

# Paiu Teofil : Weather Station

## Introducere

**Nume: Paiu Teofil**

**Grupa: 333CB**

**Indrumator: Daniel Dosaru**

## Descriere

Proiectul consta in crearea unui sistem de monitorizare a temperaturi si umidități dintr-o încăpăre si stocarea acestor informați într-o baza de date pentru a putea fi analizate in timp.

Pe lângă funcția de stocare acesta beneficiază de un ecran pentru afișajul actual, cat si o pagina web cu informațiile stocate in ultima ora de funcționare.

## Motivatie

Idea a plecat de la nevoia de cunoaștere a spațiului in care locuim si acomodarea acestuia. Pe viitor datele extrase pot fi folosite pentru controla diferite aparaturi pentru a crea un mediu comod.

Cred ca proiectul poate fi util si este o baza pentru pornirea diferitelor aplicați si automatizări ce țin de cunoașterea temperaturi si umidității din mediul înconjurător.

## Descriere generală

In cele ce urmeaza o sa detaliez schema bloc a proiectului

Celulele albastre reprezinta cele doua  $\mu\text{C}$  folosite ci anume Atmega328P (Arduino Uno R3) si ESP-8266 32bit (NodeMCU).

Celulele verzi reprezinta intrările care vor fi citite de  $\mu\text{C}$  in special semnalele butoanelor (prin întreruperi), senzorul de temperatura DTH11 (printr-un protocol definit in datasheet-ul sau) si o fotorezistenta intr-un ansablu divizor de tensiune cu ajutorului functionalitati DAC.

Celula galbena reprezinta iesirea  $\mu C$  reprezentata de un ecran LCD 16x02 pe care se vor afisa citirile actuale ale senzorului de temperatura.



## Hardware Design

In cele ce urmeaza o sa detaliez partea hardware a proiectului prezentand componentele cablajul si varianta finala a acestei etape.

Pentru realizarea designului hardware am folosit un tool numit Fritzing care mi-a oferit posibilitatea de a importa module proprii.

## Lista Componente

- 1 x Arduino Uno R3 cu Atmega328P
- 1 x NodeMCU cu ESP-8266 32bit
- 1 x DTH11 senzor
- 1 x fotorezistenta
- 3 x Butoane
- 5 x Rezistenta 1k $\Omega$
- 1 x Rezistenta 2k $\Omega$
- 3 x Rezistenta 10k $\Omega$
- 4 x Capacitorare ceramice 47nF
- 1 x LCD 1602 I2C
- 1 x Adaptor baterie 9V
- 30 x male-male jumping wires
- 4 x male-female jumping wires

## Conectivitate

- P2 : Se conecteaza buton pentru schimbare valori unitati de masura a temperaturi.
- P3 : Se conecteaza buton pentru activarea schimbari ceasului si selectia orei si minutelor.
- P4 : Se conecteaza buton pentru incrementarea valori orei/minutelor.
- P5 : Se conecteaza la portul TX al NODEMCU-ului.
- P6 : Se conecteaza la portul RX al NODEMCU-ului.
- P7 : Se conecteaza la pinul data al senzorului DTH22.
- P10: Se conecteaza prin modul PWM la pinul jumper A al modulului I2C backpack.
- A5 : Se conecteaza la pinul SCL a modulului I2C backpack
- A3 : Se conecteaza la fotorezistenta si se citeste valoarea analogica cu ajutorul DAC.
- A4 : Se conecteaza la pinul SDA a modulului I2C backpack

