

Material Suplementar para “Estudo experimental do efeito *Faraday* usando *Arduíno* e determinação da constante de *Verdet* em líquidos”

Apêndice B

Sketch ou código Fonte em C++ escrito Arduíno IDE, para a automação do dispositivo Faraduíno.

```
1 // Incluindo as Bibliotecas necessarias
2 #include <OneWire.h>
3 #include <DallasTemperature.h>
4 // Porta D2 do pino conectada ao sensor DS18B20
5 #define ONE_WIRE_BUS 2
6 // Define uma instancia da biblioteca oneWire para comunica o com o sensor
7 OneWire oneWire (ONE_WIRE_BUS);
8 DallasTemperature sensors (&oneWire);
9 // endere o do sensor DS18B20s cc:db:a7:3d:ec:18
10 uint8_t sensor1[8] = { 0x28, 0xFF, 0x1C, 0x7B, 0xA6, 0x16, 0x05, 0x77 };
11 // Sensibilidade do fotosensor OPT101 para a luz laser vermelha S = 470 m V W
12 float Sensibilidade = 0.470;
13 // Sensibilidade o sensor de corrente por efeito Hall, S = 66 m V A
14 float SensibilidadeC = 0.66;
15 float FC = 0.12763148; // fator de calibra o.
16
17 void setup () {
18 /* inicializa a serial */
19 Serial.begin (115200);
20 sensors.begin ();
21 }
22
23 void loop () {
24 float Idc= 0.0;
25 float Plight = 0.0;
26 // Chamando as fun es para fazer as medidas de temperatura, corrente e
27 // potencia radiante.
28 // L a informa o do sensor de temperatura modelo DS18B20.
29 sensors.requestTemperatures();
30 //Aqui calculamos a corrente real, fazendo uma media de 500 amostras, aplicando
31 // a equa o 4.2.
32 Idc = (calculoCorriente (500) -1.089100) / FC);
33 //aqui calculamos a potencia radiante que atinge o fotosensor OPT101.
34 Plight = calculolight (500);
35 // Mostrando (enviando) os resultados das medi es para o serem mostrados no
36 // monitor serial.
37 // Temperatura no liquido
38 printTemperature (sensor1);
39 Serial.print (",");
40 Serial.print (Idc, 3);
41 Serial.print (",");
42 Serial.print (Plight, 4);
43 Serial.println ( );
44 delay (80); // retardo do funcionamento do Ardu no em 80 ms.
45 }
46 // Fun es para detec o da corrente, potencia radiante e temperatura.
```

```

44 float calculoCorriente(int numeroMuestras)
45 {
46     float leerAcs712 = 0;
47     float intensidad = 0;
48     for(int i=0;i<numeroMuestras;i++)
49     {
50         leerAcs712 = analogRead(A6) * (5.30 / 1023.0); // Lee a tens o no sensor na
                    porta A6.
51         intensidad =intensidade+(leerAcs712-2.5) / SensibilidadC; //Calculo da
                    corrente.
52     }
53     intensidad=intensidade/numeroMuestras; // m dia de acordo com o n mero de
                    amostras.
54     return(intensidade);
55 }
56
57 // Fun o que mede a pot ncia no fotosensorOPT101, conectado na porta
                    anal gica A7.
58 float calculolight(int numeroMuestras)
59 {
60     float leerlight = 0;
61     float Powerlight = 0.0;
62     for(int i=0; i < numeroMuestras; i++)
63     {
64         // Lee a tens o do sensor OPT101 e a transforma em volts.
65         leerlight = analogRead (A7) * (5.0 / 1023.0);
66         Powerlight = Powerlight + (leerlight/Sensibilidad); // c lculo da pot ncia
                    radiante em mW.
67     }
68     Powerlight = Powerlight / numeroMuestras;
69     return( Powerlight); // retornamos a vari vel de sa da da pot ncia radiante
                    em mW.
70 }
71 // Fun o que mede a temperatura no sensor DS18B20, conectado na porta digital
                    D2.
72 void printTemperature(DeviceAddress deviceAddress)
73 {
74     float tempC = sensors.getTempC (deviceAddress);
75     Serial.print (tempC);
76     // Serial.print(" C ");
77 }

```