

User Manual

For

TCS 3200 Color Sensor (ME069)



Description:

TCS3200 Color Sensor is a complete color detector, including a TAOS TCS3200 RGB sensor chip and 4 white LEDs. The TCS3200 can detect and measure a nearly limitless range of visible colors. Applications include test strip reading, sorting by color, ambient light sensing and calibration, and color matching, to name just a few.

The TCS3200 has an array of photodetectors, each with either a red, green, or blue filter, or no filter (clear). The filters of each color are distributed evenly throughout the array to eliminate location bias among the colors. Internal to the device is an oscillator which produces a square-wave output whose frequency is proportional to the intensity of the chosen color.

Specification

- Single-Supply Operation (2.7V to 5.5V)
- High-Resolution Conversion of Light Intensity to Frequency
- Programmable Color and Full-Scale Output Frequency
- Power Down Feature
- Communicates Directly to Microcontroller
- S0~S1: Output frequency scaling selection inputs
- S2~S3: Photodiode type selection inputs
- OUT Pin: Output frequency
- Support LED lamp light supplement control
- Size: 28.4x28.4mm

PinOut

Pin	Description
GND	Power ground
OUT	Output frequency (fo).
S0	Output frequency scaling selection inputs.
S1	Output frequency scaling selection inputs.
S2	Photodiode type selection inputs
S3	Photodiode type selection inputs
Vcc	Power supply 5V/DC

Example:

Wire connection as below:

```
Vcc-----5V
GND-----GND
S0-----D3
S1-----D4
S2-----D5
S3-----D6
OUT-----D2
```

```
*****Code Begin*****
#include <TimerOne.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,20,4);

#define S0    6
#define S1    5

#define S2    4
#define S3    3

#define OUT   2
// #define LED 7

int  g_count = 0;    // count the frequency
int  g_array[3];    // store the RGB value
int  g_flag = 0;    // filter of RGB queue
float g_SF[3];      // save the RGB Scale factor
// Init TSC230 and setting Frequency.

void TSC_Init()
{
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  pinMode(OUT, INPUT);
  digitalWrite(S0, LOW); // OUTPUT FREQUENCY SCALING 2%
  digitalWrite(S1, HIGH);
}
```

```
// Select the filter color//
void TSC_FilterColor(int Level01, int Level02)
{
  if(Level01 != 0)
    Level01 = HIGH;

  if(Level02 != 0)
    Level02 = HIGH;

  digitalWrite(S2, Level01);
  digitalWrite(S3, Level02);
}

void TSC_Count()
{
  g_count ++ ;
}

void TSC_Callback()
{
  switch(g_flag)
  {
    case 0:
      Serial.println("->WB Start");
      TSC_WB(LOW, LOW);
      break;
    case 1:
      Serial.print("->Frequency R=");
      //lcd.setCursor(0,0);
      //lcd.print("Start");
      Serial.println(g_count);
      g_array[0] = g_count;
      TSC_WB(HIGH, HIGH);
      break;
    case 2:
      Serial.print("->Frequency G=");
      Serial.println(g_count);
      g_array[1] = g_count;
      TSC_WB(LOW, HIGH);
      break;
    case 3:
      Serial.print("->Frequency B=");
      Serial.println(g_count);
```

IDUINO for maker's life

```
        Serial.println("->WB End");
        g_array[2] = g_count;
        TSC_WB(HIGH, LOW);
        break;
    default:
        g_count = 0;
        break;
    }
}

void TSC_WB(int Level0, int Level1) //White Balance
{
    g_count = 0;
    g_flag ++;
    TSC_FilterColor(Level0, Level1);
    Timer1.setPeriod(1000000);
}

void setup()
{
    TSC_Init();
    lcd.init();
    delay(100);
    lcd.backlight();
    Wire.begin();
    delay(100);
    lcd.setCursor(14,0);
    lcd.print("Color");
    lcd.setCursor(0,3);
    lcd.print("S0:2 S1:3 S2:4 S3:5 OUT:6 LED:-");
    Serial.begin(9600);
    Timer1.initialize();           // defaulte is 1s
    Timer1.attachInterrupt(TSC_Callback);
    attachInterrupt(0, TSC_Count, RISING);
    delay(4000);
    for(int i=0; i<3; i++)
        Serial.println(g_array[i]);
    g_SF[0] = 255.0/ g_array[0];    //R Scale factor
    g_SF[1] = 255.0/ g_array[1] ;  //G Scale factor
    g_SF[2] = 255.0/ g_array[2] ;  //B Scale factor

    Serial.println(g_SF[0]);
    Serial.println(g_SF[1]);
    Serial.println(g_SF[2]);
}
```

```
//for(int i=0; i<3; i++)
// Serial.println(int(g_array[i] * g_SF[i]));
}
void loop()
{
  g_flag = 0;
  for(int i=0; i<3; i++)

  {
    Serial.println(int(g_array[i] * g_SF[i]));
    //lcd.setCursor(0,1);
    //lcd.print(int(g_array[i] * g_SF[i]));
  }
  lcd.setCursor(0,1);
  lcd.print(int(g_array[0] * g_SF[0]));
  lcd.setCursor(6,1);
  lcd.print(int(g_array[1] * g_SF[1]));
  lcd.setCursor(12,1);
  lcd.print(int(g_array[2] * g_SF[2]));
  delay(4000);
  Clean2004();
}

void Clean2004()
{
  lcd.setCursor(0,1);
  lcd.print("                ");
  lcd.setCursor(0,2);
  lcd.print("                ");
}
*****Code End*****
```