

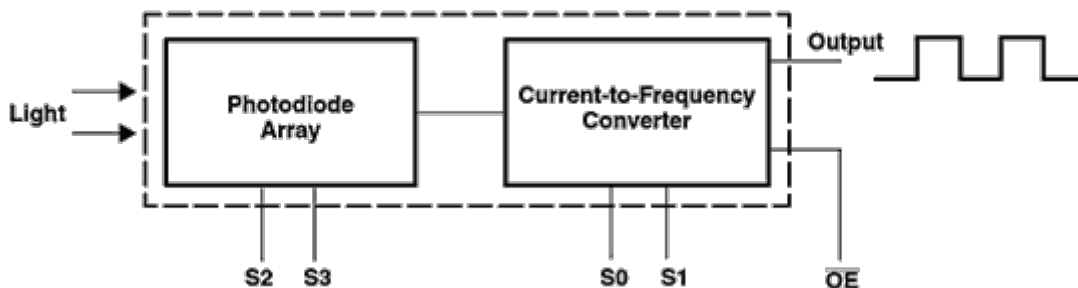
Sensor de color TCS3200: Descripción y código para Arduino



- Aclarar antes que este módulo no se comunica por I2C pero está incluido aquí por su relación con los sensores de color y el Arduino.

Chip de color TCS3200

- EL TCS3200 es un sensor óptico que permite detectar el color de un objeto ubicado en frente de él. Podemos conectarlo este sensor con facilidad a un autómata o procesador como Arduino.
- Internamente, el TCS3200 está formado por una matriz de fotodiodos de silicóna junto con un convertor de frecuencia, en un único integrado CMOS.
- La matriz dispone de 8 x 8 fotodiodos de 110 µm, de los cuales 16 tienen filtros azules, 16 verdes, 6 rojos, y 16 no tienen filtro. Los fotodiodos están distribuidos de forma que minimizan el efecto la incidencia no uniforme de la luz.
- La salida del TCS3200 es una onda cuadrada del 50% duty, cuya frecuencia es proporcional a la intensidad luminosa. La tensión de alimentación del sensor es de 2.7V a 5.5V.



- Frecuentemente el TCS3200 se distribuye en módulos que incorporan dos o cuatro LED de luz blanca y un protector de plástico. El objetivo de ambas medidas es minimizar los efectos de la iluminación ambiente en la medición.
- Pese a sus especificaciones y elementos para eliminar la luz ambiente, el TCS3200 no es capaz de medir de forma precisa el color RGB de un objeto, o la temperatura de color de una fuente luminosa.
- Sin embargo, podemos emplearlo para distinguir entre colores básicos. Por ejemplo, podemos emplearlo para reconocer el color de una tarjeta o un objeto, y guiar a un robot en un recorrido.
- Si buscas un sensor de color deberías mirar el TCS34725, un sensor RGB más moderno y avanzado que el TCS3200.

Montaje y código para Arduino

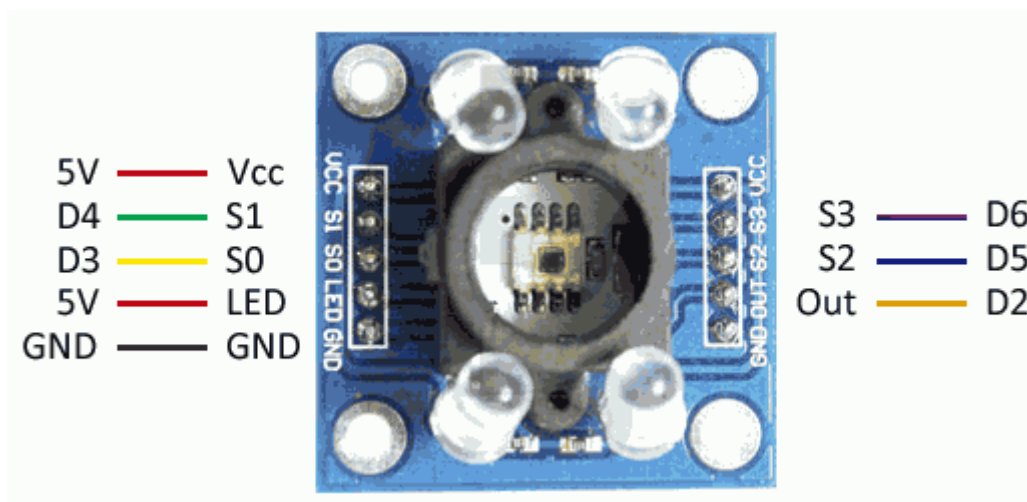
- El TCS3200 tiene cuatro entradas digitales S0, S1, S2, y S3, y una salida digital Out. Para conectarlo a Arduino necesitaremos emplear al menos 3 pines digitales.
- En primer lugar debemos alimentar el módulo conectando los pines Gnd y Vcc del TCS3200, respectivamente, a Gnd y Vcc de Arduino.
- Los pines S0 y S1 controlan la frecuencia de la salida y la desactivación del módulo. Los conectamos a dos salidas digitales, o podemos conectarlas a 5V si no queremos poder apagar el módulo.

