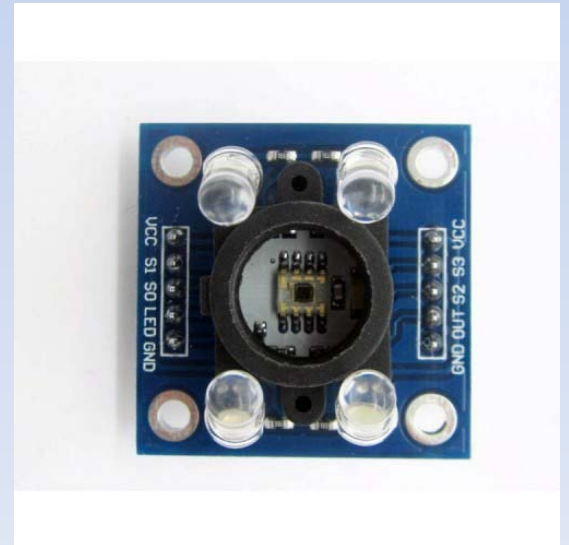


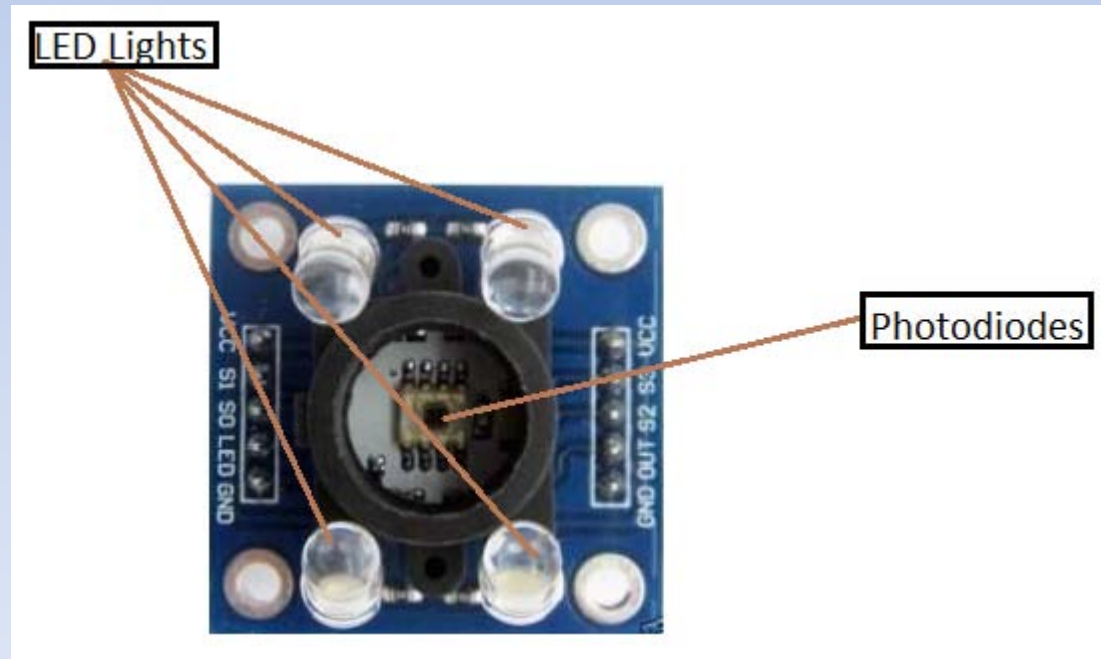
TCS230

Sensore di colore



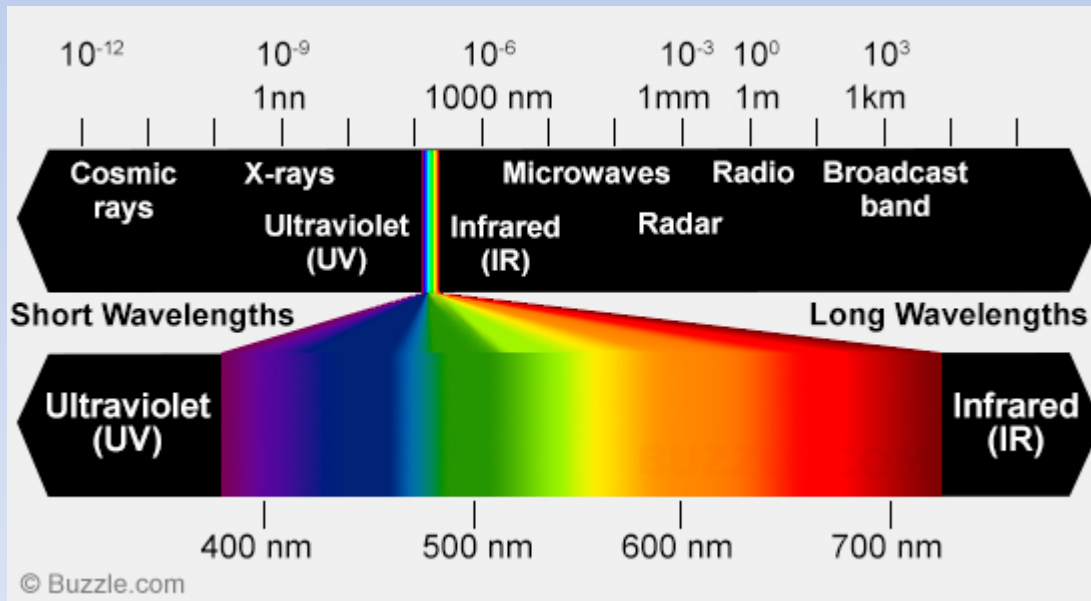
Principio di funzionamento

- Il dispositivo è formato da una matrice 8x8 di fotodiodi con filtri di colori così composti:
 - 16 fotodiodi con filtro rosso
 - 16 fotodiodi con filtro verde
 - 16 fotodiodi con filtro blu
 - 16 fotodiodi senza filtriI fotodiodi di ciascun gruppo sono collegati tra loro in parallelo. I led illuminano l'oggetto e la Luce riflessa colpisce i fotorilevatori



Principio di funzionamento

- Quando viene illuminato un oggetto con luce bianca, l'oggetto assorbe le frequenze luminose in base alla sua particolare forma o materiale e riflette il resto delle frequenze luminose
- Il risultato è il colore che viene percepito dal nostro occhio ed è dato dalla sovrapposizione delle varie frequenze.



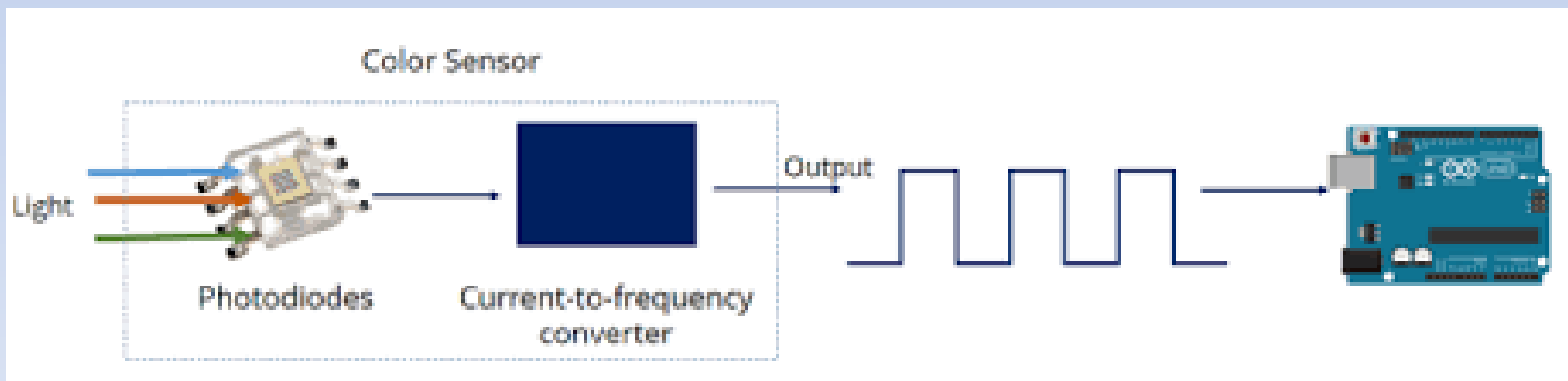
Principio di funzionamento

- Ad ogni colore è associata una frequenza del segnale luminoso
- Esistono colori primari e colori secondari
- I colori secondari sono dati dalla sovrapposizione dei colori primari: rosso, verde, blu
- I colori primari non sono dati dalla sovrapposizione di altri



Principio di funzionamento

- Un convertitore corrente-frequenza trasforma l'intensità di corrente in segnale ad onda quadra con frequenza direttamente proporzionale all'intensità di corrente



Pin

- Tutti i fotodiodi dello stesso colore sono collegati in blocchi di 16
- I quattro pin vengono posti come OUTPUT nel programma di Arduino ma, c'è un quinto pin che viene posto come INPUT ed è quello che porta le informazioni al microcontrollore
- Due pin S2 ed S3 sono di controllo ed in base ai loro livelli logici, si può scegliere quali colori rilevare

S2	S3	colore
L	L	Rosso
L	H	Blu
H	L	Nessun filtro
H	H	Verde

Pin

- S0 ed S1 servono a scalare la frequenza
- La tabella per scalare la frequenza è la seguente
- Il settaggio dei pin viene fatto nel setup

