

Universitatea Tehnică din Cluj-Napoca

Catedra de Calculatoare și Tehnologia Informației

Aplicație Web pentru supravegherea unei sere

Student: Trifu Diana Maria

Grupa: 30233

Îndrumător de proiect: Itu Răzvan

Data: 5 ianuarie 2021

MINISTERUL EDUCAȚIEI NAȚIONALE



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Cuprins

1. INTRODUCERE	ERROR! BOOKMARK NOT DEFINED.
2. IMPLEMENTARE	4
3. SCHEMA CIRCUITULUI	11
4. REZULTATE	ERROR! BOOKMARK NOT DEFINED.
BIBLIOGRAFIE	12
ANEXĂ	16

1. Introducere

Pentru acest proiect am ales să dezvolt o aplicație web care oferă utilizatorului informații legate de diverși parametrii importanți ai seriei acestuia, cum ar fi temperatura, umiditatea din atmosferă și umiditatea solului. Pentru realizarea acestui proiect am folosit următoarele componente: o placă Arduino Mega 2560, un modul Lua ESP8266MOD Wireless 802.11 b/g/n Wi-Fi Direct ESP8266 CP2102, 3 led-uri tricolor cu catod comun, un senzor temperatură și umiditate digital DHT11, un modul senzor umiditate sol – higrimetru, 9 rezistori 220R, 30 x fire dupont tată-tată, 10 x fire supont mamă-tată și două breadboard-uri. Server-ul web creat cu ajutorul modului Lua ESP8266MOD Wireless va putea fi accesat cu orice dispozitiv care are un browser în rețeaua locală.

2. Implementare

În cele ce urmează voi prezenta pe scurt componentele utilizate pentru realizarea acestui proiect, dar și metoda în care am ales să implementez soluția aleasă.



Placă Arduino Mega 2560



Modul senzor umiditate sol - higrmetru



Senzor temperatură și umiditate digital DHT11



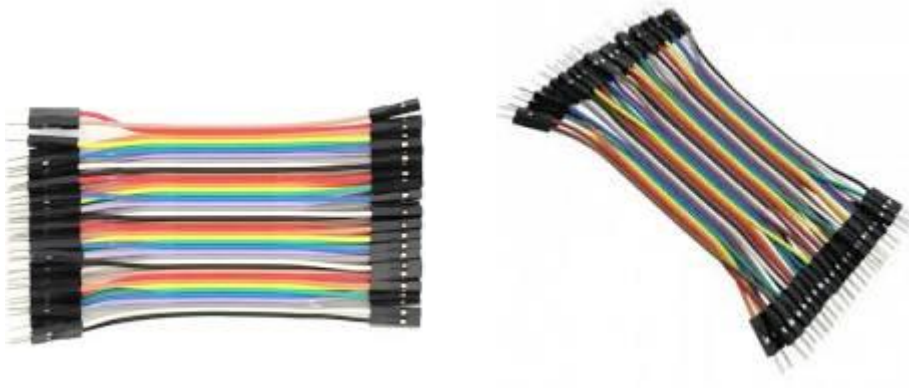
Modul Lua ESP8266MOD Wireless



Led tricolor cu catod comun



Rezistor 220R



Fire dupont tată-tată și mamă-tată

Pentru implementarea am folosit IDE-ul Arduino. Proiectul conține 2 fișiere cu extensia “.ino”: unul pentru modulul Lua Modul Lua ESP8266MOD Wireless, iar cel de-al doilea pentru placa Arduino.

```
//COD PLACA ARDUINO MEGA 2560
#include "dht.h"
#define dht_apin A0

dht DHT;
//transmitem catre modulul ESP8266
String str;
int moisturePin = A1;
int sensorValue =0;
int percentM = 0;
//led temperatura
int redPin = 9;
int greenPin = 10;
int bluePin = 11;
//led umiditate
int redPin1 = 12;
int greenPin1 = 13;
int bluePin1 = 8;
//led umiditate sol
int redPin2 = 2;
int greenPin2 = 3;
int bluePin2 = 4;

void setup() {

    Serial.begin(115200);
    Serial1.begin(115200);
    delay(500);//Delay to let system boot
    Serial.println("Wait for accessing sensor...\n\n");
    delay(1000);//Wait before accessing Sensor

}
```

În fișierul care conține codul care se încarcă pe placa Arduino am inclus o bibliotecă specială(„dht.h”) pentru folosirea senzorului de temperatură și umiditate digital DHT11. Apoi am definit portul A0 de pe plăcuță ca fiind cel care preia datele de la senzor. Avem nevoie string-ul **str** pentru a transmite datele citite de senzor către modulul ESP8266 Wireless, de parametrul **DHT** de tip „dht” pentru senzorul temperatură și umiditate, câte 3 parametri de tip „int” pentru fiecare led(cate unul pentru fiecare pin al led-ului) și 3 parametri de tip „int”: **moisturePin** pentru port-ul care preia datele modulului senzor umiditate sol, **sensorValue** pentru stocarea valorii transmise de senzor și **percentM** unde se va stoca valoarea senzorului sub forma unui procentaj, pentru a fi mai relevantă valoarea acestuia pentru utilizator. Pentru a transforma valoarea citită de senzorul de umiditate sol într-un procent am folosit funcția **map**. Primul parametru este valoarea pe care dorim să o transformăm în procent, cel de-al doilea este valoarea pe care senzorul o citește într-un mediu uscat, iar cel de-al treilea parametru este valoarea citită de senzor în apă. Programul care se execută odată la 10 secunde accesează cei doi senzori, printează în SerialMonitor valorile preluate de acestea, iar mai apoi formează un string cu aceste informații pe care le transmite modulului ESP8266. Totodată, fiecărei parametru transmis utilizatorului în aplicație îi corespunde un led care îl atenționează pe utilizator dacă un parametru nu are o valoare potrivită mediului în care trebuie să se dezvolte plantele sale. În cazul în care un parametru se află în afara intervalului normal de valori, led-ul corespunzător va fi roșu, iar în cazul în care valoarea parametrului este una normală, led-ul corespunzător va fi verde.

```
void loop(){
    DHT.read11(dht_apin);
    //umiditatea
    Serial.print("Current humidity = ");
    Serial.print(DHT.humidity);
    Serial.print(" % ");
    if(DHT.humidity > 50.00 && DHT.humidity < 75.00){
        digitalWrite(greenPin1, HIGH);
        digitalWrite(redPin1, LOW);
        digitalWrite(bluePin1, LOW);
    }else{
        digitalWrite(greenPin1, LOW);
        digitalWrite(redPin1, HIGH);
        digitalWrite(bluePin1, LOW);
    }
    //temperatura
    Serial.print("temperature = ");
    Serial.print(DHT.temperature);
    Serial.print(" C ");
    if(DHT.temperature > 20.00 && DHT.temperature < 4
        digitalWrite(greenPin, HIGH);
        digitalWrite(redPin, LOW);
        digitalWrite(bluePin, LOW);
    }else{
        digitalWrite(greenPin, LOW);
        digitalWrite(redPin, HIGH);
        digitalWrite(bluePin, LOW);
    }

    //umiditatea solului
    sensorValue = analogRead(moisturePin);
    percentM = map(sensorValue, 1017, 244, 0 , 100);
    Serial.print("soil moisture = ");
    Serial.print(percentM);
    Serial.println(" % ");
    Serial.print(sensorValue);
    if(percentM > 50 && percentM < 80){
        digitalWrite(greenPin2, HIGH);
        digitalWrite(redPin2, LOW);
        digitalWrite(bluePin2, LOW);
    }else{
        digitalWrite(greenPin2, LOW);
        digitalWrite(redPin2, HIGH);
        digitalWrite(bluePin2, LOW);
    }

    //transmitem datele catre modulul ESP8266 Wireless
    str = String(DHT.temperature) + String(DHT.humidity) + String(percentM);
    Serial.println(str);

    //asteptam 10 secunde pana sa reacesam senzorii si sa recitim datele
    delay(10000);
}
```

În fișierul care conține codul care se încarcă pe modulul ESP8266 Wireless sunt incluse mai multe biblioteci necesare pentru crearea server-ului Web. Au fost declarate două constante: **ssid** și **password**, acestea reprezentând numele și parola rețelei locale de WIFI, acestea fiind necesare creării server-ului web. Mai sunt declarate și 5 variabile globale folosite pentru preluarea datelor de la placa Arduino și pentru transmiterea acestora spre vizualizarea utilizatorului: **data**, **t**, **h**, **hs**, **c**. În acest fișier se afla și codul care construiește pagina web la care utilizatorii vor avea acces. Fiecare parametru al serei va fi transmis împreună cu un logo, numele acestuia, valoarea sa și unitatea de măsură corespunzătoare. Am făcut și o funcție care returnează un string pentru afișarea valorilor parametrilor(processor). În partea de set-up a codului are loc conectarea modulului Wireless la rețeaua locală. După conectare, în SerialMonitor va fi printată adresa IP la care va fi disponibilă pagina web care conține informațiile legate de seră. Programul care se execută în buclă verifică dacă sunt date disponibile, iar dacă sunt se așteaptă intrarea mai multor caractere în buffer. După ce mai multe caractere s-au adunat în buffer, string-urilor care stochează cele 3 informații le sunt atribuite string-uri nule., pentru a fi pregătite pentru actualizarea valorilor preluate de senzori. Mai apoi, string-ul **data** este împărțit în substring-uri, reprezentând cei 3 parametri care sunt transmiși utilizatorilor.

```
// COD ESP8266 NODE MCU
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <Hash.h>
#include <ESPAsyncTCP.h>
#include <ESPAsyncWebServer.h>

//nume și parolă locală Wifi network
const char* ssid = "Home_2.4";
const char* password = "@H2311d!";

//datele de pe senzori primite de la placuta arduino
String data = " "; //intreaga informatie transmisa de placa Arduino catre modul
String t = " "; //temperatura
String h = " "; //umiditate aer
String hs = " "; //umiditate sol
char c;
```



```

<p>
<i class="fas fa-thermometer-half" style="color:#059e8a;"></i>
<span class="dht-labels"> Temperature </span>
<span id="temperature">%TEMPERATURE%</span>
<sup class="units">&degC</sup>
</p>

<p>
<i class="fas fa-tint" style="color:#00add6;"></i>
<span class="dht-labels">Humidity</span>
<span id="humidity">%HUMIDITY%</span>
<sup class="units">%</sup>
</p>

<p>
<i class="fas fa-seedling" style="color:#059e8a;"></i>
<span class="dht-labels">Soil moisture</span>
<span id="moisture">%MOISTURE%</span>
<sup class="units">%</sup>
</p>

```

```

void setup() {

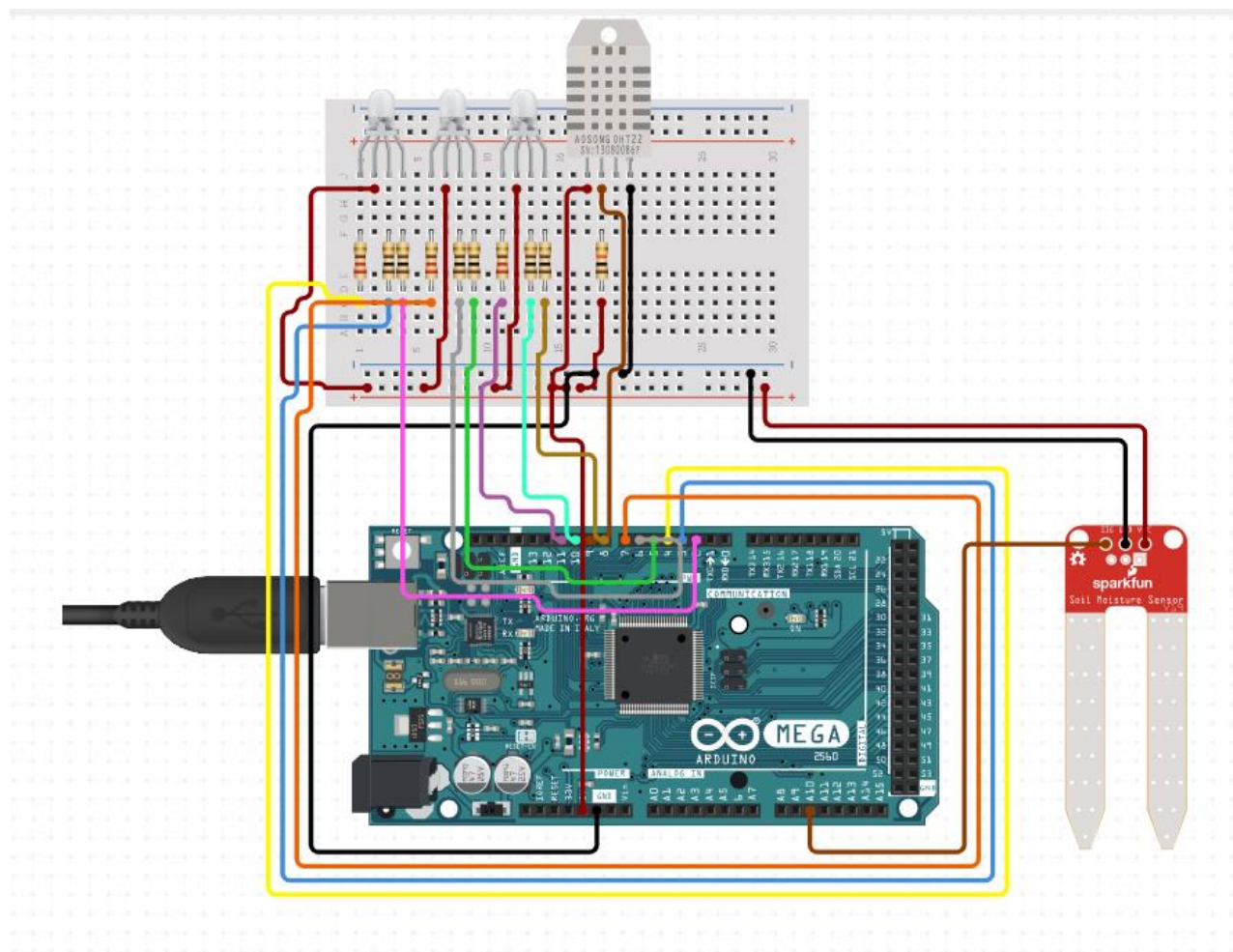
  Serial.begin(115200);
  // Connect to Wi-Fi
  WiFi.enableInsecureWEP(true);
  WiFi.begin(ssid, password);
  Serial.println("Connecting to WiFi");
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println(".");
  }
}

