



**UNIVERSITATEA
TEHNICĂ
DIN CLUJ-NAPOCA**

Detector de Culori cu Modul Bluetooth

Proiectare cu Microprocesoare

Student: Goron Gesica

Grupa: 30235

FACULTATEA DE AUTOMATICA
ȘI CALCULATOARE

26 Decembrie 2023

Cuprins

1	Introducere	2
1.1	Scopul proiectului	2
1.2	Componente utilizate	2
2	Componente și Materiale	2
3	Montaj Hardware	3
3.1	Diagrama de Circuit	3
3.2	Descriere	3
4	Funcționalități	3
4.1	Descriere cod	3
4.2	Solutia propusa	4
5	Interfața Bluetooth	6
6	Exemple de Utilizare	6
6.1	Arduino IDE	6
6.2	Detector	6
6.3	Bluetooth	6
7	Bibliografie	6

1 Introducere

1.1 Scopul proiectului

Proiectul constă în dezvoltarea unui detector de culori utilizând o placă Arduino, un senzor de culoare și un modul Bluetooth pentru a identifica și comunica culorile detectate către un dispozitiv mobil.

1.2 Componente utilizate

- Placă Arduino Uno
- Senzor de culoare TCS3200
- Modul Bluetooth HC-05
- Telefon mobil cu Android
- Breadboard
- Fire de conexiune
- 3 Leduri
- Rezistențe

2 Componente și Materiale

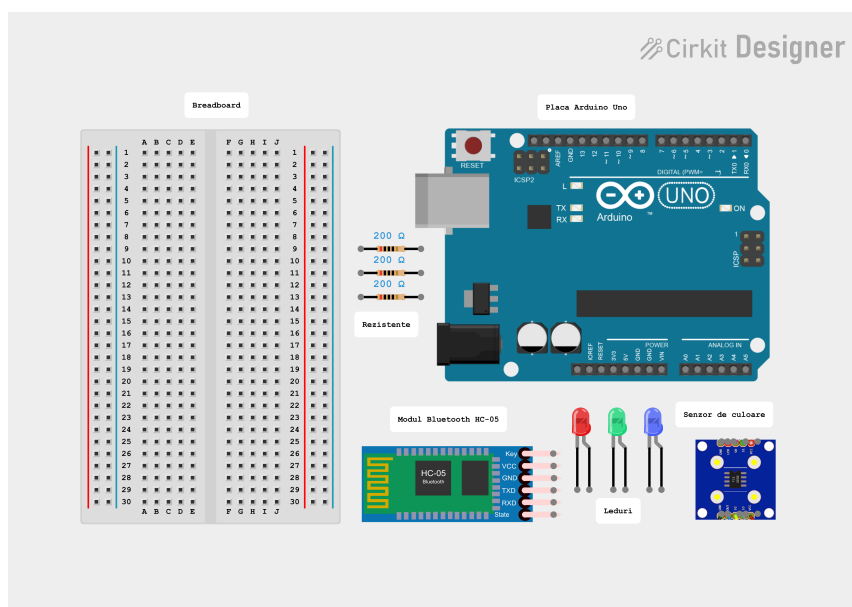


Figura 1: Componente

Placă Arduino Uno: Platforma principală pentru controlul detectorului de culori și interacțiunea cu componentele.

Senzor de culoare TCS3200: Detectează culorile și transmite informațiile către Arduino pentru procesare.

Modul Bluetooth HC-05: Permite comunicarea între Arduino și dispozitivul mobil pentru transmiterea datelor despre culori.

Breadboard, fire de conexiune, rezistențe: Utilizate pentru conexiuni fizice și stabilirea circuitului necesar funcționării dispozitivului.