S0	S1	frequenza
L	L	Nessuna potenza
L	H	2%
H	L	20%
H	H	100%

Esempio

```
#define S0 4
#define S1 5
#define S2 6
#define S3 7
#define sensorOut 8
int redFrequency = 0;
int greenFrequency = 0;
int blueFrequency = 0;

void setup() {
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);

  pinMode(sensorOut, INPUT);

  // Setting frequency scaling to 20%
  digitalWrite(S0,HIGH);
  digitalWrite(S1,LOW);
  Serial.begin(9600);
}

void loop() {
  digitalWrite(S2,LOW);
  digitalWrite(S3,LOW); //per rilevare il rosso
  redFrequency = pulseIn(sensorOut, LOW);
  Serial.print("R = ");
  Serial.print(redFrequency);
  delay(100);
  digitalWrite(S2,HIGH);
  digitalWrite(S3,HIGH);//per rilevare il verde
  greenFrequency = pulseIn(sensorOut, LOW);
  Serial.print(" G = ");
  Serial.print(greenFrequency);
  delay(100);
  digitalWrite(S2,LOW);
  digitalWrite(S3,HIGH);//per rilevare il blu
  blueFrequency = pulseIn(sensorOut, LOW);
  Serial.print(" B = ");
  Serial.println(blueFrequency);
  delay(100);
}
```


Il colore letto viene riprodotto da un RGB con arduino

```
#define S0 4
#define S1 5
#define S2 6
#define S3 7
#define sensorOut 8
#define DELAY 1000
int redFrequency = 0;
int greenFrequency = 0;
int blueFrequency = 0;
int redColor = 0;
int greenColor = 0;
int blueColor = 0;
void setup() {
  pinMode(3, OUTPUT);//B
  pinMode(10, OUTPUT);//G
  pinMode(11,OUTPUT);//R
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  pinMode(sensorOut, INPUT);
  digitalWrite(S0,HIGH);
  digitalWrite(S1,LOW);
  Serial.begin(9600);
}
```

```
void loop() {
  digitalWrite(S2,LOW);
  digitalWrite(S3,LOW);
  redFrequency = pulseIn(sensorOut, LOW);
  redColor = map(redFrequency, 70, 120, 255,0);
  Serial.print(" R = ");
  Serial.print(redColor);
  delay(100);
  digitalWrite(S2,HIGH);
  digitalWrite(S3,HIGH);
  greenFrequency = pulseIn(sensorOut, LOW);
  greenColor = map(greenFrequency, 100,199, 255, 0);
  Serial.print(" G = ");
  Serial.print(greenColor);
  delay(100);
  digitalWrite(S2,LOW);
  digitalWrite(S3,HIGH);
  blueFrequency = pulseIn(sensorOut, LOW);
  blueColor = map(blueFrequency, 38, 84, 255, 0);
  Serial.print(" B = ");
  Serial.print(blueColor);
  delay(100);
  analogWrite(3,blueColor);
  analogWrite(6,greenColor);
  analogWrite(9,redColor);
  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!");}
```

Il colore letto viene riprodotto da un RGB con stm32

```
#include "mbed.h"
#include <Timer.h>
#define COLOR_SIZE 5
Serial command(USBTX, USBRX);
DigitalIn button(USER_BUTTON);
DigitalOut led(LED1);
DigitalIn OUT(D3);//A5 BC1
DigitalOut S0(D4);//A1 VD1
DigitalOut S1(D5);//A2 NG
DigitalOut S2(D6);//A4 BC2
DigitalOut S3(D7);//A3 VD2
PwmOut mb(D8);
PwmOut mg(D9);
PwmOut mr(D10);
unsigned int lectura(void);
void borrar(void);
Timer tiempo;
    int Rojo = 0;
    int Azul = 0;
    int Verde = 0;
    int Amarillo = 0;
    int Negro = 0;
    int Blanco = 0;
int main()
{
    unsigned int Red = 0;
    unsigned int Blue = 0;
    unsigned int Green = 0;
    while(1){
        S0 = 1;
        S1 = 1;
        //Lectura del rosso
        S2 = 0;
        S3 = 0;
        Red = lectura();
        //Lectura del blue
        S2 = 0;
        S3 = 1;
        Blue = lectura();
        //Lectura del verde
        S2 = 1;
        S3 = 1;
        Green = lectura();
        command.printf("Red: %d, Green: %d, Blue: %d ",Red,Green,Blue);
        wait(0.5);
        mr.period_ms(25); //dalla lettura della frequenza di ogni colore

        mr.write(10/Red);
        mg.period_ms(50);
        mg.write(10/Green);
        mb.period_ms(30);
        mb.write(10/Blue);}

    unsigned int lectura(void){
        unsigned int Resultado = 0;
        while(OUT){}
        while(!OUT){}
        while(OUT){}
        tiempo.start();
        while(!OUT){}
        tiempo.stop();
        Resultado = tiempo.read_us();
        tiempo.reset();
        return(Resultado);
    }
    void borrar(void){
        Rojo = 0;
        Azul = 0;
        Verde = 0;
        Amarillo = 0;
        Negro = 0;
        Blanco = 0;
        return;
    }
}
```