Denumire

proiect: **Grădina inteligentă - sistem de irigație automatizat**

**implementat la sera de la Grădina Botanică Bucov**

****

Elev: Dragu Ioan Alexandru

Școala: Colegiul Național “Mihai Viteazul”, Ploiești

Clasa: 9

Profesor îndrumător: Rîpeanu Luminița

Oraș: Ploiești

Județ: Prahova

**Capitolul I. Utilitate practică**

La ce ne este necesară Grădina Inteligentă?

Ideea a venit în urma unei vizite la Grădina Botanică Bucov, unde personalul grădinii mi-a spus ca o plantă pentru a trăi și a se dezvolta are nevoie de lumină, căldură si apa, una din resursele importante ale vieții. Acolo am constatat că plantele sufereau din cauza lipsei de apă, deoarece personalul grădinii era insuficient. Totodată, udarea manuala 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.

Așa că am decis să creez Gradina Inteligentă.

Problema pe care am luat-o în considerare este sustenabilitatea, în sensul de a utiliza și păstra rezervele de apă, precum și facilitarea mai ușoară a studiilor asupra plantelor. Prin crearea acestui sistem am dorit să elimin cantitatea de apă risipită în udarea ineficientă a plantelor de la Grădina Botanică și, în același timp, să suplinesc resursa umană necesară udării manuale a acestora.

***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.***

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

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 întârziate, 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 gandit 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 ajutorului unui sistem Arduino de solar tracking care încarcă un acumulator de 9V.

**Capitolul III. Electronică**

**Placa Arduino UNO**

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

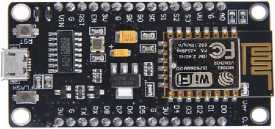
**SPECIFICAȚII TEHNICE**

• 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) of which 0.5 KB used by bootloader  
• 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.

Putere de ieșire + 19,5 dBm în modul 802.11b

Suport 802.11: b / g / n.

Conexiuni maximale concurente TCP: 5

**Placa relee intermediare**

Folosesc o placă cu relee intremediare pentru a putea comanda valvele/motoare/pompe

**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 si 30V, 10A DC

Mod de conexiune la terminalul de control: NC (Normal inchis), COM (Comun), NO (Normal deschis)

Indicatori LED: Verde pentru releu activat, Rosu pentru releu dezactivat

Compatibilitate: Suporta Windows 7, XP, 32-bit si 64-bit fara necesitatea de drivere suplimentare

Tip USB: HID (Human Interface Design)

Dimensiuni mm: 72 x 66mm

**Valve electromagnetice**

Valva electromagnetica ce permite deschiderea circuitului în momentul aplicării unei tensiuni de 12VDC cu ajutorul unui solenoid.Modelul este Normal Inchis (la aplicare tensiune 12V se deschide).

**SPECIFICAȚII TEHNICE:**

Tensiune alimentare: 12 VDC

Curent recomandat:0.6-1.5A

Marime racord: 1/2"

Material: Metal + plastic

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**

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).

### Denumirile pinilor:

pin 1: Vss;  
pin 2: Vdd;  
pin 3: Vo - contrast LCD (între Vss și Vdd V);  
pin 4: RS;  
pin 5: R/W;  
pin 6: E - enable;  
pin 7 - 14: DB0 - DB7;  
pin 15: V+ backlight;  
pin 16: V- backlight;

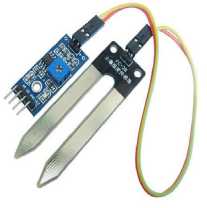
**Tipuri de senzori folosiți**

**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 trasmise sunt corecte.



Senzor capacitiv 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 apa.

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 apa presetat la 10l/ora devine activ și apa ajunge la rădacina 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 apa presetat al 5l/oră devine activ și apa ajunge la radacina 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 apa presetat al 2l/oră devine activ și apa ajunge la rădacina plantelor.

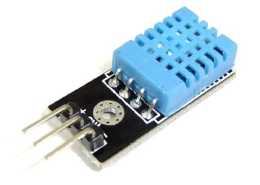
Când valoarea indicată de senzor este peste 30%, electrovalva este oprită.



