**SYSTEM REQUIREMENTS**

The TCS3200 color sensor can detect a wide variety of colors based on their wavelength.This sensor is specially useful for color recognition projects such as color matching,color sorting,test strip reading and much more.

The TCS3200 color sensor -shown in the figure below- uses a TAOS TCS3200 RGB sensor chip to detect color. It contains four LED's that light up the object in front of it.



**Specifications**

Here are the sensor specification:

* Power:2.7V to 5.5V
* Size:28.4\*28.4mm (1.12\*1.12”)
* Interface:digital TTl
* High-resolution conversion of light intensity to frequency
* Programmable color and full-scale output frequency
* Communicates directly to micro-controller

The TCS3200 has an array of photo diodes with 4 different filters. A photo diode is simply a semi conductor device that converts light to current. The sensor has:

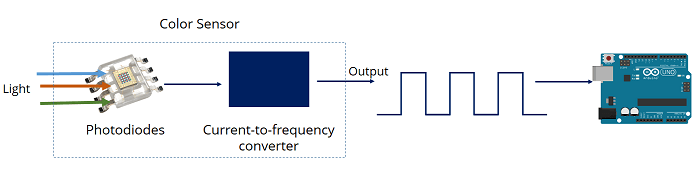
* 16 photo diodes with red filter- sensitive to red wavelength
* 16 photo diodes with green filter- sensitive to green wavelength
* 16 photo diodes with blue filter- sensitive to blue wavelength
* 16 photo diodes without filter.

If we tale a closer look at TCS3200 chip we can see the different filters.



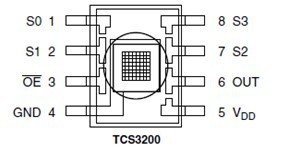
By selectively choosing the photo diodes filters readings,we’re able to detect the intensity of the different colors. The sensor has a current-to-frequency converter that converts the photo diodes readings into a square wave with a frequency that is proportional to the light intensity of the chosen color. This frequency is then read by the arduino

- this is shown in the figure below.



**Pin out**

Here is the sensors pin out

****

|  |  |  |
| --- | --- | --- |
| **Pin Name** | **I/O** | **Description** |
| GND (4) |  | Power supply ground |
| OE (3) | I | Enable for output frequency (active low) |
| OUT (6) | O | Output frequency |
| S0, S1（1，2） | I | Output frequency scaling selection inputs |
| S2, S3（7，8） | I | Photo diode type selection inputs |
| VDD（5） |  | Voltage supply |

To select the color read by the photo diode, we use the control pins S2 and S3. As the photo diodes are connected in parallel, setting the S2 and S3 LOW and HIGH in different combinations allows you to select diffrent photo diodes.

|  |  |  |
| --- | --- | --- |
| **Photo diode type** | **S2** | **S3** |
| Red | LOW | LOW |
| Blue | LOW | HIGH |
| No filter (clear) | HIGH | LOW |
| Green | HIGH | HIGH |

**Frequency scaling**

Pins S0 and S1 are used for scaling the output frequency. It can be scaled to the following preset values: 100%, 20% or 2%. Scaling the output frequency is useful to optimize the sensor readings for various frequency counters or micro controllers. Take a look at the table below:

|  |  |  |
| --- | --- | --- |
| **Output frequency scaling** | **S0** | **S1** |
| Power down | L | L |
| 2% | L | H |
| 20% | H | L |
| 100% | H | H |

For the Arduino, it is common to use a frequency scaling of 20%. So, you set the S0 pin to HIGH and the S1 pin to LOW.

**SYSTEM DESIGN**

We are going to detect colors with the Arduino and the TCSP3200 color sensor.  This sensor is accurate, and works fine for detecting colors in simple projects.

