

- Balașanu Ionuț
- Lupu Vasile-Ciprian
- Gurzun Andrei-Răzvan

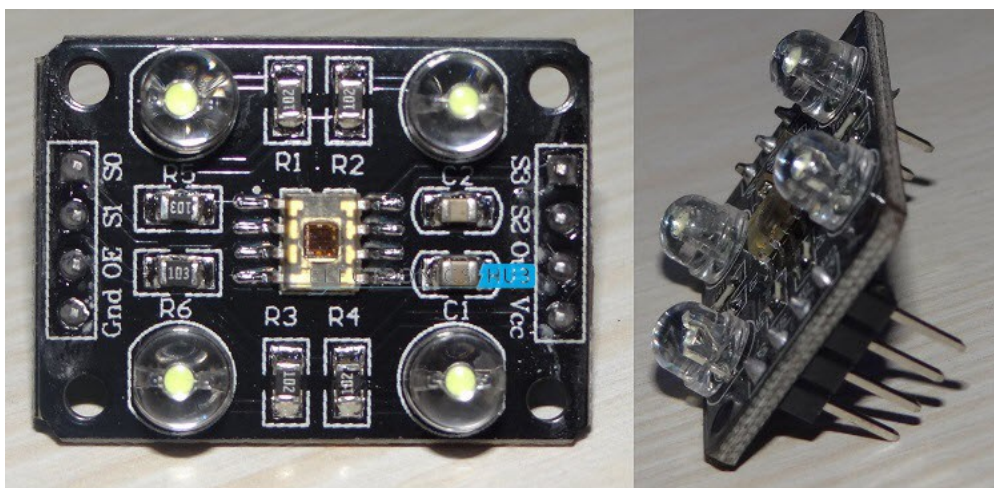
TEMA: **Color Detector**

Rezumat:

Pentru aprofundarea și explorarea noastră, a plăcuței Raspberry Pi model 3B+ vom realiza un proiect în scopul detecției de culoare. În acest scop vom aduna cunoștințe și despre senzorul de culori TCS 3200/230. Așadar, ne propunem implementarea unui detector de culori prin utilizarea celor 2 piese de bază.

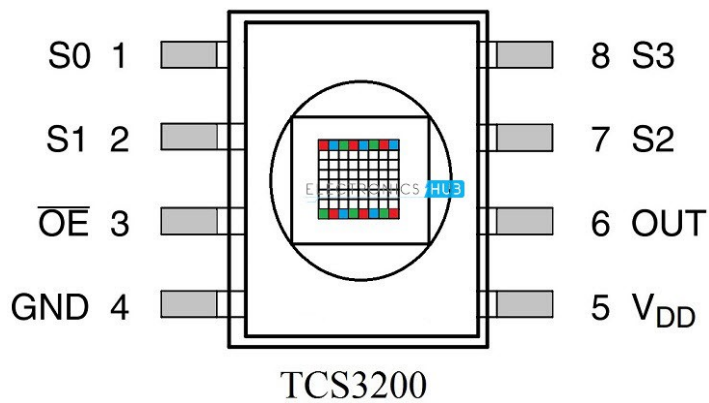
Senzorul de culoare detectează sau simte culoarea, simplist la prima vedere, însă modul în care detectează culori este parte care complică lucrurile. Există aplicații precum detectarea obiectelor, sortarea produselor, urmărirea unui obiect în care detecția de culoare joacă un rol esențial. În acest proces, senzorul și placa hardware sunt cele două componente în jurul cărora se dezvoltă proiectul final.

Există mai multe tipuri de senzori compatibili cu Raspberry dar, noi vom folosi și vom învăța TCS 230 de la TAOS (Texas Advanced Optoelectronic Solutions). În continuare vom vorbi despre elemente de bază și funcționalitatea senzorului TCS.



TCS 230 este unul dintre cei mai la îndemână senzori disponibili pe piață, pe care studenții și pasionații îl folosesc. În esență, este un convertor de lumină la frecvență, adică pe baza culorii și a intensității luminii care cad pe el, frecvența semnalului de intrare sau de ieșire variază.

Diagrama PIN pentru senzor:



Descrierea pinilor:

S0 si S1: S0 si S1 sunt intrări de scalare/masurare a frecventei semnalului de ieşire. Utilizandu-i putem scala frecventa de iesire la una dintre cele trei valori prestabilite. Tabelul corespunzator lui S0 si S1 este dat in sectiunea urmatoare.

OE: Pinul de activare a iesirii. Este un pin activ pe LOW.

GND: Pin de alimentare electrică.

VDD: pin de alimentare (de obicei, + 5V).

OUT: Pin de iesire, care produce un obiect patrat cu un ciclu de functionare de 50%, iar frecventa undei patrate este proportionala cu intensitatea luminii.

S2 și S3: S2 și S3 sunt pinii de selectare a foto-diodei .

Tabel S0 si S1:(diferite combinatii intre S0 si S1 pentru scalarea corespunzatoare a frecventei de iesire)

S ₀	S ₁	Output Frequency Scaling (f _o)	Typical full-scale Frequency
L	L	Power Down	-----
L	H	2%	10 - 12 KHz
H	L	20%	100 - 120 KHz
H	H	100%	500 - 600 KHz

Tabel S2 si S3:

S ₂	S ₃	Photodiode Type
L	L	Red
L	H	Blue
H	L	Clear (no filter)
H	H	Green

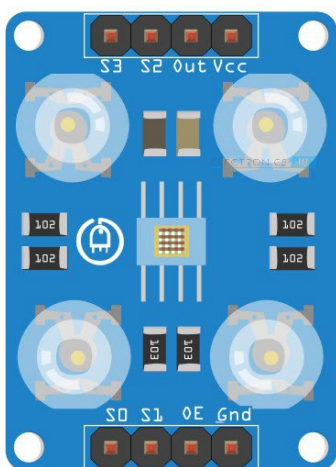
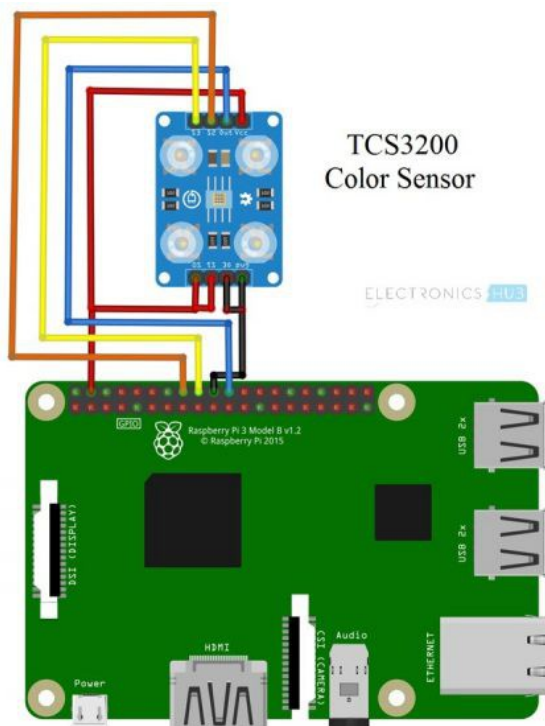
Resurse materiale:

- >Raspberry Pi model 3B+
- >TCS230 senzor culoare
- > led RGB
- > 3 rezistente de 1kΩ
- > Condensator de 1000uF
- >Bradboard

Raport sintetic:

Pentru a demonstra detectarea culorilor, vom folosi un LED RGB. Acest LED RGB va străluci în aceeași culoare, a cărui obiect este prezent în apropierea senzorului.

Circuitul conexiunii dintre Raspberry si TCS



Asa cum se observa si in imaginile de mai sus, senzorul TCS are 8 pini.
Conexiunea se face in felul urmatoar:

S1 si S0 la +5V, GND-ul sensorului la GND-ul placutei.

Vdd-ul sensorului la +5V, OUT la GPIO25(pin 22) al Raspberry-ului.

S2 si S3 la GPIO23 si GPIO24 (16 si 18).

Avand conexiunea corecta putem inainta, incarcand mai intai codul pentru calibrarea culorilor. Prin acest cod vom obtine valorile de prag pentru RGB.

```
import RPi.GPIO as
GPIO
import time

s2 = 23
s3 = 24
signal = 25
NUM_CYCLES = 10

def setup():
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(signal, GPIO.IN,
pull_up_down=GPIO.PUD_UP)
    GPIO.setup(s2, GPIO.OUT)
    GPIO.setup(s3, GPIO.OUT)
    print("\n")

def loop():
    temp = 1
    while(1):

        GPIO.output(s2, GPIO.LOW)
        GPIO.output(s3, GPIO.LOW)
        time.sleep(0.3)
        start = time.time()
        for impulse_count in range(NUM_CYCLES):
            GPIO.wait_for_edge(signal, GPIO.FALLING)
```

```

        duration = time.time() - start          #seconds to
run for loop
        red = NUM_CYCLES / duration           #in Hz
        print("red value - ",red)

        GPIO.output(s2,GPIO.LOW)
        GPIO.output(s3,GPIO.HIGH)
        time.sleep(0.3)
        start = time.time()
        for impulse_count in range(NUM_CYCLES):
            GPIO.wait_for_edge(signal, GPIO.FALLING)
            duration = time.time() - start
            blue = NUM_CYCLES / duration
            print("blue value - ",blue)

        GPIO.output(s2,GPIO.HIGH)
        GPIO.output(s3,GPIO.HIGH)
        time.sleep(0.3)
        start = time.time()
        for impulse_count in range(NUM_CYCLES):
            GPIO.wait_for_edge(signal, GPIO.FALLING)
            duration = time.time() - start
            green = NUM_CYCLES / duration
            print("green value - ",green)
            time.sleep(2)

def endprogram():
    GPIO.cleanup()

if __name__ == '__main__':

    setup()

    try:
        loop()

    except KeyboardInterrupt:
        endprogram()

