

Smart Lights

Autor: Popescu Eduard-Gabriel

Grupa: 335CA

Introducere

Se va folosi un led RGB care isi va modifica culoarea si intensitatea luminoasa in functie de caracteristicile mediului in care se afla. Valoarea citita de senzori va dicta un canal de culoare al led-ului, astfel incat de fiecare data cand o valoare se va modifica, aceasta se va reflecta, de asemenea, in lumina emisa de led.

Descriere generală

Exista doua led-uri care descriu conditiile incaperii: un led principal si un led secundar.

Senzorul DHT11 va citi temperatura si umiditatea incaperii. Temperatura va controla canalul red al led-ului principal, iar umiditatea va controla canalul blue. Temperaturile in intervalul [0, 40] grade Celsius sunt mapate intr-o intensitate a led-ului corespunzatoare intervalului [0, 255]. Orice temperatura mai mare de 40 de grade va fi reprezentata de 255, intensitatea maxima a ledului. Indexul de umiditate are valori de la [0, 100] si va fi mapat intr-o valoare din intervalul [0, 75]. Aceasta combinatie va permite led-ului sa aiba o culoarea violet placuta.

Senzorul de prezenta PIR va detecta miscarea unei persoane care trece prin fata lui si va modifica culoarea led-ului principal in verde.

Led-ul secundar se va aprinde in momentul in care gradul de iluminare al camerei este scazut si va avea intotdeauna o culoarea determinata de temperatura si de indexul de umiditate al camerei. In cazul obisnuit, led-ul principal si secundar vor avea aceeasi culoare. Diferenta apare in momentul in care senzorul de prezenta va detecta o miscare si va schimba culoarea led-ului principal in verde. Acesta este singurul caz in care culorile led-urilor vor fi diferite, intrucat led-ul secundar are rolul de a mari gradul de iluminare al camerei.



Hardware Design

- Arduino Uno R3 ATmega328P
- Breadboard
- Led RGB : Anod Comun
- Senzor prezenta PIR
- Senzor temperatura si umiditate DHT11
- Fotorezistor 5528 LDR



Software Design

Pentru scrierea codului m-am folosit de arduino web editor, care are aceleasi functionalitati cu ide-ul arduino nativ.

Dispozitivul va astepta 30s la pornire pentru a acorda senzorilor timpul necesar pentru a se acomoda la conditiile camerei dupa care la fiecare 500ms va citi face o masurare a parametrilor camerei.

Am folosit biblioteca "DHT.h" pentru a comunica cu senzorul DHT11.

In functia setup() setez pinii la care sunt conectati led-urile ca output si ii pun pe low.

In functia loop() citesc temperatura si indexul de umiditate indicate de DHT11 si verific daca valorile sunt valide. In cazul in care nu sunt, senzorul va mai astepta 500ms. Daca totul este in regula, calculam intensitatea luminoasa a led-ului si il aprindem. De asemenea, verificam daca senzorul PIR a

detectat miscare, iar in cazul in care a facut-o, setam culoarea led-ului principal sa fie verde. In momentul in care gradul de iluminare al camerei scade, se va aprinde si led-ul secundar.

```
#include <DHT.h>

#define DHTPIN 7
#define DHTTYPE DHT11

#define mainLed 0
#define auxLed 1

#define pinMainLedRed 11
#define pinMainLedGreen 10
#define pinMainLedBlue 9

#define pinAuxLedRed 6
#define pinAuxLedGreen 5
#define pinAuxLedBlue 3

#define PIRPIN 4

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  dht.begin();
  pinMode(pinMainLedRed, OUTPUT);
  pinMode(pinMainLedGreen, OUTPUT);
  pinMode(pinMainLedBlue, OUTPUT);

  pinMode(pinAuxLedRed, OUTPUT);
  pinMode(pinAuxLedGreen, OUTPUT);
  pinMode(pinAuxLedBlue, OUTPUT);

  setColorRGB(0, 0, 0, mainLed);
  setColorRGB(0, 0, 0, auxLed);

  // Wait for the sensors to acomodate to the conditions of the room
  delay(30000);
}

void loop() {
  // Wait a few seconds between measurements.
  delay(500);

  // Reading temperature or humidity takes about 250 milliseconds
  float h = dht.readHumidity();
  // Read temperature as Celsius
  float t = dht.readTemperature();

  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t)) {
```

```
    return;
}

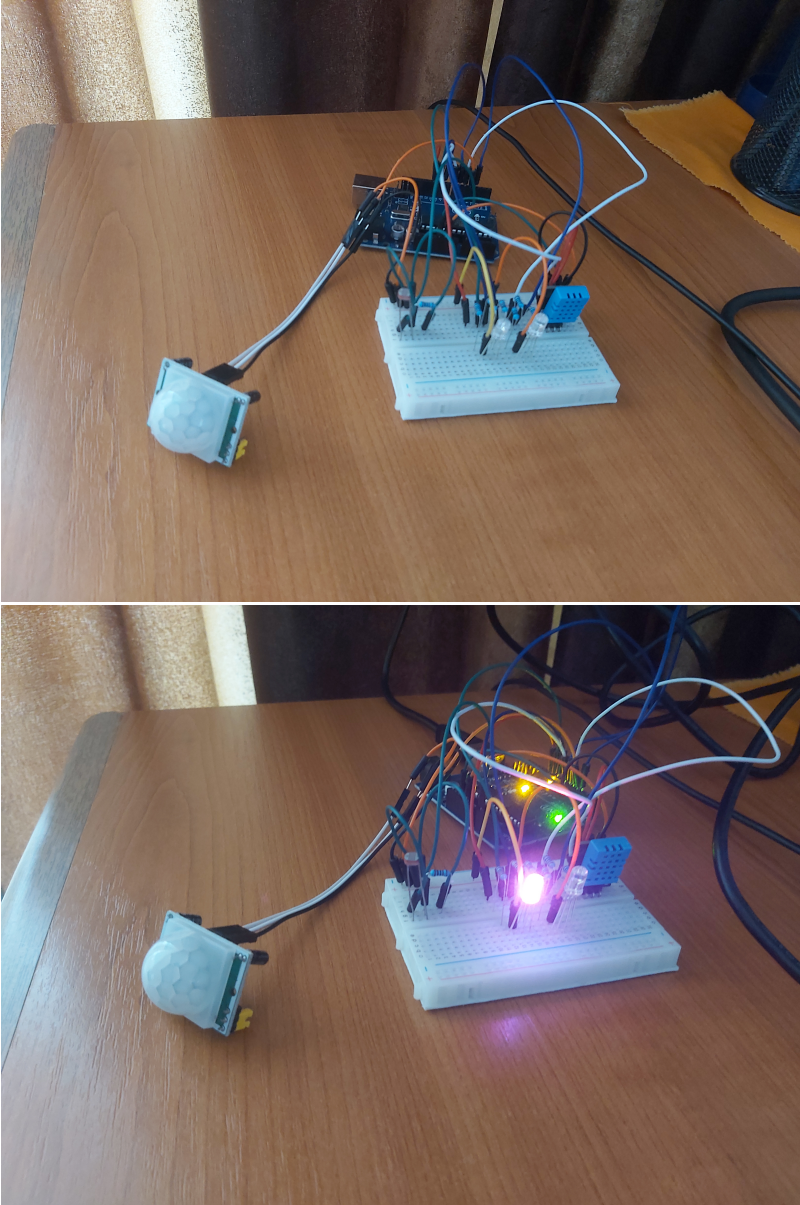
int red = (int)(t * 6.375);
if (red > 255) {
    red = 255;
}
int blue = (int)(h * 0.75);
if (blue > 75) {
    blue = 75;
}
setColorRGB(red, 0, blue, mainLed);

int light = analogRead(A0);
if (light < 15) {
    setColorRGB(red, 0, blue, auxLed);
} else {
    setColorRGB(0, 0, 0, auxLed);
}

int pirValue = digitalRead(PIRPIN);
if (pirValue == 1) {
    setColorRGB(0, 255, 0, mainLed);
    delay(2000);
}
}

void setColorRGB(unsigned int red, unsigned int green, unsigned int blue,
unsigned int type) {
    if (type == mainLed) {
        analogWrite(pinMainLedRed, 255 - red);
        analogWrite(pinMainLedGreen, 255 - green);
        analogWrite(pinMainLedBlue, 255 - blue);
    } else if (type == auxLed) {
        analogWrite(pinAuxLedRed, 255 - red);
        analogWrite(pinAuxLedGreen, 255 - green);
        analogWrite(pinAuxLedBlue, 255 - blue);
    }
}
```

Rezultate obtinute



Demo

Concluzii

A fost interesant sa realizez un proiect care presupune integrarea unor senzori. Partea hardware a fost o experienta noua pentru mine. Desi in cadrul laboratoarelor am lucrat in tinkercard, lucrul hands on e diferit. E necesara o anumita responsabilitate pentru a nu arde piesele, lucru care nu se simte atat de mult intr-un program software de simulare. Pentru imbunatatirea proiectului m-am gandit ca led-urile pot fi inlocuite cu o banda de led-uri rgb care se pot programa sa lumineze in diferite moduri.

Download

[smart_lights.zip](#)

Bibliografie/Resurse

- [PIR Sensor Usage](#)
- [Arduino Eagle Library](#)
- [DHT Library](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2021/alazar/smart_lights

Last update: **2021/05/28 15:52**

