**Simulare unei sere care controlează temperatura și umiditatea**

În cadrul proiectului meu, m-am concentrat asupra unei soluții practice pentru monitorizarea și controlul mediului dintr-o seră, cu scopul de a optimiza condițiile de creștere a plantelor.

Componentele folosite de mine sunt următoarele (toate au fost achiziționate de pe https://sigmanortec.ro/):

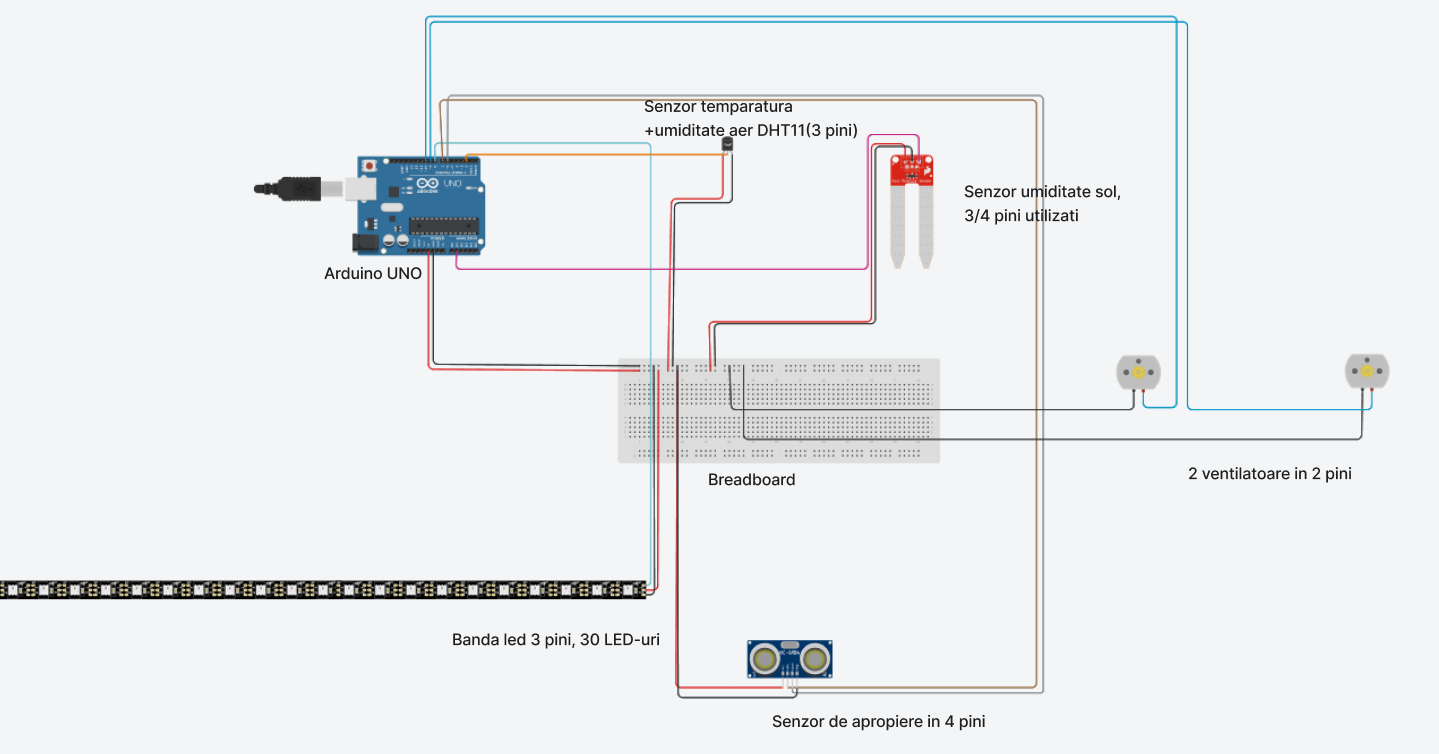
* ***Placa dezvoltare UNO R3 Arduino compatibil, AtMega328p;***
* ***Senzor temperatura si umiditate DHT11;***
* ***Breadboard 830 puncte;***
* ***Fire Dupont 30cm, Tata-Tata;***
* ***Fire Dupont 30cm, Tata-Mama;***
* ***Senzor umiditate sol, higrometru;***
* ***Senzor ultrasunete HC-SR04;***
* ***Suport senzor ultrasunete HC-SR04;***
* ***Ventilator 12V 40mm 2 pini - 2buc;***
* ***Banda LED adresabila, RGB, WS2812, 60led/m, 10cm, Neagra - 30 led.***