```

Odata aflate valorile, putem construi codul pentru detectia culorilor. Am implementat un cod pentru culorile rosu, verde si albastru, bineinteles, codul se poate extinde si pentru celelalte culori.

```
import RPi.GPIO as GPIO
```

```
import time
```

```
s2 = 23
```

```
s3 = 24
```

```
l_r = 22
```

```
l_b = 27
```

```
l_g = 17
```

```
signal = 25
```

```
NUM_CYCLES = 10
```

```
def setup():
```

```
    GPIO.setmode(GPIO.BCM)
```

```
    GPIO.setup(signal,GPIO.IN, pull_up_down=GPIO.PUD_UP)
```

```
    GPIO.setup(s2,GPIO.OUT)
```

```
    GPIO.setup(s3,GPIO.OUT)
```

```
    GPIO.setup(l_r,GPIO.OUT)
```

```
    GPIO.output(l_r,1)
```

```
    GPIO.setup(l_g,GPIO.OUT)
```

```
    GPIO.output(l_g,1)
```

```
    GPIO.setup(l_b,GPIO.OUT)
```

```
    GPIO.output(l_b,1)
```

```
    print("\n")
```

```
def ready():
```

```
    time.sleep(1.25)
```

```
    GPIO.output(l_r,0)
```

```
    GPIO.output(l_g,1)
```

```
    GPIO.output(l_b,1)
```

```
    time.sleep(1.25)
```

```
    GPIO.output(l_r,1)
```

```
    GPIO.output(l_g,0)
```

```
    GPIO.output(l_b,1)
```

```
time.sleep(1.25)
GPIO.output(l_r,1)
GPIO.output(l_g,1)
GPIO.output(l_b,0)
time.sleep(1.25)
GPIO.output(l_r,1)
GPIO.output(l_g,1)
GPIO.output(l_b,1)
```

```
def loop():
```

```
    temp = 1
```

```
    while(1):
```

```
        GPIO.output(s2,GPIO.LOW)
        GPIO.output(s3,GPIO.LOW)
        time.sleep(0.3)
        start = time.time()
        for impulse_count in range(NUM_CYCLES):
            GPIO.wait_for_edge(signal, GPIO.FALLING)
            duration = time.time() - start
            red = NUM_CYCLES / duration
```

```
        GPIO.output(s2,GPIO.LOW)
        GPIO.output(s3,GPIO.HIGH)
        time.sleep(0.3)
        start = time.time()
        for impulse_count in range(NUM_CYCLES):
            GPIO.wait_for_edge(signal, GPIO.FALLING)
            duration = time.time() - start
            blue = NUM_CYCLES / duration
```

```
GPIO.output(s2,GPIO.HIGH)
GPIO.output(s3,GPIO.HIGH)
time.sleep(0.3)
start = time.time()
for impulse_count in range(NUM_CYCLES):
    GPIO.wait_for_edge(signal, GPIO.FALLING)
duration = time.time() - start
green = NUM_CYCLES / duration
```

```
if green<8000 and blue<10000 and red>12000:
```

```
    print("red")
```

```
    GPIO.output(l_r,0)
```

```
    GPIO.output(l_g,1)
```

```
    GPIO.output(l_b,1)
```

```
    temp=1
```

```
elif red<12000 and blue<12000 and green>12000:
```

```
    print("green")
```

```
    GPIO.output(l_r,1)
```

```
    GPIO.output(l_g,0)
```

```
    GPIO.output(l_b,1)
```

```
    temp=1
```

```
elif green<12000 and red<8000 and blue>12000:
```

```
    print("blue")
```

```
    GPIO.output(l_r,1)
```

```
    GPIO.output(l_g,1)
```

```
    GPIO.output(l_b,0)
```

```
    temp=1
```

```
elif red>5000 and green>5000 and blue>5000 and temp==1:
```

```
    print("place the object.....")
```

```
    GPIO.output(l_r,1)
```

```
    GPIO.output(l_g,1)
```



```

GPIO.output(l_b,1)
temp=0

def endprogram():
    GPIO.cleanup()

if __name__=='__main__':

    setup()
    ready()
    try:
        loop()

    except KeyboardInterrupt:
        endprogram()

```

Scopul acestui proiect simplu este de a intelege interfata Raspberry Pi si cum putem face o aplicatie de detectare a culorilor folosind Raspberry Pi și TCS230.

Valorile de referinta, din cel de-al doilea script Python sunt dependente de iluminatul din jur. Deci, ei ar putea sa nu functioneze perfect in anumite incaperi.

Utilizare practica:

- Control RGB cu iluminare LED
- Verificarea culorii obiectului
- Sortarea produselor
- Automatizare industrială
- Imprimare comercială
- Sanatate si fitness