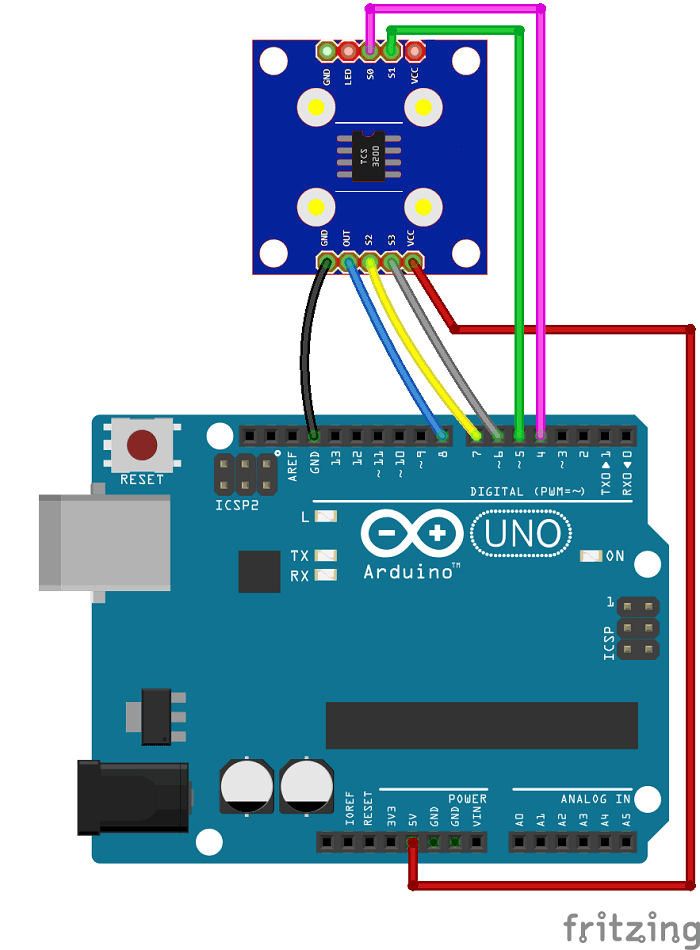
### **Parts required**

Here’s the parts required for this project:

* TCS3200 or TCS230 color sensor
* Arduino UNO
* Jumper wires

### **Schematic**

Wiring the TCSP3200 sensor to your Arduino is pretty straightforward.

n

Here are the connections between the TCSP3200 and the Arduino:

* **S0**: digital pin 2
* **S1**: digital pin 3
* **VCC**: 5V
* **S2**: digital pin 4
* **S3**: digital pin 5
* **OUT**: digital pin 10

**IMPLEMENTATION**

This next sketch maps the frequency values to RGB values (that are between 0 and 255).

**CODE:**

// TCS3200 pins wiring to Arduino  
#define S0 2  
#define S1 3  
#define S2 4  
#define S3 5  
#define sensorOut 10  
  
// Stores frequency read by the photodiodes  
int redFrequency = 0;  
int greenFrequency = 0;  
int blueFrequency = 0;  
  
// Stores the red. green and blue colors  
int redColor = 0;  
int greenColor = 0;  
int blueColor = 0;  
  
void setup() {  
  // Setting the outputs  
  pinMode(S0, OUTPUT);  
  pinMode(S1, OUTPUT);  
  pinMode(S2, OUTPUT);  
  pinMode(S3, OUTPUT);  
    
  // Setting the sensorOut as an input  
  pinMode(sensorOut, INPUT);  
    
  // Setting frequency scaling to 20%  
  digitalWrite(S0,HIGH);  
  digitalWrite(S1,LOW);  
    
  // Begins serial communication  
  Serial.begin(9600);  
}  
  
void loop() {  
  // Setting RED (R) filtered photodiodes to be read  
  digitalWrite(S2,LOW);  
  digitalWrite(S3,LOW);  
    
  // Reading the output frequency  
  redFrequency = pulseIn(sensorOut, LOW);  
  // Remaping the value of the RED (R) frequency from 0 to 255  
  redColor = map(redFrequency, 70, 129, 255,0)0  
    
  // Printing the RED (R) value  
  Serial.print("R = ");  
  Serial.print(redColor);  
  delay(100);  
    
  // Setting GREEN (G) filtered photodiodes to be read  
  digitalWrite(S2,HIGH);  
  digitalWrite(S3,HIGH);  
    
  // Reading the output frequency  
  greenFrequency = pulseIn(sensorOut, LOW);  
  // Remaping the value of the GREEN (G) frequency from 0 to 255;  
  greenColor = map(greenFrequency, 100, 199, 255, 0);  
    
  // Printing the GREEN (G) value    
  Serial.print(" G = ");  
  Serial.print(greenColor);  
  delay(100);  
   
  // Setting BLUE (B) filtered photodiodes to be read  
  digitalWrite(S2,LOW);  
  digitalWrite(S3,HIGH);  
    
  // Reading the output frequency  
  blueFrequency = pulseIn(sensorOut, LOW);  
  // Remaping the value of the BLUE (B) frequency from 0 to 255  
  blueColor = map(blueFrequency, 38, 84, 255, 0);  
    
  // Printing the BLUE (B) value   
  Serial.print(" B = ");  
  Serial.print(blueColor);  
  delay(100);  
  
  // Checks the current detected color and prints  
  // a message in the serial monitor  
  if(redColor > greenColor && redColor > blueColor){  
      Serial.println(" - RED detected!");  
  }  
  if(greenColor > redColor && greenColor > blueColor){  
    Serial.println(" - GREEN detected!");  
  }  
  if(blueColor > redColor && blueColor > greenColor){  
    Serial.println(" - BLUE detected!");  
  }  
}

Open the serial monitor at a baud rate of 9600.

To distinguish between different colors we have three conditions:

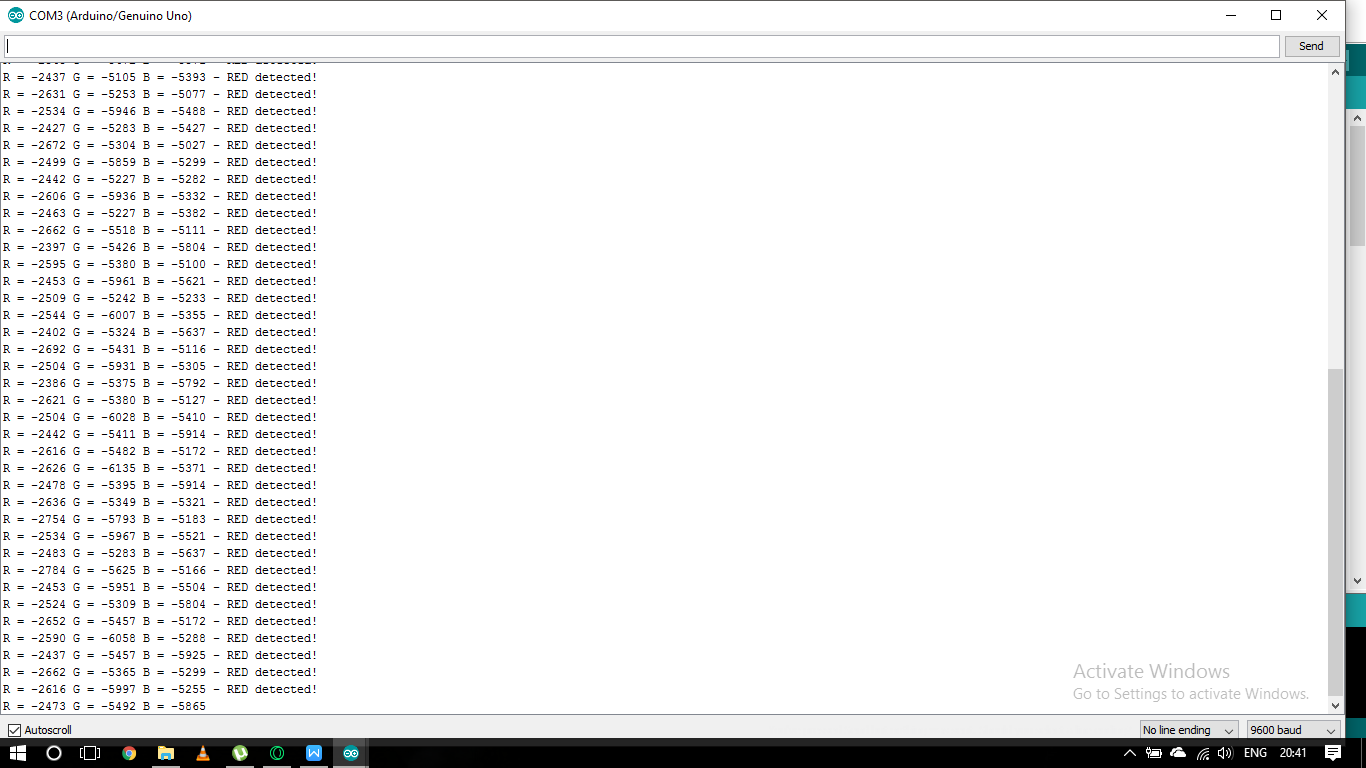
* When the R is the maximum value (in RGB parameters) we know we have a red object
* When G is the maximum value, we know we have a green object
* When B is the maximum value, we know we have a blue object

