

Automatización, de la identificación de sustancias mediante el cálculo del coeficiente de absorción, usando Arduino

German, Jorge.¹ Acosta, Noemi.²

¹ Cuarto. Facultad de Física. Universidad de la Habana. Cuba.

² Cuarto. Facultad de Física. Universidad de la Habana. Cuba.

E-mail: jorge.german@estudiantes.fisica.uh.cu nacosta@estudiantes.fisica.uh.cu

Tutor(es): Dr. N.Lopez¹,

¹ Facultad de Física.

Resumen

El siguiente trabajo representa una guía, para la construcción de un dispositivo con el cual se podrá calcular los coeficientes de absorción de distintas sustancias de forma casi automática. Para ello hicimos uso de un Arduino como corazón del proyecto, el cual se encarga de procesar las ordenes que son emitidas a los distintos dispositivos para realizar el cambio de longitud de onda incidente y la muestra a analizar. Para demostrar la funcionalidad del prototipo calculamos el coeficiente de absorción, para luz blanca y roja, de una sustancia la cual conocemos de antemano y comparamos los resultados con los reportados en la literatura. Obteniendo resultados bastantes acercados a la realidad.

Palabras Clave: Arduino, Coeficiente de Absorción.

1. Dispositivos necesarios para el montaje experimental

A continuación pasaremos a exponer cada elemento utilizado en el proyecto, su función y algunas de sus características más importantes.

1.1 Arduino Genuino UNO

El Arduino Uno Fig.1 es una placa de microcontrolador de código abierto basado en el microchip ATmega328P y desarrollado por Arduino.cc. La placa está equipada con conjuntos de pines de E/S digitales y analógicas que pueden conectarse a varias placas de expansión y otros circuitos. La placa tiene 14 pines digitales, 6 pines analógicos y programables con el Arduino IDE (Entorno de desarrollo integrado) a través de un cable USB tipo B. Puede ser alimentado por el cable USB o por una batería externa de 9 voltios, aunque acepta voltajes entre 7 y 20 voltios. En nuestro proyecto se utiliza para variar la longitud de onda usada mediante un receptor infrarrojo y dar las ordenes para mover el motor de paso y así cambiar la muestra.

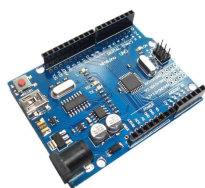


Figura 1: Placa de Arduino Genuino UNO

1.2 LED RGB con cátodo común

El LED RGB Fig.3 (red, green, blue) es un LED que en un solo empaque (encapsulado) tenemos los tres colores (rojo, verde y azul), donde podemos utilizar el PWM para generar hasta 16 millones tonos de colores (16.777.216), utilizando valores de 0-255 del color rojo y del verde y el azul Fig.2. Con este LED podemos estudiar un espectro de longitudes de onda muy amplio minimizando el uso de recursos debido a que podemos obtener todo el espectro visible con un solo dispositivo.

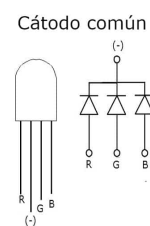


Figura 2: Diagrama electrónico LED RGB con cátodo común



Figura 3: LED RGB

1.3 Motor paso a paso 28BYJ-48

El motor paso a paso 28BYJ-48 Fig.4 posee un voltaje de funcionamiento 5V o 12V. Viene con un circuito integrado Fig.5 que esta conformado por 4 LEDs que indican cuando se excita una bobina y 4 resistencias para proteger los LEDs. Además de un Chip ULN2003 que contiene 3 transistores Darlington, 4 entradas para el controlador, por donde entran los pulsos y unos Jumpers para seleccionar el voltaje de funcionamiento (5V o 12V).

Cada paso avanza $5,625^\circ$. Tiene una caja reductora mediante engranajes $1/64$ que consiguen un paso de $5,625/64 = 0,088^\circ$. Además de una resistencia del bobinado de $50\ \Omega$ con un torque de 34 Newton Metro más o menos 35 gramos por cm y una frecuencia máxima 100Hz que equivale a un delay de 10 ms.

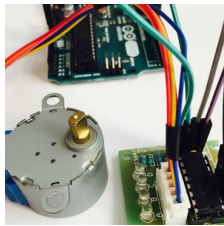


Figura 4: Motor paso a paso 28BYJ-48

Para alimentar el motor 28BYJ-48 desde una placa de Arduino, tenemos que puentear las conexiones con un jumper. Estos jumpers nos permiten seleccionar una alimentación de 5V o de 12V. El Arduino UNO trabaja con 5V como máximo por lo que usamos una fuente de corriente directa con 5V Fig.7.