**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: ± 2% RH

Acuratețea de măsurare a temperaturii: ± 0.5°

Tensiune de lucru: DC 3-5V

Consum: 2.5mA max (când se procesează datele)

Temperatura operare: -40 +80 grade Celsius

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. Senzori 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 masura temperatura solului.

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

**SPECIFICAȚII TEHNICE**



Alimentare: 3.0V până la 5.5V

Interval de temperatură: -55 °C până la + 125 °C

Precizie ridicată: ± 0,5 °C (-10 °C până la + 85 °C)

Tip senzor: DS18B20

Precizie de conversie 9 ~ 12 biți A / D

Întârziere scurtă în conversia temperaturii, maxim 750 ms

Suportă rețelele multi-point

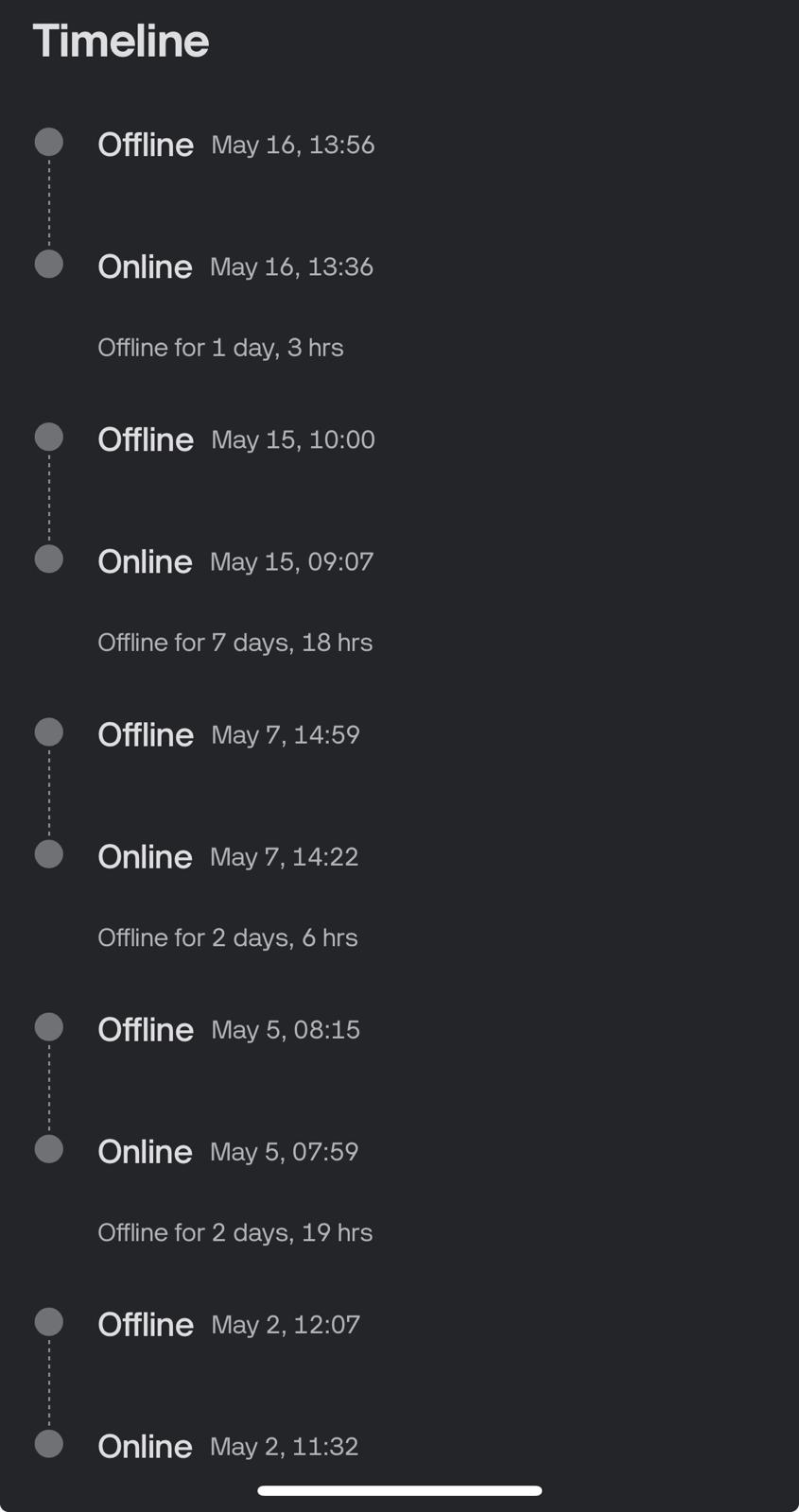
Reziztent la apă

***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.***

**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.

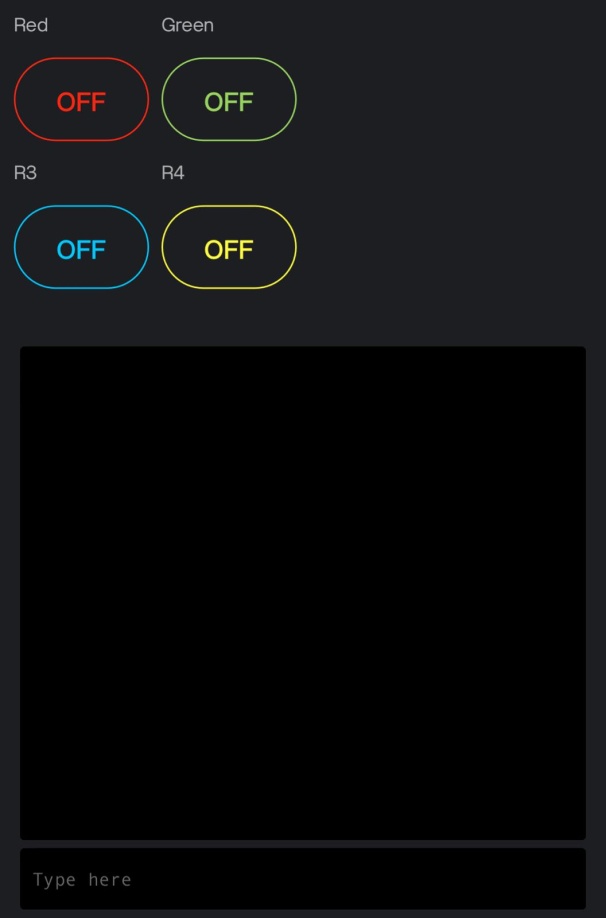
Deci, aplicația afișează pe telefon si pe calculator, în timp real, oriunde ai fi, atat datele din senzori, cat si daca val­va este deschisa sau inchisa si o poti controla de la distanta.



Istoricul sistemului, în ce zile și în ce interval orar a fost activ.



Meniul de pe calculator unde sunt afișate datele și pot fi activate zonele.



Meniul de pe telefon unde sunt afișate datele și pot fi activate zonele.

