel Leve)(branco)
16: #define led_mod0_2 7 //Pino que o led2 (Nivel Normal)(branco)
17: #define led_mod0_3 6 //Pino que o led3 (Nivel Pesado)(branco)
18:
19: #define button_start_system 5 // Pino que verifica se o sistema
   pode iniciar
20: #define button_seleciona_nivel 2 // Pino para entrada do nivel
   desejado
21: #define button_seleciona_mod0 3 // Pino para entrada do mod0
   desejado
22:
```

```
23: int incomingByte;          // a variable to read incoming serial
    data into
24: int buttonState = 0 ; // botao começa desligado
25: int addr = 0; // Posicao Inicial da Memoria
26: const int LM35 = A0; // Pino que verifica o nivel de agua
27: float temperatura; // Variável que armazenará a temperatura
    medida
28: int   saida;
29:
30: int nivel = 0;
31: int modo = 0;
32: typedef struct Estados{
33:     int modo_operacao;
34:     int nivel_agua;
35:     int dia;
36:     int mes;
37:     int ano;
38: }Estado;
39:
40: void lerEeprom(){
41:     int address = 0;
42:     byte value;
43:     while (address <= EEPROM.length()){
44:         value = EEPROM.read(address);
45:         Serial.print(address);
46:         Serial.print("\t");
47:         Serial.print(value, DEC);
48:         // avance ao próximo endereço da EEPROM
49:         address = address + 1;
50:         // há apenas 512 bytes na EEPROM, de 0 a 511, de forma que se
51:         // estivermos no endereço 512, meia-volta ao 0
52:         if (address == EEPROM.length())
53:             {
54:                 address = 0;
55:             }
56:     }
57: }
58: void ligar_bomba(int nivel_agua){
59:     switch (nivel_agua) {
60:
61:         case 0: // Tirar a agua
62:             analogWrite(bomba, 254);
```

```

63:             while(analogRead(LM35) > 1){ // Maior que 0
64:                 temperatura = analogRead(LM35);
65:                 Serial.print("Tirando agua: ");
66:                 Serial.println(temperatura);
67:             }
68:             analogWrite(bomba, 0);
69:             break;
70:
71:         case 1: // Colocar agua até o nivel1
72:             analogWrite(bomba, 254);
73:             while(analogRead(LM35) < 360){ // Menor que
35%
74:                 temperatura = analogRead(LM35);
75:                 Serial.print("Nivel1: ");
76:                 Serial.println(temperatura);
77:             }
78:             analogWrite(bomba, 0);
79:             break;
80:
81:         case 2: // Colocar agua até o nivel2
82:             analogWrite(bomba, 254);
83:             while(analogRead(LM35) < 614){ // Menor que 60%
84:                 temperatura = analogRead(LM35);
85:                 Serial.print("Nivel2: ");
86:                 Serial.println(temperatura);
87:             }
88:             analogWrite(bomba, 0);
89:             break;
90:         case 3: // Colocar agua até o nivel3
91:             Serial.println("Case 3");
92:             analogWrite(bomba, 254);
93:             while(analogRead(LM35) < 1023){
94:                 temperatura = analogRead(LM35);
95:                 Serial.print("Nivel3: ");
96:                 Serial.println(temperatura);
97:             }
98:             analogWrite(bomba, 0);
99:             break;
100:     }
101:     delay(1000);
102: }
103: void ligar_motor(int tempo){

```

```
104:     tempo = tempo * 1000;
105:     digitalWrite(motor, HIGH);
106:     Serial.println("Motor Ligado");
107:     delay(tempo);
108:     Serial.println("Motor Desligado");
109:     digitalWrite(motor, LOW);
110: }
111: void modoRapido(){
112:     // Ligar a bomba para colocar agua (Nivel);
113:     ligar_bomba(nivel); // Passando nivel 1 de agua
114:     ligar_motor(5); // Passando 5 minutos
115:     Serial.println("Molho por 5 segundos");
116:     delay(5000); // Molho
117:     ligar_motor(5); // Passando 5 minutos
118:     ligar_bomba(0); // Ligar a bomba para tirar agua; (Tirando
        agua suja)
119:     ligar_bomba(nivel); // Ligar a bomba para colcoar agua (Nivel);
        (Colocando agua limpa)
120:     ligar_motor(5); // Ligar o motor; (Enxaguando)
121:     ligar_bomba(0); // Ligar a bomba para tirar agua; (Tirando
        agua enxague)
122:     Serial.println("Centrifugando");
123:     ligar_motor(5); // Ligar o motor; (Centrifugando)
124:     Serial.println("Roupas Lavadas no modo Rapido");
125:     delay(5000);
126: }
127: boolean modoNormal(){
128:     ligar_bomba(nivel); // Passando nivel Selecionado de agua
129:     ligar_motor(10); // Ligar motor por 10 minutos
130:     delay(5000); // Molho por 5 min
131:     Serial.println("Molho por 5 segundos");
132:     ligar_motor(10); // Ligar motor por 10 minutos
133:     delay(5000); // Molho por 5 min
134:     ligar_bomba(0); // Ligar a bomba para tirar agua; (Tirando
        agua suja)
135:     ligar_bomba(nivel); // Ligar a bomba para colcoar agua (Nivel
        ); (Colocando agua limpa)
136:     ligar_motor(5); // Ligar motor por 5 minutos
137:     ligar_bomba(0); // Ligar a bomba para tirar agua; (Tirando
        agua suja)
138:     Serial.println("Centrifugando por 10 segundos");
139:     ligar_motor(10); // Ligar o motor; (Centrifugando)
```