VCC: Se conecteaza la toate dispozitivele care au nevoie

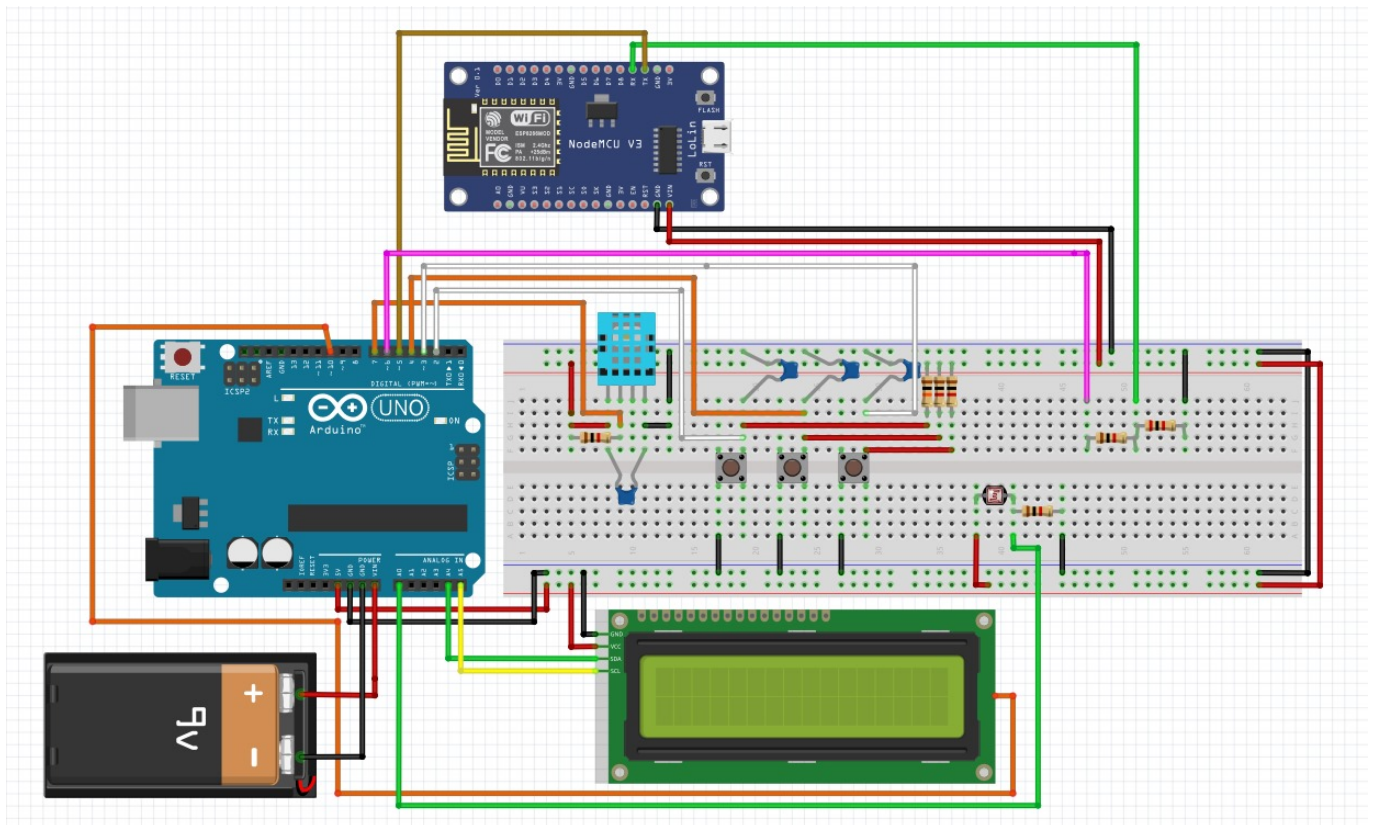
GND: Se conecteaza la toate dispozitivele care au nevoie este comun in tot circuitul

## Diagrama Circuit

Am ales sa folosesc niște filtre RC pentru a reduce efectul de debouncing al butoanelor. Pentru valorile alese am folosit un calculator online ([de aici](#)) pentru a avea un timp de aproximativ 30 de milisecunde pana la activare.

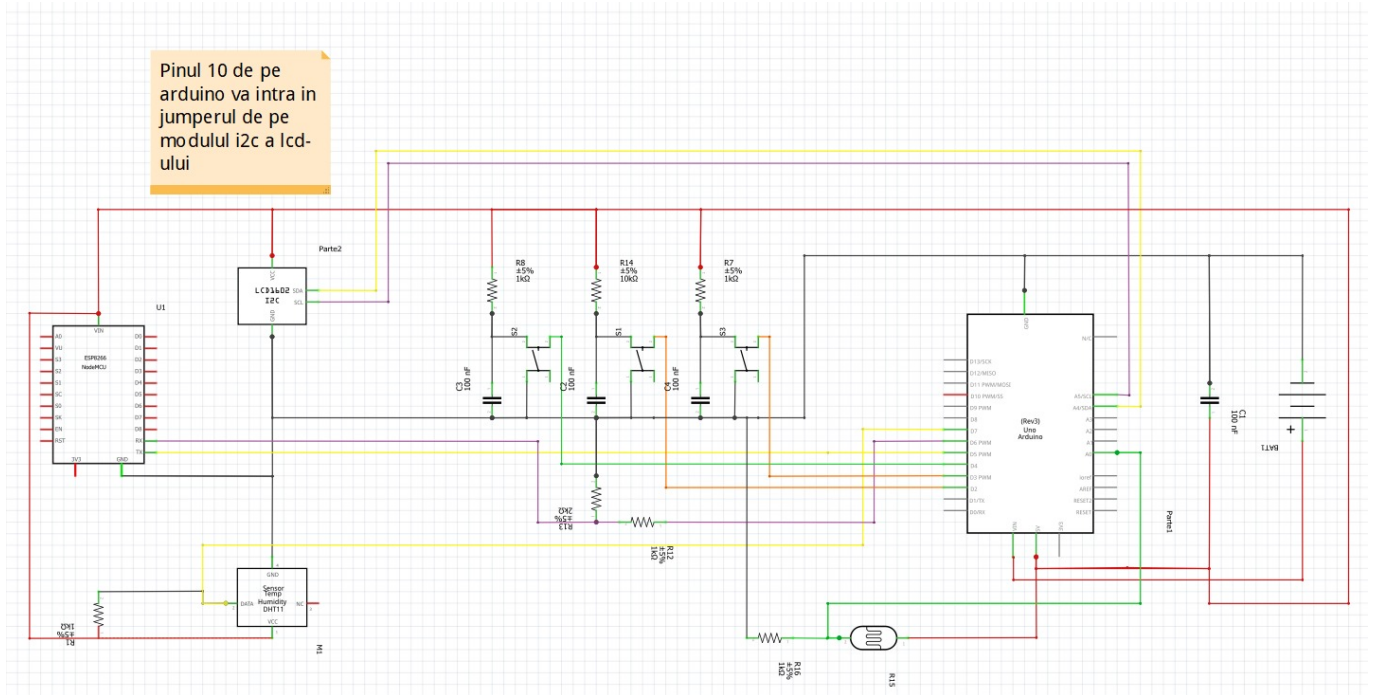
Circuitul mai dispune de voltage divider (cu un factor de divizare 2/3) pentru a realiza comunicația între Arduino (5V) și NodeMCU (3.3v) în mod sigur.

Conform datasheet-ului de la DHT11 acesta recomanda un condensator între VCC și GND pentru a filtra alimentarea și o rezistență pullup de 1kΩ pe data pin. Pentru butoane am ales rezistențe de 10kΩ pentru a tine intrarea sus.



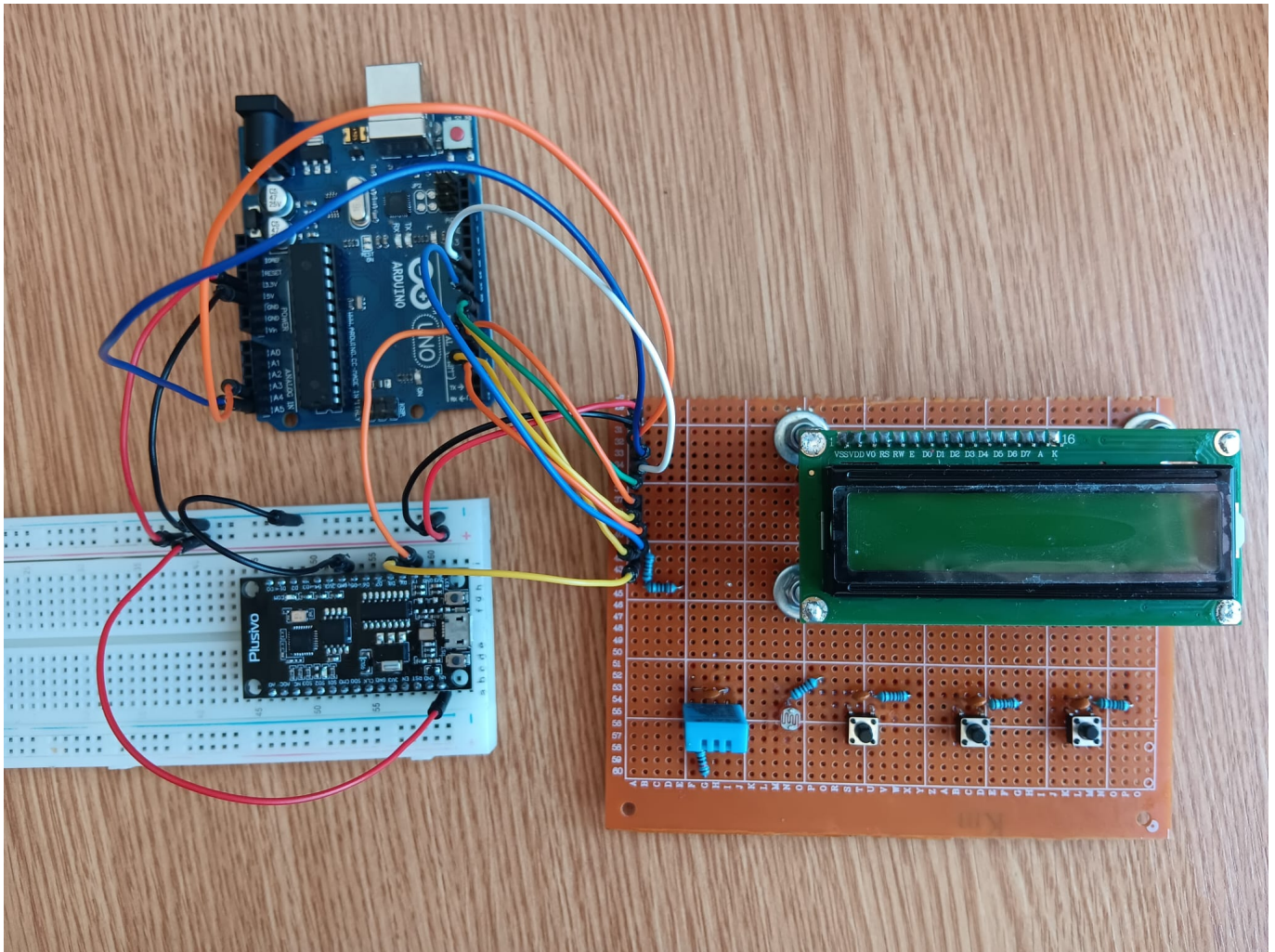
## Schema circuit

În același software ca mai sus am realizat și un cablaj al întregului proiect așa cum a fost prezentat mai sus.



## Implementare

Am ales sa folosesc un o placa de cablaj de test pe care am lipit componente si 2 headeri de pini a cate 5 intrari fiecare pentru a conecta arduinoul si placa NODE MCU. Rezultatul final al acestei etape a fost cel din poza de mai jos.



## Software Design

In aceasta parte a pagini o sa prezint firmware-ul aplicatiei si detaliera lui la nivel de structura si functionalitate. Acesta a fost dezvoltat in ARDUINO IDE cu ajutorul unor biblioteci exterioare.

## Structura

Pe partea software exista doua firmware-uri diferite unul pentru Arduino Uno R3 si unul pentru NodeMCU ESP8266 32bit:

- cel de arduino este responsabil sa comunice cu senzorul DTH11, butoanele, photorezistenta si ecranul lcd 16x02 prin modulul. Aceasta logica se encapsuleaza in clasa Board prin diferite metode si atribute care vor fi discutate mai jos. Pe langa sarcina  $\mu C$  de a comunica cu toate aceste componente, mai are taskul de a comunica serial la fiecare secunda temperatura si umiditatea pe serial la BAUD 9600.
- cel de nodemcu este responsabil sa recepteze datele comunicate si sa stocheze pana la 100 de inregistrari, sa hosteze o pagina web statica care afiseaza aceste date intr-un grafic care se

updateaza odata la 10 secunde.

## Descriere cod

Arduino Nano R3:

[board.h](#)

```
#ifndef BOARD_H
#define BOARD_H

#include <Arduino.h>

// module folosite in comunicarea cu ecranul LDC
#include <LiquidCrystal_I2C.h>
#include <Wire.h>

// modul folosit in comunicarea cu arduino-ul
#include <SimpleDHT.h>

// toata logica senzorilor si modulelor de pe arduino
class Board {
public:
    Board(uint8_t lcdAddr, uint8_t lcdCols, uint8_t lcdRows, uint8_t
backlightPin, uint8_t dhtPin,
uint8_t photoPin, uint8_t btn3Pin, uint8_t txPin);

    // getteri la variabilele private temp si humidity
    byte readTemperature();
    byte readHumidity();

    // functie initializeaza intreruperile, porturile si comunicatile
    void begin();

    // functie ce citeste valorile senzorilor si updateaza ecranul
    void update();

private:
    // obiectele ecranului si a senzorului DHT11
    LiquidCrystal_I2C lcd;
    SimpleDHT11 dht11;

    // variabile pentru ora si minute
    uint8_t hour, minute;
    unsigned long prev_millies;

    // variabilele pentru stocarea citirilor
    byte temp, humidity;
```

```
// diferiti pini ce se folosesc pentru a seta intreruperi sau a citi  
sau scrie valori  
uint8_t photoPin, backlightPin, dht_pin, btnIncrement, btnTime,  
btnTemp, txPin;  
  
// functie ce updateaza ecranul  
void updateScreen();  
  
// functii ce se ocupa de starea unei intreruperi  
void handleTime();  
void incrementX();  
void handleTemp();  
};  
  
// functile pe care le dau intreruperilor  
void changeTemp();  
void changeTime();  
  
extern uint8_t setTime;  
extern bool celsius;  
  
#endif
```

NodeMCU ESP8266 32bit:

[main.cpp](#)

```
// include biblioteca server  
#include <ESP8266WebServer.h>  
  
// trebuie sa fie 2.4Gz  
const char* ssid = "ssid_router";  
const char* password = "password";  
  
// pe portul 80 al placutei  
ESP8266WebServer server(80);  
  
// modelul unui punct  
struct SensorData {  
    float temperature;  
    float humidity;  
    unsigned long timestamp;  
};  
  
// pagina http statica  
String page = R"  
<html lang='en'>  
<head>  
    ...
```

```
</head>
<body>
    ...
</body>
</html>
)";

// dimensiunea si array-ul in care se stocheaza datele
const int dataSize = 100;
SensorData data[dataSize];

// endpoint catre pagina
void htmlIndex();

// cerere get pentru data
void handleData();

// conexiune wifi
void connectToWiFi();

// seteaza rutelele serverului si porneste
void setupServer();
```

