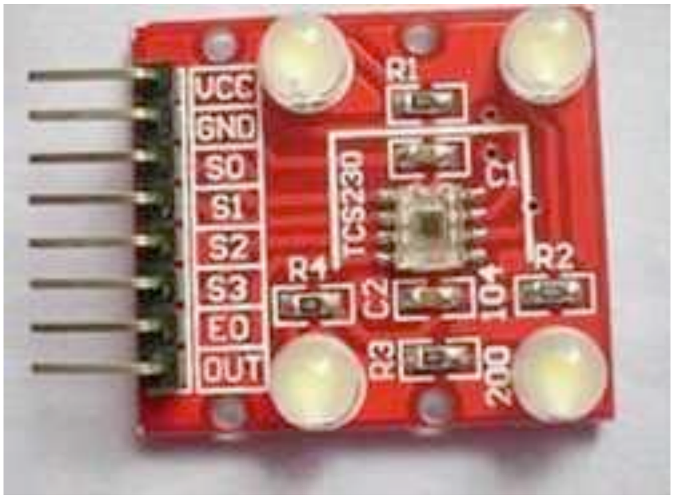
Future electronics

Color Sensor TSC230 Principle and Arduino Code



This Color Sensor is a complete color detector, including a TCS230 RGB sensor chip and 4 white LEDs. The TCS230 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.

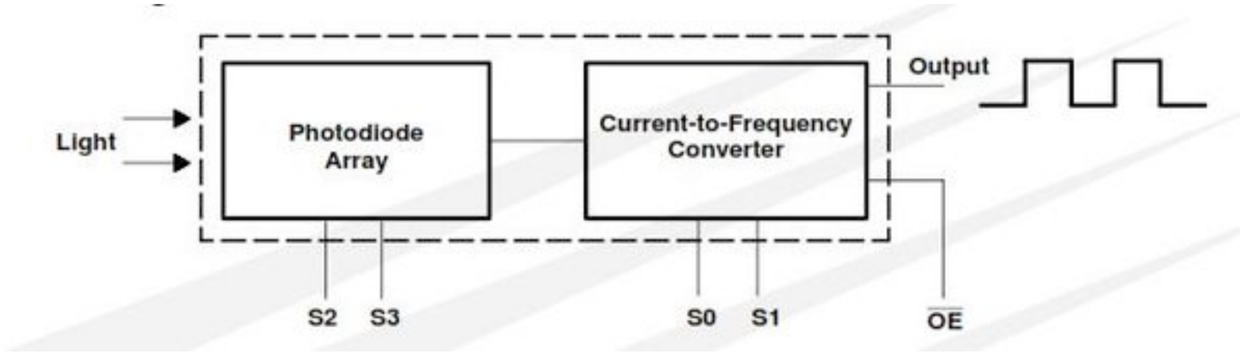
The TCS230 has an array of photodetectors, each with either a red, green, or blue filter, or no filter (clear). The filters of each color are distributed evenly throughout the array to eliminate location bias among the colors.

Internal to the device is an oscillator which produces a square-wave output whose frequency is proportional to the intensity of the chosen color.

Specifications

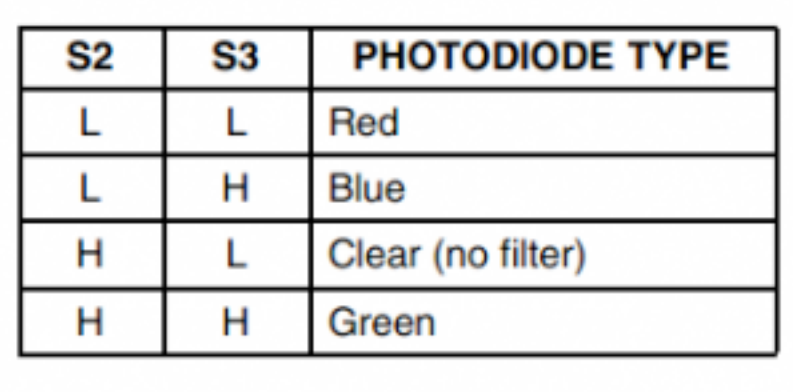
Single-Supply Operation (2.7V to 5.5V)
High-Resolution Conversion of Light Intensity to Frequency
Programmable Color and Full-Scale Output Frequency
Power Down Feature
Communicates Directly to Microcontroller/Arduino
S0~S1: Output frequency scaling selection inputs
S2~S3: Photodiode type selection inputs
OUT Pin: Output frequency
EO Pin: Output frequency enable pin (active low)

Recognition color principle of TCS230 sensor



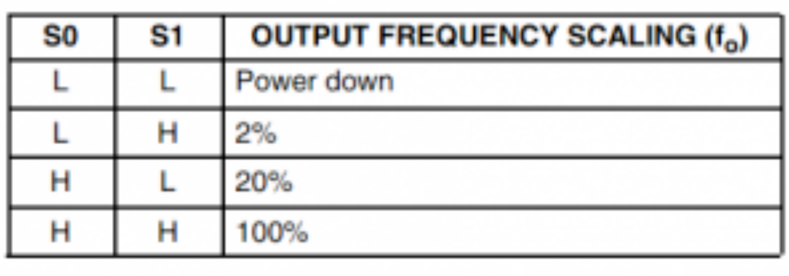
When choosing color filter, the TCS230 can allow only one particular color to get through and prevent other color. For example, when choosing the red filter, only red incident light can get through, blue and green will be prevented. So we can get the red light intensity. Similarly ,when choose other filters we can get blue or green light.

TCS230 has four photodiode types. Red , blue, green and clear, reducing the amplitude of the incident light uniformity greatly, so that to increase the accuracy and simplify the optical. When the light project to the TCS230 we can choose the different type of photodiode by different combinations of S2 and S3. Look at the form as follows.



S2	S3	PHOTODIODE TYPE
L	L	Red
L	H	Blue
H	L	Clear (no filter)
H	H	Green

TCS230 can output the frequency of different square wave. Different color and light intensity correspond with different frequency of square wave. There is a relationship between the output and light intensity. The range of the typical output frequency is 2HZ~500KHZ. We can get different scaling factor by different combinations of S0 and S1. Look at the form as follows.



S0	S1	OUTPUT FREQUENCY SCALING (f_o)
L	L	Power down
L	H	2%
H	L	20%
H	H	100%

Arduino Code and Connection

```
int s0=3,s1=4,s2=5,s3=6;
int out=2;
int flag=0;
byte counter=0;
byte countR=0,countG=0,countB=0;
void setup()
 {
 Serial.begin(115200);
 pinMode(s0,OUTPUT);
 pinMode(s1,OUTPUT);
 pinMode(s2,OUTPUT);
 pinMode(s3,OUTPUT);
 }
void TCS()
 {
 flag=0;
 digitalWrite(s1,HIGH);
 digitalWrite(s0,HIGH);
 digitalWrite(s2,LOW);
 digitalWrite(s3,LOW);
 attachInterrupt(0, ISR_INTO, LOW);
 timer0_init();

 }
void ISR_INTO()
 {
 counter++;
 }
 void timer0_init(void)
 {
  TCCR2A=0x00;
  TCCR2B=0x07;   //the clock frequency source 1024 points
  TCNT2= 100;    //10 ms overflow again
  TIMSK2 = 0x01; //allow interrupt
 }
 int i=0;
 ISR(TIMER2_OVF_vect)//the timer 2, 10ms interrupt overflow again.
Internal overflow interrupt executive function
{
    TCNT2=100;
    flag++;
 if(flag==1)
  {
    countR=counter;
    Serial.print("red=");
    Serial.println(countR,DEC);
    digitalWrite(s2,HIGH);
    digitalWrite(s3,HIGH);
  }
  else if(flag==2)
   {
    countG=counter;
    Serial.print("green=");
    Serial.println(countG,DEC);
    digitalWrite(s2,LOW);
    digitalWrite(s3,HIGH);
   }
   else if(flag==3)
    {
    countB=counter;
    Serial.print("blue=");
    Serial.println(countB,DEC);
    Serial.println("\n");
    digitalWrite(s2,LOW);
    digitalWrite(s3,LOW);

    }
    else if(flag==4)
     {
     flag=0;
     }
       counter=0;
}
void loop()
 {
  TCS();
while(1);
 }
```