

## Sensor de Color TCS3200 con Arduino.

La forma en que la tecnología interactúa con el ambiente, crece de manera descomunal, hoy tenemos sensores que pueden medir todos y cada uno de los parámetros físicos; sensores de temperatura, de gas, de sonido, de campos magnéticos, etc. Esto permite que las aplicaciones en los campos de la domótica, robótica y electrónica en general crezcan al mismo ritmo.

Uno de estos sensores es el de color RGB TCS3200, este es un convertidor de luz a frecuencia que combina fotodiodos de silicio reconfigurables y una corriente de frecuencia en un solo circuito integrado. La salida es una onda cuadrada (ciclo de trabajo 50%) con una frecuencia directamente proporcional a la intensidad de luz. Las entradas y salidas digitales permiten una interfaz directa con un microcontrolador u otro conjunto de circuitos lógicos, por esta razón el sensor TCS3200 es ideal para líneas de producción, domótica, robótica, etc. En la figura N° 1 se puede observar un ejemplo de la implementación de sensores de color.

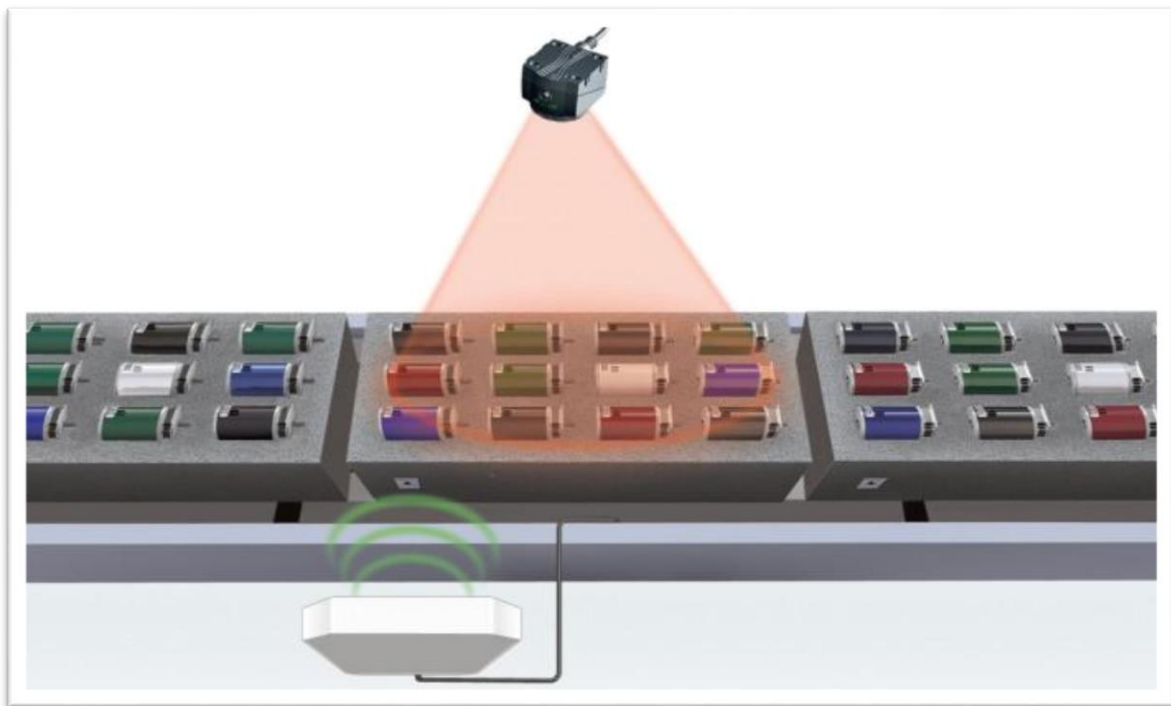


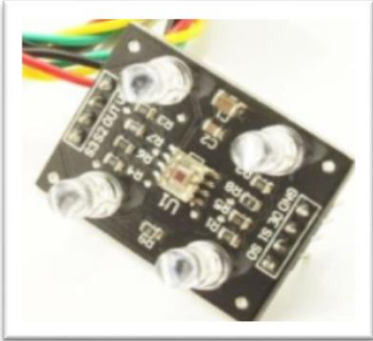

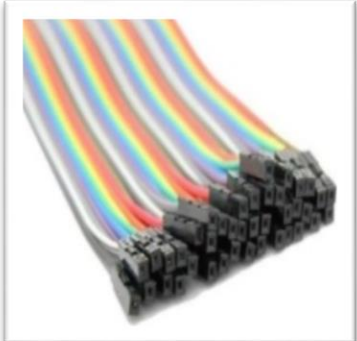
Figura n° 1: Implementación de un sensor de color para la separación de objetos en una banda.

Como se mencionó anteriormente existen sensores que se encargan de detectar los colores primarios: el rojo, el verde y el azul; su implementación en distintos equipos electrónicos varía dependiendo de las necesidades que se tengan. Para ello es importante saber cómo está formado el sensor y como es su funcionamiento.

El sensor TCS3200 es un convertidor de luz a frecuencia, el cual detecta la luz de color con la ayuda de una serie de fotodiodos de 8x8, de tal manera que 16 fotodiodos tienen filtro azul, 16 fotodiodos tienen filtro verde, 16 fotodiodos tienen filtro rojo y 16 fotodiodos son sin filtro.

