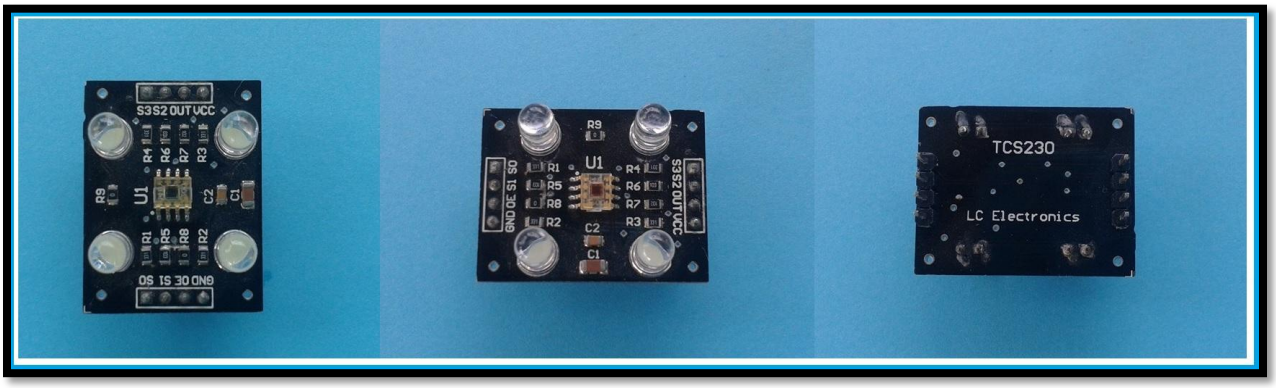
TCS230 COLOR RECOGNITION MODULE



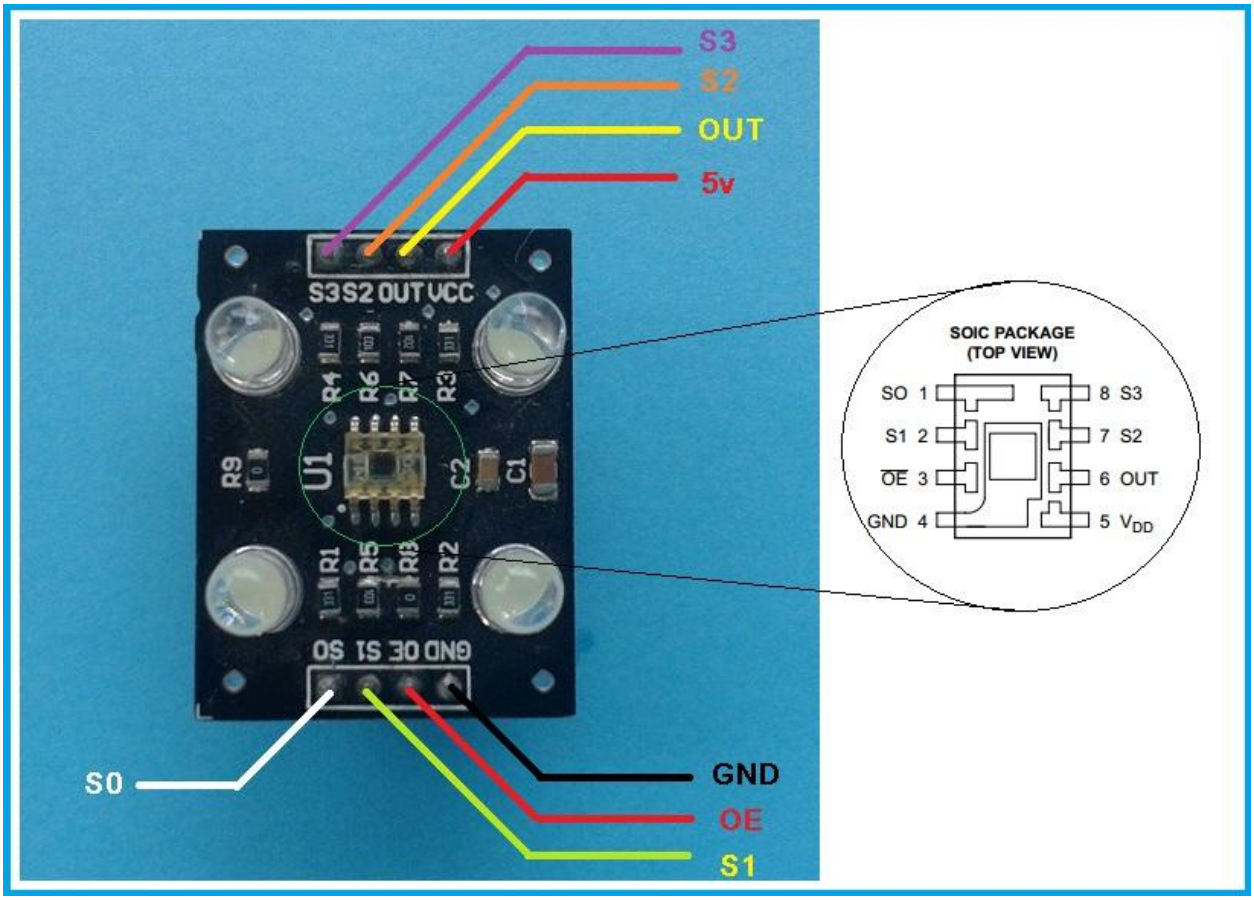
Product Description:

The TCS230 programmable color light-to-frequency converter combines configurable silicon photodiodes and a current-to-frequency converter on single monolithic CMOS integrated circuit. The output is a square wave (50% duty cycle) with frequency directly proportional to light intensity (irradiance). The full-scale output frequency can be scaled by one of three preset values via two control input pins. Digital inputs and digital output allow direct interface to a microcontroller or other logic circuitry. Output enable (OE) places the output in the high-impedance state for multiple-unit sharing of a microcontroller input line. The light-to-frequency converter reads an 8 x 8 array of photodiodes. Sixteen photodiodes have blue filters, 16 photodiodes have green filters, 16 photodiodes have red filters, and 16 photodiodes are clear with no filters. The four types (colors) of photodiodes are interdigitated to minimize the effect of non-uniformity of incident irradiance. All 16 photodiodes of the same color are connected in parallel and which type of photodiode the device uses during operation is pin-selectable. Photodiodes are 120 µm x 120 µm in size and are on 144-µm centers.

Product Specification:

- High-Resolution Conversion of Light Intensity to Frequency
- Programmable Color and Full-Scale Output Frequency
- Communicates Directly With a Microcontroller
- Single-Supply Operation (2.7 V to 5.5 V)
- Power Down Feature
- Nonlinearity Error Typically 0.2% at 50 kHz
- Stable 200 ppm/°C Temperature Coefficient
- Low-Profile Surface-Mount Package

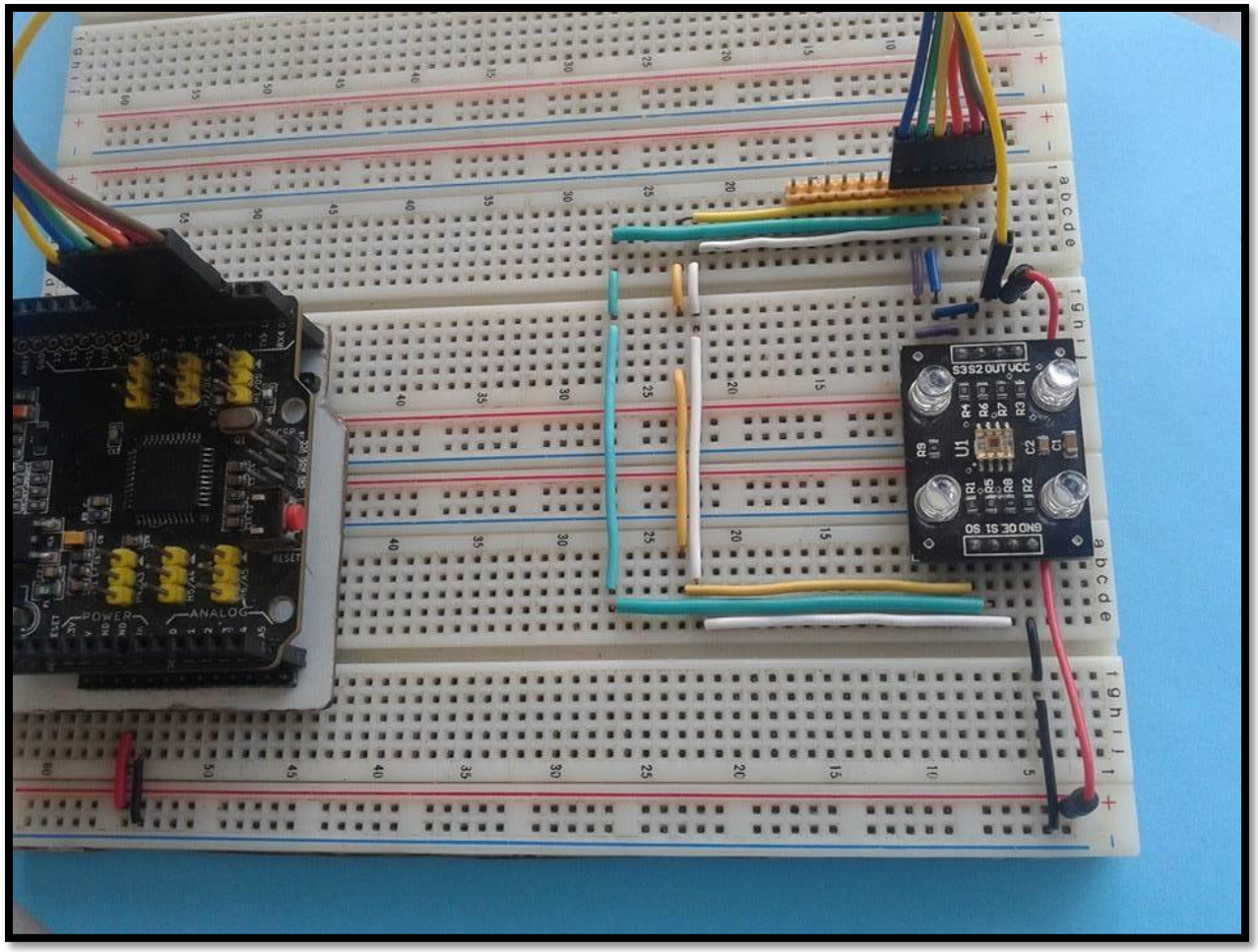
Pin Configuration and System Diagram:



How to test:

First connect the pins of the TCS230 module to the Arduino as indicated in the table:

TCS230 Module	Arduino
OE	Digital Pin 2
S0	Digital Pin 3
S1	Digital Pin 4
S2	Digital Pin 5
S3	Digital Pin 6
OUT	Digital Pin 8
VCC	5V
GND	GND



Wiring of the module to the Arduino

Here's a sample program that can detect 3 colors, Red, Green and Blue:

Sample Program:

```
//=========================================================================
//                                TCS230 CONNECTIONS
//=========================================================================
const int outputEnabled = 2;          // write LOW to turn on Note, may not be hooked up.
const int s0 = 3;                     // sensor pins
const int s1 = 4;
const int s2 = 5;
const int s3 = 6;
const int nLED = 7;                   // illuminating LED
const int out = 8;                    // TCS230 output
//=========================================================================
//                              Variables to store color values
//=========================================================================
int red = 0;
int green = 0;
int blue = 0;
//=========================================================================
void setup()
{
  pinMode(outputEnabled, OUTPUT);
  pinMode(s0, OUTPUT);
  pinMode(s1, OUTPUT);
  pinMode(s2, OUTPUT);
  pinMode(s3, OUTPUT);
  pinMode(nLED, OUTPUT);
  pinMode(out, INPUT);
  Serial.begin(9600);
```