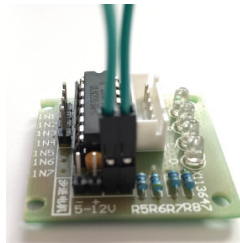


Figura 5: Circuito integrado

El objetivo de este motor es poder cambiar de muestra de forma automatizada y asegurando que las cubetas queden en la misma posición y así disminuir posibles errores en la medición

1.4 Modulo KY-022 Sensor Receptor Infrarrojo IR

El módulo receptor Arduino IR KY-022 Fig.6 reacciona a la luz infrarroja de 38 kHz. Este módulo KY-022 consta de un receptor IR 1838, una resistencia de $1\ k\Omega$ y un LED. La función de este receptor es recibir la ordenes para cambiar la longitud de onda e intercambiar las muestras a analizar.

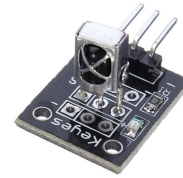


Figura 6: Sensor Receptor Infrarrojo IR

1.5 Fuente de voltaje de 5V

Como fuente se podría haber usado cualquier fuente de corriente directa. Nosotros utilizamos la xj173321 Fig.7 debido a que era la que teníamos disponible. La función de esta es suministrar la corriente al motor y así no forzar al Arduino y poder alargar su vida.



Figura 7: Fuente de CD

1.6 Foto-detector

Usamos un celular Xiaomi Mi A2 lite Fig.8 el cual se utiliza como foto-detector, procesando los valores de irradiancia obtenidos mediante la aplicación phyphox. Estos valores se registraron desde una Laptop marca ASUS, modelo X453SA, y se exportaron a formato excel para luego ser procesados.



Figura 8: Dispositivo usado como foto-detector

2. Explicación del montaje

El montaje general Fig.9 esta formado como se ve en la figura por cuatro componentes principales. La fuente, la cual alimenta el motor con un voltaje de 5V. El motor, que es el encargado de cambiar las muestras introducidas en las dos cubetas, haciendo un giro de 180° . Para ver las conexiones mas de cerca podemos ver en la Fig.10 como la tierra y el VCC del motor de paso están conectadas a las de la fuente y a su vez la tierra de la fuente conectada a la del Arduino. En el

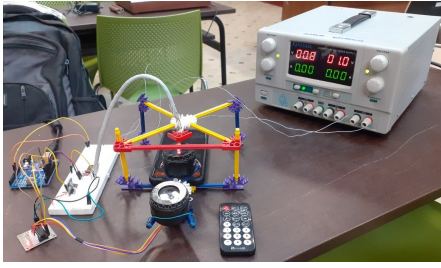


Figura 9: Montaje experimental completo

motor también podemos ver las conexiones encargadas de llevar a cabo el movimiento de las cubetas.

La tercera componente es la fuente de luz (LED). Para el funcionamiento del LED colocamos una resistencia de 680Ω antes de cada entrada, que va conectada al Arduino; para controlar el voltaje que recibe el LED, además de conectamos el común a tierra.

La cuarta componente es el IR Reciver el cual tiene tres conexiones, una tierra, un VCC y la tercera que es por la cual se transmite la señal al Arduino para luego ser interpretada y usada de acuerdo a como se programo. Dichas señales se usan para cambiar la longitud de onda que emite el LED y variar la muestra a estudiar.

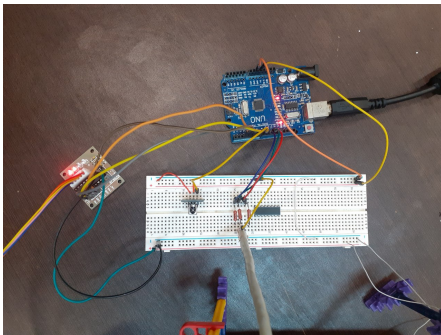


Figura 10: Conexiones en el Protoboard

3. Aplicación Física del Proyecto

3.1 Explicación Teórica

En nuestro proyecto, para luz roja, obtenemos valores de intensidad luminosa al hacer pasar la luz por dos cubetas. Una con agua y otra con una sustancia a investigar. El objetivo final es calcular el coeficiente de absorción y poder identificar cual es la sustancia que estamos investigando. Durante 10s tomamos los valores que llegan al foto-detector (Teléfono móvil) y exportamos dichos valores a formato .xlsx luego a partir de ellos sacar el valor medio y así poder trabajar con un valor lo más cercano a la realidad posible. Al realizar lo antes explicado para la cubeta con agua obtenemos el valor I_0 e I para la cubeta con la sustancia y teniendo el valor del espesor de la cubeta fija, que en este caso es 20mm, determinamos el coeficiente de absorción

