Stație meteorologică IoT monitorizată prin intermediul unei interfețe web

PROIECT RC

Autor: **Emanuel Robert LUCACI**

Conducător științific: **ȘL. dr. ing. Valentin SITA**

Cuprins

[1 Introducere](#_Toc155287478) 2

[1.1 Context general](#_Toc155287479) 2

[1.2 Obiective](#_Toc155287480) 2

[1.3 Specificații](#_Toc155287481) 3

[2 Studiu bibliografic](#_Toc155287482) 5

[3 Analiză, proiectare, implementare](#_Toc155287483) 6

[3.1 Proiectare și implementare hardware](#_Toc155287481) 6

[3.2 Proiectare și implementare software](#_Toc155287481) 7

**4 MANUAL DE UTILIZARE**………………………………………………………………………………………**12**

[4.1 Setare la prima rulare](#_Toc155287481) 12

[4.2 Folosirea stației meteorologice](#_Toc155287481) 13

[5 Concluzii](#_Toc155287484) 15

[5.1 Rezultate obținute](#_Toc155287485) 15

[5.2 Direcții de dezvoltare](#_Toc155287486) 15

[6 Bibliografie](#_Toc155287495) 16

# Introducere

## Context general

Internet of things(IoT) reprezintă un termen general care acoperă orice domeniu în care o serie de dispozitive sunt conectate între ele cu ajutorul internetului pentru a putea colecta și transfera date. Un sistem IoT are patru componente principale: senzorii, partea de comunicare, modul de transmitere a informației și procesarea/afișarea datelor către utilizatori.

În ultimii ani, IoT a cunoscut întrebuințări din ce în ce mai frecvente în toate sectoarele industriale, de la comerț până la asistență medicală și multe altele. Un bun exemplu poate fi considerat viitorul oraș inteligent, care cu ajutorul tehnologiei IoT poate rezolva diferite probleme precum: reglementarea traficului rutier, folosirea sistemelor de iluminat inteligente pentru a economisi energie sau chiar gestionarea deșeurilor urbane.

Realizarea unui proiect cu o astfel de tehnologie aduce numeroare beneficii, printre care: eficiență operațională îmbunătățită, posibilitatea de monitorizare și management de resurse, costuri reduse de producție și capacitatea de a îmbunătăți continuu experiența utilizatorilor.

În această lucrare se va urmări dezvoltarea unei stații meteorologice IoT care permite monitorizarea informațiilor obținute de la senzori atât prin intermediul unui display LCD, cât și prin internetul unei interfețe web. Acest proiect permite astfel monitorizarea exactă a diferiți parametrii precum: temperatura, umiditatea, luminozitatea și factorul de precipitație.

În cele ce urmează se susțin obiectivele propuse pentru acest proiect, urmate de specificațiile necesare și detalierea modului de implementare a lucrării. În final se realizează un manual de utilizare și un set de concluzii cu privire la rezultatele obținute și la posibilele direcții de dezvoltare.

## Obiective

Obiectivul lucrării constă în dezvoltarea unei aplicații IoT ce permite monitorizarea web a temperaturii, umidității, luminozității și a factorului de precipitație în mediul în care se folosește dispozitivul. Totodată, există și alte obiective generale ale stației meteorologice, printre care:

1. **Familiarizare cu concepte de electronică și programare**

* înțelegerea modului de funcționare a plăcii de dezvoltare arduino cu ethernet shield cât și modul de măsurare a valorilor aferente fiecărui sensor
* programare în mediul arduino ide folosind limbajul C++

1. **Colectarea și afișare datelor legate de vreme**

* datele colectate de la sensori sunt transmise mai departe pentru a putea fi afișate utilizatorului atât pe un display cât și pe o interfață web

1. **Eficiență energetică și cost de producție redus**
2. **Capacitatea de modificare și personalizare ulterioară a proiectului**
3. **Măsurarea precisă a parametrilor dintr-o locație**

* Uneori predicțiile meteo disponibile online nu sunt exacte

## Specificații

În realizarea acestui proiect s-au folosit următoarele tipuri de resurse:

1. Hardware:

* senzor de temperatură DHT11 pentru măsurarea temperaturii și umidității <https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>
* senzor de detectare a picăturilor de ploaie <https://www.openhacks.com/uploadsproductos/rain_sensor_module.pdf>
* senzor detector de lumină

<https://5.imimg.com/data5/XP/IG/AR/SELLER-1225448/ldr-sensor-module.pdf>

* display 16x2 cu interfață serială I2C pentru afișarea parametrilor local prin folosirea a doar 4 fire <http://handsontec.com/dataspecs/module/I2C_1602_LCD.pdf>
* placă de dezvoltare Arduino Uno folosită pentru a controla senzorii și pentru a transmite datele spre display și pe interfața web <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>
* modul Arduino Ethernet Shield pentru a permite conectarea la rețea și comunicare pe un mediu web

<https://www.farnell.com/datasheets/1682241.pdf>

* adițional se folosesc și leduri, fire și un breadboard

1. Software:

* Mediul de dezvoltare software Arduino Ide care are rolul de a încărca codul în memoria plăcii
* codul sursă încărcat în memoria existentă a plăcii de dezvoltare Arduino care are rolul de a prelua datele primite de la senzori pentru a le afișa ulterior pe display și pe interfața web
* bibliotecile adiționale utilizate în cod pentru buna funcționare a tuturor sarcinilor: <DHT.h> pentru folosirea diferitor funcții asupra senzorului de temperatură și umiditate DHT11, <LiquidCrystal\_I2C.h> pentru a putea folosi display-ul alături de modului I2C și <Ethernet.h> alături de <SPI.h> pentru a putea comunica și transmite date în rețea

1. Cunoștinte de bază necesare:

* principii de bază legate de electronică
* programare în limbajul C++
* cunoștințe minimale în HTML pentru realizarea interfeței web

# Studiu bibliografic

La baza acestui proiect se află o serie de resurse științifice care au fost studiate cu scopul de a înțelege și îmbogății cunoștințele necesare realizării unei astfel de lucrări. Astfel, în următoarele rânduri se vor expune materialele folosite:

1. Înțelegerea conceptului de IoT:

* <https://www.researchgate.net/publication/351003790_Internet_of_Things_IoT>: acest articol a fost realizat de către autorul Radouan Ait Mouha în anul 2021 și are scopul de a familiariza cititorii cu privire la ce înseamnă Internet of Things, ce fel de arhitecturi sunt folosite, componentele ce pot apărea într-un proiect IoT și ce tipuri de aplicații pot fi create

1. Folosirea plăcii de dezvoltare Arduino:

* <https://www.uio.no/studier/emner/matnat/ifi/IN1060/v21/arduino/arduino-projects-book.pdf>: cartea oficială în format electronic care explică la nivel fundamental cele mai importante concepte Arduino: de la explicarea elementelor existente pe placă și până la realizarea diferitor teme pentru acomodarea cu o serie vastă de componente; acestea din urmă sunt explicate la nivel electronic și există de asemenea cod sursă pentru a facilita mediul de învățare

1. Foile de catalog care explică modul de funcționare a senzorilor folosiți:

* <https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>: pentru senzorul de temperatură și umiditate DHT11
* <https://www.openhacks.com/uploadsproductos/rain_sensor_module.pdf>: pentru senzorul de detectare a picăturilor de ploaie
* <https://5.imimg.com/data5/XP/IG/AR/SELLER-1225448/ldr-sensor-module.pdf>: pentru senzorul de lumină

1. Alte componente utilizate:

* <http://handsontec.com/dataspecs/module/I2C_1602_LCD.pdf>: display 1602 cu modul I2C pentru a afișa datele primite de la senzori
* <https://www.farnell.com/datasheets/1682241.pdf>: Ethernet Shield pentru conectarea în rețea

1. Pentru software:

* <https://www.arduino.cc>: site-ul oficial arduino care conține exemple și explicații cu privire la codul introdus în mediul Arduino Ide

# Analiză, proiectare, implementare

## Proiectare și implementare hardware

Schema de mai sus este o reprezentare grafică a proiectului realizat în fritzing, aceasta fiind creată cu scopul de a permite o urmărire mai ușoară a conexiunilor realizate. Pentru a putea controla și obține date de la senzori, se alege folosirea unui Arduino Uno alături de modulul Ethernet Shield care va fi montat peste placa de dezvoltare. La nivel de comandă, pinii sunt configurați astfel:

* A3: este pinul de intrare care va primi date de la senzorul de ploaie; acesta transmite valori cuprinse între 0 și 1023, astfel se poate configura modalitatea de a detecta apariția precipitațiilor, în condițiile în care nu există picături de ploaie pe placă, se transmite valoarea maximă 1023 iar aceasta scade în funcție de cât de intense sunt precipitațiile în mediul de testare
* A4 și A5: acționează ca pinii SDA(Serial DAta line), respectiv SCL(Serial CLock line) pentru modulul I2C conectat la display; acești pini au ca scop transmiterea atât a semnalului de clock cât și a datelor pentru a putea fi afișate
* D2: este pinul de alimentare al unuia dintre leduri, acest led va fi oprit în momentul în care senzorul LDR detectează lumină iar în absența luminii, ledul se va porni
* D3: este pinul de intrare al senzorului de lumină LDR, prin intermediul acestui pin vom primi una dintre valorile 0 și 1 în funcție de prezența sau absența luminii din mediul de testare al dispozitivului
* D4: este al doilea led prezent în proiect, acesta va semnala dacă există sau nu precipitații; când valoarea transmisă de senzorul de ploaie este cuprinsă între [700,1023] ledul va fi oprit iar între [0,700) acesta se va porni, evident aceste valori de trecere de la o stare la alta nu sunt fixe, ci pot fi modificate conform preferințelor fiecărui proiect
* D5: este pinul de intrare la senzorului DHT prin intermediul căruia se vor transmite valorile temperaturii și umidității din mediu
* D10, D11, D12: deși accesibili prin intermediul Ethernet Shield-ului, acești pini sunt de regulă folosiți de placa de rețea pentru a comunica cu Arduino Uno
* 5V și GND: sunt pinii folosiți care permit alimentarea tuturor componentelor prin intermediul plăcii de dezvoltare care, la rândul ei, va fi alimentată fie prin cablu USB conectat un calculator, fie printr-o baterie externă/sursă care asigură 5V

## Proiectare și implementare software

Pentru a putea avea control asupra componentei hardware, se va folosi mediul de implementare Arduino Ide pentru a scrie codul necesar care va fi ulterior încărcat în memoria plăcii de dezvoltare.

#include <LiquidCrystal\_I2C.h>

#include <DHT.h>

#include <DHT\_U.h>

#include <Ethernet.h>

#include <SPI.h>

#define DHT11PIN 5

#define DHTTYPE DHT11

//Temp and humidity

DHT temp(DHT11PIN,DHTTYPE);

LiquidCrystal\_I2C lcd(0x27,16,2);

//Rain In/OUT

int rain\_input=A0;

int LED\_rain=4;

//Light In/Out

int light\_input=3;

int LED\_light=6;

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

IPAddress ip(172,27,164,81); //ip to modify

EthernetServer server(80);

În liniile de cod anterioare se fac setările necesare: se introduc bibliotecile folosite, se inițializează pinii de intrare/ieșire aleși anterior la proiectarea hardware, se inițializează modulul LCD, se setează o adresă generică de mac alături de o adresă de IP și se pornește serverul ethernet pe portul ales 80.

void setup() {

  Ethernet.init(10);

  Ethernet.begin(mac,ip);

  server.begin();

  Serial.begin(9600);

  lcd.init();

  lcd.backlight();

  pinMode(rain\_input,INPUT);

  pinMode(LED\_rain,OUTPUT);

  pinMode(light\_input,INPUT);

  pinMode(LED\_light,OUTPUT);

  temp.begin();

}

În funcția setup se inițializează modulul de ethernet cu adresa de mac și ip-ul ales, se pornește serverul, serial monitor-ul, lcd-ul și se definește fiecare pin dacă este de intrare sau ieșire.

void loop() {

  lcd.clear();

  int value\_rain = analogRead(rain\_input);

  int value\_light = digitalRead(light\_input);

  int hum\_temp = temp.read(DHT11PIN);

  float hum = temp.readHumidity();

  float temper = temp.readTemperature();

  String isRaining;

În funcția loop se creează variabilele care se încarcă cu valorile primite de la toți senzorii.

EthernetClient client = server.available();

  if (client)

    {

      boolean currentLineIsBlank = true;

      while (client.connected ( ) )

        {

          if (client.available ( ) )

            {

              char character = client.read ( );

              Serial.write(character);

              if (character == '\n' && currentLineIsBlank)

                {

                  client.println ("HTTP/1.1 200 OK");

                  client.println ("Content-Type: text/html");

                  client.println ("Connection: close");

                  client.println ("Refresh: 5");

                  client.println ( );

                  client.println ("<!DOCTYPE HTML>");

                  client.println ("<html>");

                  client.print ("<Title>Arduino Weather Station</Title>");

                  client.print ("<h1>Weather Station</h1>");

                  client.print ("<h4>Temperature in C: ");

                  client.print (temper);client.print("C");

                  client.print ("</h4><h4>Humidity: ");

                  client.print (hum);client.print("%");

                  client.print ("</h4><h4>Rain factor is:");

                  client.print (value\_rain);

                  client.print ("</h4><h4>Is it raining? ");

                  if(value\_rain<700){

                    isRaining="YES";

                    client.print(isRaining);

                     }

                     else

                     {

                        isRaining="NO";

                        client.print(isRaining);

                     }

                  client.print ("</h4><h4>Is it sunny outside? ");

                  if(value\_light)

                    client.print("No, its dark");

                    else

                    client.print("Yes, it's sunny outside");

                  client.println ("<br />");

                  client.println ("</html>");

                  break;

                }

                if ( character == '\n')

                  {

                    currentLineIsBlank = true;

                  }

                else if (character != '\r')

                  {

                    currentLineIsBlank = false;

            }

        }

    }

    delay(1);

    client.stop();

  }

În această etapă, se setează serverul ca fiind accesibil și cu ajutorul acestei secvențe de cod se creează o simplă pagină web al cărei limbaj HTML e transmis prin printări succesive. După cum se poate observa, se afișează valorile tuturor parametrilor și apar și două mesaje adiționale: unul spune dacă plouă sau nu afară în funcție de valoarea analog primită de la senzor iar celălalt mesaj semnalează prezența sau lipsa luminii în mediul de testare folosit. Un aspect important de menționat este că valorile transmise sunt reactualizate la fiecare 5 secunde.

Serial.print("Rain factor: ");

  Serial.println(value\_rain);

  Serial.println("Is it raining?");

  if(value\_rain<700){

    digitalWrite(LED\_rain,HIGH);

    isRaining="YES";

    Serial.println(isRaining);

  }

  else

  { digitalWrite(LED\_rain,LOW);

    delay(100);

    isRaining="NO";

    Serial.println(isRaining);

  }

//////LIGHT/////////

  if(value\_light==1){

    digitalWrite(LED\_light,HIGH);

    Serial.println("It's dark outside");

  }

  else

  {

    digitalWrite(LED\_light,LOW);

    Serial.println("It's light outside");

  }

/////HUMIDITY and TEMPERATURE/////////

  Serial.print("Humidity (%): ");

  Serial.println(hum, 2);

  delay(100);

  Serial.print("Temperature  (C): ");

  Serial.println(temper, 2);

Această secvență de cod este complet opțională pentru că în etapele incipiente ale proiectului, afișarea datelor se realiza exclusiv în Serial Monitor din mediul de dezvoltare Arduino Ide. Decizia de a păstra încă această secvență stă la baza faptului că, în cazul în care se dorește reproducerea proiectului de către un cititor, pentru simplitate se poate folosi doar această secțiune pentru afișarea datelor.

///LCD///

  lcd.setCursor(0,0);

  lcd.print("T:");

  lcd.print(temper, 2);

  lcd.setCursor(0,1);

  lcd.print("H:");

  lcd.print(hum, 2);

  lcd.setCursor(9,0);

  lcd.print("L:");

  lcd.print(value\_light);

  lcd.setCursor(9,1);

  lcd.print("R:");

  lcd.print(value\_rain);

   delay(3000);

}

Această ultimă secvență de cod afișează valorile de la senzori pe display-ul incorporat în proiect pentru a putea folosi stația meteorologică chiar și în lipsa unei conexiuni la rețea.

# Manual de utilizare

## Setare la prima rulare

Pentru a putea folosi stația meteorologică, este necesar accesul la componenta hardware realizată. Pentru a introduce secvența de cod pentru prima dată în memoria plăcii de dezvoltare, se va utiliza un cablu USB pentru a se putea conecta la un computer. Totodată, este necesară instalarea mediului de lucru Arduino Ide, disponibil pe site-ul oficial. Înainte de a introduce și rula codul, este nevoie să introducem bibliotecile folosite, asftel se navighează în secțiunea Sketch –> Include Library –> Manage Libraries conform figurii 4.1.

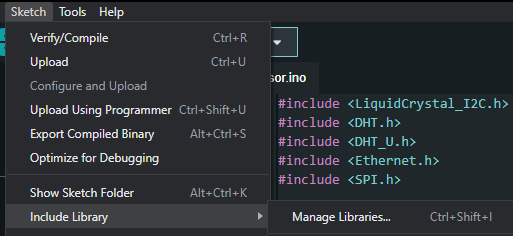


Figura 4.1. Introducerea bibliotecilor

Din acest punct se va deschide un tab nou de unde se vor căuta și instala toate bibliotecile necesare rulării programului.

Următorul pas înainte de încărcarea programului constă în modificarea adresei ip din cod. În cazul în care nu există un număr mai mare de proiecte IoT conectate la rețea, cel mai ușor mod de a obține adresa ip dorită este de a accesa din Arduino Ide secțiunea Files –> Examples –> Ethernet –> DhcpAdressPrinter iar în urma rulării acestui exemplu se poate identifica adresa ip necesară pentru ca programul să funcționeze în mediul web.

O alternativă a metodei anterioare constă în accesarea setărilor de internet din Network and Internet –> Network Connections de unde se accesează opțiunea Properties a rețelei de internet. În continuare, se urmărește figura 4.2: se selectează opțiunea TCP/IPv4 care va deschide o fereastră nouă de unde se va selecta opțiunea ,,Use the following IP adress”. În cazul în care primele 3 seturi de valori din ip vă sunt necunoscute, acestea se pot obține deschizând un command prompt și utilizarea comenzii ,,ipconfig”. Ultimul set de cifre din IP este ales de către utilizator între intervalul 0 și 255 conform preferințelor iar subnet mask-ul se va introduce automat.

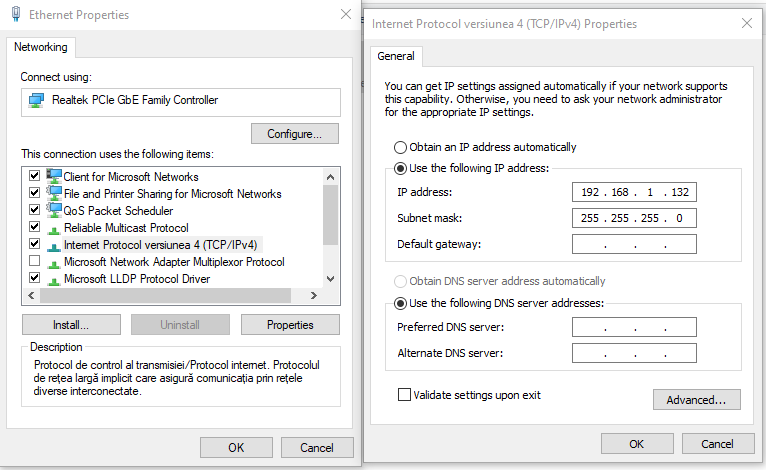


Figura 4.2. Setarea adresei IP

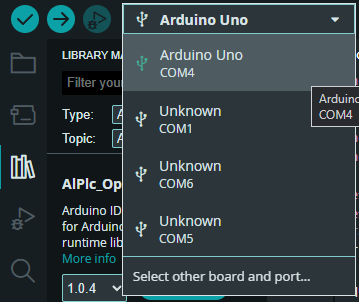
Următorul pas constă în încărcarea codului sursă pe placa de dezvoltare, lucru care se poate realiza prin selectarea plăcii folosite și selectarea portului conform figurii 4.3 și în final se încarcă codul folosind butonul de upload.