## Biblioteci utilizare

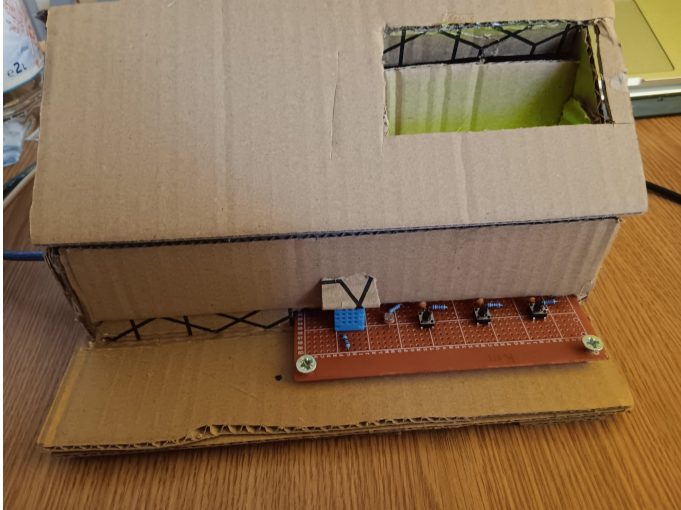
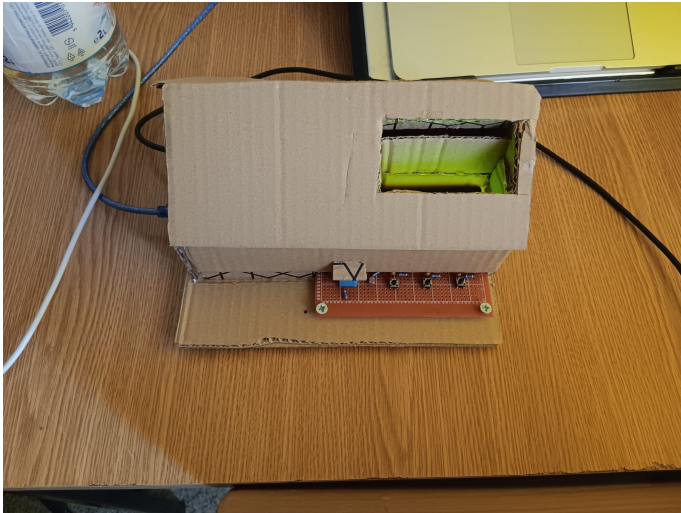
- LiquidCrystal\_I2C.h
- Wire.h
- SimpleDHT.h
- Arduino.h
- ESP8266WebServer.h

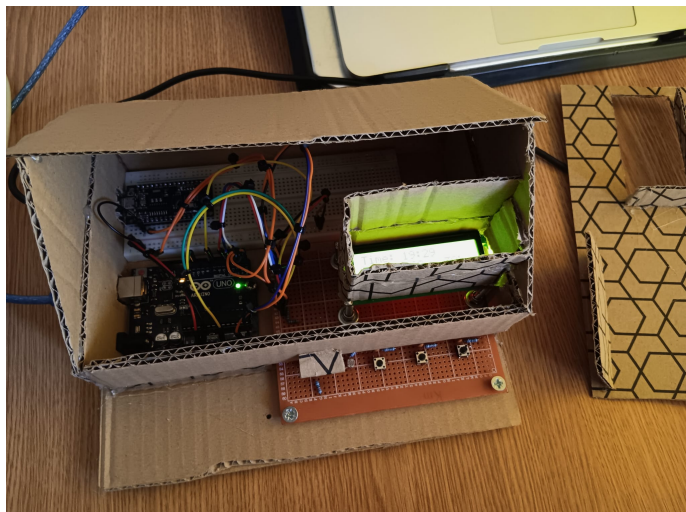
## Rezultate Obținute

In urma realizarii proiectului se pot vizualiza rezultatele obtinute la urmatorul video:

De asemenea o fotografie cu acesta:







si interfata web:



## Download

Toate fisierele care au fost realizate in acest proiect se regasesc in arhiva de mai jos care cuprinde:

- un fisier README care explica pasi de importare a bibliotecile externe si de compilare si rulare.
- un director LIBS care contine bibliotecile externe.
- un director src care contine codul proiectului (partea software).
- un fisier PM\_final.fzz care contine schema proiectului (partea hardware).
- un fisier PM.drawio care contine diagrama de context.

Arhiva se afla aici → [Arhiva.zip](#)

## Bibliografie/Resurse

- [Documentatia in format pdf](#)
- <https://garrysblog.com/2022/06/30/dimming-16x2-and-20x4-lcd-displays/>
- <https://www.youtube.com/watch?v=EfzZOiNBQml&t=1s>
- <https://pdfcoffee.com/guidepdf-7-pdf-free.html>

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

Permanent link:  
<http://ocw.cs.pub.ro/courses/pm/prj2024/ddosaru/teofil.paiu>

Last update: **2024/05/21 18:02**

