

Capitolul I. Utilitate practică

I.1 La ce ne este necesară Grădina Inteligentă?

Ideea a venit în urma unei vizite la Grădina Botanică Bucov, unde personalul grădinii mi-a spus că o plantă pentru a trăi și a se dezvolta are nevoie de lumină, căldură și apă, una din resursele importante ale vieții. Acolo am constatat că plantele sufereau din cauza lipsei de apă, deoarece personalul grădinii era insuficient. Totodată, udarea manuală a plantelor, pe lângă risipa apei, ducea și la o udare ineficientă care putea produce rău plantelor, pentru că nu exista un control al umidității.



I.2 Research

În urma unui research am aflat că anul 2022 a fost unul dintre cei mai secetoși ani, recolta de porumb scăzând cu 40%, iar recolta de semințe de floarea-soarelui scăzând cu 30%. Din datele transmise de Ministerul Agriculturii și Dezvoltării rurale, am constatat că 1,3 milioane de hectare au fost afectate de secetă din 37 de județe și municipiul București.

Mai exact, potrivit Administrației Naționale de Meteorologie, anul 2022 a fost al treilea cel mai calduros an din istoria măsurărilor meteorologice din România, temperatura medie anuală fiind de 11.77°C, iar abaterea termică de 1.55°C față de media perioadei 1981-2010. Datele au fost preluate de la 129 de stații meteorologice cu șir complet în perioada 1961 - 2022 (Fig. 1).

Cei mai calduroși 5 ani din perioada 1900-2022 sunt: 2019, 2020, 2022, 2015 și 2007, iar intervalul 2012-2022 reprezintă perioada de 11 ani consecutivi cea mai calduroasă, fapt care confirmă în mod evident tendința de creștere a temperaturii aerului și în țara noastră.

Abateri pozitive au fost înregistrate în nouă din cele 12 luni ale anului, temperatura medie lunară pe țară fiind mai mare decât mediana intervalului de referință standard (1991 - 2020) cu valori cuprinse între 0,7 °C (mai) și 2,6 °C (decembrie). În restul lunilor, abaterea a fost negativă și a avut valori între 0,1 °C, în septembrie și 1,8 °C, în martie (Fig. 2).

Nr. crt.	An	Temperatura medie pe țară (°C)	Abaterea față de mediana intervalului de referință 1981-2010 (°C)	Abaterea față de mediana intervalului de referință 1991 – 2020 (°C)
1	2019	12,14	1,92	1,40
2	2020	11,88	1,66	1,14
3	2022	11,77	1,55	1,04
4	2015	11,72	1,50	0,98
5	2007	11,67	1,45	0,93
6	2018	11,57	1,35	0,84
7	2014	11,36	1,14	0,62
8	1994	11,35	1,13	0,62
9	2009	11,28	1,06	0,54
10	2013	11,23	1,01	0,50

Fig. 1. Topul celor mai calzi zece ani din perioada 1900 – 2022

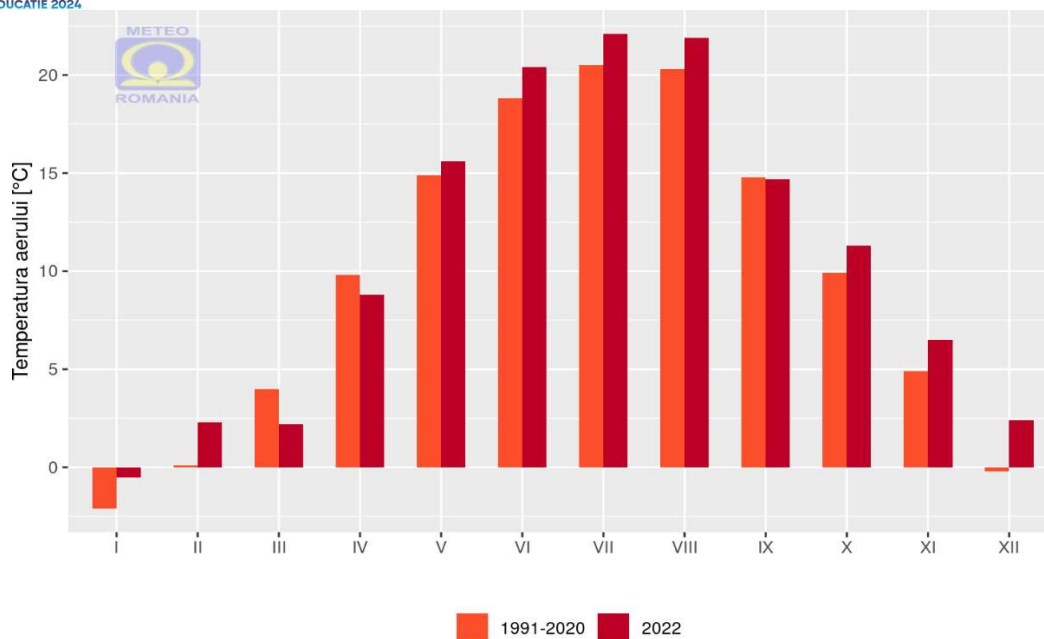


Fig. 2. Evoluția temperaturii medii lunară, medie pe țară, din România, în anul 2022, comparativ cu mediana intervalului climatologic standard (1991 - 2020)

I.3 Studiu de piață

Analizand ofertele existente pe piață, am ajuns la concluzia că dacă aș comanda unul dintre sistemele existente pe piață care s-ar potrivi cerințelor specialiștilor de la Sera, costul acestuia ar fi de aproximativ 2500 lei, neavând senzori care să transmită în timp real parametrii necesari dezvoltării plantelor.

Așa că am decis să creez Grădina Inteligentă.

În urma discuțiilor avute cu specialiștii de la Grădina Botanică, plantele din seră au fost împărțite în 3 zone: zona 1 - plante iubitoare de apă (cantitate mare de apă), zona 2 - plante obișnuite (cantitate medie de apă), zona 3 - zona aridă (cantitate mică de apă).

Prin implementarea unor senzori de umiditate ai solului capacitivi care să transmită sistemului inteligent informații de la fiecare zonă de udare, plantele primesc apa necesară dezvoltării în funcție de caracteristicile lor, eliminând riscul udării în exces sau întârziată, ceea ce duce la moartea lor.

De asemenea, am implementat și un sistem de duze pentru pulverizarea apei, care imită ploaia.

Capitolul II. Mecanică

II.1 Complexitate

Sistemul pe care l-am gândit l-am asimilat cu un robot, el fiind dezvoltat pe o **placă arduino UNO** (plăcuța fiind creierul sistemului) la care am conectat:

- în loc de motoare am folosit un sistem de **electrovalve**, 3 pentru zonele de udare și 1 pentru duzele de pulverizare. Electrovalvele sunt acționate de către o plăcuța cu **relee intermediare**, deoarece outputurile de la plăcuța UNO nu suportau alimentarea de putere data de bobinele electrovalvei.

- **senzori** pe care i-am ales în funcție de necesitățile plantelor, astfel:

1. senzori de umiditate sol
2. senzori de temperatura și umiditate aer
3. senzori de temperatura sol

- **display local LCD** (doar pe macheta) pe care se afișează:

- temperatura și umiditatea aerului din sera
- temperatura solului
- umiditatea pe fiecare zonă
- starea sistemului

- **plăcuță Wi-Fi ESP8266**, care transmite datele serverului Blynk

II. 2 Eficiența în construcție

Sistemul poate funcționa cu ajutorul unei surse de alimentare 230VAC/12VDC sau cu ajutorul unui sistem Arduino de solar tracking care încarcă un acumulator de 9V.



Fig. 3 – Sistem solar tracking

II.3 Sistemul de solar tracking

Sistemul de solar tracking este dezvoltat pe o placuta Arduino UNO la care am conectat:

- Două **servouri** pentru orientarea panoului solar astfel încât să aibă cât mai multă lumină
- Modul de alimentare cu litiu alimentat de energie solară și prin cablu USB
- Modul de încărcare a telefonului
- Display LCD I2C
- Senzori :
 - senzor de intensitate a luminii
 - fotorezistori
 - senzor de temperatură și umiditate aer
 - buzzer
 - buton

Capitolul III. Electronică

III.1 Componente Grădina Inteligentă

Placa Arduino UNO

Placa de dezvoltare UNO R3 cu microcontroller ATmega328p, necesară controlării procesului.



Fig. 4 - Placa Arduino UNO

Specificații tehnice

- Microcontroller: ATmega328p
- USB Chip: CH340G
- Tensiune alimentare USB: 5V
- Tensiune alimentare conector DC: 8-12V
- Pini digitali I/O: 14 (6 suporta iesire PWM)
- Pini analogici: 6
- Curent maxim pe pin I/O: 40 mA
- Memorie Flash: 32 KB (ATmega328)
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Frecventa: 16 MHz

Placa Wi-Fi Esp8266

Modulul ESP8266 transmite datele de la plăcuța Arduino serverului Blynk.

Specificații tehnice

- Voltaj: 3.3V.
- Wi-Fi Direct (P2P), soft-AP.
- Consum curent: 10uA ~ 170mA.
- Memorie flash atașabilă: maxim 16MB (normal 512K).
- Protocolul TCP / IP integrat.
- Procesor: Tensilica L106 pe 32 de biți.
- Viteza procesorului: 80 ~ 160MHz.
- RAM: 32K + 80K.
- GPIO: 17 (multiplexate cu alte funcții).
- Analogic la digital: 1 intrare cu rezoluție de 1024 pas.



Fig. 5 – Placa Wi-Fi Esp 8266

- Putere de ieșire + 19,5 dBm în modul 802.11b
- Suport 802.11: b / g / n.
- Conexiuni maxime concurente TCP: 5

Placa releee intermediare

O placă cu releee intermediare este folosită pentru a putea comanda valvele.

Specificații tehnice:

- Tensiune de alimentare: 24VDC
- Canale: 4
- Tensiune releu: 24VDC
- Interfata: USB integrat
- IC control: ATmega 8A
- IC control releu: ULN2803
- Tip USB: Type-B
- Tip relee: Songle 24V
- Rating releu: 250V, 10A AC și 30V, 10A DC
- Mod de conexiune la terminalul de control: NC (Normal închis), COM (Comun), NO (Normal deschis)
- Indicatori LED: Verde pentru releu activat, Rosu pentru releu dezactivat
- Compatibilitate: Suporta Windows 7, XP, 32-bit și 64-bit fără necesitatea de drivere suplimentare
- Tip USB: HID (Human Interface Design)
- Dimensiuni mm: 72 x 66mm

Optimus Cyber



Fig. 6 – Placa releee

Valve electromagnetice

Valva electromagnetica ce permite deschiderea circuitului în momentul aplicării unei tensiuni electrice. Modelul este Normal Închis (la aplicarea tensiunii se deschide).

Specificații tehnice:

- Tensiune alimentare: 12 VDC
- Curent recomandat: 0.6-1.5A
- Marime racord: 1/2"
- Material: Metal + plastic



Fig. 7 – Valva electromagnetica

- Presiune: 0.02- 0.8Mpa
- Temperatura maxima lichid: 100°C
- Mod operare: Normally Closed
- Tip valva: diafragma
- Folosire: apa si lichide cu viscozitate redusa
- Dimensiuni mm: 78 x 60mm
- Diametru interior: 13.5mm
- Diametru exterior: 20mm

Display LCD 2x16 I2C

Modulul LCD 2004 afișează datele de la senzori și starea fiecărei zone.

Specificații tehnice:

- Tensiune de alimentare: 5V;
- Curent: 2 mA;
- Tensiune alimentare backlight: 4.2V;
- Curent lumina de fundal: 250mA (MAX).



Fig. 8 – Ecran LCD

III.2 Tipuri de senzori folosiți

Senzorii i-am ales in functie de necesitățile plantelor, astfel:

1. Senzori pentru control umiditate sol

Inițial am folosit senzori clasici la care am observat că după un timp de utilizare apare fenomenul de coroziune și nu mai transmit datele reale.

I-am inlocuit cu senzori capacitivi din plastic care au durata mare de viață și nu mai sunt afectați de coroziune, iar informațiile transmise sunt corecte.



Fig. 9 - Senzor capacitiv

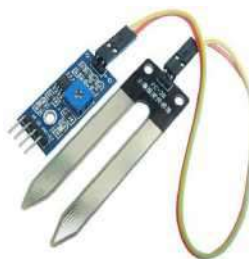


Fig.10 - Senzor clasic

Calibrare senzori umiditate sol

Am luat valorile afișate de senzori într-un mediu cu 0% umiditate și într-un mediu cu 100% umiditate reprezentat de un pahar cu apă.

După aceea, am suprapus intervalele determinate de aceste valori cu intervalul 0-100 pentru a afișa umiditatea fiecărei zone în procente.

Zona 1 – zona umiditate mare

Dacă valoarea scade sub 70%, electrovalva este pornită și sistemul compus din furtune și duze cu debit de apă presetat la 10l/ora devine activ și apa ajunge la rădăcina plantelor.

Când valoarea indicată de senzor este peste 70%, electrovalva este oprită.

Zona 2 – zona umiditate medie

Dacă valoarea scade sub 50%, electrovalva este pornită și sistemul compus din furtune și duze cu debit de apă presetat la 5l/oră devine activ și apa ajunge la rădăcina plantelor.

Când valoarea indicată de senzor este peste 50%, electrovalva este oprită.

Zona 3 – zona aridă

Dacă valoarea scade sub 30%, electrovalva este pornită și sistemul compus din furtune și duze cu debit de apă presetat la 2l/oră devine activ și apa ajunge la rădăcina plantelor.

Când valoarea indicată de senzor este peste 30%, electrovalva este oprită.

2. Senzori de temperatură și umiditate aer

DHT22 este un senzor digital de temperatură și umiditate. Folosește un senzor capacitiv pentru umiditate și un termistor pentru a măsura temperatura aerului.

Specificații tehnice:

- Dimensiune: 28mm X 12mm X 10mm
- Intervalul de măsurare a umidității: 0 - 100 RH
- Acuratețea de măsurare a umidității: $\pm 2\%$ RH
- Acuratețea de măsurare a temperaturii: $\pm 0.5^\circ$
- Tensiune de lucru: DC 3-5V
- Consum: 2.5mA max (când se procesează datele)
- Temperatura operare: $-40 +80$ grade Celsius



Fig. 11 – Senzor de temperatura si umiditate aer

Cu ajutorul acestui senzor, afișăm temperatura și umiditatea aerului din interiorul serei, parametrii ce sunt necesari specialiștilor în a analiza dezvoltarea plantelor și a putea face studii asupra acestora.

3. Senzor de temperatura sol

Senzorul de temperatură bazat pe cipul DS18B20, care constă dintr-o sondă rezistentă la apă și un fir lung de 1m, folosit pentru a măsura temperatura solului.

În funcție de valorile afișate de acest senzor, specialiștii pot să realizeze studii mai ușoare asupra plantelor.

Specificații tehnice:

- Alimentare: 3.0V până la 5.5V
- Interval de temperatură: -55°C până la $+125^\circ\text{C}$
- Precizie ridicată: $\pm 0,5^\circ\text{C}$ (-10°C până la $+85^\circ\text{C}$)
- Tip senzor: DS18B20
- Precizie de conversie 9 ~ 12 biți A / D
- Întârziere scurtă în conversia temperaturii, maxim 750 ms
- Suportă rețelele multi-point
- Rezistent la apă



Fig. 12 – Senzor de temperatura sol

Grădina inteligentă este un sistem complet autonom, nu necesită intervenție umană, ba mai mult el transmite în timp real informații despre parametrii necesari dezvoltării plantelor.

III.3 Componente kit solar tracking

Servo

Un servo este un dispozitiv de acționare electronic, care utilizează măsurători ale poziției, vitezei și cuplului pentru a controla un proces automat.

Specificații tehnice:

- Tensiune de operare: 4.8 ~ 6.0V
- Curenct: 0.19A@5V, 0.24A@6V
- Viteza de funcționare: 0.12sec / 60 grade (4.8V) ~ 0.1 sec / 60 grade (6.0V)
- Cuplu: 1,6 kg / cm (4,8 V)
- Interval de temperatură: -30 ~ + 60 °C
- Lungime cablu: 25cm
- Servo tip: servo analogic
- Dimensiune: 23X12.2X29MM



Fig. 13 - Servo

Buzzer

Acest modul este constituit dintr-un buzzer piezoelectric activ, ce generează un sunet atunci când se depășește o valoare predefinită. Produce un sunet de aproximativ 2,5KHz.

Specificații tehnice:

- Tensiune de alimentare: 1.5 ~ 15V DC
- Gama de generare a tonurilor: 1.5 ~ 2.5kHz
- Dimensiuni: 18.5mm x 15mm

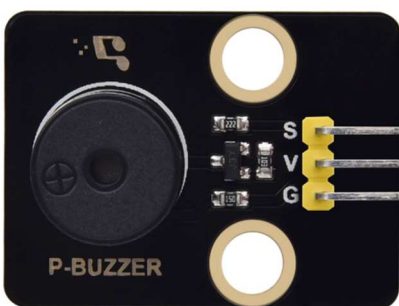


Fig. 14 - Buzzer

Fotorezistor

Fotorezistorul este un element pasiv de circuit căruia îi variază rezistența în funcție de intensitatea luminii ce ajunge pe el.

Specificații tehnice:

- Consum 100 mW
- Temperatură - 30°C - +70°C
- Valoarea spectrală 540 nm
- Rezistența la lumină 10 - 20 KΩ

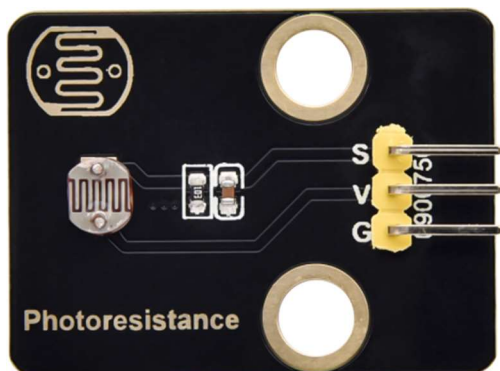


Fig. 15 - Fotorezistor

Capitolul IV. Software

Placa Arduino preia datele de la senzori, le analizează, iar în funcție de acestea, decide dacă le dă plantelor apă sau nu. Totodată, datele sunt trimise plăcuței Wi-Fi, care se conectează la rețeaua locală și le transmite mai departe serverului Blynk unde acestea sunt afișate.

Deci, aplicația afișează pe telefon și pe calculator, în timp real, oriunde ai fi, atât datele din senzori, cât și dacă valva este deschisă sau închisă și o poți controla de la distanță.

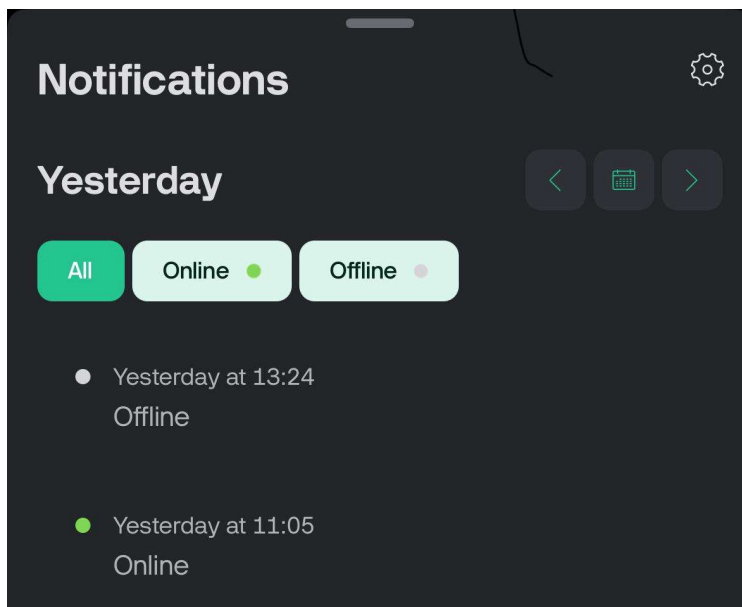


Fig. 16 - Istoric activitate sistem

```
TSol=  
29.87  
H%Zona1=  
0  
Valva1Pornit  
H%Zona2=  
0  
Valva2Pornit  
HAer=  
37.00  
TAer=  
28.90  
TSol=  
29.81  
H%Zona1=  
0  
Valva1Pornit  
H%Zona2=  
0  
Valva2Pornit
```

Type here

Fig. 17 – Date afisate pe telefon

Terminal

```
TSol=  
29.87  
H%Zona1=  
0  
Valva1Pornit  
H%Zona2=  
0  
Valva2Pornit  
HAer=
```

Type here

Fig. 18 – Date afisate pe calculator

IV.1 Cod Placuta Arduino

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
//declarare ecran LCD

#include <DHT.h>
#define DHTPIN 12
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
//declarare senzor de umiditate si temperatura aer

#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 13
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
//declarare senzor de temperatura sol

const int AirValue0 = 520;
const int WaterValue0 = 220;
int soilvalue0 = 0;
int soilpercent0 = 0;
//parametrii zona 1

const int AirValue1 = 520;
const int WaterValue1 = 220;
int soilvalue1 = 0;
int soilpercent1 = 0;
//parametrii zona 2

const int AirValue2 = 520;
const int WaterValue2 = 220;
int soilvalue2 = 0;
int soilpercent2 = 0;
//parametrii zona 3

const int valva1 = 9;
const int valva2 = 11;
const int valva3 = 10;
//declarare valve
```



```
void setup() {  
  Serial.begin(9600);  
  pinMode(valva1, OUTPUT);  
  pinMode(valva2, OUTPUT);  
  pinMode(valva3, OUTPUT);  
  dht.begin();  
  //initializare valve si senzor de umiditate si temperatura aer  
  
  lcd.init();  
  lcd.backlight();  
  //initializare ecran LCD  
  
  lcd.print("Smart Garden");  
  lcd.setCursor(0, 1);  
  lcd.print("Alex Dragu");  
  delay(3000);  
  lcd.clear();  
  //afisarea unui mesaj la pornire  
}  
  
void loop() {  
  lcd.clear();  
  soilvalue0 = analogRead(A0); //citire senzor de umiditate sol zona 1  
  soilpercent0 = map(soilvalue0, AirValue0, WaterValue0, 0, 100); //transformarea datelor in  
  intervalul 0-100  
  if (soilpercent0 >= 100)  
  {  
    soilpercent0 = 100;  
  }  
  else if (soilpercent0 <= 0)  
  {  
    soilpercent0 = 0;  
  }  
  Serial.println("H%Zona1="); Serial.println(soilpercent0);  
  lcd.setCursor(0, 1);  
  lcd.print("H%Zona1=");  
  lcd.setCursor(8, 1);  
  lcd.print(soilpercent0); //afisare umiditate zona 1  
  delay(3000);  
  lcd.clear();  
  
  if (soilpercent0 <= 60)  
  {
```

```

digitalWrite(valva1, HIGH);
Serial.println("Valva1 Pornit");
lcd.setCursor(0, 1);
lcd.print("Valva1 Pornit");
//daca umiditatea zonei scade sub 60%, valva se deschide
}
else
{
    digitalWrite(valva1, LOW);
    //altfel valva este inchisa
}
delay(3000);
lcd.clear();

soilvalue1 = analogRead(A1); //citire senzor de umiditate sol zona 2
soilpercent1 = map(soilvalue1, AirValue1, WaterValue1, 0, 100); //transformarea datelor in
intervalul 0-100
if (soilpercent1 >= 100)
{
    soilpercent1 = 100;
}
else if (soilpercent1 <= 0)
{
    soilpercent1 = 0;
}
Serial.println("H%Zona2="); Serial.println(soilpercent1);
lcd.setCursor(0, 1);
lcd.print("H%Zona2=");
lcd.setCursor(8, 1);
lcd.print(soilpercent1); //afisare umiditate zona 2
delay(3000);
lcd.clear();

if (soilpercent1 <= 50)
{
    digitalWrite(valva2,HIGH);
    Serial.println("Valva2 Pornit");
    lcd.setCursor(0, 1);
    lcd.print("Valva2 Pornit");
    //daca umiditatea zonei scade sub 50%, valva se deschide
}
else
{

```

```
digitalWrite(valva2,LOW);  
//altfel valva este inchisa  
}  
delay(3000);  
lcd.clear();
```

```
float h = dht.readHumidity();  
float t = dht.readTemperature();  
//citire senzor de umiditate si temperatura aer
```

```
Serial.println("HAer="); Serial.println(h);  
lcd.setCursor(0, 1);  
lcd.print("HAer=");  
lcd.setCursor(5, 1);  
lcd.print(h);  
delay(3000);  
lcd.clear();  
//afisare umiditate aer
```

```
if (h > 75) {  
    digitalWrite(valva3, HIGH);  
    lcd.setCursor(0, 1);  
    lcd.print("Valva3Pornit");  
    //daca umiditatea aerului e mai mare decat 75%, un ventilator se porneste  
}  
else digitalWrite(valva3, LOW); //altfel se inchide  
delay(3000);  
lcd.clear();
```

```
Serial.println("TAer="); Serial.println(t);  
lcd.setCursor(0, 1);  
lcd.print("TAer=");  
lcd.setCursor(5, 1);  
lcd.print(t);  
delay(3000);  
lcd.clear();  
//afisare temperatura aer
```

```
sensors.requestTemperatures();  
Serial.println("TSol="); Serial.println(sensors.getTempCByIndex(0));  
lcd.setCursor(0, 1);  
lcd.print("TSol=");  
lcd.setCursor(5, 1);
```

```
lcd.print(sensors.getTempCByIndex(0));  
delay(3000);  
lcd.clear();  
//afisare temperatura sol  
}
```

IV.2 Cod Placuta Wi-Fi

```
#include <LiquidCrystal_I2C.h>  
LiquidCrystal_I2C lcd(0x27, 16, 2);  
//declarare ecran LCD  
  
#include <DHT.h>  
#define DHTPIN 12  
#define DHTTYPE DHT11  
DHT dht(DHTPIN, DHTTYPE);  
//declarare senzor de umiditate si temperatura aer  
  
#include <OneWire.h>  
#include <DallasTemperature.h>  
#define ONE_WIRE_BUS 13  
OneWire oneWire(ONE_WIRE_BUS);  
DallasTemperature sensors(&oneWire);  
//declarare senzor de temperatura sol  
  
const int AirValue0 = 520;  
const int WaterValue0 = 220;  
int soilvalue0 = 0;  
int soilpercent0 = 0;  
//parametrii zona 1  
  
const int AirValue1 = 520;  
const int WaterValue1 = 220;  
int soilvalue1 = 0;  
int soilpercent1 = 0;  
//parametrii zona 2  
  
const int AirValue2 = 520;  
const int WaterValue2 = 220;  
int soilvalue2 = 0;
```

```
int soilpercent2 = 0;
//parametrii zona 3

const int valva1 = 9;
const int valva2 = 11;
const int valva3 = 10;
//declarare valve

void setup() {
  Serial.begin(9600);
  pinMode(valva1, OUTPUT);
  pinMode(valva2, OUTPUT);
  pinMode(valva3, OUTPUT);
  dht.begin();
  //initializare valve si senzor de umiditate si temperatura aer

  lcd.init();
  lcd.backlight();
  //initializare ecran LCD

  lcd.print("Smart Garden");
  lcd.setCursor(0, 1);
  lcd.print("Alex Dragu");
  delay(3000);
  lcd.clear();
  //afisarea unui mesaj la pornire
}

void loop() {
  lcd.clear();
  soilvalue0 = analogRead(A0); //citire senzor de umiditate sol zona 1
  soilpercent0 = map(soilvalue0, AirValue0, WaterValue0, 0, 100); //transformarea datelor in
  intervalul 0-100
  if (soilpercent0 >= 100)
  {
    soilpercent0 = 100;
  }
  else if (soilpercent0 <= 0)
  {
    soilpercent0 = 0;
  }
  Serial.println("H%Zona1="); Serial.println(soilpercent0);
  lcd.setCursor(0, 1);
```

```
lcd.print("H%Zona1=");
lcd.setCursor(8, 1);
lcd.print(soilpercent0); //afisare umiditate zona 1
delay(3000);
lcd.clear();
```

```
if (soilpercent0 <= 60)
{
    digitalWrite(valva1, HIGH);
    Serial.println("Valva1 Pornit");
    lcd.setCursor(0, 1);
    lcd.print("Valva1 Pornit");
    //daca umiditatea zonei scade sub 60%, valva se deschide
}
else
{
    digitalWrite(valva1, LOW);
    //altfel valva este inchisa
}
delay(3000);
lcd.clear();
```

```
soilvalue1 = analogRead(A1); //citire senzor de umiditate sol zona 2
soilpercent1 = map(soilvalue1, AirValue1, WaterValue1, 0, 100); //transformarea datelor in
intervalul 0-100
```

```
if (soilpercent1 >= 100)
{
    soilpercent1 = 100;
}
else if (soilpercent1 <= 0)
{
    soilpercent1 = 0;
}
Serial.println("H%Zona2="); Serial.println(soilpercent1);
lcd.setCursor(0, 1);
lcd.print("H%Zona2=");
lcd.setCursor(8, 1);
lcd.print(soilpercent1); //afisare umiditate zona 2
delay(3000);
lcd.clear();
```

```
if (soilpercent1 <= 50)
```

```
{
    digitalWrite(valva2,HIGH);
    Serial.println("Valva2Pornit");
    lcd.setCursor(0, 1);
    lcd.print("Valva2Pornit");
    //daca umiditatea zonei scade sub 50%, valva se deschide
}
else
{
    digitalWrite(valva2,LOW);
    //altfel valva este inchisa
}
delay(3000);
lcd.clear();

float h = dht.readHumidity();
float t = dht.readTemperature();
//citire senzor de umiditate si temperatura aer

Serial.println("HAer="); Serial.println(h);
lcd.setCursor(0, 1);
lcd.print("HAer=");
lcd.setCursor(5, 1);
lcd.print(h);
delay(3000);
lcd.clear();
//afisare umiditate aer

if (h > 75) {
    digitalWrite(valva3, HIGH);
    lcd.setCursor(0, 1);
    lcd.print("Valva3Pornit");
    //daca umiditatea aerului e mai mare decat 75%, un ventilator se porneste
}
else digitalWrite(valva3, LOW); //altfel se inchide
delay(3000);
lcd.clear();

Serial.println("TAer="); Serial.println(t);
lcd.setCursor(0, 1);
lcd.print("TAer=");
lcd.setCursor(5, 1);
lcd.print(t);
```



```

delay(3000);
lcd.clear();
//afisare temperatura aer

sensors.requestTemperatures();
Serial.println("TSol="); Serial.println(sensors.getTempCByIndex(0));
lcd.setCursor(0, 1);
lcd.print("TSol=");
lcd.setCursor(5, 1);
lcd.print(sensors.getTempCByIndex(0));
delay(3000);
lcd.clear();
//afisare temperatura sol
}

```

IV.3 Cod panou solar

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
//declarare ecran LCD

#include <BH1750.h>
BH1750 lightMeter;
//declarare senzor de intensitate lumina

#include <dht11.h>
dht11 DHT;
#define DHT11_PIN 7
//declarare senzor de umiditate si temperatura aer

#include <Servo.h>
Servo lr_servo; //declarare servo axa X
Servo ud_servo; //declarare servo axa Y

int lr_angle = 90; //declarare unghi initial servo axa X
int ud_angle = 10; //declarare unghi initial servo axa Y

int l_state = A0;
int r_state = A1;
int u_state = A2;

```

```
int d_state = A3;  
//declarare pini voltaj servo  
  
int buzzer = 6; //declarare pin buzzer  
int lr_servopin = 9; //declarare pin control servo axa X  
  
int ud_servopin = 10; //declarare pin control servo axa Y  
int buttonPin = 2; //declarare pin button  
  
int error = 15; //declarare eroare  
  
int accuracy = 1; //declarare acuratete  
  
int light; //declarare variabila intensitate lumina  
int temperature; //declarare variabila temperatura  
int humidity; //declarare variabila umiditate  
  
void setup() {  
  Serial.begin(9600);  
  Wire.begin();  
  lightMeter.begin();  
  //initializare senzor de intensitate lumina  
  
  lr_servo.attach(lr_servopin); //setare pin de control servo axa X  
  ud_servo.attach(ud_servopin); //setare pin de control servo axa Y  
  pinMode(l_state, INPUT);  
  pinMode(r_state, INPUT);  
  pinMode(u_state, INPUT);  
  pinMode(d_state, INPUT);  
  
  lcd.init();  
  lcd.backlight();  
  //initializare ecran LCD  
  
  lr_servo.write(lr_angle);  
  delay(1000);  
  //setare unghi initial servo axa X  
  
  ud_servo.write(ud_angle);  
  delay(1000);  
  //setare unghi initial servo axa Y  
}
```

```
void loop() {
    servo();
    //setare unghiuri servouri

    Light();
    //citire intensitate lumina

    Dht();
    //citire umiditate si temperatura aer

    LCD();
    //afisare date pe ecran LCD
}

void servo(){
    int L = analogRead(l_state);
    int R = analogRead(r_state);
    int U = analogRead(u_state);
    int D = analogRead(d_state);
    //citire voltaj

    if (abs(L - R) > error && L >= R)
    {
        //daca eroarea nu este in intervalul dorit si servoul este orientat spre dreapta

        lr_angle -= accuracy;
        //unghiul se ajusteaza

        if (lr_angle < 0) lr_angle = 0;
        //limitare unghi servo

        lr_servo.write(lr_angle);
        delay(10);
        //seteaza unghiul servoului
    }
    else if (abs(L - R) > error && L < R)
    {
        //daca eroarea nu este in intervalul dorit si servoul este orientat spre stanga

        lr_angle += accuracy;
        //unghiul se ajusteaza

        if (lr_angle > 180) lr_angle = 180;
```

```
//limitare unghi servo

lr_servo.write(lr_angle);
delay(10);
//seteaza unghiul servoului
}
else lr_servo.write(lr_angle); //altfel se seteaza unghiul servoului

if (abs(U - D) > error && U >= D)
{
//daca eroarea nu este in intervalul dorit si servoul este orientat spre jos

ud_angle -= accuracy;
//unghiul se ajusteaza

if (ud_angle < 0) ud_angle = 0;
//limitare unghi servo

ud_servo.write(ud_angle);

delay(10);
//seteaza unghiul servoului
}
else if (abs(U - D) > error && U < D)
{
//daca eroarea nu este in intervalul dorit si servoul este orientat spre stanga

ud_angle += accuracy;
//unghiul se ajusteaza

if (ud_angle > 180) ud_angle = 180;
//limitare unghi servo

ud_servo.write(ud_angle);
delay(10);
//seteaza unghiul servoului
}
else ud_servo.write(ud_angle); //altfel se seteaza unghiul servoului
}

void LCD() {
lcd.setCursor(0, 0);
lcd.print("Light:");
lcd.setCursor(6, 0);
lcd.print(light);
}
```

```

lcd.setCursor(11, 0);
lcd.print("lux");
//afisare intensitate lumina
    lcd.setCursor(0, 1);
    lcd.print(temperature);
    lcd.setCursor(2, 1);
    lcd.print("C");
//afisare temperatura aer

    lcd.setCursor(5, 1);
    lcd.print(humidity);
    lcd.setCursor(7, 1);
    lcd.print("%");
//afisare umiditate aer

    lcd.setCursor(11, 1);
    lcd.print("res:");
    lcd.setCursor(15, 1);
    lcd.print(accuracy);
//afisare acuratete servouri
}
void Light(){
    light = lightMeter.readLightLevel();
    //citire intensitate lumina

}
void Dht(){
    temperature = DHT.temperature;
    //citire temperatura aer

    humidity = DHT.humidity;
    //citire umiditate aer
}
void adjust_accuracy() {
    tone(buzzer, 800, 100);
    delay(10);
    //redare sunet prin buzzer
    if (!digitalRead(buttonPin))
    {
        if(accuracy < 5) accuracy++;
        else accuracy = 1;
        //modificare acuratete servouri
    }
}

```

IV.4 Link catre GitHub

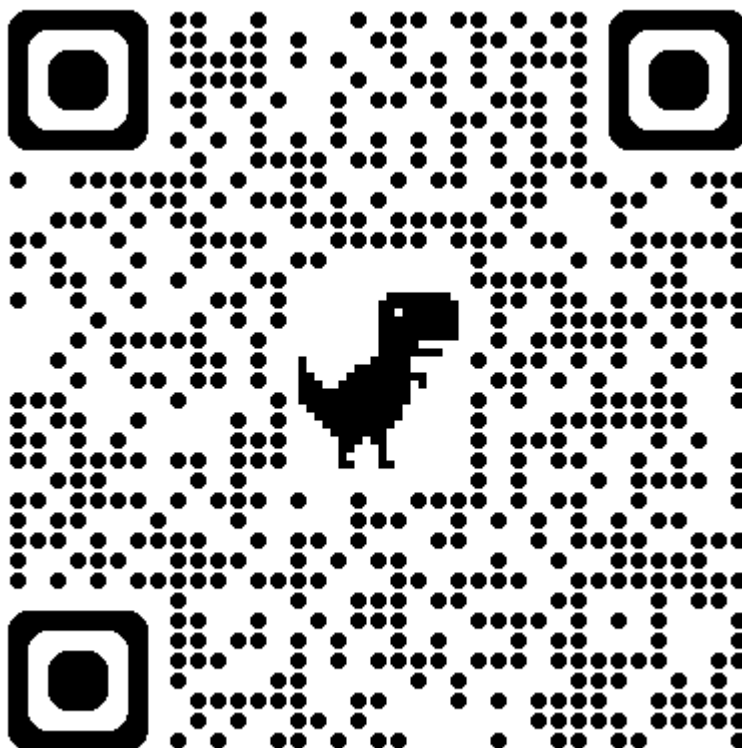
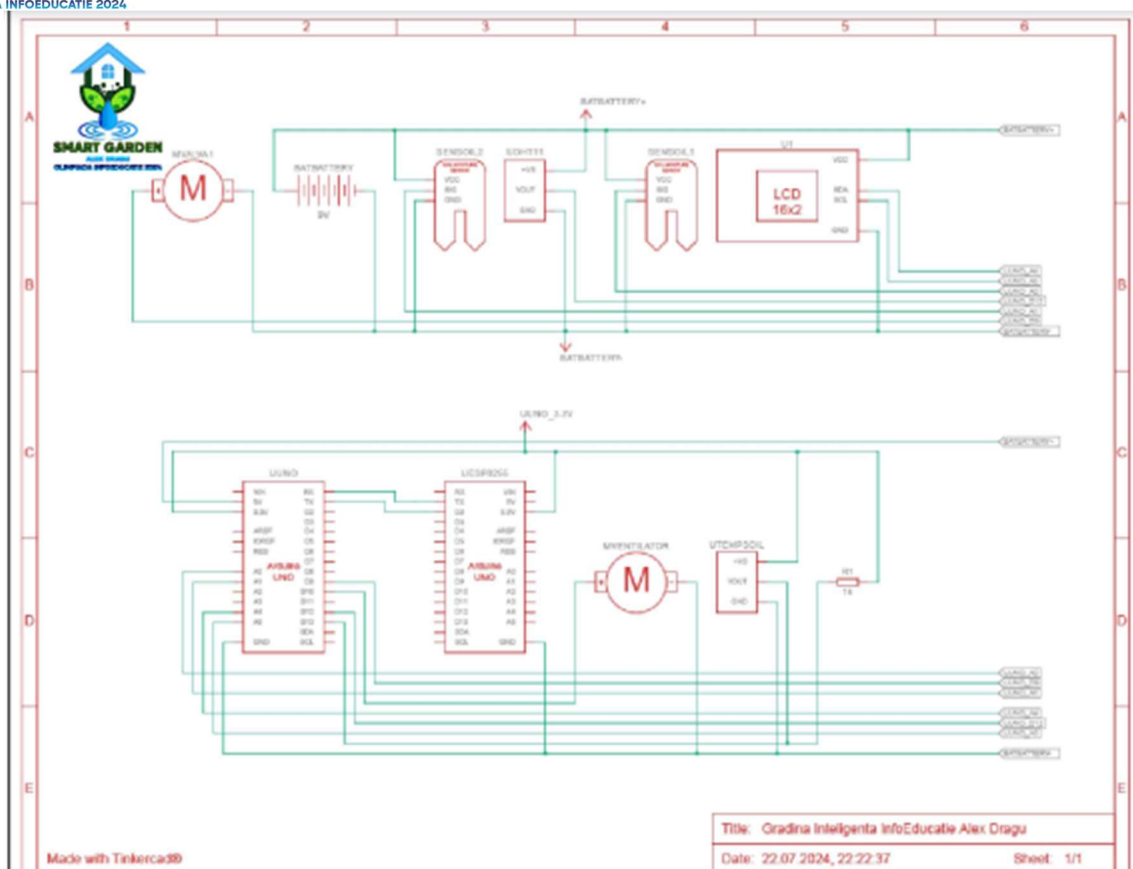


Fig. 19 - Link catre GitHub



Componente macheta:

- Tablou comanda
- Afisaj LED
- Relee comanda
- Doze legaturi
- Electrovalva apa NC
- Senzori capacitivi umiditate sol
- Sonda temperatura sol
- Senzor umiditate si temperatura aer
- Ventilator/motor
- Senzor rezistiv umiditate sol
- Componente retea de irigare

Fig. 20 – Schema electrică și componente machetă

Capitolul VI. Concluzii

Având în vedere că apa este un element vital vieții și trebuie utilizată rațional, sistemul gândit și implementat de mine, numit “Grădina inteligentă”, are rolul de a economisi resursele de apă și resursele umane.

Acest sistem poate fi aplicat în orice domeniu legat de creșterea și dezvoltarea plantelor (agricultura, sere, grădini și plante de apartament).

Capitolul VII. Posibilități de îmbunătățire

- Transformarea grădinii într-o casă inteligentă, adăugând mai mulți senzori cum ar fi senzori de gaz, detecție de mișcare sau o cameră de transmisie live
- Posibilitatea de adăugare a cât mai mulți senzori pentru o eventuală redistribuire a plantelor
- Posibilitatea de a adăuga sisteme de control pentru echipamente de pompare a apei (senzori de nivel pentru rezervoare de apă, relee comandă pompe de apă)

Capitolul VIII. Reportaje:

https://tvrinfo.ro/echipa-de-robotica-a-colegiului-na-ional-mihai-viteazul-din-ploie-ti-a-realizat-un-sistem-inteligent-de-iriga-ii-folosit-la-gradina-botanica-din-bascov_926211/

<https://fb.watch/pqLYnkCqgx/>

<https://www.observatorulph.ro/social/2678234-elevii-cnvm-ploiesti-au-introdus-un-sistem-inteligent-de-irigatii-la-gradina-botanica-bucov>

<https://republicanews.ro/sistem-inteligent-de-irigatii-pentru-gradina-botanica-video>

https://youtu.be/NUcoy_RShCg?feature=shared

<https://stirileprotv.ro/video/cum-a-fost-transformata-gradina-botanica-bucov-intr-o-sera-inteligenta-inovatia-lansata-de-elevii-unei-echipe-de-robotica/62328319/>

Capitolul IX. Screenshots