Luego, utilizando un convertidor de corriente a frecuencia, las lecturas de los fotodiodos se convierten en una onda cuadrada con una frecuencia directamente proporcional a la intensidad de la luz. Finalmente, usando la placa Arduino, podemos leer la salida de onda cuadrada y obtener los resultados del color.

## Material Utilizado.

Electrónico		
Sensor de color TCS3200.  SKU: SR0005	Arduino uno RV3.  SKU: AR0016	Dupont M/H de 20 cm.  SKU: CS0006

## Diagrama de Conexión.

Los 16 fotodiodos de cada filtro se conectan en paralelo, por lo que con los dos pines de control S2 y S3 podemos seleccionar cuál de ellos será leído. Entonces, por ejemplo, si queremos detectar el color rojo, podemos usar los 16 fotodiodos filtrados en rojo ajustando los dos pines al nivel lógico bajo según la tabla N° 1.

S2	S3	Photodiode Type
L	L	Red
L	H	Blue
H	L	Clear (no filter)
H	H	Green

Tabla N° 1: Tabla de estados para detectar los colores.

El sensor tiene dos pines de control más, S0 y S1 que se utilizan para escalar la frecuencia de salida. La frecuencia se puede escalar a tres valores predeterminados diferentes de 100%, 20% o 2%. Esta función de escalado de frecuencia permite que la salida del sensor se perfeccione para varios contadores de frecuencia o microcontroladores. Ver tabla N° 2.

<i>S0</i>	<i>S1</i>	<i>Output Frequency Scaling</i>
L	L	Power down
L	H	2%
H	L	20%
H	H	100%

Tabla N° 2: Tabla de estados para el porcentaje del ciclo de trabajo.

Además, posee los pines o conectores VCC, GND, OUT y OE. En la figura N° 2 se puede observar las conexiones del sensor de color RGB TCS3200.

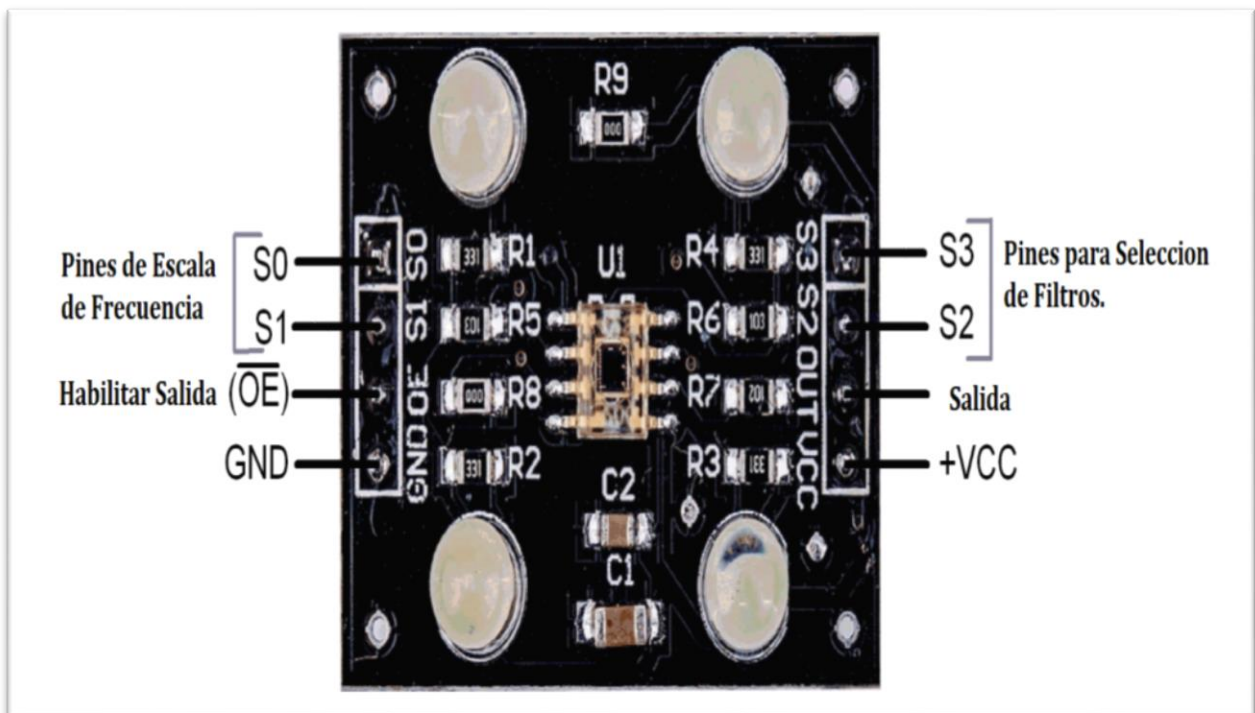


Figura n° 2: Pines de conexión del sensor RGB TCS3200.

Para probar el sensor, se conecta al Arduino uno como se muestra en la figura N° 3 y se describe a continuación:

## Alimentación del Circuito.

- Conecte el pin 5V del Arduino UNO al pin VCC del sensor de color.
- Conecte el pin GND del Arduino UNO al pin GND del sensor.

## Conexión del Sensor TCS3200.

- Conecte el pin GND del Arduino UNO al pin OE del sensor de color.
- Conecte el pin digital 8 del Arduino UNO al pin S0 del sensor de color.
- Conecte el pin digital 9 del Arduino UNO al pin S1 del sensor de color.
- Conecte el pin digital 10 del Arduino UNO al pin OUT del sensor de color.
- Conecte el pin digital 11 del Arduino UNO al pin S3 del sensor de color.
- Conecte el pin digital 12 del Arduino UNO al pin S2 del sensor de color.

## Conexión de los leds.

- Conecte el pin digital 2 del Arduino UNO al pin Ánodo del led rojo.
- Conecte el pin digital 3 del Arduino UNO al pin Ánodo del led verde.
- Conecte el pin digital 4 del Arduino UNO al pin Ánodo del led rojo.
- Conecte el pin GND a un extremo de una resistencia de 330Ω.
- Conecte el otro extremo de la resistencia a la misma línea que los cátodos de los leds previamente conectados, puede auxiliarse de una protoboard.