Figura 4.3. Alegerea plăcii și portului adecvat

Toți acești pași se realizează doar la prima rulare a proiectului sau în cazul în care dorim să folosim stația meteo într-o locație diferită.

## Folosirea stației meteorologice

Pentru a conecta dispozitivul la internet, este necesar să asigurăm o conexiune la router print-un cablu ethernet și să alimentăm placa de dezvoltare printr-un cablu USB. Alimentarea se poate face direct de la un computer însă pentru a economisi energie și pentru a asigura o mai bună mobilitate a proiectului, se poate alimenta cu o sursă sau cu o baterie externă standard. Odată alimentat, pe display se afișează valorile primite de la senzori după cum se poate observa în figura 4.4.



Figura 4.4. Valorile afișate pe display

Totodată, dacă intrăm în orice browser de internet și introducem adresa ip în bara de căutare, se va putea observa interfața web a aplicației care ne arată atât parametrii de temperatură, umiditate și factor de ploaie, cât și o serie de mesaje care ne înștințează dacă în mediul de testare există precipitații și dacă este lumină. Reactualizarea parametrilor se face în mod automat odată la 5 secunde,

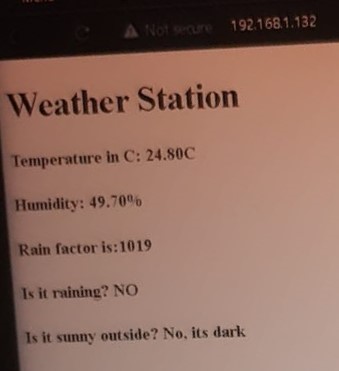


Figura 4.5. Interfața web

# Concluzii

## Rezultate obținute

În această lucrare de stație meteorologică s-au atins obiectivele setate, reușindu-se asftel crearea unei modalități simple și eficiente de a monitoriza diferiții parametrii de mediu. Pe lângă acestea, costul redus al producerii, acuratețea măsurătorilor și posibilitatea de a modifica sau adăuga elemente face ca acest proiect să devină un dispozitiv folositor în viața de zi cu zi.

În urma realizării acestui proiect nu se obține doar un dispozitiv IoT folositor, ci se pot dobândi diferite abilități și informații precum:

* folosirea mediul de dezvoltare Arduino
* înțelegerea modului de funcționare a diferiților senzori
* familiarizare cu elemente electronice

## Direcții de dezvoltare

Cu toate că obiectivele propuse au fost îndeplinite, acest proiect are capacitatea de a fi dezvoltat în viitor în numeroase moduri:

* adăugarea mai multor senzori, precum senzori de vânt și presiune atmosferică
* crearea unui aspect de produs de vreme ce această lucrare este un prototip
* stocarea datelor și transmiterea acestora într-un model AI(Artificial Intelligence) care să fie capabil să aducă anumite preziceri cu privire a vremii în funcție de evoluția parametrilor în timp

# Bibliografie

[1] S. Fitzgerald and M. Shiloh, Arduino Projects Book, 2012

[2] R.A. Mouha, Internet of Things (IoT) Journal of Data Analysis and Information Processing, 2021

[3] Bella, H. & Khan, Mohammed & Naidu, M. & Jayanth, Digumarti & Khan, Yasir, Developing a Sustainable IoT-based Smart Weather Station for Real Time Weather Monitoring and Forecasting, E3S Web of Conferences, 2023

[4] Temilola, Adepoju & Oladele, Matthias & Kasali, Abdulwakil & Fabiyi, Gbenga, Development of a Low-Cost Arduino-Based Weather Station, FUOYE Journal of Engineering and Technology, 2023