

# Smart lights

## Introducere

*Smart lights* reprezinta un **sistem distribuit** implementat cu scopul de a schimba luminile ambientale pentru a reflecta temperatura din spatiul in care se afla senzorul.

*Functionalitate:* **detectarea miscarii** pentru aprinderea luminilor, **masurarea temperaturii** din incapere, afisarea valorii acesteia si **modificarea intensitatii si culorii luminilor** in functie de temperatura, precum si inchiderea acestora la un interval de timp atunci cand nu se mai detecteaza miscare.

## Descriere generală

Sistemul distribuit este bazat pe 3 module independente fizic care comunica intre ele: **modulul 1 trimite date catre modulul 2**, iar **modulul 3 trimite date catre modulul 2**.

**Modulul 1** reprezinta un Arduino Uno conectat la un **senzor de miscare** si un **modul nRF**. Acesta detecteaza miscarea prin senzorul infrarosu pasiv si transmite informatia prin unde radio catre modulul 2.

**Modulul 3** reprezinta o alta placuta Arduino Uno conectata la un **senzor de temperatura**, un **display lcd** si un **modul bluetooth**. Acesta are rolul de a masura temperatura (si umiditatea) din incapere si de a transmite valorile masurate prin bluetooth catre modulul 2. Totodata, afiseaza pe display-ul lcd rezultatele masuratorii. Display-ul lcd are un **modul I2C** prin intermediul caruia se conecteaza la placuta Arduino.

**Modulul 2** reprezinta tot o placuta Arduino conectata la un **led**, un **modul nRF** si un **modul bluetooth**. Rolul sau este de a aprinde led-ul cand primeste informatia corespunzatoare de la modulul 1 si de a seta culoarea si intensitatea luminii in functie de informatia primita de la modulul 3. Placa este conectata la modulele bluetooth si nRf pentru a fi **receptor** pentru modulul 3, respectiv modulul 1.



Printre obiectivele urmarite se numara:




- Crearea unui sistem distribuit pentru a permite amplasarea senzorilor, a display-ului si a led-urilor in *locuri diferite*.
- *Minimizarea* consumului de putere.

# Hardware Design

## Lista piese:

- 3x Arduino Uno
- Motion Sensor HC-SR501
- Temperature Sensor DHT11
- 2x Bluetooth Module HC-05
- 2x RF Module nRF24L01
- Display LCD 1602
- I2C Module for LCD 1602
- LED RGB
- 3x Battery & battery holder
- 3x Small Breadboard
- 3x Resistor 220Ω
- Wires

## Scheme electrice:

Modul	Schema
Modulul 1	
Modulul 2	
Modulul 3	

# Software Design

Fiecare modul al sistemului distribuit are propriul sau cod sursa.

Aspecte comune in fisierele sursa:

- Deoarece senzorul de miscare are nevoie de un timp pentru a se calibra, setup-ul modulelor este facut sa dureze aproximativ 30 de secunde, folosind functia de delay.

```
for(int i = 0; i < calibrationTime; i++)
    delay(1000);
```

- Modulele ce comunica prin RF (1 si 2), au implementata o structura numita Package cu doua campuri: state - indica starea senzorului de miscare (1 atunci cand detecteaza miscarea, 0 atunci cand nu o detecteaza) si id - folosit pentru a identifica pachetele trimise.
- Modulele bluetooth au fost setate cu rol de master/slave folosind comenzi AT.
- O biblioteca Arduino, LowPower.h, este utilizata pentru a face ca modulul 1 si modulul 3 sa transmita date o data la 8 secunde, iar in rest sa intre in modul powerDown, cu scopul de a economisi putere.

## Modulul 1:

- In functia setup se initializeaza modulul nRF cu parametrii precum canalul si pipe-ul pe care se

desfasoara comunicatia.

```
myRadio.begin();
myRadio.setChannel(115);
myRadio.openWritingPipe(pipe);
```

- In functia loop se citesc datele de pe pin-ul senzorului de miscare, se introduc in structura creata si se trimit catre celalalt modul. Se trimite doar atunci cand apare o schimbare in starea pinului, iar dupa ce s-a detectat miscare, placuta Arduino intra in modul low power pentru 8 secunde.

```
LowPower.powerDown(SLEEP_8S, ADC_OFF, BOD_OFF);
```

## Modulul 2:

- In functia setup este setat modulul nRF similar cu setarea acestuia pentru modulul 1.
- In functia loop se verifica primirea datelor prin bluetooth sau prin RF.

```
if (myRadio.available()) {
    myRadio.read(&data, sizeof(data));
}

if(Serial.available() > 0) {
    value = Serial.read();
}
```

- Culoarea led-ului RGB este setata folosind functia analogWrite in functia setColor.

```
void setColor(int R, int G, int B) {
    analogWrite(red, R);
    analogWrite(green, G);
    analogWrite(blue, B);
}
```

## Modulul 3:

- Pentru LCD este folosita biblioteca LiquidCrystal\_I2C.h. Pentru DHT11 se foloseste biblioteca dht11.h.
- In functia setup este initializat lcd-ul si este efectuata o prima masuratoare si afisare.
- In functia loop se citește temperatura masurata de catre DHT folosind functia DHT11.read, iar apoi este afisata pe LCD. Este trimisa valoarea masurata prin bluetooth, folosind interfata seriala, iar dupa trimiterea unei valori Arduino intra in modul low power pentru 8 secunde, folosind functia powerDown de mai sus.

```
int chk = DHT11.read(DHT11PIN);
lcd.clear();
lcd.setCursor(0,0);// set the cursor on the first row and column
lcd.print("Humidity: ");
lcd.print(DHT11.humidity); // print the humidity
lcd.print("%");
lcd.setCursor(0,1); // set the cursor on the second row and first column
lcd.print("Temperature: ");
lcd.print(DHT11.temperature); // print the temperature
```

```
lcd.print((char)223); // degree symbol
```

## Rezultate Obținute

<https://youtu.be/Mroa0k1NPPg>

## Concluzii si sugestii imbunatatire

Sistemul distribuit permite amplasarea modulelor in diferite pozitii ale camerei, la distante ce depind de specificatiile hardware ale modulelor nRF si bluetooth.

Consumul de putere a fost minimizat prin utilizarea bibliotecii Arduino mentionate in descrierea implementarii.

Functionalitatea proiectului poate fi imbunatatita prin adaugarea unui senzor de intensitate luminoasa si reglarea luminii led-ului in functie de acesta.

Pentru a obtine luminozitate mai buna, pot fi adaugate mai multe led-uri sau alte tipuri de led.

## Download

[smart-lights.zip](#)

## Bibliografie/Resurse

- Modul 1 (Motion sensor):

<https://www.circuitmagic.com/arduino/pir-motion-sensor-with-arduino/>

<https://www.electronicwings.com/sensors-modules/pir-sensor>

<https://www.circuitmagic.com/arduino/pir-motion-sensor-with-arduino/>

- Modul 3 (DHT11, LCD):

<https://projecthub.arduino.cc/arcaegecengiz/12f621d5-055f-41fe-965d-a596fcc594f6>

<https://www.instructables.com/Arduino-Temp-and-Humidity-DHT11-LCD-1602A-KY-015-S/>

[https://github.com/johnrickman/LiquidCrystal\\_I2C](https://github.com/johnrickman/LiquidCrystal_I2C)

- nRF:

<https://www.electronics-lab.com/project/communication-between-two-arduinios-using-nrf24l01/>

<https://github.com/nRF24/RF24>

- Bluetooth:

<https://www.hackster.io/embeddedlab786/how-to-configure-and-pair-two-hc-05-bluetooth-modules-fa8ce2>

<https://howtomechatronics.com/tutorials/arduino/how-to-configure-pair-two-hc-05-bluetooth-module-master-slave-commands/>

[https://www.youtube.com/watch?v=I2qFXSe0W3w&ab\\_channel=AcedInventor](https://www.youtube.com/watch?v=I2qFXSe0W3w&ab_channel=AcedInventor)

- Led:

<https://www.circuitstoday.com/interface-common-anode-and-common-cathode-rgb-leds-with-arduino>

- Low Power:

<https://learn.sparkfun.com/tutorials/reducing-arduino-power-consumption>

<https://www.gammon.com.au/forum/?id=11497>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/danielid/smart-lights>



Last update: **2023/05/29 11:34**