

Smart Greenhouse

Introducere

Proiectul intentioneaza sa faca cresterea si ingrijirea plantelor mai usoara, prin mentinerea factorilor optimi pentru bunastarea acestora, intr-un mediu controlat si automatizat. Probleme precum udarea prea des sau prea rar a plantelor vor fi inlaturate, deoarece proiectul va avea grija sa mentina umiditatea ideala a solului, dar va controla si nivelul de lumina si temperatura din interiorul serei. Acest proiect este perfect pentru iubitorii de natura care, din diferite motive, nu pot oferi un mediu perfect pentru ingrijire, dar totusi viseaza sa creasca propriile plante.

Descriere generală

Folosindu-se de datele oferite de un senzor de umiditate a solului, proiectul va decide daca plantele necesita apa si va porni un sistem de irigare daca solul este uscat. Cu ajutorul unui senzor de temperatura, daca gradele sunt crescute in interiorul serei, usile serei se vor deschide printr-un sistem actionat de un servomotor. In plus, pe baza informatiilor primite de la un senzor de luminozitate, va decide daca plantele necesita lumina in plus si, in situatia aceasta, va porni o banda de LEDuri.



Hardware Design

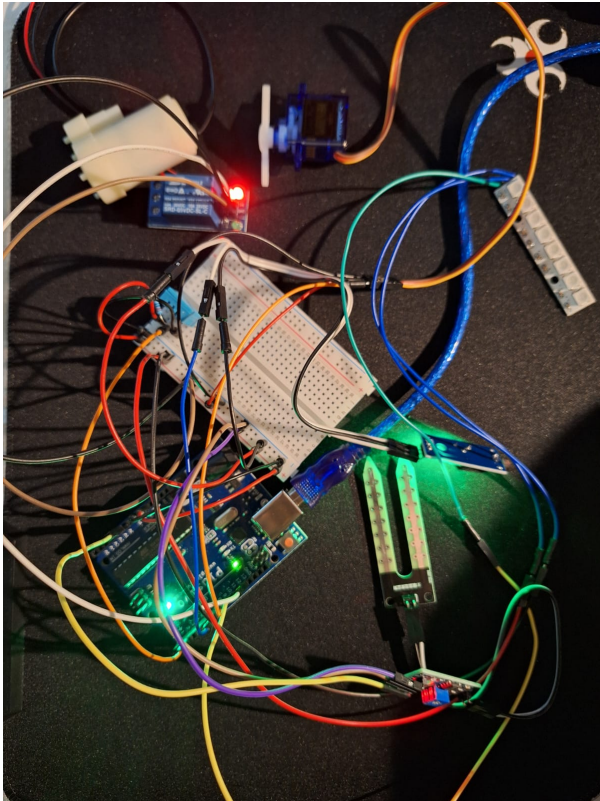
Lista pieselor necesare:

- Placa de dezvoltare Arduino Uno
- Modul cu senzor de umiditate a solului
- Senzor de temperatura DHT11
- Modul cu fotorezistor
- Pompa de apa
- Releu
- Banda de LEDuri WS2812
- Micro servomotor 9g
- Fire
- Rezistente
- Baterie pentru alimentarea pompei de apa

Schema electrica:



Cablajul proiectului:



Conectarea pinilor:

- Pinul A0 al placii Arduino este conectat la senzorul de umiditate a solului intrucat acesta intoarce date in format analog.
- Pinul D2 este conectat la senzorul de temperatura DHT11, care intoarce date in format digital.
- Pinul D7 este conectat la modulul photorezistor, care intoarce date digitale.
- Pinul D9 este conectat la servomotor, care functioneaza prin semnal PWM.
- Pinul D11 este conectat la (banda) de LED-uri, care pot fi comandate prin semnal PWM.
- Pinul D13 este conectat la releu, care actioneaza pompa de apa.

Functionarea unei componente din proiect:

In videoclipul [acesta](#) demonstrez functionalitatea modului cu fotorezistor. Atunci cand pe suprafata fotorezistorului cade lumina, banda LED este stinsa, iar atunci cand nu primeste suficienta lumina, se aprind LEDurile.

Concepte folosite din laboratoare

- **Timere**: Am folosit un timer pentru schimbarea culorii LEDurilor la un interval de timp fixat (10 secunde).
- **PWM**: Am folosit PWM pentru controlul servomotorului.
- **ADC**: Am folosit ADC pentru citirea semnalului analog dat

de senzorul de umiditate.

