



**UNIVERSIDADE FEDERAL DO RIO DE JANEIRO**  
Instituto de Física  
Programa de Pós-Graduação em Ensino de Física  
Mestrado Nacional Profissional em Ensino de Física



Circuito e programa do módulo Arduino de leitura da Célula de Ar

Hercilio Pereira Cordova  
&  
Helio Salim de Amorim ; Carlos Augusto Domingues Zarro

Material instrucional associado à dissertação de mestrado de Hercílio Pereira Cordova, apresentada ao Programa de Pós-Graduação em Ensino de Física da Universidade Federal do Rio de Janeiro.

Rio de Janeiro  
Setembro de 2016

# Circuito e programa do módulo Arduino de leitura da Célula de Ar

Hercilio Pereira Cordova

&

Helio Salim de Amorim; Carlos Augusto Domingues Zarro

## 1 Introdução

O circuito e programa do módulo Arduino permitem realizar leitura da pressão interna da Célula de Ar ao mesmo tempo em que as franjas são deslocadas. Assim um sensor de pressão colocado dentro da Célula de Ar transmite via cabo as informações de pressão dentro da Célula, ao passo que um *resistor sensível à luz* (LDR) transmite via cabo ao Arduino a intensidade da franja, o software faz a leitura do deslocamento da franja e a pressão naquele instante, salvando na memória um par de dados. O primeiro é o número da franja deslocada (1, 2, 3...50) e a pressão em Pascal no momento em que a franja mudou de claro para escuro. A memória interna do Arduino comporta até 50 pares de valores, embora no experimento sejam movidas no máximo 20 franjas.

O uso deste módulo torna o experimento de medida do índice de refração do ar em função da pressão muito prático e rápido, além de permitir uma conexão com um computador de forma que os dados podem ser carregados diretamente em uma planilha de Excel.

O diagrama também conta com um display de LCD, que permite visualizar os dados de pressão e franja durante o experimento ou no término das medidas, bastando acessar a memória. A vantagem do uso do display é que elimina o uso do computador em sala de aula, reduzindo o custo e tornando o experimento mais prático. Já as tabelas e gráficos (ver figura 1) podem ser feitos no quadro pelos alunos, a fim de exercitar o manuseio de dados experimentais.

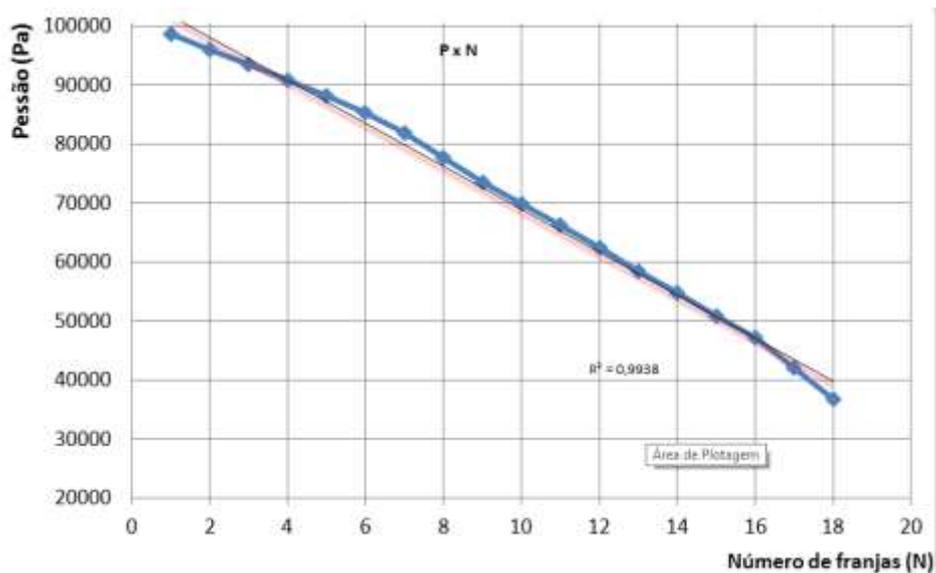


Figura 1. Exemplo de gráfico gerado pelo experimento.

## 2 Arranjo Experimental

O circuito para o uso do Arduino é apresentado na figura 2, onde o esquema representa a posição dos componentes em uma matriz de contato (protoboard). O *buzzer* tem a finalidade de emitir um som a cada vez que um par de dados é coletado, no início e fim da coleta de dados. O display de LCD permite ler os dados e instruções durante o experimento.

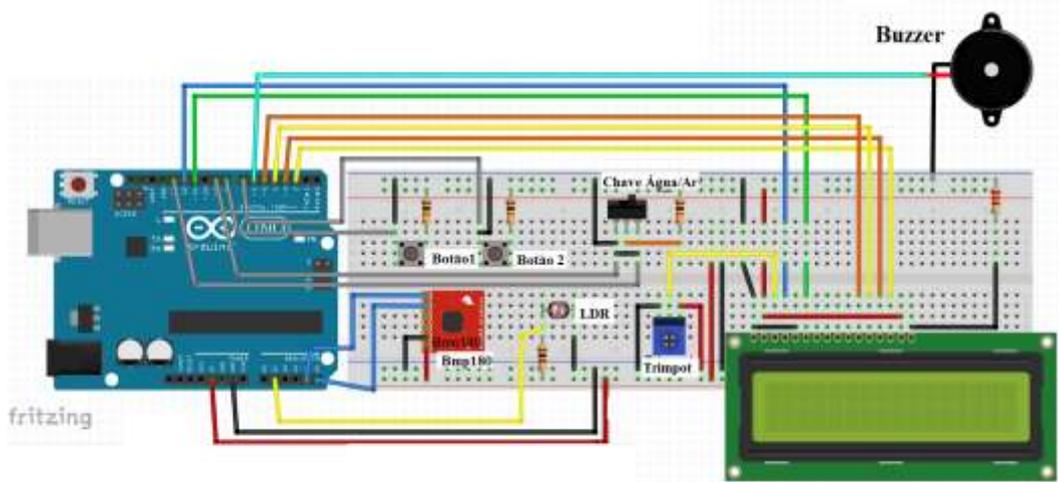


Figura 2. Circuito do módulo Arduino.

A figura 3 apresenta o circuito em uma matriz de contato para testes, sendo que o ideal é acondicionar o Arduino e demais circuitos em uma caixa (ver figura 4). Contudo o circuito exposto permite aos alunos interagir com a eletrônica, pois ficam cientes do papel de cada componente, e não só verem o módulo como uma caixa preta.

Caso seja optado pela matriz de contato é recomendado adquirir um conjunto de

fios com conectores nas pontas (jumper), estes fios tornam a ligação entre os componentes muito mais rápida e precisa, além de manter uma organização, já que costumam vir em pacotes com 60 unidades em diversas cores.