```

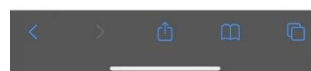
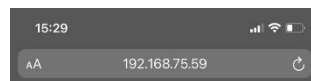
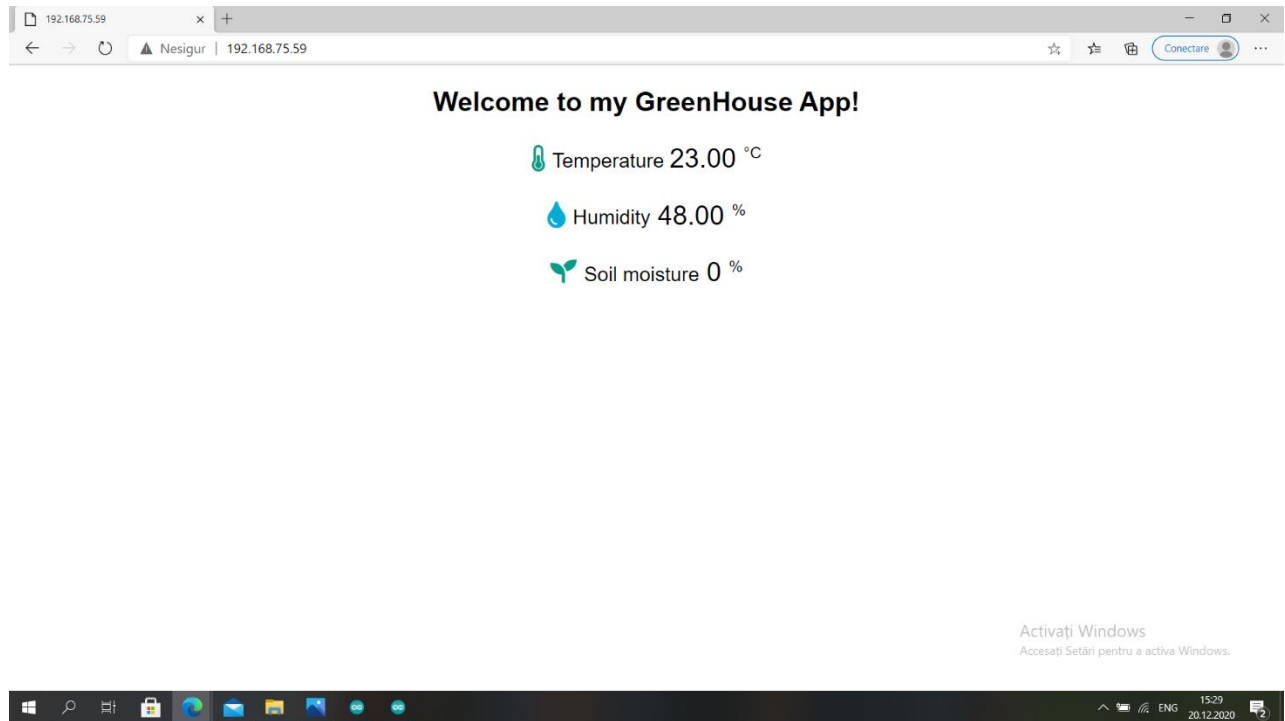
```
void loop() {  
  
    //avem cel puțin un caracter în buffer-ul serial  
    if(Serial.available()){  
        //așteptăm să intre mai multe caractere în buffer  
        delay(50);  
        //curățăm ce s-a stocat la ultimul update în string-ul în care preluăm informația din buffer  
        t = " ";  
        h = " ";  
        hs = " ";  
        data = " ";  
        //preluăm datele pe care placa Arduino le transmite serial  
        while(Serial.available()){  
            c = Serial.read();  
            data.concat(c);  
            t = data.substring(1,6);  
            h = data.substring(6,11);  
            hs = data.substring(11,13);  
        }  
        Serial.println(t);  
        Serial.println(h);  
        Serial.println(hs);  
    }  
}
```

3. Schema circuitului

Deoarece nu a fost posibilă în editor conexiunea dintre placa Arduino și modulul ESP8266 voi spune pe scurt legăturile dintre acestea. Se va face legătura de la pin-ul TX de la modulul ESP8266 la pin-ul RX1 de pe placa și de la pin-ul RX al modulului la pin-ul TX1, iar pin-ul GND la pin-ul GND al plăcii.



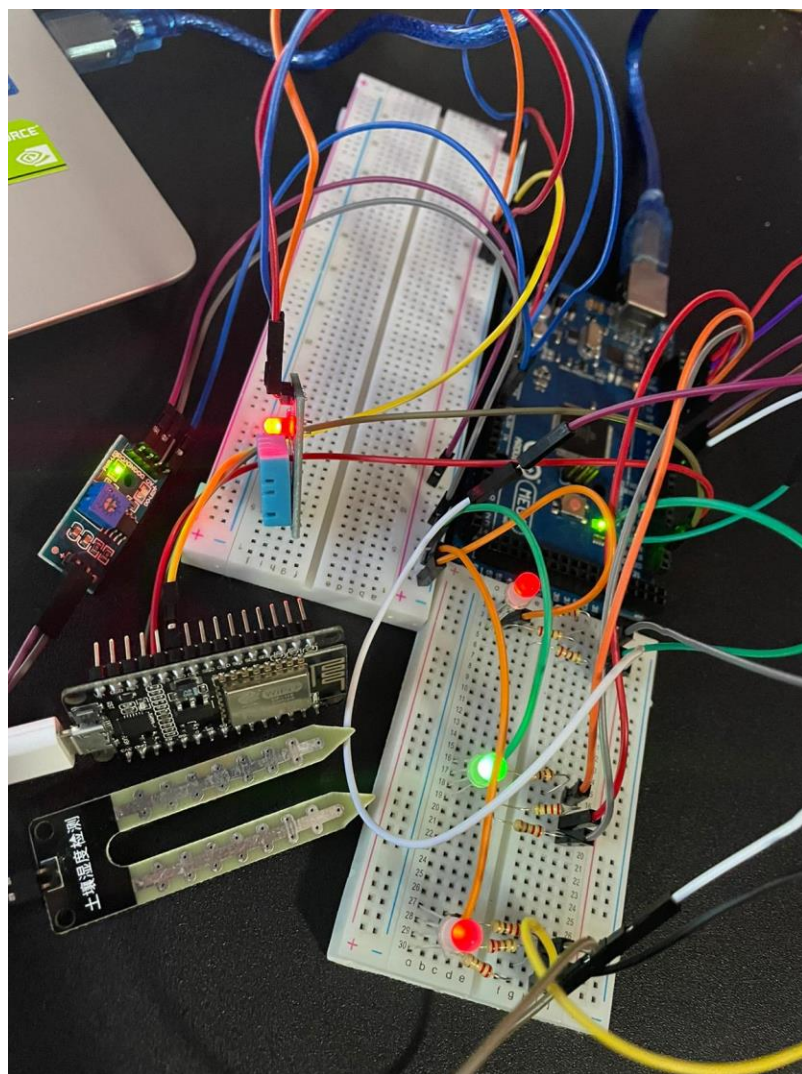
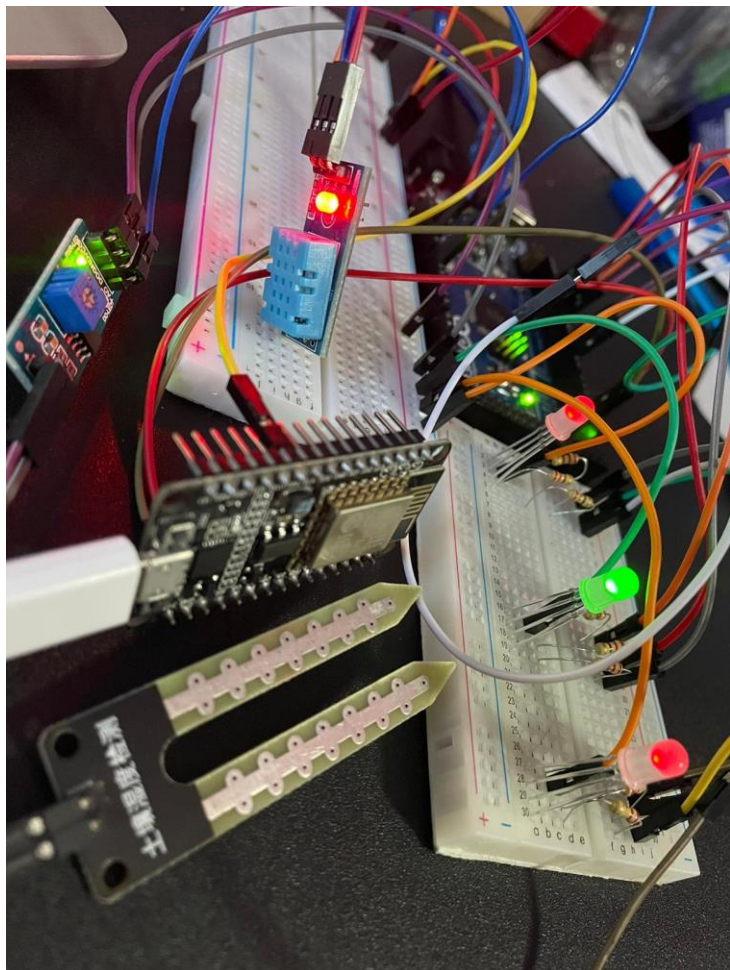
4. Rezultate

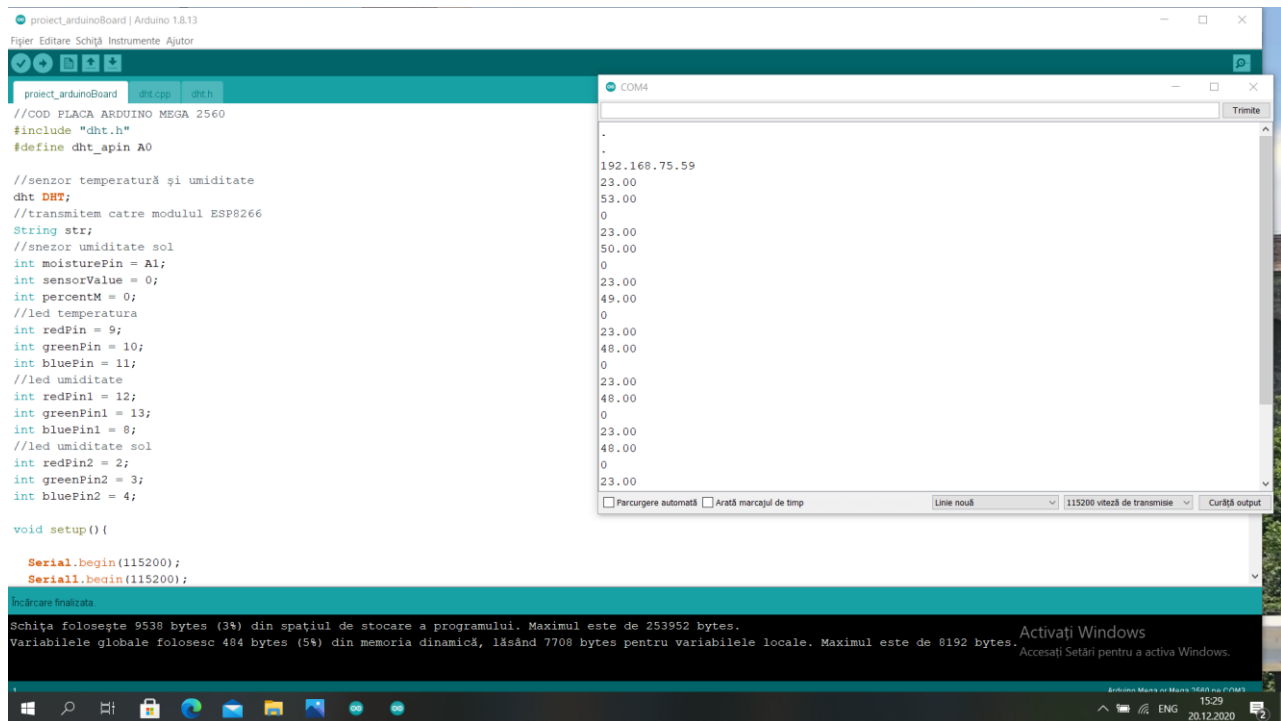


MINISTERUL EDUCAȚIEI NAȚIONALE



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA





proiect_arduinoBoard | Arduino 1.8.13
Figier Editare Schiță Instrumente Ajutor

proiect_arduinoBoard dht.cpp dht.h

```
//COD PLACA ARDUINO MEGA 2560
#include "dht.h"
#define dht_apin A0

//senzor temperatură și umiditate
dht DHT;
//transmitem catre modulul ESP8266
String str;
//senzor umiditate sol
int moisturePin = A1;
int sensorValue = 0;
int percentM = 0;
//led temperatura
int redPin = 9;
int greenPin = 10;
int bluePin = 11;
//led umiditate
int redPin1 = 12;
int greenPin1 = 13;
int bluePin1 = 8;
//led umiditate sol
int redPin2 = 2;
int greenPin2 = 3;
int bluePin2 = 4;

void setup() {
  Serial.begin(115200);
  Serial1.begin(115200);
}
```

Încărcare finalizată.

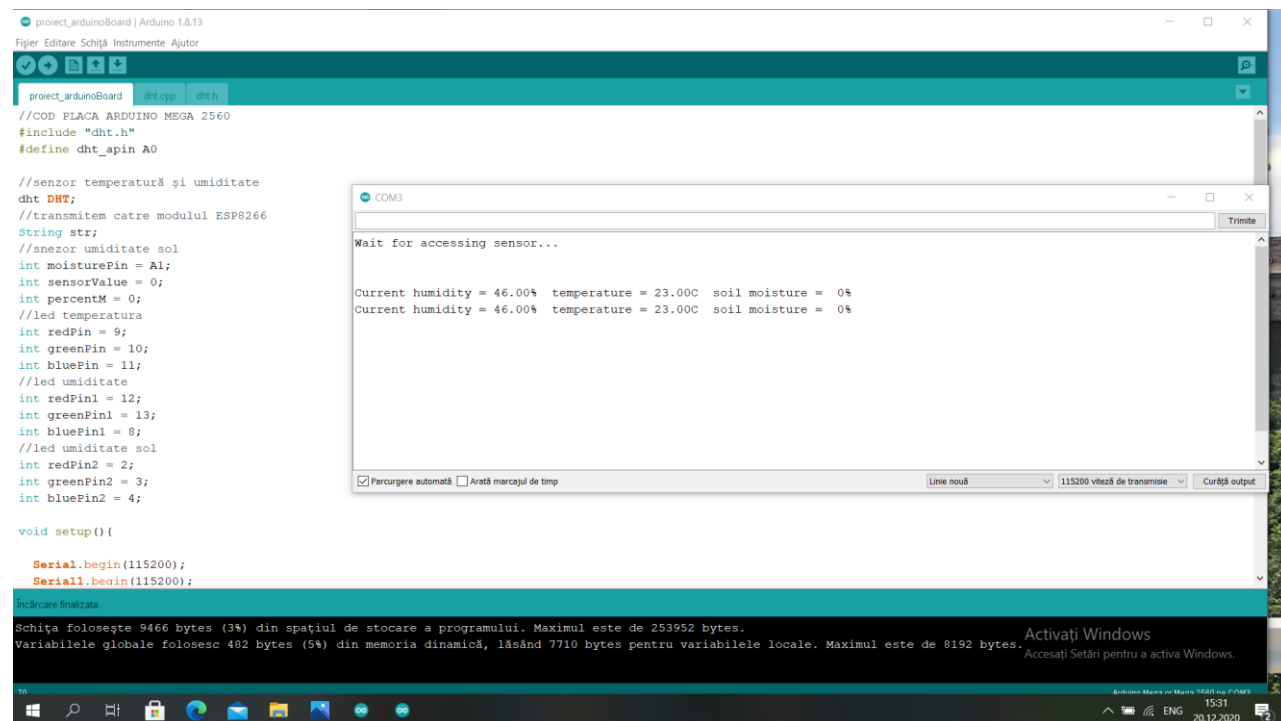
Schița folosește 9538 bytes (3%) din spațiul de stocare a programului. Maximul este de 253952 bytes.
Variabilele globale folosesc 484 bytes (5%) din memoria dinamică, lăsând 7708 bytes pentru variabilele locale. Maximul este de 8192 bytes.

COM4

```
192.168.75.59
23.00
53.00
0
23.00
50.00
0
23.00
49.00
0
23.00
48.00
0
23.00
48.00
0
23.00
48.00
0
23.00
```

Parcurgere automată ☐ Arestă marcajul de timp ☐ Linie nouă 115200 viteză de transmisie

Activați Windows
Accesați Setări pentru a activa Windows.



proiect_arduinoBoard | Arduino 1.8.13
Figier Editare Schiță Instrumente Ajutor

proiect_arduinoBoard dht.cpp dht.h

```
//COD PLACA ARDUINO MEGA 2560
#include "dht.h"
#define dht_apin A0

//senzor temperatură și umiditate
dht DHT;
//transmitem catre modulul ESP8266
String str;
//senzor umiditate sol
int moisturePin = A1;
int sensorValue = 0;
int percentM = 0;
//led temperatura
int redPin = 9;
int greenPin = 10;
int bluePin = 11;
//led umiditate
int redPin1 = 12;
int greenPin1 = 13;
int bluePin1 = 8;
//led umiditate sol
int redPin2 = 2;
int greenPin2 = 3;
int bluePin2 = 4;

void setup() {
  Serial.begin(115200);
  Serial1.begin(115200);
}
```

Încărcare finalizată.

Schița folosește 9466 bytes (3%) din spațiul de stocare a programului. Maximul este de 253952 bytes.
Variabilele globale folosesc 482 bytes (5%) din memoria dinamică, lăsând 7710 bytes pentru variabilele locale. Maximul este de 8192 bytes.

COM3

```
Wait for accessing sensor...

Current humidity = 46.00% temperature = 23.00C soil moisture = 0%
Current humidity = 46.00% temperature = 23.00C soil moisture = 0%
```

Parcurgere automată ☒ Arestă marcajul de timp ☐ Linie nouă 115200 viteză de transmisie

Activați Windows
Accesați Setări pentru a activa Windows.

Bibliografie

- [1] <https://www.robotshop.com/media/files/pdf/arduinomega2560datasheet.pdf>
- [2] <https://www.youtube.com/watch?v=IPOHARgRLE0>
- [3] <https://sites.google.com/site/arduinoelectronicasiprogramare/arduino-si-senzori/dht>

Anexă

```
//COD PLACA ARDUINO MEGA 2560
#include "dht.h"
#define dht_apin A0

dht DHT;
//transmitem catre modulul ESP8266
String str;
int moisturePin = A1;
int sensorValue =0;
int percentM = 0;
//led temperatura
int redPin = 9;
int greenPin = 10;
int bluePin = 11;
//led umiditate
int redPin1 = 12;
int greenPin1 = 13;
int bluePin1 = 8;
//led umiditate sol
int redPin2 = 2;
int greenPin2 = 3;
int bluePin2 = 4;

void setup(){

  Serial.begin(115200);
  Serial1.begin(115200);
  delay(500);//Delay to let system boot
  Serial.println("Wait for accessing sensor...\n\n");
  delay(1000);//Wait before accessing Sensor

}

void loop(){
  DHT.read11(dht_apin);
```



```
//umiditatea
Serial.print("Current humidity = ");
Serial.print(DHT.humidity);
Serial.print("% ");
if(DHT.humidity > 60.00){
    digitalWrite(greenPin1, HIGH);
    digitalWrite(redPin1, LOW);
    digitalWrite(bluePin1, LOW);
}else{
    digitalWrite(greenPin1, LOW);
    digitalWrite(redPin1, HIGH);
    digitalWrite(bluePin1, LOW);
}
//temperatura
Serial.print("temperature = ");
Serial.print(DHT.temperature);
Serial.print("C ");
if(DHT.temperature > 20.00){
    digitalWrite(greenPin, HIGH);
    digitalWrite(redPin, LOW);
    digitalWrite(bluePin, LOW);
}else{
    digitalWrite(greenPin, LOW);
    digitalWrite(redPin, HIGH);
    digitalWrite(bluePin, LOW);
}
//umiditatea solului
sensorValue = analogRead(moisturePin);
percentM = map(sensorValue, 1017, 244, 0 , 100);
Serial.print("soil moisture = ");
Serial.print(percentM);
Serial.println("% ");
Serial.print(sensorValue);
if(percentM > 60.00){
    digitalWrite(greenPin2, HIGH);
    digitalWrite(redPin2, LOW);
    digitalWrite(bluePin2, LOW);
}else{
```

```

    digitalWrite(greenPin2, LOW);
    digitalWrite(redPin2, HIGH);
    digitalWrite(bluePin2, LOW);
}
str = String(DHT.temperature) + String(DHT.humidity) + String(percentM);
Serial1.println(str);

//asteptam 10 secunde pana sa reacesam senzorii si sa recitim datele
delay(10000);

}

```

```

// COD ESP8266 NODE MCU
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <Hash.h>
#include <ESPAsyncTCP.h>
#include <ESPAsyncWebServer.h>

//nume și parolă locală Wifi network
const char* ssid = "Home_2.4";
const char* password = "@H2311d!";

//datele de pe senzori primite de la placuta arduino
String data = " "; //intreaga informatie transmisa de placa Arduino catre modul
String t = " "; //temperatura
String h = " "; //umiditate aer
String hs = " "; //umiditate sol
char c;

// Create AsyncWebServer object on port 80
AsyncWebServer server(80);

```

//WEB PAGE CODE

```

const char index_html[] PROGMEM = R"rawliteral(
<!DOCTYPE HTML><html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.2/css/all.css"
      integrity="sha384-
      fnmOCqbTIWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr"
      crossorigin="anonymous">
<style>
html {
font-family: Arial;
display: inline-block;
margin: 0px auto;
text-align: center;
}
h2 { font-size: 2.0rem; }
p { font-size: 2.0rem; }
.units { font-size: 1.2rem; }
.dht-labels{
font-size: 1.5rem;
vertical-align:middle;
padding-bottom: 15px;
}
</style>
</head>
<body>
<h2>Welcome to my GreenHouse App!</h2>

<p>
<i class="fas fa-thermometer-half" style="color:#059e8a;"></i>
<span class="dht-labels"> Temperature </span>
<span id="temperature">%TEMPERATURE%</span>
<sup class="units">&degC</sup>
</p>

<p>

```

```

<i class="fas fa-tint" style="color:#00add6;"></i>
<span class="dht-labels">Humidity</span>
<span id="humidity">% HUMIDITY%</span>
<sup class="units">%</sup>
</p>

<p>
<i class="fas fa-seedling" style="color:#059e8a;"></i>
<span class="dht-labels">Soil moisture</span>
<span id="moisture">% MOISTURE%</span>
<sup class="units">%</sup>
</p>

</body>
</html>)>rawliteral";

```

// Replaces placeholder with DHT values

```

String processor(const String& var){
  if(var == "TEMPERATURE"){
    return String(t);
  }
  else if(var == "HUMIDITY"){
    return String(h);
  }
  else if(var == "MOISTURE"){
    return String(hs);
  }
  return String();
}

```

```

void setup(){

```

```

  Serial.begin(115200);
  // Connect to Wi-Fi
  WiFi.enableInsecureWEP(true);
  WiFi.begin(ssid, password);
  Serial.println("Connecting to WiFi");

```

MINISTERUL EDUCAȚIEI NAȚIONALE



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

```

while (WiFi.status() != WL_CONNECTED) {
  delay(1000);
  Serial.println(".");
}

// Print ESP8266 Local IP Address
Serial.println(WiFi.localIP());

// Route for root / web page
server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
  request->send_P(200, "text/html", index_html, processor);
});
server.on("/temperature", HTTP_GET, [](AsyncWebServerRequest *request){
  request->send_P(200, "text/plain", String(t).c_str());
});
server.on("/humidity", HTTP_GET, [](AsyncWebServerRequest *request){
  request->send_P(200, "text/plain", String(h).c_str());
});
server.on("/moisture", HTTP_GET, [](AsyncWebServerRequest *request){
  request->send_P(200, "text/plain", String(hs).c_str());
});

// Start server
server.begin();
}

//cod program
void loop() {

  //avem cel putin un caracter in buffer-ul serial
  if(Serial.available()){
    //asteptam sa intre mai multe caractere in buffer
    delay(50);
    //curatam ce s-a stocat la ultimul update in string-ul in care preluam informatia din buffer
    t = " ";
    h = " ";
    hs = " ";
    data = " ";
  }
}

```

```
//preluam datele pa care placa Arduino le transmite serial
while(Serial.available()){
    c = Serial.read();
    data.concat(c);
    t = data.substring(1,6);
    h = data.substring(6,11);
    hs = data.substring(11,13);
}
Serial.println(t);
Serial.println(h);
Serial.println(hs);
}

}
```