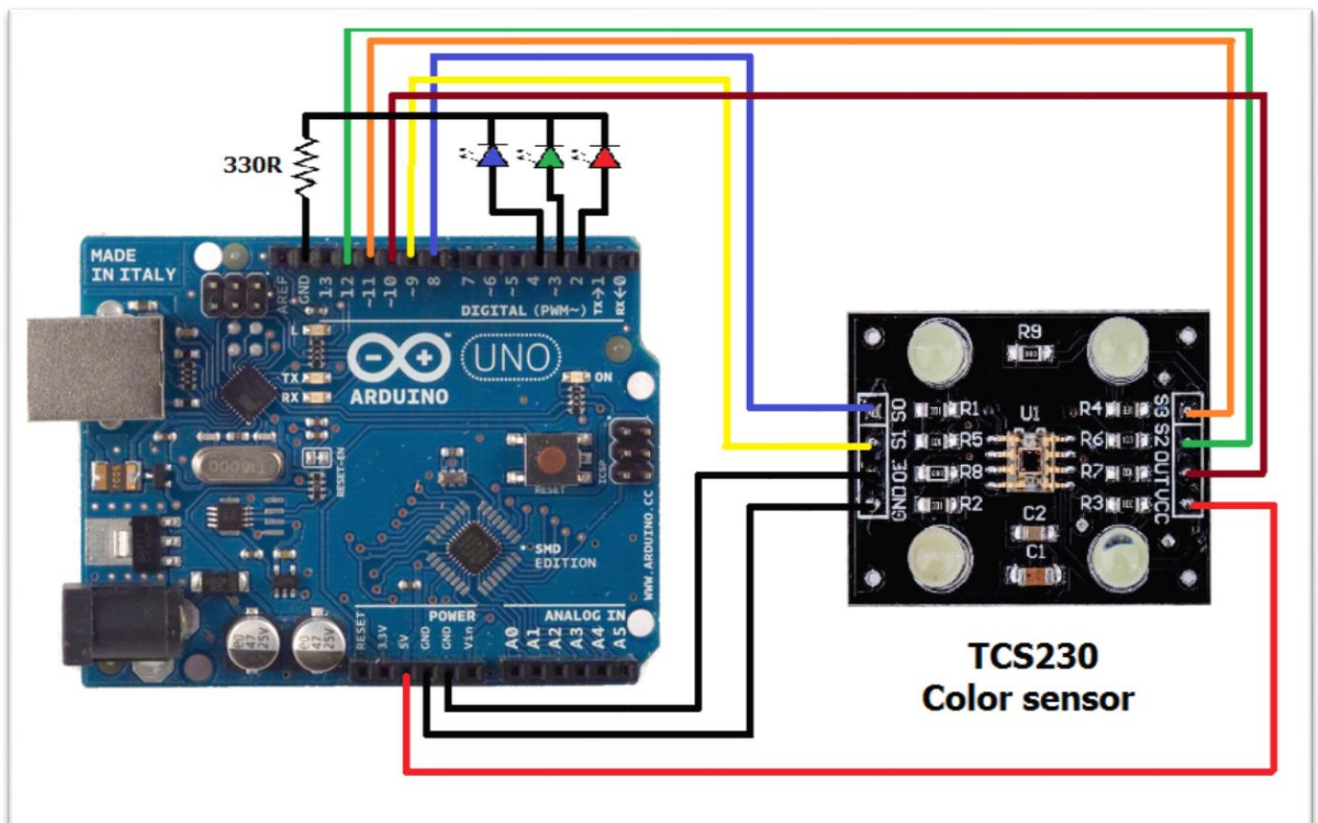


Figura N° 3: Diagrama de Conexión del sensor de color.

## Código Usado.

Este es el código que se usó en la tarjeta Arduino para el funcionamiento correcto del sensor de color TCS3200. Para programar es necesario contar con el programa IDE de Arduino.

```
const int s0 = 8;
const int s1 = 9;
const int s2 = 12;
const int s3 = 11;
const int out = 10;

int redLed = 2;    // pines de conexión de los leds
int greenLed = 3;
int blueLed = 4;

int red = 0;       // Variables
int green = 0;
int blue = 0;

void setup()
{
  Serial.begin(9600);
  pinMode(s0, OUTPUT);
  pinMode(s1, OUTPUT);
  pinMode(s2, OUTPUT);
  pinMode(s3, OUTPUT);
  pinMode(redLed, OUTPUT);
  pinMode(greenLed, OUTPUT);
  pinMode(blueLed, OUTPUT);
  pinMode(out, INPUT);

  digitalWrite(s0, HIGH);
  digitalWrite(s1, HIGH);
}

void loop()
{
  color();

  Serial.print("R Intensity:");
  Serial.print(red, DEC);
  Serial.print(" G Intensity: ");
  Serial.print(green, DEC);
  Serial.print(" B Intensity : ");
  Serial.print(blue, DEC);

  if (red < blue && red < green && red < 20)
  {
    Serial.println(" - (Red Color)");
    digitalWrite(redLed, HIGH); //Encender led
    digitalWrite(greenLed, LOW);
  }

  digitalWrite(blueLed, LOW);
}

else if (blue < red && blue < green)
{
  Serial.println(" - (Blue Color)");
  digitalWrite(redLed, LOW);
  digitalWrite(greenLed, LOW);
  digitalWrite(blueLed, HIGH); // Encender led
}

else if (green < red && green < blue)
{
  Serial.println(" - (Green Color)");
  digitalWrite(redLed, LOW);
  digitalWrite(greenLed, HIGH); // Encender led
  digitalWrite(blueLed, LOW);
}

else
{
  Serial.println();
}

delay(300);
digitalWrite(redLed, LOW);
digitalWrite(greenLed, LOW);
digitalWrite(blueLed, LOW);
}

void color()
{
  digitalWrite(s2, LOW);
  digitalWrite(s3, LOW);

  red = pulseIn(out, digitalRead(out) == HIGH ?
LOW : HIGH);
  digitalWrite(s3, HIGH);

  blue = pulseIn(out, digitalRead(out) == HIGH ?
LOW : HIGH);
  digitalWrite(s2, HIGH);

  green = pulseIn(out, digitalRead(out) == HIGH ?
LOW : HIGH);
}
```



## Imágenes de Funcionamiento.

En las figuras N° 4, N° 5 y N° 6 se puede observar el funcionamiento del sensor conectado al Arduino, preniendo los leds dependiendo del color de la llanta detectado.

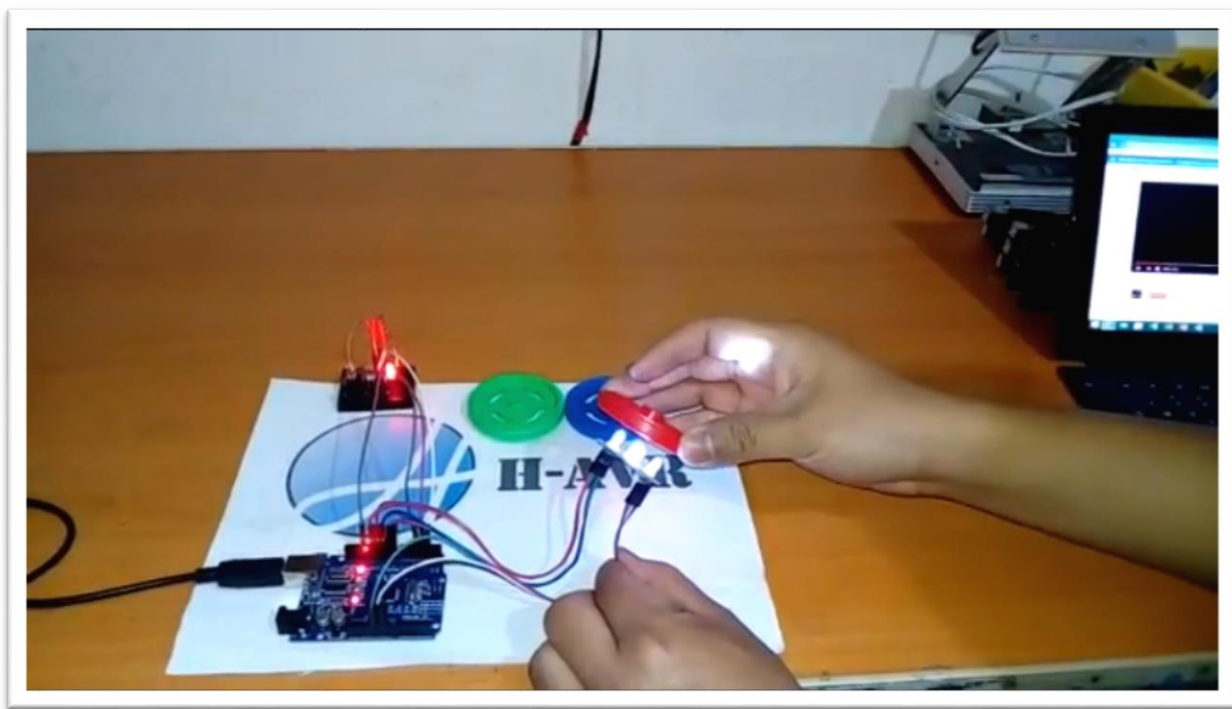


Figura N° 4: Detección del color rojo.

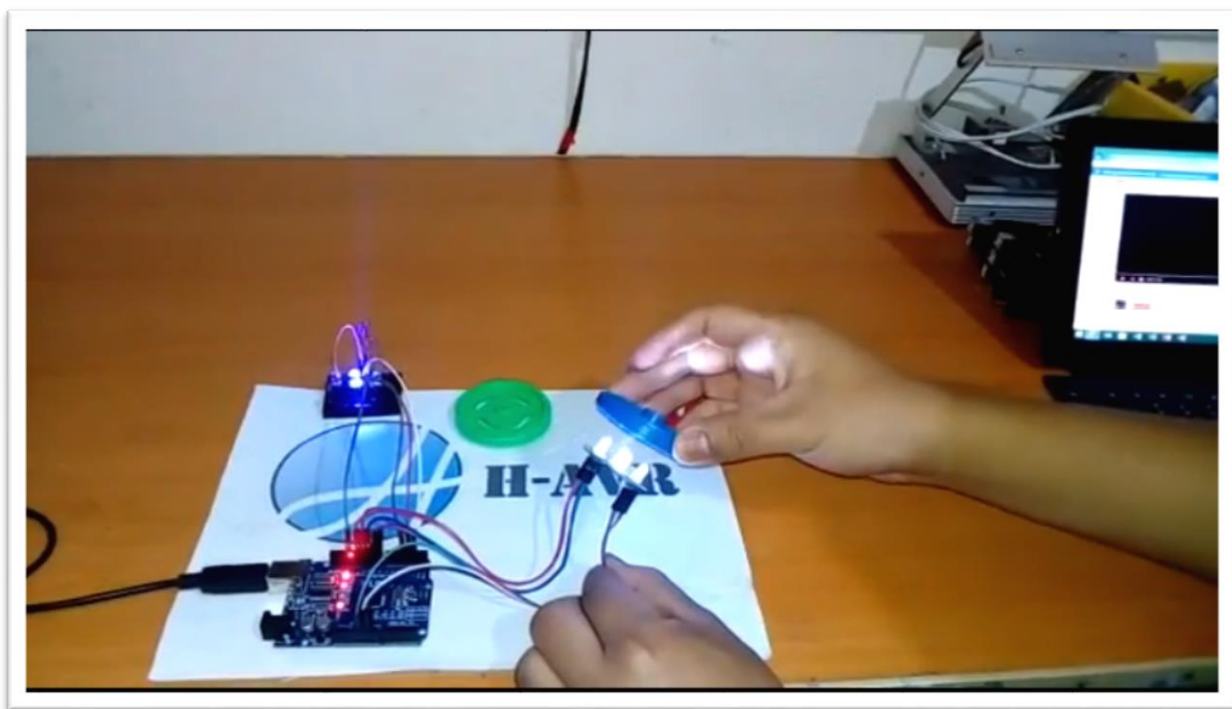


Figura N° 5: Detección del color azul.

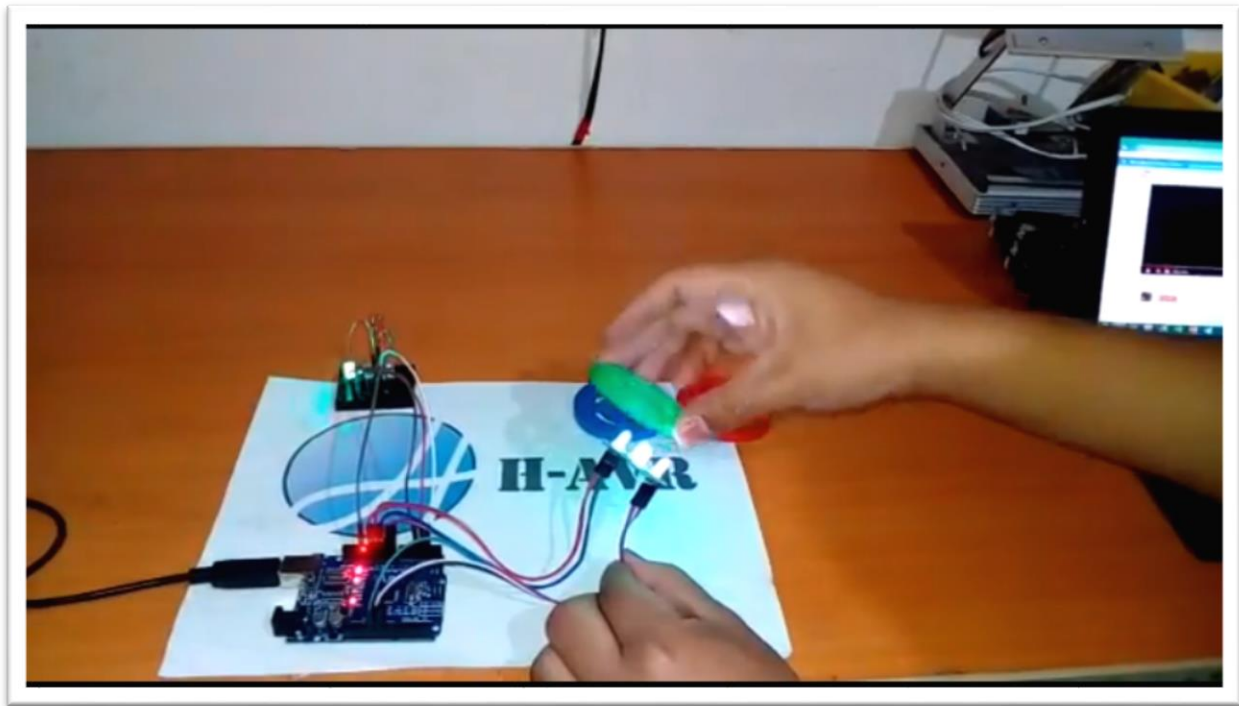


Figura N° 6: Detección del color verde.

## Conclusiones.

Al término de esta pequeña práctica demostrativa de como conectar un sensor de color a un Arduino uno, se puede observar que la tabla de valores que visualiza previamente es importante para poder hacer que el filtro del color correspondiente se active, como se menciona, este cuenta con una matriz de fotodiodos de 8x8. También cabe aclarar que con la variación del ciclo del trabajo de nuestra señal cuadrada podemos obtener el espectro de colores conectando a la salida un led RGB.

## Contacto.

- <http://www.h-avr.mx/>

## Video del Funcionamiento.

- <https://www.youtube.com/watch?v=YN89W3kcHyQ>

## Donde Comprar:



**mercado  
libre**



**H-AVR**  
ELECTRÓNICA



**H-AVR**  
ELECTRÓNICA