Now, place something in front of the sensor. It should print in your serial monitor the color detected: red, green or blue.

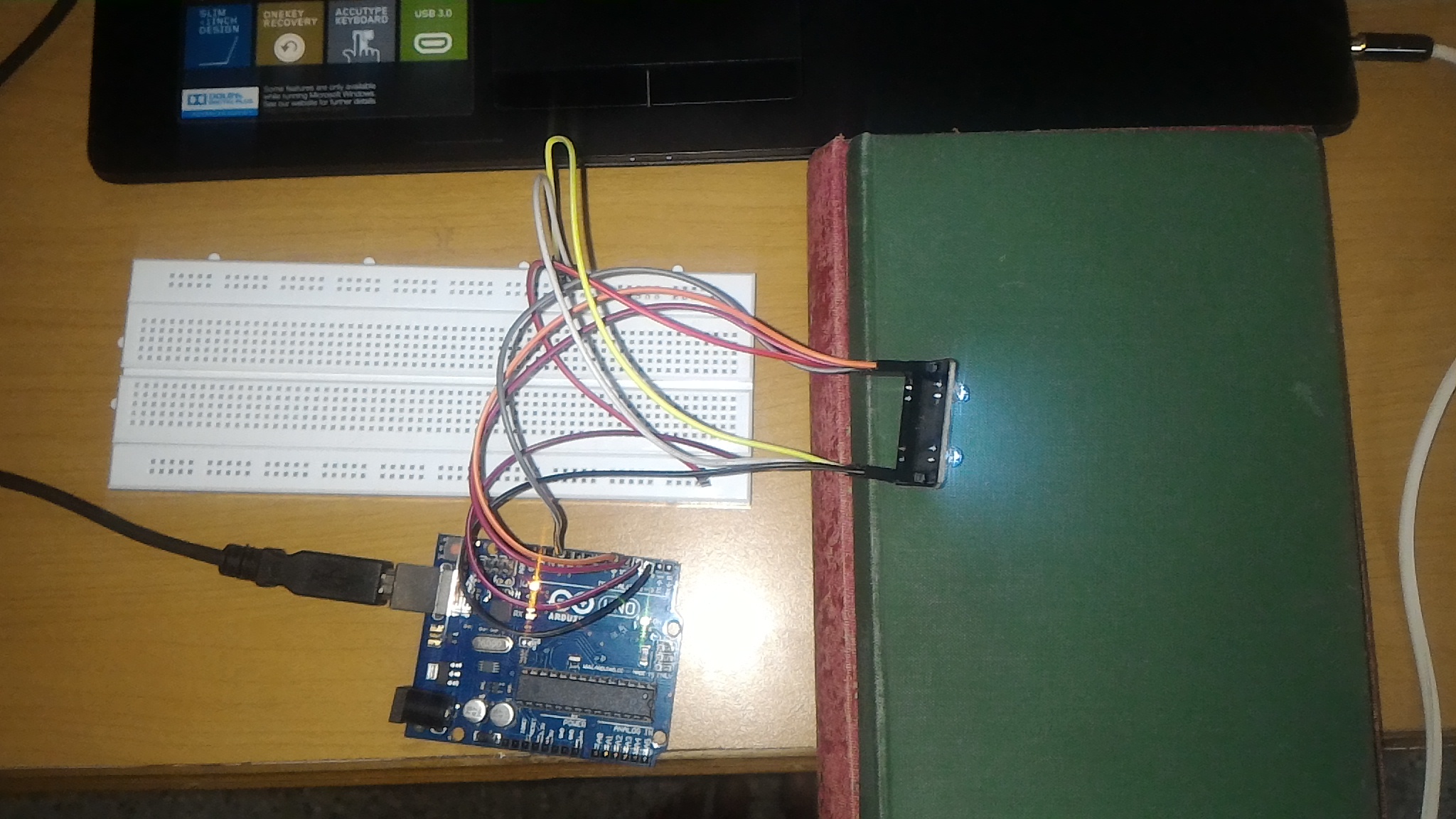