Software Design

Mediu de dezvoltare

Pentru dezvoltarea codului am folosit VSCode impreuna cu PlatformIO.

Librării și surse 3rd-party

1. **<SimpleDHT.h>**: pentru folosirea senzorului DHT11

Aceasta biblioteca a fost folosita pentru citirea datelor oferite de senzorul de temperatura. Cu ajutorul acesteia am reusit sa preiau informatii despre temperatura din interiorul serei.

2. **<Servo.h>**: pentru controlul servomotorului

Aceasta biblioteca a fost folosita pentru controlarea servomotorului ce are rolul de a deschide fereastra serei atunci cand temperatura din interior este crescuta. Biblioteca ofera o interfata pentru controlul pozitiei servomotorului pe baza unui unghi dat.

3. **<MsTimer2.h>**: pentru folosirea timerului 2

Biblioteca aceasta este folosita pentru schimbarea culorii LEDurilor la un interval de timp dat. Biblioteca inglobeaza in functiile sale configurarea timerului 2 al placii cu o rezolutie de 1ms.

4. **<WS2812.h>**: pentru controlul LEDurilor

Am folosit aceasta biblioteca pentru controlul benzii de LEDuri RGB. In cadrul bibliotecii se gasesc functionalitati ce permit setarea culorii fiecarui LED in parte.

Variabile folosite in cod

- Constante pentru descrierea pinilor de pe placa Arduino folositi in cod;
- Constante pentru pragurile senzorilor (temperatura, procent umiditate), pozitiile servomotorului, culorile LEDurilor;
- Variabile pentru starea componentelor.

Algoritmi și structuri implementate

void setup():

- initializeaza pinii si alte componente necesare programului

void changeColor():

- functie ajutatoare pentru timer; cicleaza prin culorile prestabilite pentru LEDuri

void tempServo():

- verifica temperatura primita de la senzorul DHT si actualizeaza starea servomotorului daca este cazul, rezultand in deschiderea sau inchiderea ferestrei serei.

void photoresLEDs():

- verifica daca fotorezistorul detecteaza prezenta luminii si in functie de aceasta aprinde sau stinge LEDurile. De fiecare data cand sunt aprinse, ciclul de culori va incepe de la lumina alba.

void humidityRelay():

- calculeaza procentajul de uscare al solului pe baza datelor primite analog de la senzorul de umiditate si activeaza pompa de apa daca acesta este crescut.

void loop():

- apeleaza pe rand tempServo, photoresLEDs si humidityRelay pentru a asigura functionarea celor trei caracteristici principale ale proiectului.

Rezultate Obținute

Proiectul este finalizat!

In [acest](#) videoclip demonstrez functionalitatea senzor umiditate + pompa de apa.

[Aici](#) prezint cum functioneaza fotorezistorul + LEDurile.

Iar in [aceasta](#) inregistrare demonstrez functionarea senzorului de temperatura + servomotor.

Concluzii

Am reusit sa finalizez proiectul cu toate functionalitatile pe care le-am planuit initial. A fost o provocare, dar si un proces interesant si satisfactor, deoarece am avut ocazia sa ma implic in aspecte tehnice si conceptuale cu care nu m-am intalnit prea des în cadrul facultatii. Mi-a placut sa invat cum sa conectez diferite componente intre ele si sa vad cum codul scris de mine produce efecte in viata reala.

Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună 😊.

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul).
Exemplu: Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

https://drive.google.com/file/d/1vpRT06CqU83oDTfOaHkoniJPTcKVU-3N/view?usp=drive_link

Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

Bibliografie/Resurse

https://www.youtube.com/watch?v=dJJAQxyryoQ&ab_channel=CircuitBasics

- pentru utilizarea DHT11

https://www.optimusdigital.ro/ro/electronica-de-putere-module-cu-releu/13084-modul-releu-cu-un-canal-comandat-cu-5-v.html?search_query=releu+5v&results=108

- pentru conectarea si utilizarea releului

https://www.optimusdigital.ro/ro/optoelectronice-bare-cu-leduri/753-bara-de-led-uri-rgb-ws2812-cu-8-led-uri.html?search_query=banda+led&results=23

- pentru conectarea si utilizarea LEDurilor

https://www.youtube.com/watch?v=SfmHNb5QAzc&ab_channel=educ8s.tv

- pentru conectarea si utilizarea servomotorului

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/avaduva/miruna.dolineanu>



Last update: **2024/05/27 06:53**