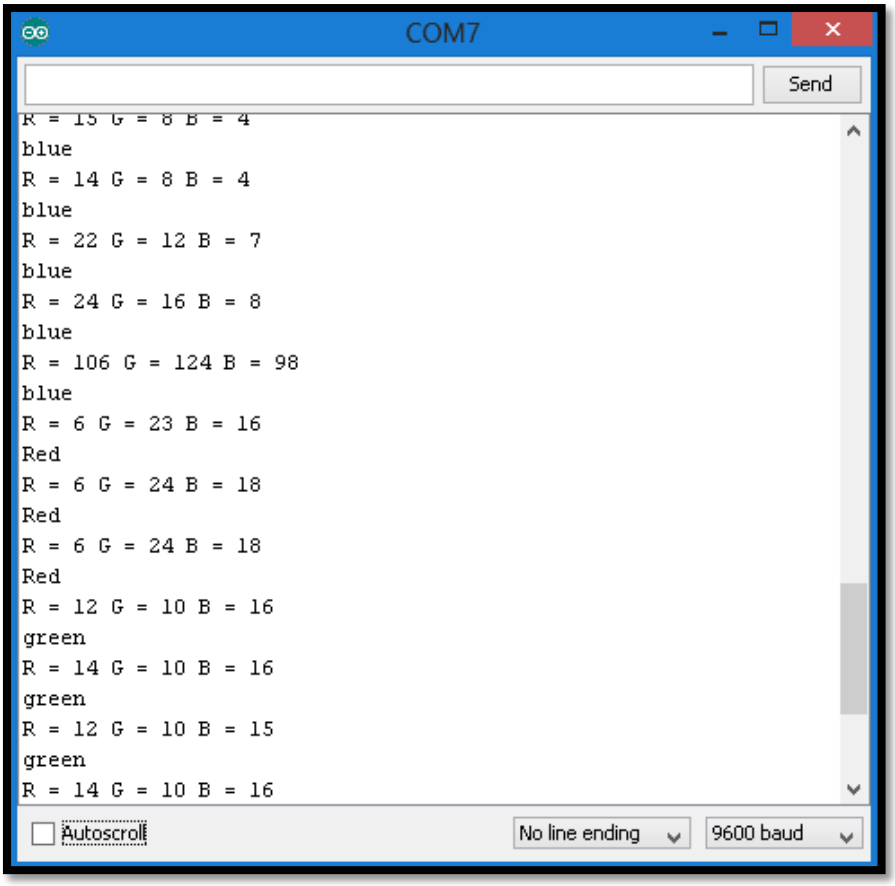
```
digitalWrite(outputEnabled, LOW);
                                                    //Set Frequency scaling to largest value
  digitalWrite(s0, HIGH);
  digitalWrite(s1, HIGH);
  digitalWrite(nLED, LOW);
}

void loop()
{
  color();
  Serial.print("R = ");
  Serial.print(red, DEC);
  Serial.print(" G = ");
  Serial.print(green, DEC);
  Serial.print(" B = ");
  Serial.print(blue, DEC);
  Serial.println();
                                                    //Simple logic to test for color
  if (red < blue && red < green)
  {
  Serial.println("Red");
  }
  else if (blue < red && blue < green)
  {
  Serial.println("blue");
  }
  else
  {
  Serial.println("green");
  }
```

```
delay(1000);
}

void color() {
  digitalWrite(s2, LOW);
  digitalWrite(s3, LOW);
                                        //count OUT, pRed, RED
  red = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
  digitalWrite(s3, HIGH);
                                        //count OUT, pBLUE, BLUE
  blue = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
  digitalWrite(s2, HIGH);
                                        // count OUT, pGreen, GREEN
  green = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
}
```

Result:



This sample code will display 3 values for Red, Green and Blue respectively, when a colored material is place on top of the sensor. This values will adjust according to the color of the material. The one with the lowest value will determine its color.