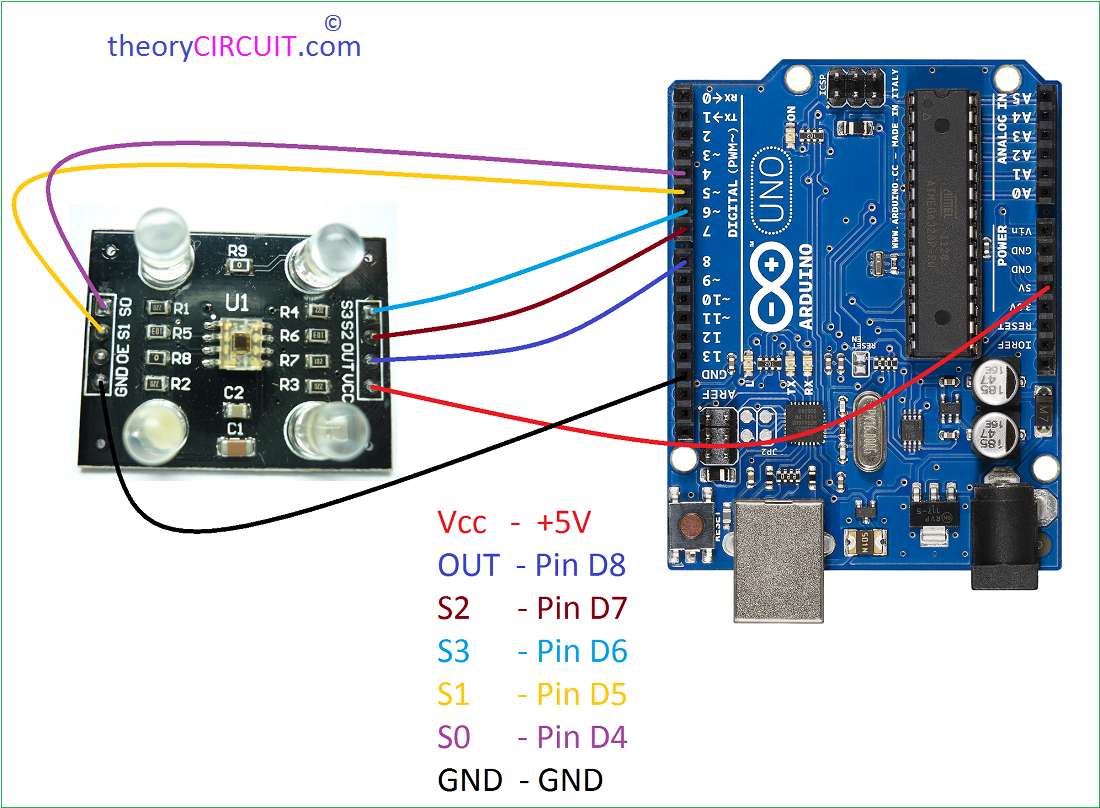
PROJECT

RGB Color Detector Using TCS3200 Sensor Module

This project is used for detecting primary colors (red, green and blue, or RGB)—colors that are physically available in LEDs in one package; for example, common cathode or common-cathode RGB LED. We can display primary colors and also generate specific colors by modifying the Arduino code. The project demonstrates the basic interfacing of TCS3200 sensor, Arduino Uno and common-cathode RGB LED.