Senzorul DHT11 va fi utilizat pentru măsurarea temperaturii și umidității din interiorul serii, asigurând un mediu optim pentru creșterea plantelor. Senzorul de umiditate a solului va furniza informații despre starea solului, permițând monitorizarea și ajustarea nivelului de udare. Senzorul de apropiere va detecta prezența unei persoane în apropierea serei și poate fi folosit pentru a activa banda LED atașată deasupra serei.

Arduino Uno preia datele de la toți senzorii și le procesează într-o secvență de cod programată. Aceasta va include o rutină de comparare a valorilor citite cu pragurile prestabilite. Atunci când pragurile sunt atinse sau depășite, Arduino va controla ventilatoarele pentru reglarea temperaturii sau va activa banda LED pentru iluminarea suplimentară

Afișajul către utilizator poate fi realizat prin intermediul unui monitor serial, unde valorile citite de senzori și starea sistemului vor fi afișate în timp real. Astfel, utilizatorul poate monitoriza și interveni manual în sistem, dacă este necesar.

Am atașat schema montajului meu, dar țin sa fac câteva precizări: datorita spațiului de dezvoltare folosit (adică thinkercad), nu am putut face o schema 1 la 1 cu ce este în realitate, dar am explicat în poză ce ar reprezenta fiecare senzor



Acesta este codul introdus de mine in Arduino IDE. Ce verifica acesta?

1. Daca temperatura este una optima, intre 20-24 grade Celsius. Daca temperatura este <20 si >30, banda led se va face rosie pentru a avertiza faptul ca temperature este optima, iar daca temperatura >25 grade Celsius, vor porni ventilatoarele.
2. Daca umiditatea este scazuta, la fel, se va aprinde banda led pentru a semnala ca ceva nu este bine.
3. Senzorul de apropiere citeste distanta dintre noi si acesta, iar daca aceasta <30cm, atunci va porni banda LED pentru a putea vedea in interiorul serei.

Cod:

#include <DHT.h>

#include <Adafruit\_NeoPixel.h>

#define PIN\_LED 8

#define NUM\_LED 30

#define PIN\_TRIG 7

#define PIN\_ECHO 6

#define DHTPIN 2

#define DHTTYPE DHT11

const int umiditateSolPin = A1;

const int umiditateSolUscat = 385;

const int umiditateSolUd = 1021;

const int pinVentilator1 = 9;

const int pinVentilator2 = 10;

Adafruit\_NeoPixel strip = Adafruit\_NeoPixel(NUM\_LED, PIN\_LED, NEO\_GRB + NEO\_KHZ800);

DHT dht(DHTPIN, DHTTYPE);

unsigned long previousProxReadTime = 0;

unsigned long previousTempHumidityReadTime = 0;

void setup() {

Serial.begin(9600);

Serial.println("Citire date de la senzori: ");

dht.begin();

strip.begin();

strip.show();

pinMode(PIN\_TRIG, OUTPUT);

pinMode(PIN\_ECHO, INPUT);

pinMode(pinVentilator1, OUTPUT);

pinMode(pinVentilator2, OUTPUT);

}

void loop() {

unsigned long currentMillis = millis();

// Citire senzor de apropiere la fiecare 0.5 secunde

if (currentMillis - previousProxReadTime >= 500) {

previousProxReadTime = currentMillis;

long duration, distance;

digitalWrite(PIN\_TRIG, LOW);

delayMicroseconds(2);

digitalWrite(PIN\_TRIG, HIGH);

delayMicroseconds(10);

digitalWrite(PIN\_TRIG, LOW);

duration = pulseIn(PIN\_ECHO, HIGH);

distance = (duration / 2) / 29.1;