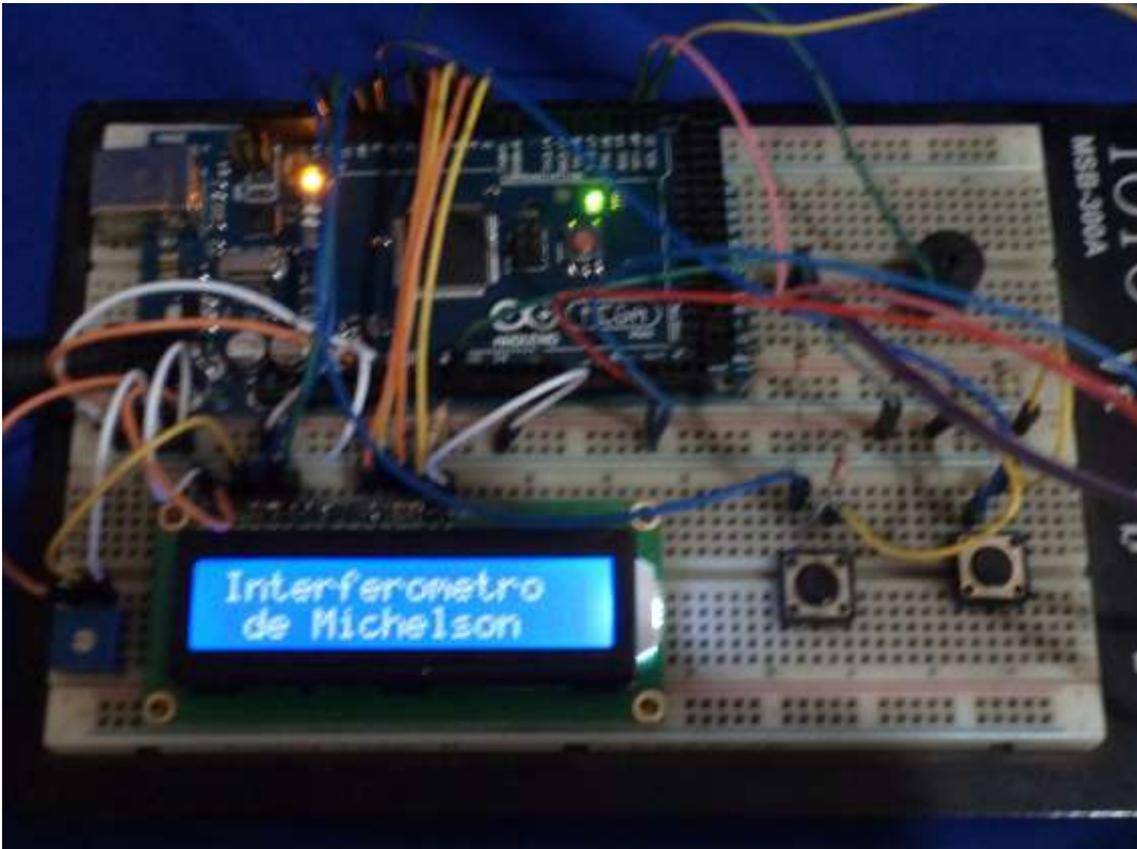


Figura 3. Módulo Arduino em uma matriz de contato (protoboard).



Figura 4, Modulo Arduino acondicionado em uma caixa.

Os componentes eletrônicos usados são de fácil obtenção, tanto em lojas de eletrônicas como em site de venda de componentes eletrônicos pela internet. Os componentes necessários são:

- 01 Arduino Uno;
- 01 sensor de pressão e temperatura Bmp180;
- 01 buzzer;
- 01 display de LCD 16x2;
- 01 trimpot de 10K $\Omega$ ;
- 02 chave push boton;
- 01 chave de 3 posição (somente se for medir o índice de refração da água);
- 01 termistor NTC de 10K $\Omega$  (somente se for medir o índice de refração da água);
- 04 resistores de 10K $\Omega$  1/8W;
- 01 resistores de 2,2K $\Omega$  1/8W;
- 01 LDR de 5 mm;
- 07 m de fio paralelo fino, para conectar o LDR ao Arduino;
- Fios diversos.

Estes componentes permitem realizar o experimento do índice de refração do ar em função da pressão e também o índice de refração da água em função da temperatura, contudo o ultimo não foi abordado, embora o código tenha essa função funcionando, ela é selecionada mediante seleção na chave de três posições.

Um ponto importante no código é a função de auto calibração. Devido à aplicação do interferômetro em locais com iluminação diferente e até mesmo a variação da intensidade do LASER usado, o LDR pode apresentar valores diferentes para considerar uma mudança de franja. Assim a rotina de auto calibração analisa a menor e maior intensidade a qual o LDR está submetido e estabelece um ponto para considerar que a franja mudou de claro para escuro ou escuro para claro. Para essa calibração é preciso que o LDR esteja no centro da figura de interferência e as franjas sejam movidas. O inicio e fim da calibração do LDR é orientado com uso de mensagens no display de LDC e avisos sonoros.

O código fonte ou sketch do programa usado pelo módulo Arduino é apresentado no final deste roteiro, sendo que o programa usado para programar o Arduino é fornecido gratuitamente pela própria fabricante do Arduino no site [www.arduino.cc](http://www.arduino.cc).

O código está com comentários em cada linha de comando a fim de orientar o professor sobre os passos do programa.

### **Código fonte do Arduino (Sketch)**

Para programação do módulo Arduino Uno foi usado a versão 1.6.7 do software Arduino.

Foi usada uma variação do C++.

```
/*Para o bmp180  
* Fio vermelho = Vcc5 V  
* Fio preto = GND
```

```

* Fio verde = SDA, porta analogica A4 para arduino UNO
* Fio branca = SCL, porta analogica A5 para arduino UNO*/
/*Para o LDR
* conectado na porta analogica A1*/
/* Para o LM35
* Conectado na porta analogica A0*/
#include <Wire.h> // referente ao sensor bmp180
#include <Adafruit_BMP085.h> // referente ao sensor bmp180
Adafruit_BMP085 bmp;
#include <avr/eeprom.h> // referente a gravação na EEPROM
#include <EEPROM.h> // referente a gravação na EEPROM
#include <Streaming.h> // referente a gravação na EEPROM
#include <LiquidCrystal.h> //Inclui a biblioteca do LCD
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); //Configura os pinos do Arduino para se comunicar
com o LCD
// variaveis e pins usados___Inicio
int LABEL = 1;
int franja_N = 0;
int valor = 0;
int botao = 8; //botão para iniciar a medida
int botao2 = 7; // para calibrar o sensor LDR
int chaveAr = 13; // Ir para medir refração do ar
int chaveAgua = 9 ; // ir para medir refração da agua
int som = 6; // aciona o bip
int ldr = 1; // calibração do ldr
int claro = 0; //variavel de controle
int escuro = 1; //variavel de controle
int memo = 0; // valor de posição de memoria
int N = 0; //variavel de controle
int N2 = 0; //variavel de controle
int zero = 0; //variavel de controle
int calibrado; // registra se o ldr foi calibrado
float pressao = 0; //variavel de pressão
float temperatura; // variavel de temperatura
float memoX; //variavel dmemoria da pressão
unsigned int maximo = 0; // ddp maxima no ldr
unsigned int minimo = 1020; // ddp minimo no ldr
unsigned int intensidade = 0; //variavel de controle
const int LM35 = A0; // Pino Analógico onde vai ser ligado ao pino 2 do LM35
const float base_Celcius = 0.4887585532746823069403714565; //Base de conversão
para Graus Celsius ((5/1023) * 100)
//const int sensor=0;
// variaveis e pins usados___Fim
void setup()
{
Serial.begin(9600); // ativa a comunicação com monitor de dados
Serial.println("CLEARDATA");// ativa comunicação com o PLX.DAQ
Serial.println("LABEL,Tempo,Pressao,Temperatura, Franja,");// Joga na planilha de
excel
pinMode (botao, INPUT); // botão 1

```

```

pinMode (botao2, INPUT); // botão 2
pinMode (chaveAr, INPUT); // chave de alavanca para opção Ar
pinMode (chaveAgua, INPUT); // chave de alavanca para opção agua
pinMode (som, OUTPUT); // buzzer
lcd.begin(16, 2); //Inicia o LCD com dimensões 16x2(Colunas x Linhas)
bmp.begin(); // ativa o sensor bmp180
pinMode(A0, INPUT);
calibrado = 0; // verifica se o ldr já foi calibrado
}
void loop()
{
if (calibrado == 0) // verifica se o ldr foi calibrado
{
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Calibre o LDR");
lcd.setCursor(0, 1);
lcd.print("Aperte o botao 2");
for ( int x = 0; x < 1000; x++)
delay(1);
{
if (digitalRead(botao2) == 1)
{
digitalWrite(som, HIGH);
delay(100);
digitalWrite(som, LOW);
calibrado = 1;
calibrando();
}}}
if (calibrado == 1)
{
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Interferometro");
lcd.setCursor(0, 1);
lcd.print(" de Michelson");
delay(1000);
for (int w = 0; w < 10000; w++)
{
if (digitalRead(chaveAr) == 1)
{
Ar();
}
}
if (digitalRead(chaveAgua) == 1)
{
Agua();
}
}
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Mova a chave");

```

```

lcd.setCursor(0, 1);
lcd.print("A P/agua B P/Ar");
delay(1000);
for (int f = 0; f < 10000; f++)
{
delay(1);
if (digitalRead(chaveAr) == 1)
{
Ar();
}
if (digitalRead(chaveAgua) == 1)
{
Agua();
} } }
void Ar() // rotina para leitura de franja e gravacao na EEPROM da pressao
{
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Refracao do Ar");
lcd.setCursor(0, 1);
lcd.print("com a pressao");
delay(3000);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Inicie o vacuo");
lcd.setCursor(0, 1);
lcd.print("ao som do Pib");
delay(3000);
tone(6, 1000, 2000);
for (unsigned long x = 0; x < 90000; x++)
{
if (analogRead(ldr) > maximo && claro == 1 && franja_N < 50)
{
pressao = (bmp.readPressure());
franja_N++;
N2 = franja_N;
claro = 0;
escuro = 1;
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("P =");
lcd.setCursor(4, 0);
lcd.print(pressao);
lcd.setCursor(11, 0);
lcd.print("Pa");
lcd.setCursor(0, 1);
lcd.print("Franja numero ");
lcd.print(franja_N);
memo = memo + 5;
eeprom_write_float( (float *) memo, pressao ); //escreve na memoria o valor da pressao