**Cod Placuță Arduino**

#**include** <DHT.h>

#**define** DHTPIN 12

#**define** DHTTYPE DHT22

DHT **dht**(DHTPIN, DHTTYPE);

//declarare senzor de umidititate 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

}

void **loop**() {

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); //afisare umiditate zona 1

delay(2000);

**if** (soilpercent0 <= 60)

{

digitalWrite(valva1, HIGH);

Serial.println("Valva1Pornit");

//daca umiditatea zonei scade sub 60%, valva se deschide

}

**else**

{

digitalWrite(valva1, LOW);

//altfel valva este inchisa

}

delay(2000);

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); //afisare umiditate zona 2

delay(2000);

**if** (soilpercent1 <= 50)

{

digitalWrite(valva2, HIGH);

Serial.println("Valva2Pornit");

//daca umiditatea zonei scade sub 50%, valva se deschide

}

**else**

{

digitalWrite(valva2, LOW);

//altfel valva este inchisa

}

delay(2000);

soilvalue2 = analogRead(A2); //citire senzor de umiditate sol zona 3

soilpercent2 = map(soilvalue2, AirValue2, WaterValue2, 0, 100); //transformarea datelor in intervalul 0-100

**if** (soilpercent2 >= 100)

{

soilpercent2 = 100;

}

**else** **if** (soilpercent2 <= 0)

{

soilpercent2 = 0;

}

Serial.println("H%Zona3="); Serial.println(soilpercent2); //afisare umiditate zona 3

delay(2000);

**if** (soilpercent2 <= 40)

{

digitalWrite(valva3, HIGH);

Serial.println("Valva3Pornit");

//daca umiditatea zonei scade sub 40%, valva se deschide

}

**else**

{

digitalWrite(valva3, LOW);

//altfel valva este inchisa

}

delay(2000);

float h = dht.readHumidity();

float t = dht.readTemperature();

//citire senzor de umiditate si temperatura aer

Serial.println("HAer="); Serial.println(h); //afisare umiditate aer

delay(2000);

Serial.println("TAer="); Serial.println(t); //afisare temperatura aer

delay(2000);

sensors.requestTemperatures(); //citire senzor temperatura sol

Serial.println("TSol="); Serial.println(sensors.getTempCByIndex(0)); //afisare temperatura sol

delay(2000);

}

**Cod Placuță Esp8266**

#**define** BLYNK\_TEMPLATE\_ID "xxxxx"

#**define** BLYNK\_DEVICE\_NAME "Gradina Inteligenta"

#**define** BLYNK\_AUTH\_TOKEN "xxxxx"

//declarare nume device, token de autentificare si id-ul template-ului

#**define** BLYNK\_PRINT Serial

//declarare canal de comunicare catre Blynk

#**include** <ESP8266WiFi.h>

#**include** <BlynkSimpleEsp8266.h>

#**include**<SoftwareSerial.h>

SoftwareSerial **mySerial**(D1,D2);

//declarare placuta Esp8266

char auth[] = BLYNK\_AUTH\_TOKEN;

char ssid[] = "xxxxx";

char pass[] = "xxxxx";

//declarare token de autentificare, numele retelei Wi-Fi si parola acesteia

#**define** wifiLed LED\_BUILTIN

//declarare led de pe placuta Wi-Fi

BlynkTimer timer;

//declarare unui timer cu ajutorul caruia verificam daca serverul Blynk este conectat

void **checkBlynkStatus**() {

//verificam daca serverul Blynk este conectat

bool isconnected = Blynk.connected();

**if** (isconnected == false) {

Serial.println("Blynk Not Connected");

digitalWrite(wifiLed, HIGH);

//daca nu este conectat, ledul de pe placuta Wi-Fi se aprinde

}

**if** (isconnected == true) {

digitalWrite(wifiLed, LOW);

Serial.println("Blynk Connected");

//altfel se stinge

}

}

void **setup**()

{

Serial.begin(115200);

mySerial.begin(9600);

//initializare placuta Esp8266

pinMode(wifiLed, OUTPUT);

//initializare led

Blynk.begin(auth, ssid, pass);

//initializare server Blynk

timer.setInterval(2000L, checkBlynkStatus);

//la fiecare 2 secunde verificam daca serverul este conectat

delay(1000);

}

void **loop**()

{

Blynk.run();

//pornire server Blynk

timer.run();

//pornire timer

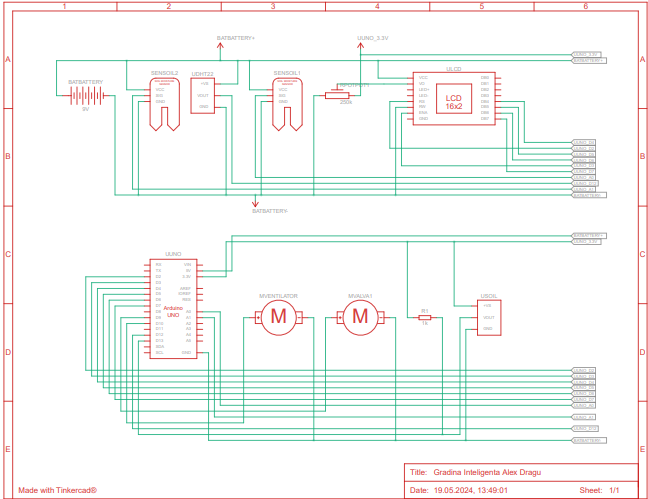
String msg = mySerial.readStringUntil('\r'); //citire date

Blynk.virtualWrite(V8,msg); //afisare date

delay(2000);

}

**Diagramă cablaj**

****