// Dacă distanța este mai mică de 30 cm, se va aprinde banda LED în alb

if (distance < 30) {

colorWipe(strip.Color(71, 69, 23), 10);

} else {

colorWipe(strip.Color(0, 0, 0), 50);

}

}

// Citim temperatura și umiditatea din sol la fiecare 5 secunde

if (currentMillis - previousTempHumidityReadTime >= 2000) {

previousTempHumidityReadTime = currentMillis;

float temperatura = dht.readTemperature();

float umiditate = dht.readHumidity();

int umiditateSol = analogRead(umiditateSolPin);

int umiditateSolCalibrata = map(umiditateSol, umiditateSolUd, umiditateSolUscat, 0, 100);

umiditateSolCalibrata = constrain(umiditateSolCalibrata, 0, 100);

Serial.print("Temperatura: ");

Serial.print(temperatura);

Serial.print(" °C | Umiditate: ");

Serial.print(umiditate);

Serial.print(" % | Umiditate sol: ");

Serial.println(umiditateSolCalibrata);

// Dacă umiditatea din sol scade sub 30% sau temperatura este mai mică de 20°C sau mai mare de 30°C, banda LED va fi roșie

if (umiditateSolCalibrata < 30 || temperatura < 20 || temperatura > 30) {

colorWipe(strip.Color(255, 0, 0), 10);

}

// Dacă temperatura depășește 25 grade Celsius, pornim ventilatoarele

if (temperatura > 25.0) {

Serial.println("Temperatura depășește 26°C - Ventilatoare activate!");

// Activează ventilatoarele

digitalWrite(pinVentilator1, HIGH);

digitalWrite(pinVentilator2, HIGH);

} else {

// Dezactivează ventilatoarele

digitalWrite(pinVentilator1, LOW);

digitalWrite(pinVentilator2, LOW);

}

}

}

void colorWipe(uint32\_t color, int wait) {

for(int i = 0; i < strip.numPixels(); i++) {

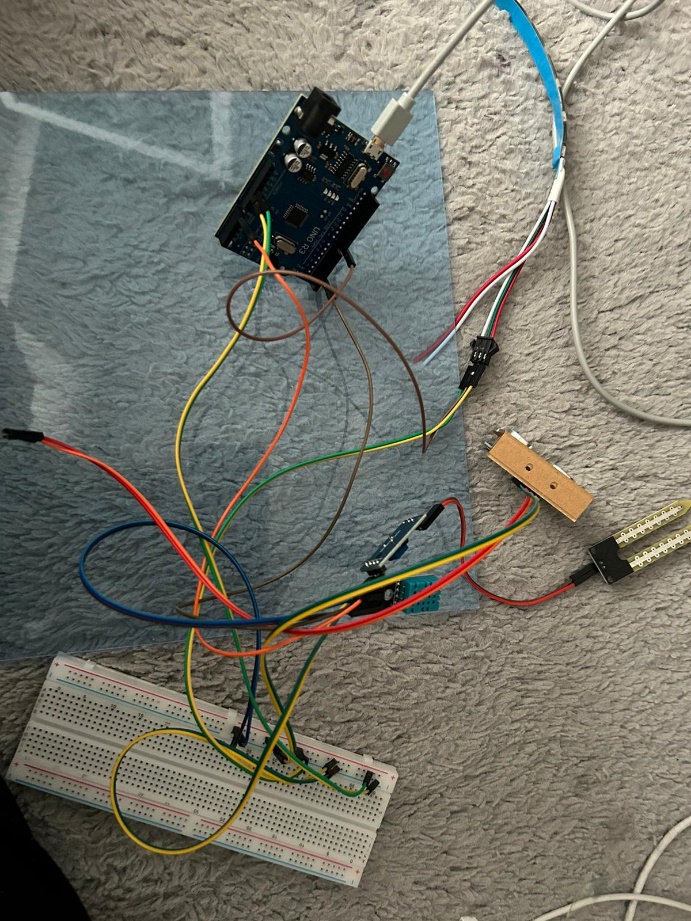
strip.setPixelColor(i, color);

}

strip.show();

delay(wait);

}

Poze realizate pe parcursul montării proietului:

