

Dispozitiv detectie culoare obiect

Proiect 2

Realizatori: STEFAN Costin-Cristian 433Aa
IONETE George-Iulian 433Aa

Profesor coordonator: Valentin Stoica

Cuprins

Descriere proiect.....	3
Placa de dezvoltare Uno R3 ATmega328P	
Descriere.....	4
Caracteristici.....	4
Microcontroller-ul ATmega328	
Descriere.....	5
Caracteristici.....	5
Modul senzor de culoare	
Descriere.....	6
Caracteristici.....	6
Montaj	
Montajul schematic	7
Montajul fizic.....	8
Partea de software	
Arduino IDE	9
Python.....	11

DESCRIERE PROIECT

Scopul proiectului este acela de a crea un dispozitiv care detectează culoarea obiectelor folosind un senzor de culoare, montat la o placă de dezvoltare Arduino, și afisează rezultatul pe ecranul calculatorului prin intermediul unui software.

Componentele de bază folosite în proiect sunt: placă de dezvoltare Arduino Uno R3 ATmega328, cu microcontrollerul ATmega328 și un modul senzor de culoare, cu chip-ul TSC230 încorporat pe placă. Acesta este legat prin intermediul unor fire la placa de dezvoltare Arduino, iar aceasta la rândul ei este alimentată printr-un cablu usb direct de la computer. Soft-ul este încărcat prin intermediul aceluiași cablu.

Partea software a proiectului este realizată în două medii de dezvoltare. Primul este reprezentat de Arduino IDE, software prin intermediul căruia se citește semnalul dreptunghiular pe care îl transmite montajul, se alege frecvența de lucru, în cazul nostru 2%. Prin intermediul diferitelor combinații de LOW și HIGH pentru porturile S2, S3 putem găsi frecvența pentru fiecare culoare, astfel :

- S2 – LOW, S3 – LOW -> Red
- S2 – LOW, S3 – HIGH -> Blue
- S2 – HIGH, S3 – HIGH -> Green

Valorile astfel obținute nu reprezintă valorile RGB, întrucât acestea pot atinge valori de până la 1000 așa că acestea trebuie mapate între valorile 0 și 255, unde 255 reprezintă frecvența cea mai mică, iar 0 frecvența cea mai mare. Valorile acestea reprezintă ieșirea din primul mediu de dezvoltare. Programul din cadrul Arduino IDE are nevoie de calibrare continuă dacă luminile ambientale se modifică constant.

Al doilea mediu de dezvoltare este IDLE, un IDE pentru limbajul de programare python. Pentru ca acesta să poată accesa datele primite prin porturi folosim biblioteca pyserial. Culoarea este afișată prin intermediul bibliotecii turtle. În programul realizat în python, se citesc valorile trimise mai departe de Arduino IDE printr-o variabilă de tip string, se aleg cele trei variabile R, G, B se convertesc în valori hexazecimale, se aranjează într-un string de tipul hexazecimal (exemplu: #000000), apoi sunt pasate ca parametru pentru funcția din biblioteca turtle care actualizează culoarea de fundal din cadrul ferestrei astfel create. Pentru ca programul să nu returneze eroare la fiecare recepționare de date mai problematice (de exemplu valori care nu au fost bine mapate, adică valori negative sau care depășesc 255) au fost introduse niște condiții.

Placa de dezvoltare Arduino Uno R3 ATmega328P

DESCRIERE: Arduino Uno R3 este o placă de dezvoltare bazată pe microcontrollerul ATmega328. Are 14 pini de intrare/ieșire (dintre care 6 pot fi folosiți ca ieșiri PWM), 6 intrări analog, un oscilator de 16MHz, o conexiune USB, mufă de alimentare, și un buton de reset. Poate fi alimentat direct de la calculator, de la portul USB, prin intermediul unei baterii de 9V sau a unui alimentator de 9V.

CARACTERISTICI TEHNICE:

- Microcontroller ATmega328
- Tensiune de operare: 5V
- Tensiune de alimentare recomandată: 7-12V
- Limită de tensiune: 6-20V
- Pini intrare/ieșire digitali: 14 (dintre care 6 pot oferi ieșire PWM)
- Pini analogici de intrare: 6
- Memorie Flash 32 KB
- SRAM 2 KB
- EEPROM 1 KB
- Frecvență de lucru: 16 MHz



Microcontroller-ul ATmega328

DESCRIERE:

Microchip Technology ATmega328 Microcontrolere AVR[®] pe 8 biți (MCU) sunt dispozitive RISC de înaltă performanță, care combină 32KB ISP Flash memory cu capacități de citire în timp, scriere în timp, 1KB EEPROM, 2KB SRAM, 23 linii I / O de uz general, 32 registre de lucru cu scop general, USART programabil în serie și multe altele.

MCm-urile ATmega328 execută instrucțiuni puternice într-un singur ciclu de ceas, permițând dispozitivului să realizeze randamente care se apropie de 1 MIPS pe MHz în timp ce echilibrează consumul de energie și viteza de procesare. Aceste MCU Microchip sunt proiectate pentru a fi utilizate în automatizarea industrială și în automatizarea caselor și a clădirilor.

CARACTERISTICI TEHNICE:

- Microcontroller AVR[®] de 8 biți , de înaltă performanță
- Advanced RISC Architecture
- 131 Instrucțiuni puternice
- Până la 20 MIPS cu 20 MHz
- Multiplicator cu 2 cicluri On-chip
- Numărul de pini: 28/32
- Max I/O Pins: 23
- SPI: 2
- UART: 1
- ADC: 8 canale, rezoluție pe 10 biți
- Comparatori analogici: 1
- Flash (Kbytes): 32
- EEPROM (Kbytes): 1
- SRAM (Kbytes): 2
- Temp. Interval: -40 până la 85 ° C



Modul senzor de culoare

DESCRIERE:

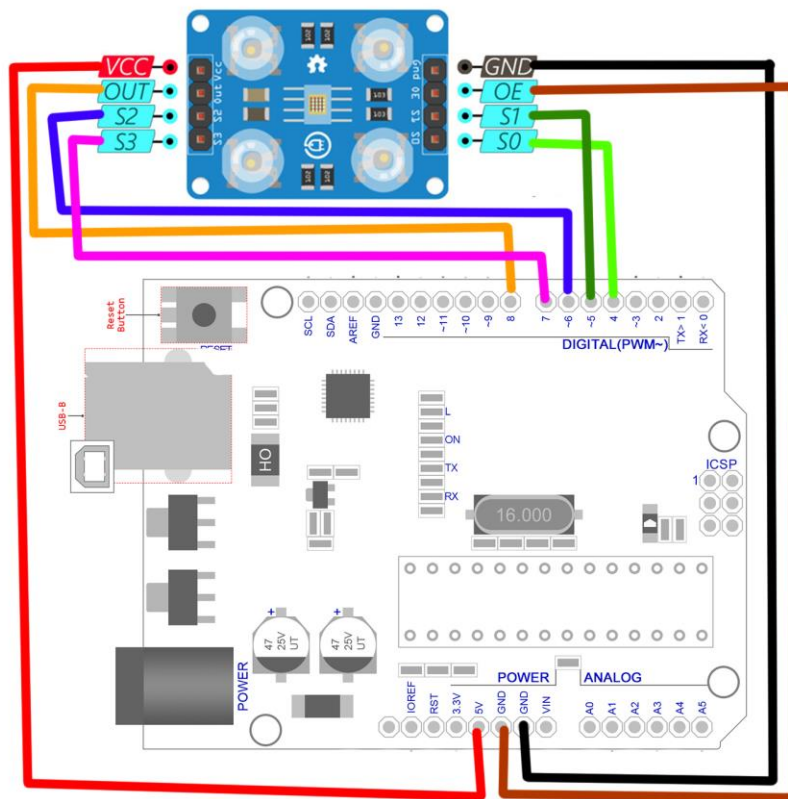
Acest modul captează lumina color cu ajutorul unui chip TSC230 cu o matrice de fotodiode de 8 x 8. Apoi, folosind un convertor curent-la-frecvență citirile de la fotodiode sunt convertite într-un semnal dreptunghiular cu o frecvență direct proporțională cu intensitatea luminii. În cele din urmă, folosind placa de dezvoltare Arduino, putem citi rezultatul semnalului și obține rezultatele pentru culoare.

CARACTERISTICI TEHNICE:

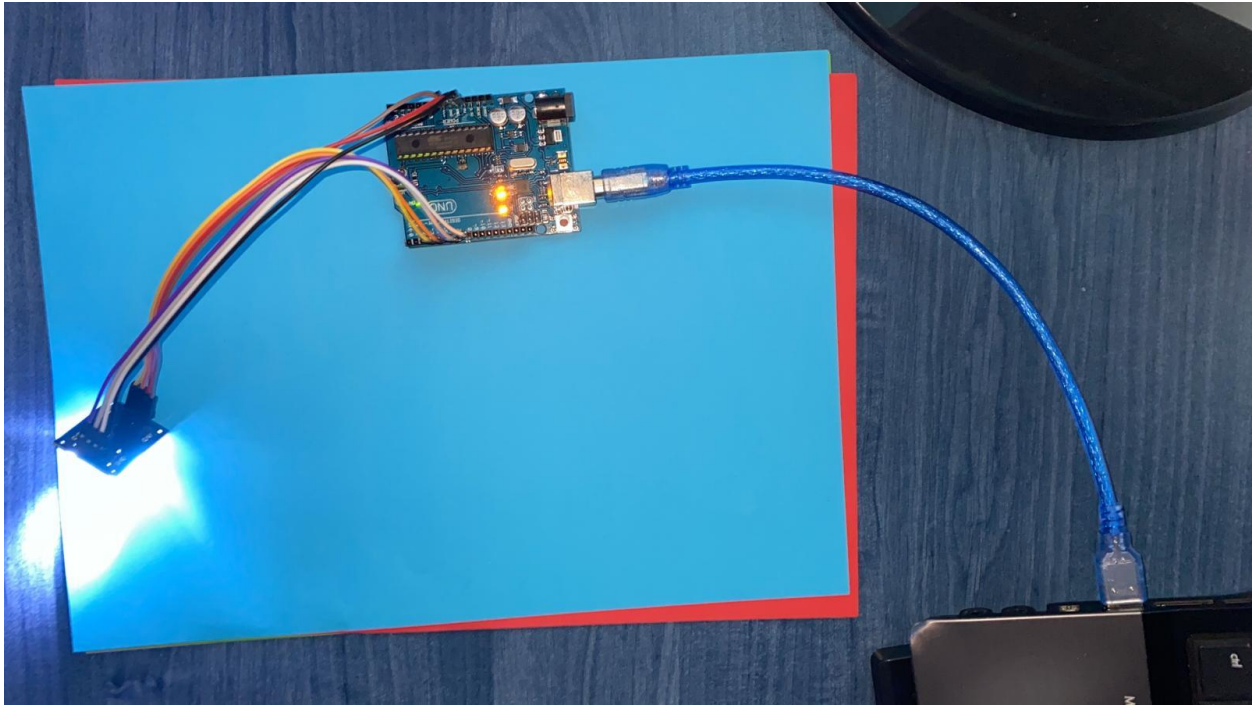
- Alimentare: 2.7V - 5.5V
- Consum: 3mA
- Dimensiuni: 32 x 25mm
- Temperaturi de lucru: de la - 40 la 70
- Pini de lucru: 8
- Canalele senzorului de culoare: RGBC (Red, Green, Blue, Clear Filter)



MONTAJUL SCHEMATIC



MONTAJ FIZIC:



PARTEA DE SOFTWARE:

1.Codul din Arduino IDE:

```
#define S0 4

#define S1 5

#define S2 6

#define S3 7

#define sensorOut 8

int redFrequency = 0;

int greenFrequency = 0;

int blueFrequency = 0;

int redColor = 0;

int greenColor = 0;

int blueColor = 0;

void setup() {

    pinMode(S0, OUTPUT);

    pinMode(S1, OUTPUT);

    pinMode(S2, OUTPUT);

    pinMode(S3, OUTPUT);

    pinMode(sensorOut, INPUT);

    digitalWrite(S0,LOW);

    digitalWrite(S1,HIGH);

    Serial.begin(9600);}
```

```

void loop() {

  digitalWrite(S2,LOW);

  digitalWrite(S3,LOW);

  redFrequency = pulseIn(sensorOut, LOW);

  redColor = map(redFrequency, 240, 530, 255,0);

  delay(100);


  digitalWrite(S2,HIGH);

  digitalWrite(S3,HIGH);

  greenFrequency = pulseIn(sensorOut, LOW);

  greenColor = map(greenFrequency, 250, 690, 255, 0);

  delay(100);


  digitalWrite(S2,LOW);

  digitalWrite(S3,HIGH);

  blueFrequency = pulseIn(sensorOut, LOW);

  blueColor = map(blueFrequency, 90, 170, 255, 0);

  Serial.println(String(redColor)+" "+String(greenColor)+" "+String(blueColor));

  //Serial.println(String(redFrequency)+" "+String(greenFrequency)+" "+String(blueFrequency));

  delay(100);

}

```

2. Codul în python:

```
import time

import serial

import turtle

b = turtle.Screen()

try:

    ser = serial.Serial(

        port='COM4',

        baudrate=9600,

        timeout=1)

except:

    print('Please check the port')R = 0

G = 0

B = 0

nr = 0

while 1:

    RGB = str(ser.readline().decode('ascii'))

    print(RGB)

    tmp=""

    new=[]

    count = 0

    for i in RGB:
```

```

if i==" ":
    if count == 0:
        R = int(float(tmp))
        count = count + 1
    elif count == 1:
        G = int(float(tmp))
        count = count + 1
    elif count == 2:
        count = 0
    new.append(tmp)
    tmp=""
elif i != "-":
    tmp=tmp+i
if tmp:
    B = int(float(tmp))
print(R)
print(G)
print(B)
if R > 255:
    R = 255
if G > 255:
    G = 255
if B > 255:
    B = 255
if R < 16:
    red = "0" + str(hex(R))[2:]

```

```

else: red = str(hex(R))[2:]

if G < 16:

    green = "0" + str(hex(G))[2:]

else: green = str(hex(G))[2:]

if B < 16:

    blue = "0" + str(hex(B))[2:]

else: blue = str(hex(B))[2:]

color = "#" + red + green + blue

if nr > 1:

    b.bgcolor(color)

nr = nr + 1

```

Consolași partea de afișaj după rularea programelor:

