

# **Proiect Măsurări şi Traductoare**

## **-Ventilator inteligent-**

Profesor coordonator: Floroian Laura  
Grupa: 4LF412  
Studenti: David Alexandra Bianca  
Coroiu Dana-Cornelia  
Ghenea Dana-Georgiana  
Florean Oana Maria

## Cuprins

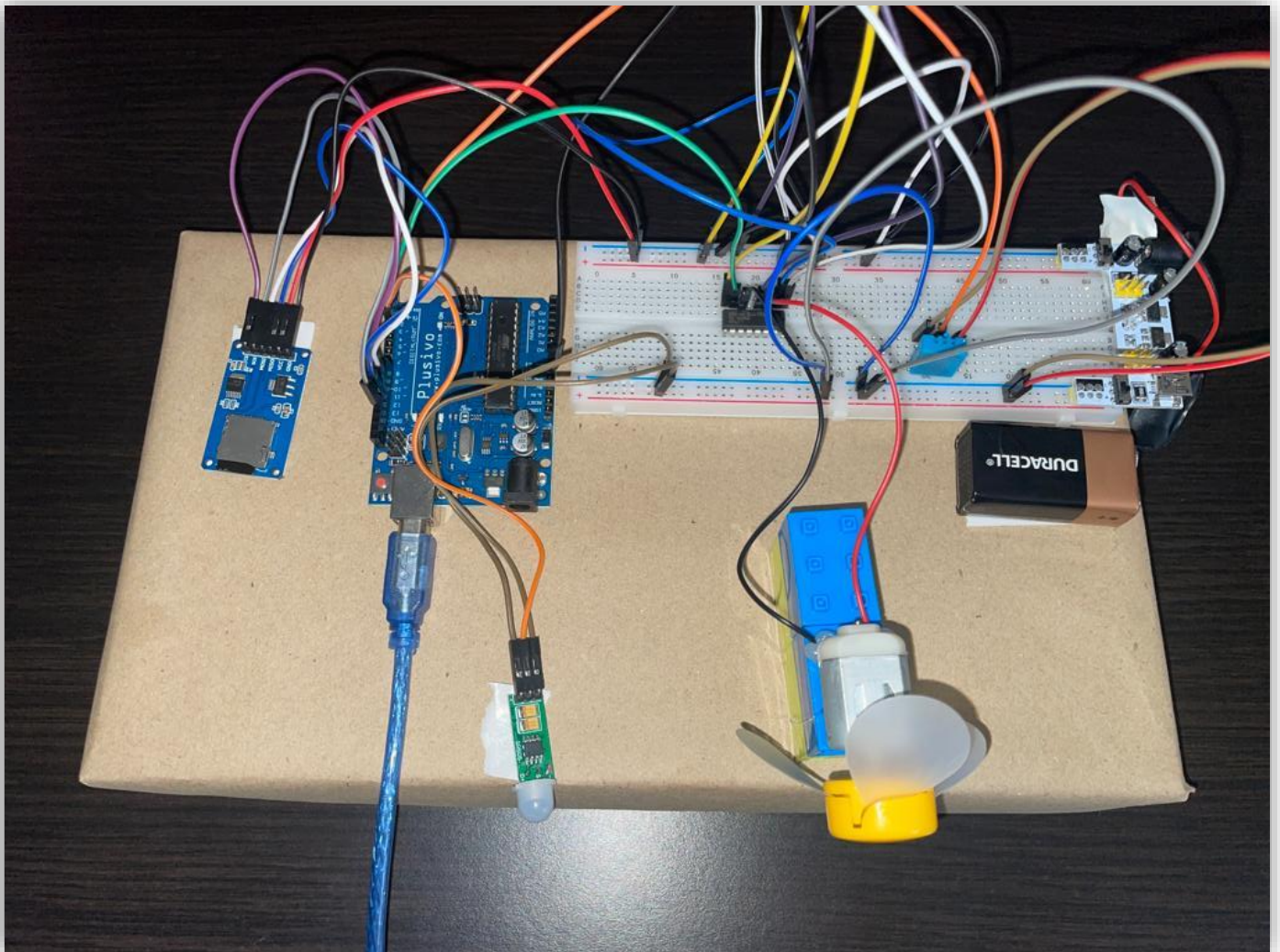
Cap. 1. Introducere .....	3
Cap. 2. Componente şi conexiuni .....	4
2.1. Componente.....	4
2.2. Conexiuni .....	5
Cap. 3. Cod.....	5
Cap. 4. Probleme întâmpinate .....	6
Cap. 5. Funcţionalitate .....	7
Cap. 6. Concluzii.....	8
Cap. 7. Bibliografie.....	9
Cap. 8. Anexe .....	9

## Cap. 1. Introducere

Controlul temperaturii într-o cameră poate fi o problemă importantă, mai ales în condiții de căldură extremă sau frig. În acest context, soluțiile automatizate de control al temperaturii pot fi de mare ajutor, prin monitorizarea constantă a acesteia și luarea de măsuri imediate în caz de necesitate.

Proiectul prezentat își propune să abordeze această problemă prin integrarea unui senzor de temperatură cu un ventilator, astfel încât acesta să pornească automat în momentul în care temperatura depășește un anumit prag. De asemenea, sistemul va înregistra aceste informații pe un card SD pentru analiza ulterioară.

Scopul acestui proiect este de a oferi o soluție eficientă și convenabilă pentru controlul temperaturii într-o cameră, prin intermediul tehnologiilor moderne.



## Cap. 2. Componente şi conexiuni

### 2.1. Componente

Pentru a realiza acest proiect am avut nevoie de următoarele piese:

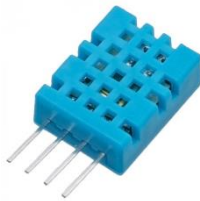
- ✓ Placă Arduino Uno



- ✓ Breadboard



- ✓ Senzor de temperatură DTH11



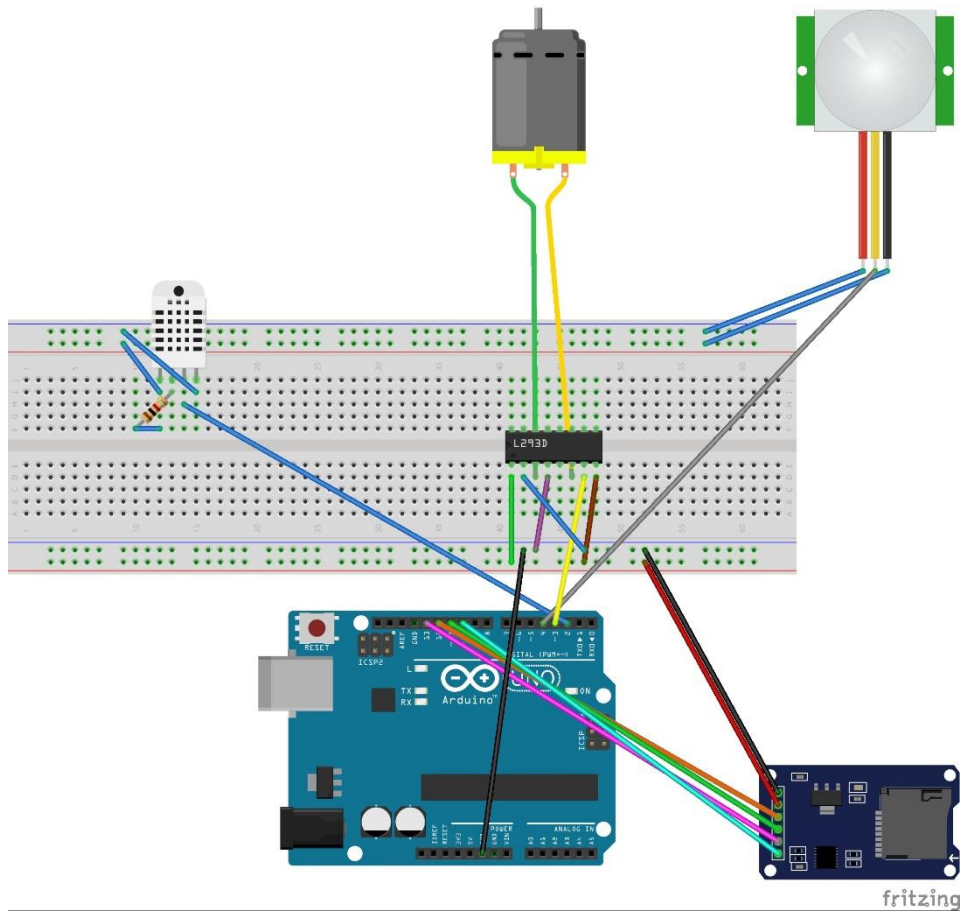
- ✓ Motor cu elice
- ✓ Card microSD + adaptor
- ✓ Senzor PIR HC-SR505



- ✓ Modul sursă de alimentare + alimentator
- ✓ Driver L293
- ✓ Rezistenţă(10kOhm)
- ✓ Fire Dupont tată-tată/mamă-tată

## 2.2. Conexiuni

Schema electrică folosită pentru a realiza proiectul a fost concepută în aplicaţia Fritzing:



## Cap. 3. Cod

Codul a fost realizat în mediul de lucru Arduino IDE. Mai jos este prezentată partea semnificativă de cod care realizează citirea temperaturii şi scrierea acesteia pe card.

```
void loop()
{
  hasMoved = digitalRead(pir);
  Serial.println(hasMoved);
  byte temp = 0;
  byte humidity = 0;
  byte data[40] = {0};

  if (dht11.read(pinDHT11, &temp, &humidity, data))
  { Serial.print("Read DHT11 failed");
    return;
  }
```

```
}  
Serial.print((float)temp);  
Serial.print(" *C, ");  
  
dataLoggerFile = SD.open("logger.txt", FILE_WRITE);  
  
if (dataLoggerFile)  
{ dataLoggerFile.print((float)temp);  
  dataLoggerFile.print("-");  
  dataLoggerFile.println(hasMoved);  
  Serial.println("Done");  
}  
  
else  
{  
  Serial.println("error opening data logger file");  
}  
dataLoggerFile.close();  
  
if((float)temp>=20)  
  digitalWrite(m,HIGH);  
else  
  digitalWrite(m,LOW);  
  
delay(1000);  
}
```

## Cap. 4. Probleme întâmpinate

În timp ce asamblam piesele, făceam legăturile şi am încărcat codul au apărut şi primele probleme:

1. Nefuncţionarea motorului:

- iniţial proiectul avea ca driver pentru motor un ansamblu alcătuit dintr-un tranzistor si o diodă;

**Soluţie:** *Am folosit un circuit integrat.*

2. Putere de alimentare prea mică:

- la început am încercat să folosim doar placa Arduino pentru alimentare , însă aceasta nu putea alimenta toate componentele;

**Soluţie:** *Am adăugat o sursă de alimentare pentru breadboard.*

3. Conectarea baterii la sursa de alimentare:

- în teorie, sursa de alimentare putea fi conectată la o baterie externă, la fel ca placa Arduino, având nevoie de un cablu.
- Însă nu dispuneam de acele cabluri.

**Soluţie:** *am lipit firele carcasei pentru bateria de 9V la sursa de alimentare.*

## Cap. 5. Funcţionalitate

În viaţa de zi cu zi, acest proiect dus la un nivel avansat poate fi folositor în următoarele situaţii:

- a) Într-o baie, pentru a ventila aerul în urma unui duş fierbinte;
- b) Într-o sală de şedinţe pentru a menţine o temperatură potrivită şi pentru a ţine evidenţa persoanelor care intră în acea sală( acest lucru fiind posibil datorita senzorului PIR);
- c) Într-un apartament pe timp de vară, acest proiect putând fi opusul unei centrale termice, răcind casa în funcţie de temperatura prestabilită de individ.

În graficul de mai jos (Fig.1) se poate observa fluctuaţia temperaturii, stocată pe cardul SD, într-o încăpere unde temperatura era peste 20°C, iar în figura 2 (Fig. 2) este reprezentată fluctuaţia temperaturii într-o încăpere unde temperatura era sub 20°C.

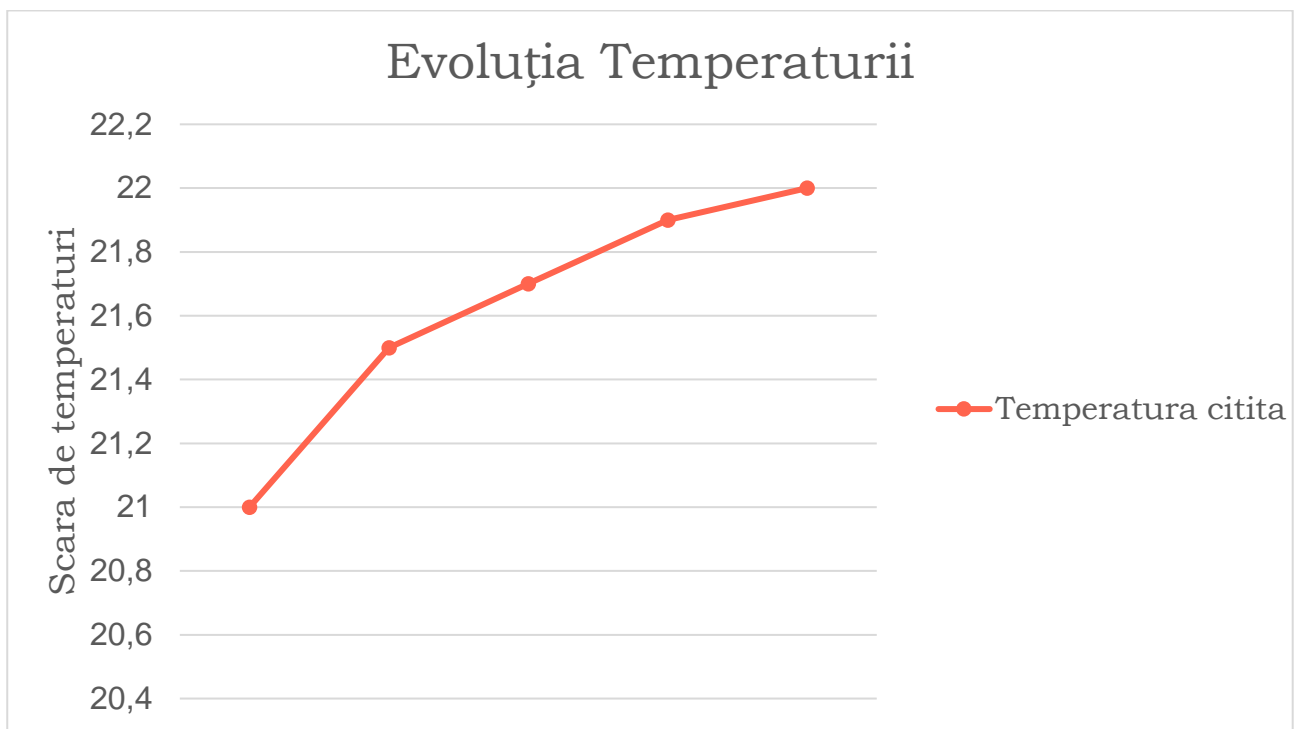


Fig. 1. Evoluția temperaturii într-o încăpere cu temperatura >20°C

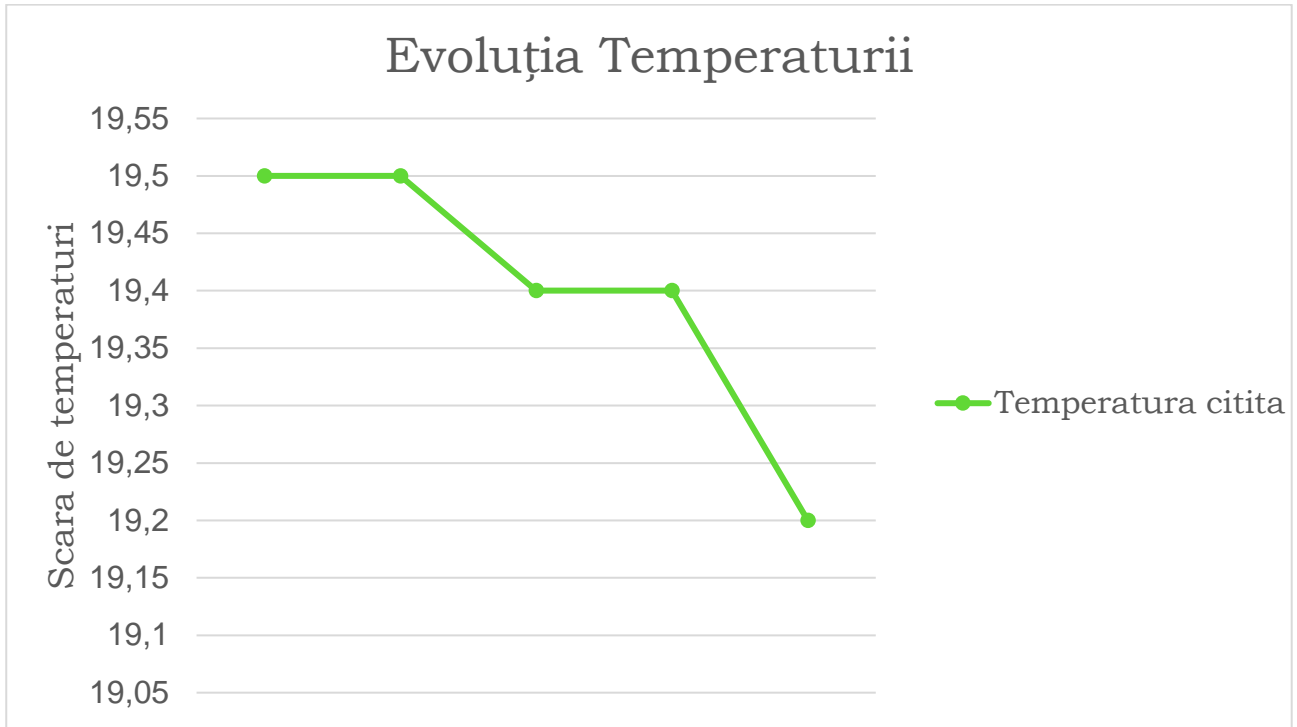


Fig. 2. Evoluţia temperaturii într-o încăpere cu temperatura < 20°C

## Cap. 6. Concluzii

Prin urmare, acest sistem permite monitorizarea continuă a temperaturii şi luarea de măsuri imediate în cazul în care temperatura depăşeşte un anumit prag prestabilit, astfel asigurând un mediu confortabil.

De asemenea, sistemul de înregistrare a datelor pe un card SD permite analiza ulterioară a fluctuaţiilor temperaturii şi identificarea eventualelor surse de variaţie, precum şi prezenţa unui obstacol în raza senzorului PIR.

Acest proiect demonstrează capacitatea de a integra mai multe tehnologii într-o soluţie coerentă şi eficientă pentru rezolvarea unor probleme practice.



## Cap. 7. Bibliografie

- [1] <https://www.arduino.cc/reference/en/language/functions/math/map/>  
[2] [guide-english.pdf](#)

## Cap. 8. Anexe

Codul folosit pentru întregul proiect:

```
#include <SPI.h>
#include <SD.h>
#include <SimpleDHT.h>

int pinDHT11 = 2;
SimpleDHT11 dht11;
int m=3;
int pir=4;
#define CS_PIN 10

File dataLoggerFile;

int hasMoved;
String dataLoggerFileName = "logger.txt";

void setup()
{
    Serial.begin(9600);

    // SD Card initialization
    pinMode(CS_PIN, OUTPUT);
    if (SD.begin()) {
        Serial.println("SD Card is ready to use.");
    } else {
        Serial.println("SD Card initialization failed");
        return;
    }
    pinMode(pir, INPUT);
    pinMode(m, OUTPUT);
    delay(500);
}

void loop()
{
```

```
hasMoved = digitalRead(pir);
Serial.println(hasMoved);
byte temp = 0;
byte humidity = 0;
byte data[40] = {0};

if (dht11.read(pinDHT11, &temp, &humidity, data))
{
    Serial.print("Read DHT11 failed");
    return;
}
Serial.print((float)temp);
Serial.print(" *C, ");

dataLoggerFile = SD.open("logger.txt", FILE_WRITE);
if (dataLoggerFile) {
    dataLoggerFile.print((float)temp);
    dataLoggerFile.print("-");
    dataLoggerFile.println(hasMoved);
    Serial.println("Done");
}
else {
    Serial.println("error opening data logger file");
}
dataLoggerFile.close();

if((float)temp>=20)
    digitalWrite(m,HIGH);
else
    digitalWrite(m,LOW);
delay(1000);
}
```