	Power down	2%	20%	100%
S0	L	L	H	H
S1	L	H	L	H

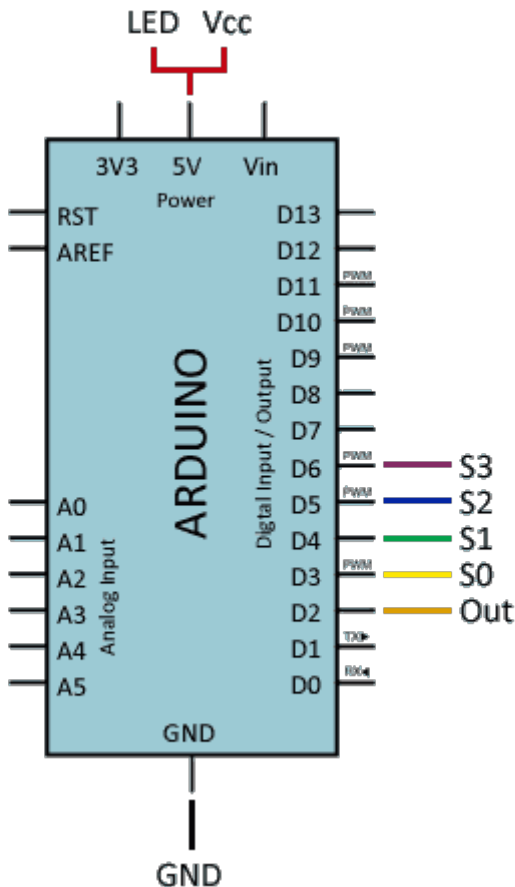
- Por otra parte, los pines S2 y S3 seleccionan el color a medir. Debemos conectarlos a dos salidas digitales de Arduino.

	Red	Blue	Clear	Green
s2	L	L	H	H
s3	L	H	L	H

- Finalmente, conectamos la salida del sensor Out a una entrada digital de Arduino por lo que la conexión empleando los pines, sería la siguiente.



- Mientras que la conexión, vista desde Arduino, sería la siguiente.



- Código.** En el siguiente ejemplo realizamos la lectura del TCS3200.
 - Para ello empleamos la función `PulseIn` para determinar la duración del pulso recibido por el sensor.

- Realizamos el proceso para cada color, y empleamos los valores obtenidos para clasificarlo como rojo, azul o verde.

Código del ejemplo

[ejm_tcs3200.ino](#)

```
//VCC—5V
//GND—GND
//S0—D3
//S1—D4
//S2—D5
//S3—D6
//OUT—D2

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

byte countRed = 0;
byte countGreen = 0;
byte countBlue = 0;

void setup() {
  Serial.begin(9600);
  pinMode(s0, OUTPUT);
  pinMode(s1, OUTPUT);
  pinMode(s2, OUTPUT);
  pinMode(s3, OUTPUT);
  pinMode(out, INPUT);
  digitalWrite(s0, HIGH);
  digitalWrite(s1, HIGH);
}

void loop() {
  getColor();
  Serial.print("Red: ");
  Serial.print(countRed, DEC);
  Serial.print("Green: ");
  Serial.print(countGreen, DEC);
  Serial.print("Blue: ");
  Serial.print(countBlue, DEC);

  if (countRed < countBlue && countGreen > 100 && countRed < 80)
  {
    Serial.println(" - Red");
  }
  else if (countBlue < countRed && countBlue < countGreen)
  {
    Serial.println(" - Blue");
  }
  else if (countGreen < countRed && countGreen < countBlue)
  {
    Serial.println(" - Green");
  }
  else {
    Serial.println("-");
  }
  delay(300);
}
```

```
void getColor()
{
    digitalWrite(s2, LOW);
    digitalWrite(s3, LOW);
    countRed = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
    digitalWrite(s3, HIGH);
    countBlue = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
    digitalWrite(s2, HIGH);
    countGreen = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
}
```

From: <https://www.euloxio.myds.me/dokuwiki/> - **Euloxio wiki**

Permanent link: https://www.euloxio.myds.me/dokuwiki/doku.php/doc:tec:lab:bus_i2c:color_tcs3200_0:inicio

Last update: **2026/03/28 18:45**