```
140:     Serial.println("Roupas Lavadas no modo Normal");
141: }
142: boolean modoPesado(){
143:     ligar_bomba(nivel); // Passando nivel 1 de agua
144:     ligar_motor(15); // Ligar motor por 15 minutos
145:     Serial.println("Molho por 10 segundos");
146:     delay(10000); // Molho
147:     ligar_motor(15); // Ligar motor por 15 minutos
148:     Serial.println("Molho por 10 segundos");
149:     delay(10000); // Molho
150:     ligar_bomba(0); // Ligar a bomba para tirar agua; (Tirando
agua suja)
151:     ligar_bomba(nivel); // Ligar a bomba para colcoar agua(Nivel
); (Colocando agua limpa)
152:     ligar_motor(5); // Ligar motor por 5 minutos
153:     ligar_bomba(0); // Ligar a bomba para tirar agua; (Tirando
agua suja)
154:     ligar_bomba(nivel); // Ligar a bomba para colcoar agua(Nivel
); (Colocando agua limpa)
155:     ligar_motor(5); // Ligar motor por 5 minutos
156:     ligar_bomba(0); // Ligar a bomba para tirar agua; (Tirando
agua suja)
157:     Serial.println("Centrifugando por 15 segundos");
158:     ligar_motor(15); // Ligar o motor; (Centrifugando)
159:     Serial.println("Roupas Lavadas no modo Pesado");
160: }
161: void ligar_Led_Nivel(){
162:     if(nivel == 0){
163:         Serial.println("Nivel 0");
164:         digitalWrite(led_nivel_1, LOW);
165:         digitalWrite(led_nivel_2, LOW);
166:         digitalWrite(led_nivel_3, LOW);
167:     }
168:     if(nivel == 1){
169:         Serial.println("Nivel 1");
170:         digitalWrite(led_nivel_1, HIGH);
171:         digitalWrite(led_nivel_2, LOW);
172:         digitalWrite(led_nivel_3, LOW);
173:     }
174:     if(nivel == 2){
175:         Serial.println("Nivel 2");
176:         digitalWrite(led_nivel_1, HIGH);
```

```
177:     digitalWrite(led_nivel_2, HIGH);
178:     digitalWrite(led_nivel_3, LOW);
179: }
180: if(nivel == 3){
181:     Serial.println("Nivel 3");
182:     digitalWrite(led_nivel_1, HIGH);
183:     digitalWrite(led_nivel_2, HIGH);
184:     digitalWrite(led_nivel_3, HIGH);
185: }
186:
187: }
188:
189: void ligar_Led_Modo(){
190:     if (modo == 0){
191:         Serial.println("Modo Desligado");
192:         digitalWrite(led_modo_1, LOW);
193:         digitalWrite(led_modo_2, LOW);
194:         digitalWrite(led_modo_3, LOW);
195:     }
196:     if(modo == 1){
197:         Serial.println("Modo Rapido");
198:         digitalWrite(led_modo_1, HIGH);
199:         digitalWrite(led_modo_2, LOW);
200:         digitalWrite(led_modo_3, LOW);
201:     }
202:     if(modo == 2){
203:         Serial.println("Modo Normal");
204:         digitalWrite(led_modo_1, HIGH);
205:         digitalWrite(led_modo_2, HIGH);
206:         digitalWrite(led_modo_3, LOW);
207:     }
208:     if(modo == 3){
209:         Serial.println("Modo Pesado");
210:         digitalWrite(led_modo_1, HIGH);
211:         digitalWrite(led_modo_2, HIGH);
212:         digitalWrite(led_modo_3, HIGH);
213:     }
214:
215: }
216: void selecionarNivel(){
217:     if (buttonState != HIGH){
218:         nivel++;
```