```

```

Serial.print(franja_N);
Serial.print(" ");
Serial.println(pressao);
//joga os dados para a planilha d Excel usando o PLX_DAQ
Serial.print("DATA,TIME,");
Serial.print(pressao);
Serial.print(",");
Serial.print(bmp.readTemperature());
Serial.print(",");
Serial.println(franja_N);
Serial.println("SET,2");
x = 0;
}
if (digitalRead(botao) == 1 || franja_N >= 50)
{
franja_N = 51;
zero = 0;
delay(500);
}
if (analogRead(ldr) < minimo && escuro == 1 && franja_N < 50)
{
escuro = 0;
claro = 1;
zero = 0;
digitalWrite(som, HIGH);
delay(15);
digitalWrite(som, LOW);
delay(15);
}
if (franja_N == 51) //zera a variavel memo para poder reproduzir no display
{
if (zero == 0)
{ memo = 0;
memoX = 0;
zero = 1;
}
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Teste finalizado");
delay(1000);
for ( int x = 0; x < 4; x++)
{
digitalWrite(som, HIGH);
delay(15);
digitalWrite(som, LOW);
delay (15);
}
delay(00);

for (int x = 0; x < 3000; x++)

```

```

{
delay(1);
if (digitalRead(botao) == 1)
{
leitura();
}}
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Aperte o botao 1");
lcd.setCursor(0, 1);
lcd.print("p/ ver os dados");
for (int x = 0; x < 3000; x++)
{
delay(1);
if (digitalRead(botao) == 1)
{
leitura();
}}} }}
void leitura()
{
for (unsigned long x = 0; x < 900000; x++)
{
if (digitalRead(botao) == 1) // apresenta os valores de pressao salvo na memoria
EEPROM no display
{
digitalWrite(som, HIGH);
delay(50);
digitalWrite(som, LOW);
memo = memo + 5;
N = N + 1;
memoX = eeprom_read_float( (float *) (memo) );
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("P =");
lcd.setCursor(4, 0);
lcd.print(memoX);
lcd.setCursor(10, 0);
lcd.print(" Pa");
lcd.setCursor(0, 1);
lcd.print("Franja numero ");
lcd.print(N);
x = 0;
if ( N > N2)
{
lcd.clear();
lcd.setCursor(0, 0);
lcd.print(" Fim dos dados");
N = 0;
memo = 0;
memoX = 0;
}
}
}
}

```

```

}
delay(500);
} }}
void Agua()
{
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Refracao da Agua");
  lcd.setCursor(0, 1);
  lcd.print("c/ a temperatura");
  delay(3000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Ligue o Resistor");
  lcd.setCursor(0, 1);
  lcd.print("ao som do Pib");
  delay(3000);
  tone(6, 1000, 2000);
  for (unsigned long x = 0; x < 90000; x++)
  {
    if (analogRead(ldr) > maximo && claro == 1 && franja_N < 50)
    {
      temperatura = 0;
      for (int i = 0; i < 1000; i++)
      {
        temperatura = temperatura + (analogRead(LM35) * base_Celcius );
      }
      temperatura = temperatura / 1000; //Media de mil leituras
      franja_N++;
      N2 = franja_N;
      claro = 0;
      escuro = 1;
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("T =");
      lcd.setCursor(4, 0);
      lcd.print(temperatura);
      lcd.setCursor(9, 0);
      lcd.print("*C");
      lcd.setCursor(0, 1);
      lcd.print("Franja numero ");
      lcd.print(franja_N);
      memo = memo + 5;
      eeprom_write_float( (float *) memo, temperatura ); //escreve na memoria o valor da
      pressao
      Serial.print(franja_N);
      Serial.print(" ");
      Serial.println(temperatura);
      //joga os dados para a planilha d Excel usando o PLX_DAQ
      Serial.print("DATA,TIME,");

```

```

Serial.print("-----");
Serial.print(",");
Serial.print(temperatura);
Serial.print(",");
Serial.println(franja_N);
Serial.println("SET,2");
x = 0;
}
if (digitalRead(botao) == 1 || franja_N >= 50)
{
franja_N = 51;
zero = 0;
delay(500);
}
if (analogRead(ldr) < minimo && escuro == 1 && franja_N < 50)
{
escuro = 0;
claro = 1;
zero = 0;
digitalWrite(som, HIGH);
delay(15);
digitalWrite(som, LOW);
delay(15);
}
if (franja_N == 51) //zera a variavel memo para poder reproduzir no display
{
if (zero == 0)
{ memo = 0;
memoX = 0;
zero = 1;
}
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Teste finalizado");
delay(1000);
for ( int x = 0; x < 4; x++)
{
digitalWrite(som, HIGH);
delay(15);
digitalWrite(som, LOW);
delay (15);
}
delay(00);
for (int x = 0; x < 3000; x++)
{
delay(1);
if (digitalRead(botao) == 1)
{
leituraAgua();
}}
}

```

```

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Aperte o botao 1");
lcd.setCursor(0, 1);
lcd.print("p/ ver os dados");
for (int x = 0; x < 3000; x++)
{
delay(1);
if (digitalRead(botao) == 1)
{
leituraAgua();
}
} } }
void leituraAgua()
{
for (unsigned long x = 0; x < 900000; x++)
{
if (digitalRead(botao) == 1) // apresenta os valores de pressão salvo na memoria
EEPROM no display
{
digitalWrite(som, HIGH);
delay(50);
digitalWrite(som, LOW);
memo = memo + 5;
N = N + 1;
memoX = eeprom_read_float( (float *) (memo) );
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("T =");
lcd.setCursor(4, 0);
lcd.print(memoX);
lcd.setCursor(9, 0);
lcd.print("*C");
lcd.setCursor(0, 1);
lcd.print("Franja numero ");
lcd.print(N);
x = 0;
if ( N > N2)
{
lcd.clear();
lcd.setCursor(0, 0);
lcd.print(" Fim dos dados");
N = 0;
memo = 0;
memoX = 0;
}
} } }
delay(500);
} } }
void calibrando() // rotina para calibra a intensidade maxima e minima no LDR
{

```

```

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Calibrando o LDR");
delay(2000);
lcd.setCursor(0, 1);
lcd.print("Movas as franjas");
digitalWrite(som, HIGH);
delay(100);
digitalWrite(som, LOW);
delay(100);
digitalWrite(som, HIGH);
delay(100);
digitalWrite(som, LOW);
for (unsigned int x = 0 ; x < 5000; x++)
{
intensidade = analogRead(ldr);
if (intensidade > maximo)
{
maximo = intensidade;
}
if (intensidade < minimo)
{
minimo = intensidade;
}
delay(1);
}
maximo = maximo - 20;
minimo = minimo + 20;
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Sensor Calibrado");
digitalWrite(som, HIGH);
delay(100);
digitalWrite(som, LOW);
delay(100);
digitalWrite(som, HIGH);
delay(100);
digitalWrite(som, LOW);
delay(100);
digitalWrite(som, HIGH);
delay(100);
digitalWrite(som, LOW);
lcd.setCursor(0, 1);
lcd.print("T=");
lcd.setCursor(2, 1);
lcd.print(maximo * 0.0049);
lcd.setCursor(6, 1);
lcd.print("V");
lcd.setCursor(9, 1);
lcd.print("t=");

```

```
lcd.setCursor(11, 1);  
lcd.print(minimo * 0.0049);  
lcd.setCursor(15, 1);  
lcd.print("v");  
delay(2000);  
loop();  
}
```

#### **4 Conclusão**

Este complemento do interferômetro de Michelson-Morley é um grande passo para experimento em sala de aula, não só pela parte didática voltada à física, mas em como a física precisa e usa de toda tecnologia e meios para poder realizar medidas mais precisas e confiáveis.

Por se tratar de um módulo muito usado, o Arduino dispõe de um grande acervo na internet de site, blogs e vídeos de como usar e programa-lo, tornando o Arduino uma boa escolha, pelo seu baixo preço e grande capacidade de implementação.