mediante la ecuación:

$$k = \frac{1}{d} \ln\left(\frac{I_0}{I}\right) \quad (1)$$

Y con el coeficiente de absorción podemos calcular la densidad óptica mediante la ecuación:

$$D = kd \quad (2)$$

3.2 Resultados y Discusión

Para procesar los datos utilizamos el programa computacional Wolfram Mathematica, para la luz roja obtuvimos los siguientes valores medios de las intensidades luminosas.

La luz roja:

$$I_0 = 199,638lux \text{ e } I = 71,697lux$$

Los cuales al sustituir en la ecuación (1) obtenemos para el coeficiente de absorción un valor de $k = 0,0512mm^{-1}$ y al sustituir k y d en la ecuación (2) obtenemos el valor de la densidad óptica como $D = 1,024$

En la práctica de laboratorio sobre absorción se utiliza la longitud de onda de máxima absorción la cual se corresponde con $\lambda = 8 \cdot 10^2 nm$, con la cual se obtiene un coeficiente de absorción $K = 0,239mm^{-1}$ y una densidad óptica de $D = 4,78$

4. Conclusiones

El resultado obtenido no se corresponde al valor real esto puede ser debido a las condiciones en las que se realizó el experimento, la luz del led no se encontraba previamente colimada; el foto-detector recibía ruido de otros sistemas como la luz de la lámpara, etc. Esta y otras causas constituyen grandes fuentes de incertidumbre en la medición. A pesar de las causas de errores antes mencionadas, el método desarrollado constituye una buena aproximación para automatizar la identificación de las sustancias, además de que tiene su aplicación práctica en la determinación de la absorción en los tejidos de la piel y en otras sustancias; las cuales no fueron posible implementarse debido a que no teníamos los materiales imprescindible para su montaje experimental.

Anexo A. Código del programa

```

1      #include <IRremote.h>
2      #include <Stepper.h>
3      int RECV_PIN = 8;
4      int bluePin = 9;
5      int greenPin = 10;
6      int redPin = 11;
7      double stepsPerRevolution = 2048;
8      int step1 = 1040;
9      Stepper myStepper(stepsPerRevolution, 6, 4, 5, 3);
10     IRrecv irrecv(RECV_PIN);
11     decode_results results;
12
13     void setup(){
14         myStepper.setSpeed(10);
15         irrecv.enableIRIn();
16         pinMode(redPin, OUTPUT);
17         pinMode(greenPin, OUTPUT);
18         pinMode(bluePin, OUTPUT);
19     }
20
21     void loop(){
22         if (irrecv.decode(&results)){
23             int value = results.value;
24
25             switch(value){
26                 case 12495: //Keypad button "1"
27                     analogWrite(redPin, 0xFF);
28                     analogWrite(greenPin, 0xFF);
29                     analogWrite(bluePin, 0xFF);
30                     break;
31
32                 case 6375: //Keypad button "2"
33                     analogWrite(redPin, 0x00);
34                     analogWrite(greenPin, 0x00);
35                     analogWrite(bluePin, 0xFF);
36                     break;
37
38                 case 31365: //Keypad button "3"
39                     analogWrite(redPin, 0xFF);
40                     analogWrite(greenPin, 0x00);
41                     analogWrite(bluePin, 0x00);
42                     break;
43
44                 case 4335: //Keypad button "4"
45                     analogWrite(redPin, 0x00);
46                     analogWrite(greenPin, 0xFF);
47                     analogWrite(bluePin, 0x00);
48                     break;
49
50                 case -15811:
51                     myStepper.step(-step1);
52                     delay(1000);
53                     break;
54
55                 case 8925:
56                     myStepper.step(step1);
57                     delay(1000);
58                     break;
59
60                 case -28561:
61                     myStepper.step(-500);
62                     delay(1000);
63                     break;
64
65                 case -8161:
66                     myStepper.step(500);
67                     delay(1000);
68                     break;
69
70                 default:
71                     int qq=0;
72             }
73             irrecv.resume();
74         } }

```