3 Montaj Hardware

3.1 Diagrama de Circuit

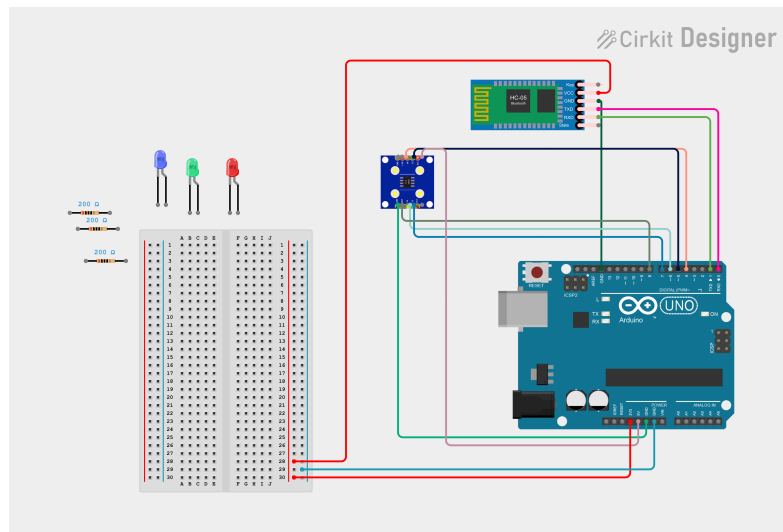


Figura 2: Conexiunea dintre componente

3.2 Descriere

Senzorul de culoare TCS3200 are pini pentru alimentare (VCC și GND), ieșirea semnalului de culoare (OUT) conectat la pinul 8 al plăcuței Arduino și patru pini pentru selectarea frecvenței culorilor (S0, S1, S2, S3).

Modulul Bluetooth HC-05 are pini pentru alimentare (VCC și GND) și pentru comunicația serială (RX și TX) care trebuie conectați la pini corespunzători ai Arduino (RX0 și TX1).

Ledurile sunt conectate la rezistente și la pinii corespunzători ai plăcuței ca în figura 1.

4 Funcționalități

4.1 Descriere cod

Bibliotecile utilizate: Se folosesc bibliotecile `SoftwareSerial.h` și `TimerOne.h` pentru comunicarea serială cu modulul Bluetooth și pentru gestionarea unui timer.

Definirea pinilor: Se definește modul de conectare a senzorului de culoare TCS3200 și a LED-urilor (pentru afișarea culorilor detectate), precum și pinii pentru comunicarea cu modulul Bluetooth.

Funcția `setup()`: Se configurează pinii ca ieșiri sau intrări, se inițializează comunicarea serială la o rată de 9600 de biți pe secundă și se setează un timer pentru a apela o anumită funcție (`callback()`) la fiecare 100 de miliseunde. Apoi se setează frecvența de lucru a senzorului pentru a citi culorile într-un anumit mod.

Funcția `callback()`: Se configurează senzorul pentru a citi frecvența culorilor roșu, verde și albastru, iar apoi se măsoară aceste frecvențe. Pe baza valorilor de frecvență măsurate, se fac verificări pentru a identifica culorile detectate și se aprind LED-urile corespunzătoare culorii identificate.

Este important de menționat că acest cod se bazează pe măsurarea frecvenței luminii reflectate de diferite culori și pe interpretarea acestor frecvențe pentru identificarea culorilor. Configurarea senzorului TCS3200 este crucială pentru a obține rezultate precise în identificarea culorilor.

4.2 Soluția propusă

```
1  #include <SoftwareSerial.h>
2  #include <TimerOne.h>
3
4  #define S0 4
5  #define S1 5
6  #define S2 6
7  #define S3 7
8  #define sensorOut 8
9
10 #define LedRed    A2
11 #define LedGreen  A0
12 #define LedBlue   A1
13
14 #define BT_RX_PIN 0
15 #define BT_TX_PIN 1
16
17 SoftwareSerial BTSerial(BT_RX_PIN, BT_TX_PIN);
18
19 int frequency=0;
20 int redfrequency = 0;
21 int greenfrequency = 0;
22 int bluefrequency = 0;
23
24 void setup() {
25     pinMode(S0, OUTPUT);
26     pinMode(S1, OUTPUT);
27     pinMode(S2, OUTPUT);
28     pinMode(S3, OUTPUT);
29     pinMode(sensorOut, INPUT);
30
31     pinMode(LedRed, OUTPUT);
32     pinMode(LedGreen, OUTPUT);
33     pinMode(LedBlue, OUTPUT);
34
35     // Setting frequency-scaling to 20%
36     digitalWrite(S0,HIGH);
37     digitalWrite(S1,LOW);
38
39     Serial.begin(9600);
40
41     Timer1.initialize(100000); // set a timer of length 100000 microseconds (or 100 ms)
42     Timer1.attachInterrupt(callback); // attach the service routine here
43 }
```

```

44
45 void callback() {
46     // Setting red filtered photodiodes to be read
47     digitalWrite(S2,LOW);
48     digitalWrite(S3,LOW);
49     // Reading the output frequency
50     redfrequency = pulseIn(sensorOut, LOW);
51
52     // Setting Green filtered photodiodes to be read
53     digitalWrite(S2,HIGH);
54     digitalWrite(S3,HIGH);
55     // Reading the output frequency
56     greenfrequency = pulseIn(sensorOut, LOW);
57
58     // Setting Blue filtered photodiodes to be read
59     digitalWrite(S2,LOW);
60     digitalWrite(S3,HIGH);
61     // Reading the output frequency
62     bluefrequency = pulseIn(sensorOut, LOW);
63
64     if (redfrequency>25 && redfrequency< 100)
65     {
66         Serial.println("ROSU");
67         digitalWrite(LedRed, HIGH);
68         digitalWrite(LedBlue, LOW);
69         digitalWrite(LedGreen, LOW);
70     }
71     else if (bluefrequency>44 && bluefrequency< 150)
72     {
73         Serial.println("ALBASTRU");
74         digitalWrite(LedBlue, HIGH);
75         digitalWrite(LedGreen, LOW);
76         digitalWrite(LedRed, LOW);
77     }
78     else if (greenfrequency>50 && greenfrequency< 180)
79     {
80         Serial.println("VERDE");
81         digitalWrite(LedGreen, HIGH);
82         digitalWrite(LedBlue, LOW);
83         digitalWrite(LedRed, LOW);
84     }
85     else if (bluefrequency> 80 && redfrequency>150 && greenfrequency>185)
86     {
87         Serial.println("NEGRU");
88         digitalWrite(LedBlue, LOW);
89         digitalWrite(LedGreen, LOW);
90         digitalWrite(LedRed, LOW);
91     }

```

```

92  else if (bluefrequency> 4 && redfrequency>13 && greenfrequency>15)
93  {
94      Serial.println("ALB");
95      digitalWrite(LedBlue, LOW);
96      digitalWrite(LedGreen, LOW);
97      digitalWrite(LedRed, LOW);
98  }
99  else
100 {
101     Serial.println("NO COLOUR DETECTION");
102     digitalWrite(LedBlue, LOW);
103     digitalWrite(LedGreen, LOW);
104     digitalWrite(LedRed, LOW);
105 }
106 }
107
108 void loop() {
109
110 }

```

5 Interfața Bluetooth

Parcurgeți următorii pași pentru a stabili o conexiune între Arduino și telefonul mobil:

1. Descărcați aplicația Serial Bluetooth Terminal, sau o altă aplicație echivalentă, pe telefonul mobil.
2. Conectați modulul Bluetooth la placa Arduino, conform instrucțiunilor de mai sus, și programați placa Arduino cu programul dat.
3. Activați funcția Bluetooth a telefonului dvs, și intrați în meniul de setări la secțiunea Bluetooth, pentru a descoperi dispozitivele disponibile.
4. Identificați numele dispozitivului dumneavoastră care ar trebui să fie de forma HC-05.
5. Inițiați operațiunea de pairing (împerechere) cu dispozitivul. Dacă vi se solicită un cod PIN, introduceți 1234.
6. Porniți aplicația terminal pe telefonul mobil. Din meniul aplicației, selectați “Devices”, și apoi alegeți dispozitivul dumneavoastră. În acest moment se va stabili conexiunea.
7. În secțiunea de Terminal, ar trebui să apară culoarea detectată.

6 Exemple de Utilizare

6.1 Arduino IDE

6.2 Detector

6.3 Bluetooth

7 Bibliografie

- <https://mihai.utcluj.ro>
- <https://howtomechatronics.com>

- <https://circuitdigest.com>