Circuit and Working



PIN CONNECTIONS

VCC to 5V

GND to GND

s0 to 4

s1 to 5

s2 to 6

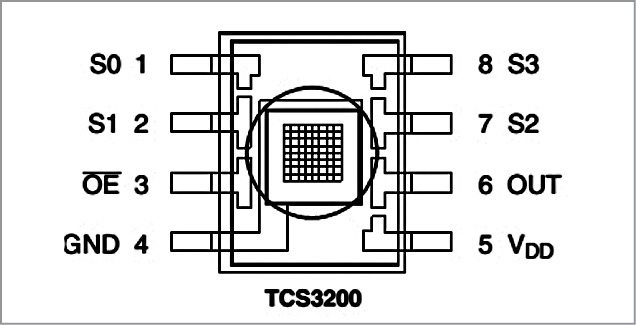
s3 to 7

OUT to 8

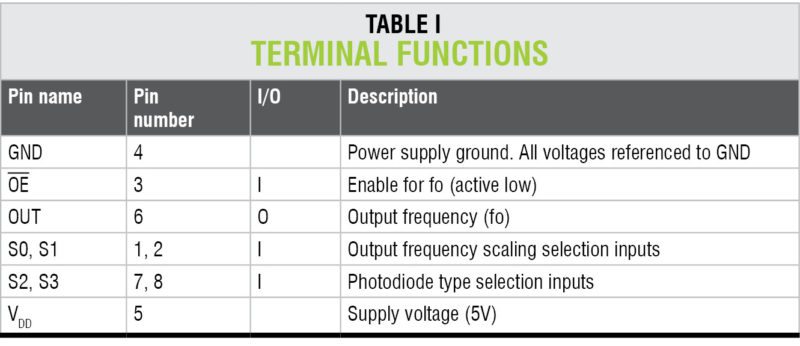
OE to GND

WORKING

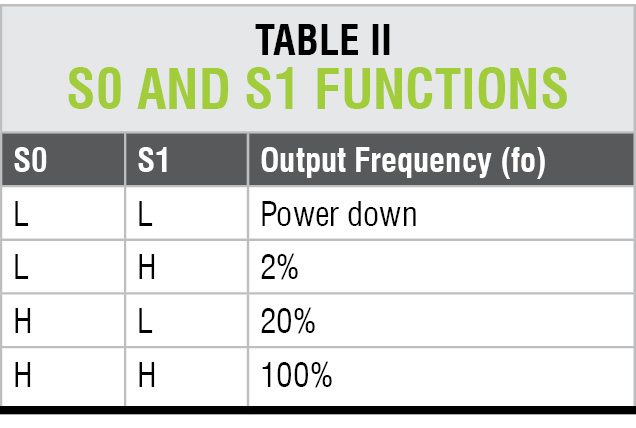
TCS3200 module has eight pins as shown in Fig. 4. This module consists of programmable color light-to-frequency converters that combine configurable silicon photodiodes and current-to-frequency converter on a single monolithic CMOS integrated circuit. Output is square-wave (50 per cent duty cycle) with frequency directly proportional to light intensity (irradiance).

Fig. 4: Pin diagram of the TCS3200 color sensor module

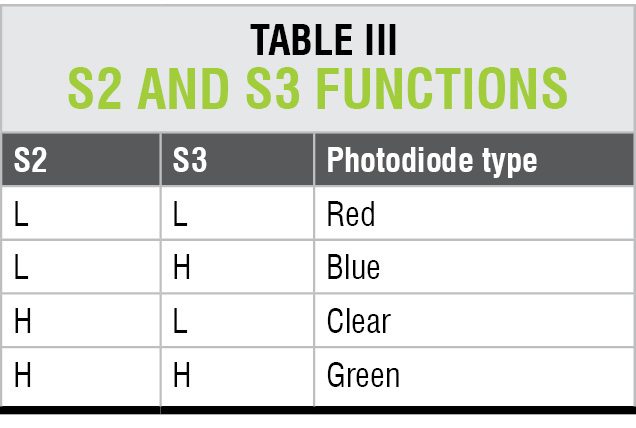
Digital inputs and outputs allow direct interface to the MCU or other logic circuitry. Output enable (OE) places the output in high-impedance state for multiple units sharing an MCU input line. In TCS3200, the light-to-frequency converter reads an 8×8 array of photodiodes. Sixteen photodiodes have blue filters, another sixteen have green, yet another sixteen have red and remaining sixteen are clear with no filters.



All photodiodes of the same color are connected in parallel. Pins S2 and S3 of TCS3200 are used to select the group of photodiodes (red, green, blue and clear) that are active. The detailed pin description is shown in Tables I, II and III, respectively.



Each sensor array in these three arrays is selected separately, depending on the requirement. Hence, it is known as a programmable sensor.



The module can be used to sense a particular color only. It contains filters for selection purpose. There is a fourth mode with no filter. With no filter, the sensor detects white light.

CODE

const int s0 = 4;

const int s1 = 5;

const int s2 = 6;

const int s3 = 7;

const int out = 8;

// LED pins connected to Arduino

int redLed = 2;

int greenLed = 3;

int blueLed = 4;

// Variables

int red = 0;

int green = 0;

int blue = 0;

void setup()

{

Serial.begin(9600);

pinMode(s0, OUTPUT);

pinMode(s1, OUTPUT);

pinMode(s2, OUTPUT);

pinMode(s3, OUTPUT);

pinMode(out, INPUT);

pinMode(redLed, OUTPUT);

pinMode(greenLed, OUTPUT);

pinMode(blueLed, OUTPUT);

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);

//Serial.println();

if (red < blue && red < green && red < 20)

{

Serial.println(" - (Red Color)");

digitalWrite(redLed, HIGH); // Turn RED LED ON

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); // Turn BLUE LED ON

}

else if (green < red && green < blue)

{

Serial.println(" - (Green Color)");

digitalWrite(redLed, LOW);

digitalWrite(greenLed, HIGH); // Turn GREEN LED ON

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);

//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);

}