```
219:   ligar_Led_Nivel();
220:
221:   if(nivel > 3){
222:       nivel = 0;
223:   }
224: }else{
225:     Serial.println("Não é possivel selecionar nivel apos
        sistema iniciado");
226: }
227: }
228: void selecionarModo(){
229:   if (buttonState != HIGH){
230:       modo++;
231:       ligar_Led_Modo();
232:       if (modo > 3){
233:           modo = 0;
234:       }
235:   }else{
236:       Serial.println("Não é possivel selecionar modo apos
        sistema iniciado");
237:   }
238: }
239: void salvarEstado(){
240:     EEPROM.write(addr, modo);
241:     EEPROM.write(addr++, nivel);
242:     EEPROM.write(addr++, 28);
243:     EEPROM.write(addr++, 03);
244:     EEPROM.write(addr++, 17);
245:     if (addr == 512) {
246:         addr = 0;
247:     }
248: }
249: void historico(){
250:     int estado;
251:     int address = 0 ;
252:
253:     Serial.println("\t\t----- HISTORICO
        -----");
254:
255:     while(address < addr){
256:         estado = EEPROM.read(address);
257:         Serial.println(" -----");
```

```
258:         if(estado == 1){ Serial.println("| Modo Rapido          |"
    );}
259:         if(estado == 2){ Serial.println("| Modo Normal          |"
    );}
260:         if(estado == 3){ Serial.println("| Modo Pesado          |"
    );}
261:         Serial.print("| Nivel de agua: "); Serial.print(EEPROM
    .read(address++));Serial.println(" |");
262:         Serial.print("| Data: "); Serial.print(EEPROM.read(
    address++));
263:         Serial.print("/"); Serial.print(EEPROM.read(address++))
    ;
264:         Serial.print("/"); Serial.print(EEPROM.read(address++))
    ; Serial.println("      |");
265:         Serial.println(" -----");
266:     }
267: }
268: void setup() {
269:     Serial.begin(9600); // inicializa a comunicação serial
270:
271:     // Inicialia os pinos de saída
272:     pinMode(bomba, OUTPUT);
273:     pinMode(motor, OUTPUT);
274:
275:     pinMode(led_nivel_1, OUTPUT);
276:     pinMode(led_nivel_2, OUTPUT);
277:     pinMode(led_nivel_3, OUTPUT);
278:
279:     pinMode(led_modos_1, OUTPUT);
280:     pinMode(led_modos_2, OUTPUT);
281:     pinMode(led_modos_3, OUTPUT);
282:
283:     //Inicializa os pinos de entrada
284:     pinMode(button_start_system, INPUT);
285:     pinMode(button_seleciona_nivel, INPUT);
286:     pinMode(button_seleciona_modos, INPUT);
287:
288:     attachInterrupt(digitalPinToInterrupt(2), selecionarNivel,
        RISING); //Selecionar Nivel
289:     attachInterrupt(digitalPinToInterrupt(3), selecionarModos,
        RISING); // Selecionar Modos
290: }
```

```
291:
292: void loop() {
293:   buttonState = digitalRead(button_start_system);
294:
295:   if (Serial.available() > 0) {
296:
297:     incomingByte = Serial.read();
298:     if (incomingByte == 'H') {
299:       historico();
300:     }
301:
302:   }
303:
304:   if (buttonState == HIGH){
305:     Serial.println("System ON");
306:     switch(modos){
307:
308:       case 1: Serial.println("Modo Rapido Seleccionado");
309:               modoRapido();
310:               salvarEstado();
311:               break;
312:       case 2: Serial.println("Modo Normal Seleccionado");
313:               modoNormal();
314:               salvarEstado();
315:               break;
316:       case 3: Serial.println("Modo Pesado Seleccionado");
317:               modoPesado(); // Salvar Estado
318:               salvarEstado();
319:               break;
320:       default: Serial.println("Nenhum Modo Seleccionado");
321:               break;
322:     }
323:
324:     buttonState = LOW;
325:
326:   }
327:   delay(50);
328: }
```

CONCLUSÃO

6.1 Conclusão

Com o projeto de construção da máquina de lavar completo e devidamente testado, foi possível notar que o uso de microcontroladores vai muito além do uso didático, estando presente nas mais diversas tarefas do cotidiano, mesmo que não sejam notados.

O uso da placa controladora Arduino teve um papel crucial na execução e conclusão do projeto, devido à sua grande versatilidade na comunicação com os mais diversos componentes e sua facilidade de uso.

Esses pequenos componentes estão presentes nos mais variados dispositivos eletrônicos, mostrando-se capazes de realizar uma enorme gama de tarefas, bastando apenas um esforço e uma certa noção de programação para transportar um problema real para um código numa determinada linguagem de programação.

Considerando o que foi proposto inicialmente, que era a construção de uma simulação do funcionamento de uma máquina de lavar automática, podemos então constatar que o projeto funcionou conforme os objetivos propostos inicialmente, após uma série de testes e ajustes realizados ao longo da execução do trabalho.

REFERÊNCIAS

ARDUINO. **Arduino/Genuino UNO**. 2016. Disponível em: <<https://www.arduino.cc/en/main/arduinoBoardUno>>. Acesso em: 27/02/2017. Citado na página 24.

ARDUINO. **PWM**. 2016. Disponível em: <<https://www.arduino.cc/en/Tutorial/PWM>>. Acesso em: 27/02/2017. Citado na página 25.