**SNAPSHOTS**

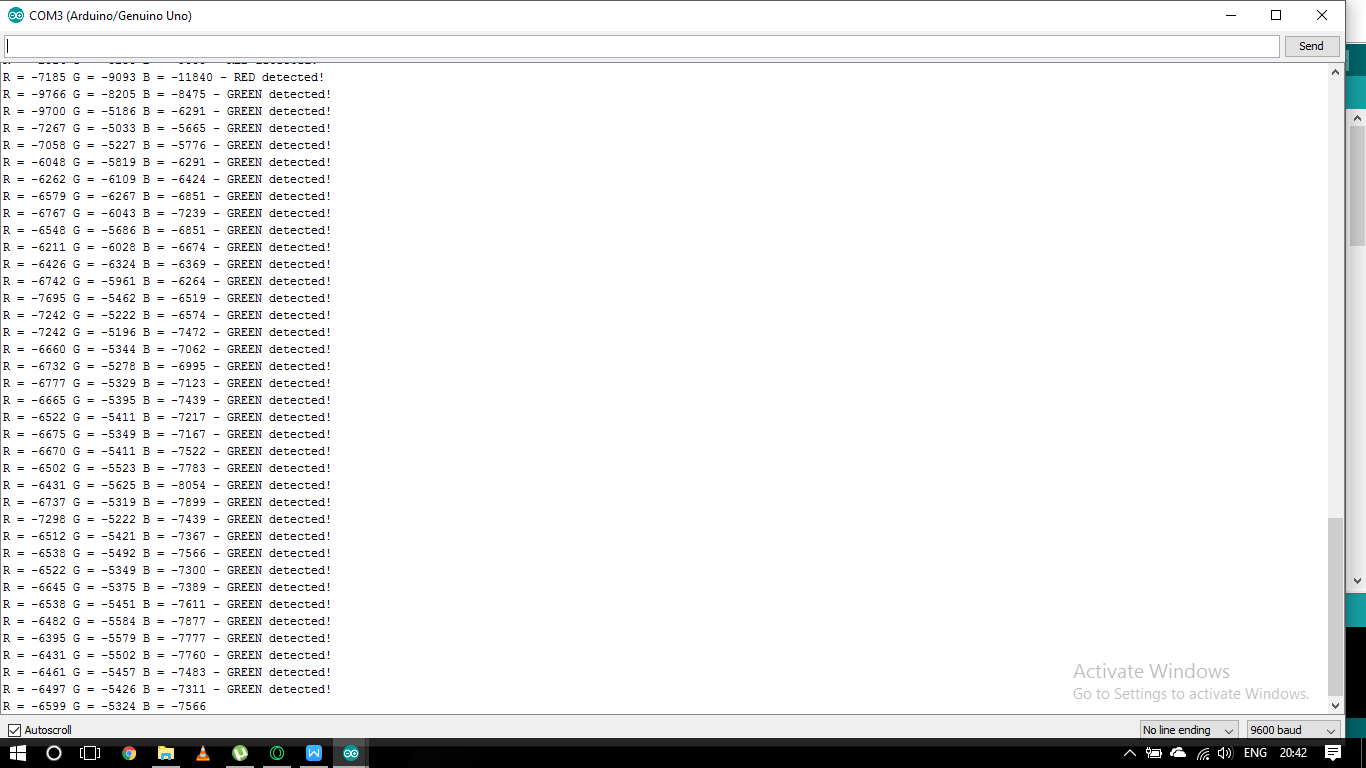
# P_20170504_203940

Detecting red color:



Detecting green color:





Detecting blue color:

