

Temperature Control

Student: Ioan Cioraneanu
Grupa: 332CB

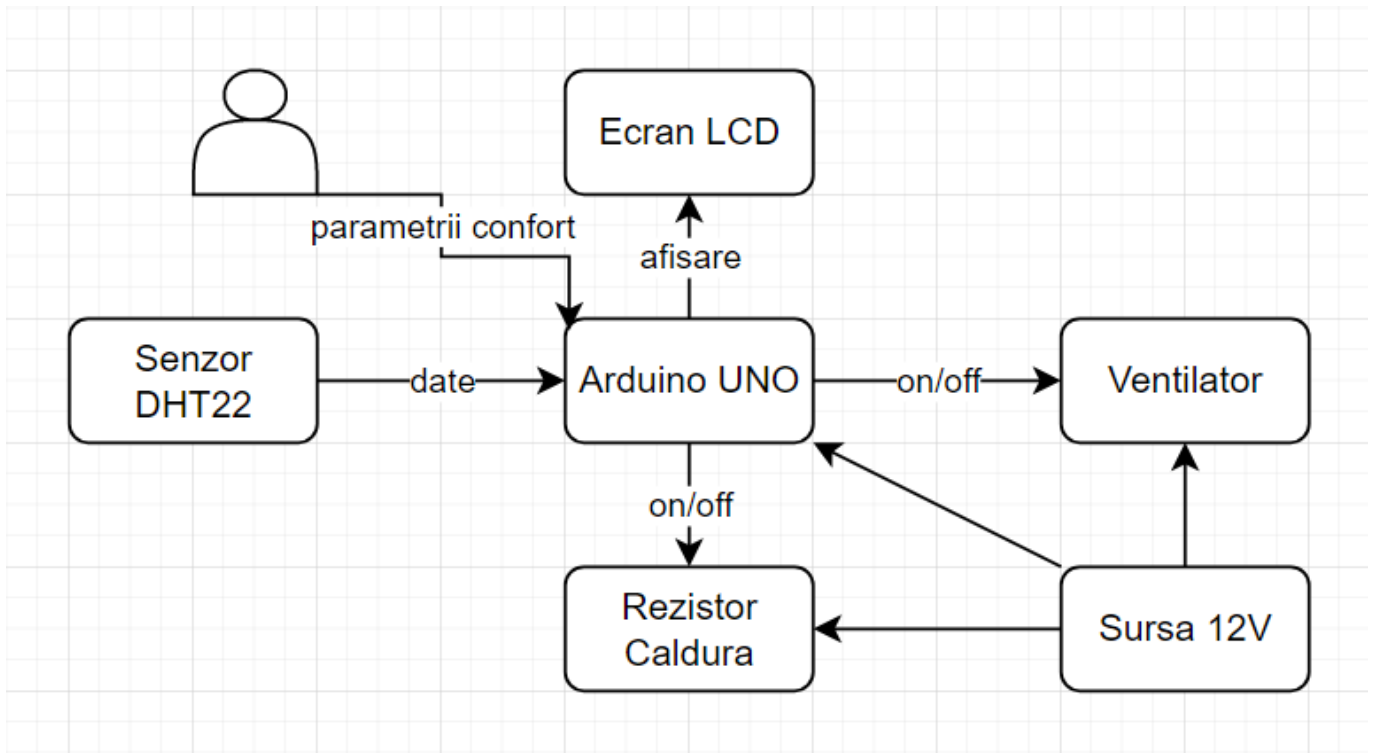
Introducere

Masurarea temperaturii si a umiditatii, afisarea lor pe un ecran LCD. Cand tempereatura nu este in parametrii de confort, un ventialor va racori camera sau un rezistor o va incalzi. Camera va fi o incinta mica pentru a se putea observa schimbarile de temperatura.

Descriere generală

Proiectul isi propune cotrolarea temperaturii intr-o incinta. Acesta se foloseste de un senzor pentru a masura temperatura, iar in cazul in care aceasta nu este in parametrii de confort va actiona incalzitorul sau ventilatorul pentru a redresa diferenta. Aplicabilitate: pentru a printa 3D se folosesc diferite materiale. Unele dintre acestea, precum abs sau nilon, au nevoie ca temperatura ambientala sa fie una ridicata si constanta. Astfel proiectul meu poate fi scalat, in functie de dimensiunile camerei necesare pentru a mentine aceasta temperatura ridicata.

Schema bloc

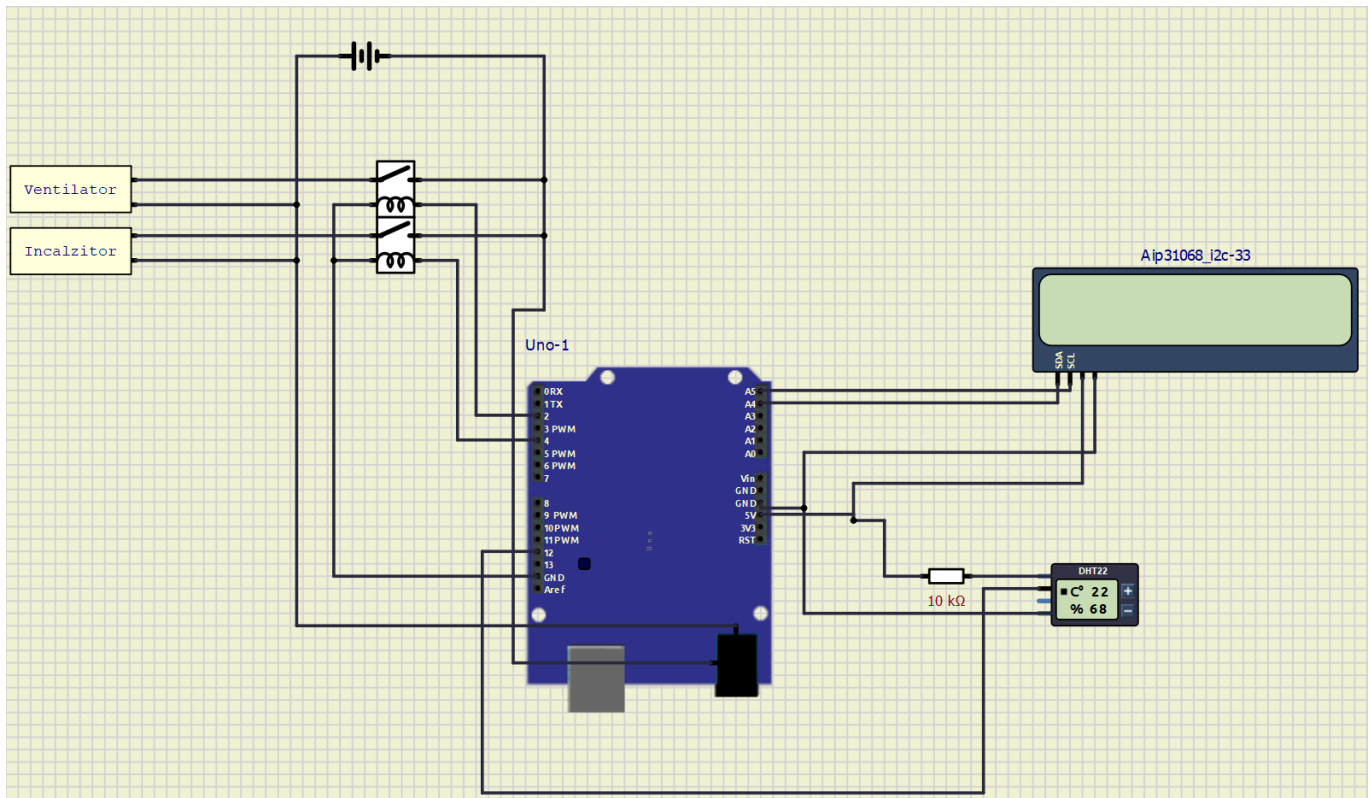


Hardware Design

Lista de piese:

- Arduino UNO
- breadboard
- ventilator
- heater + bloc aluminiu + radiator
- placa 4 rele 5V
- senzor temperatura + umiditate DHT22
- rezistoare
- fire

Schema electrica



Software Design

Dupa ce este setata temperatura de referinta circuitul functioneaza independent.

► pre-setup

- bibliotecile

```
#include <LiquidCrystal_I2C.h>
#include <DHT_U.h>
```

- se defineste senzorul de temperatura si se seteaza referinta(ref_temp)

```
#define DHTPIN 12
#define DHTTYPE DHT22
#define ref_temp 28.0
```

- se initializeaza senzorul si lcd-ul

```
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 20, 4);
```

- se definesc 2 variabile ce vor fi folosite pentru a temporiza heaterul

```
unsigned long startTime;
unsigned long currentTime;
```

► setup

- se inițializează senzorul si lcd-ul

```
dht.begin();

lcd.init(); // initialize the lcd
lcd.backlight();
lcd.setCursor(2,0);
lcd.print("Proiect PM 2022");
lcd.setCursor(0, 3);
lcd.print("Ioan Cioraneanu WOW");
delay(5000);
```

- se seteaza pinii de output

```
DDRD |= B00000000;
PORTD |= B00000100; // heater pin 2
PORTD |= B00010000; // fan pin 4

pinMode(2, OUTPUT);
pinMode(4, OUTPUT);
```

- se salveaza timpul de inceput ce va fi updatat de fiecare data cand heaterul porneste

```
startTime = millis();
```

► loop

- se citesc temperatura si umiditatea

```
float hum = dht.readHumidity();
float temp = dht.readTemperature();
if (isnan(hum) || isnan(temp)) {
  lcd.setCursor(0, 0);
  lcd.print("Read failed!");
  lcd.setCursor(0, 1);
  lcd.print("Please reset");
  return;
}
```

- se afiseaza temperatura, umiditatea si temperatura de referinta

```
lcd.setCursor(0,0);
lcd.print("Tmp: Hum: REF_TMP:");
lcd.setCursor(0,1);
lcd.print(temp);
lcd.setCursor(6,1);
lcd.print(hum);
lcd.setCursor(13, 1);
lcd.print(ref_temp);
```

- daca temperatura este mai mare se porneste ventilatorul

```
if (temp > ref_temp) {  
  PORTD &= B11101111;  
}  
else {  
  PORTD |= B00010000;  
}
```

- daca temperatura este mai mica si au trecut 10 secunde de la ultima alimentare se porneste incalzitorul

```
currentTime = millis();  
  
if (temp < ref_temp && currentTime - startTime > 10000) {  
  PORTD &= B11111011;           // start the heater  
  delay(500);                   // keep the heater on for 0.5 seconds  
  startTime = millis();        // measure the time to pause the heater  
  PORTD |= B00000100;         // stop the heater  
  
}
```

Bibliotecile

- *DHT.h* ⇒ folosirea senzorului DHT22 pentru înregistrarea temperaturii și umidității
- *LiquidCrystal_I2C.h* ⇒ folosirea display-ului LCD cu modul I2C

Mediu de Dezvoltare

Mediul de dezvoltare folosit în realizarea proiectului: **Arduino IDE**.

Rezultate Obținute

Am realizat 2 incinte. In una din ele se afla arduino, releele de control si sursa si in cea de-a doua termometrul, ventilatorul si incalzitorul. Pe incalzitor am prins si un bloc de aluminiu pe care am lipit un radiator pentru a disipa mai bine caldura in incinta.

Download

[temperature_control_cioraneanu_ioan.zip](#)

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2022/crisip/temperaturecontrol>



Last update: **2022/06/01 14:06**