

**Universitatea
Transilvania
din Brașov**

**FACULTATEA DE INGINERIE ELECTRICĂ
ȘI ȘTIINȚA CALCULATOARELOR**

PROIECT DE DIPLOMĂ

Conducător științific:

Prof.dr.ing COTFAS Daniel Tudor

Prof.dr.ing COTFAS Petru Adrian

Absolvent:

GRIGORAȘ Victor-Emil

BRAȘOV 2023

Departamentul Electronică și Calculatoare
Programul de studii: Electronică Aplicată

GRIGORAȘ Victor-Emil

Sistem de monitorizare al unei sere alimentat cu energie regenerabilă

Conducător științific:

Prof.dr.ing COTFAS Tudor Daniel

Prof.dr.ing COTFAS Petru Adrian

Brașov 2023

Cuprins

Cuprins

Lista de figuri, tabele și coduri sursă.....	5
Lista de acronime	7
1. Introducere	8
1.1 Generalități	8
1.2 Tema proiectului.....	9
1.3 Scopul.....	9
1.4 Obiectivele urmărite	10
1.5 Stadiul actual.....	10
2. Arhitectură hardware	16
2.1 Placa de dezvoltare ESP32.....	17
2.2 Senzorii utilizați, specificații, integrare și testare	19
2.2.1 Senzorul de lumina LDR	19
2.2.2 Senzorul de umiditate și temperatură DHT11	20
2.2.3 Senzorul higrometric capacitiv de umiditate a solului SEN0193	21
2.3 Componente necesare.....	22
2.3.1 Servomotoare MG996R.....	22
2.3.2 Panou fotovoltaic cu celule policristaline	23
2.3.3 Controller încărcare solară	25
2.3.4 Modul Display Oled SSD1306	26
2.3.5 Modul releu 2 canale comandat pe 5V	27
2.3.6 Pompă apă/aer cu diafragmă, 6-12V, R385	28
2.3.7 LED UV (Ultraviolete)	29
2.4 Baterie Litiu-Ion 12V	30
2.5 Sistemul final.....	31
2.5.1 Schema electrică a sistemului.....	31
2.5.2 Testarea componentelor	33
3. Arhitectură software	37

3.1	Programare senzori.....	37
3.1.1	Configurare senzor de lumina LDR	37
3.1.2	Configurare senzor de umiditate a solului	42
3.1.3	Programarea propriu zisă în vederea acționării sistemului.....	43
3.2	Pagina web pentru monitorizarea serei.....	44
3.2.1	Realizarea paginii web.....	44
4.	Simulare Simulink	46
4.1	Matlab Simulink.....	47
4.2	Modul fotovoltaic	47
4.3	Convertorul Buck.....	49
4.4	Algoritm P&O MPPT (Perturbare și observare).....	50
4.5	Semnalul de radiație.....	52
4.6	Sistemului de încărcare al bateriei.....	53
4.7	Concluziile acestei simulări.....	56
5	Rezultate experimentale	56
6	Concluzii	61
7	Bibliografie.....	62
	Rezumat	64
	Abstract	65
	Anexe.....	66

LISTA DE FIGURI, TABELE ȘI CODURI SURSĂ

FIGURI

- Figura 1. Sistemul SOLARINO
- Figura 2. Aplicația de control a sistemului SOLARINO
- Figura 3. Grafic pentru temperatură realizat de sistemul SOLARINO
- Figura 4. Interfața aplicației BlueMonitor SMART Greenhouse instalată pe telefon mobil, tabletă, laptop
- Figura 5. Schema bloc a sistemului
- Figura 6. Pini ESP32
- Figura 7. Senzorul și simbolurile LDR
- Figura 8. Senzorul de umiditate și temperatură DHT11
- Figura 9. Senzorul higrometric capacitiv de umiditate SEN0193
- Figura 10. Servomotorul MG996R
- Figura 11. Curbele de iradiere ale panoului utilizat
- Figura 12. Caracteristicile panoului în funcție de temperatură
- Figura 13. Panou fotovoltaic 10W
- Figura 14. Modul control încărcare solară PWM
- Figura 15. Display Oled SSD1306
- Figura 16. Modul releu 2 canale comandat pe 5V
- Figura 17. Pompă apă/aer cu diafragmă, 6-12V, R385
- Figura 18. LED UV 395-400nm
- Figura 19. Conectarea BMS-ului la baterie
- Figura 20. Schema electrică a sistemului
- Figura 21. Placa de prototipare cu cei 4 LDR
- Figura 22. Placa de prototipare pe a 2-a față
- Figura 23. Schema circuitului celor 4 LDR
- Figura 24. Construcție seră
- Figura 25. Sistemul final (1)
- Figura 26. Sistemul final (2)
- Figura 27. Schema electrică pentru RLDR
- Figura 28. Dependența intensității luminoase de tensiunea aplicată pe LDR
- Figura 29. Dependența intensității luminoase de tensiunea aplicată pe pinul de ADC

- Figura 30. Testarea senzorilor
- Figura 31. Pagina Web a sistemului
- Figura 32. Schema bloc a sistemului de încărcare Simulink
- Figura 33. Circuitul echivalent al unui sistem fotovoltaic
- Figura 34. Parametri panoului fotovoltaic Simulink
- Figura 35. Caracteristica I-V și P-V pentru temperatură constantă
- Figura 36. Convertorul Buck Simulink
- Figura 37. Algoritmul de bază P&O MPPT
- Figura 38. P&O MPPT algoritm Simulink
- Figura 39. Programarea MPPT-ului în Matlab
- Figura 40. Graficul semnalului de radiație
- Figura 41. Simularea sistemului de încărcare al bateriei
- Figura 42. Graficul de încărcare al bateriei
- Figura 43. Calculul eficienței
- Figura 44. Comparația dintre sistemul fix și cel de urmărire solară
- Figura 45. Imagine din timpul experimentului
- Figura 46. Sistemele de răcire și irigare oprite
- Figura 47 Sistemul de răcire pornit
- Figura 48. Sistemul de irigare pornit

TABELE

- Tabelul 1. Tensiunea de ieșire a celor patru senzori LDR
- Tabelul 2. Valorile măsurate pentru a calcula numărul de luși
- Tabelul 3. Valoarea curentului și a tensiunii sistemului fotovoltaic fix la diferite intervale de timp
- Tabelul 4. Valoarea curentului și a tensiunii sistemului fotovoltaic de urmărire solară la diferite intervale de timp
- Tabelul 5. Câștigul de putere obținut de sistemul de urmărire solară

CODURI SURSĂ

- Codul 1. Citirea senzorilor LDR și mediile valorilor senzorilor
- Codul 2. Ajustarea servo-motoarelor în funcție de lumina solară
- Codul 3. Scriptul JavaScript folosit pentru pagina web

LISTA DE ACRONIME

AC – ALTERNATING CURRENT

ADC – ANALOG DIGITAL CONVERTOR;

DC – DIRECT CURRENT

HTML – HYPER TEXT MARKUP LANGUAGE

HTTP – HYPERTEXT TRANSFER PROTOCOL

LDR – LIGHT DEPENDENT RESISTOR;

MPPT – MAXIMUM POWER POINT TRACKING

PV – PHOTOVOLTAIC;

PWM – PULSE-WIDTH MODULATION;

SPI – SERIAL PERIPHERAL INTERFACE

ST – SOLAR TRACKER

URL – UNIFORM RESOURCE LOCATOR

UV – ULTRAVIOLET

1. INTRODUCERE

Generalități

Tema proiectului

Scopul

Obiectivele urmărite

Stadiul actual

1.1 GENERALITĂȚI

În contextul actual energia este considerată a fi o resursă esențială pentru orice activitatea umană. Sursele de energie pot fi împărțite în două categorii, surse primare de energie (combustibili fosili, nucleari și alte surse care sunt limitate în timp) și surse regenerabile de energie (energie solară, eoliană, geotermică) acestea sunt disponibile în cantități nelimitate și se regenerează prin procese naturale.

Sistemele fotovoltaice pot fi utilizate pentru o gamă largă de aplicații, inclusiv pentru generarea de electricitate pentru locuințe, clădiri comerciale, clădiri industriale și ferme. Un sistem fotovoltaic tipic constă din panouri solare, un controller de încărcare, baterii și un invertor. Panourile solare pot fi montate pe acoperișe sau la sol și absorb lumina solară pentru a genera electricitate de curent continuu (DC). Invertorul are rolul de a converti curentul continuu în curent alternativ (AC). Bateria stochează excesul de energie produsă pe timpul zilei pentru a fi utilizat în perioadele fără lumină, cum ar fi noaptea. Sistemele fotovoltaice sunt ecologice, deoarece nu emit gaze nocive sau poluanți în timpul funcționării, de asemenea aceste sisteme necesită întreținerea minimă și pot avea o durată de viață de până la 25 de ani sau mai mult. Eficiența sistemelor fotovoltaice depinde de diferiți factori, inclusiv cantitatea de lumină solară primită, unghiul și orientarea panourilor solare.

Un sistem de urmărire solară este un dispozitiv care urmărește automat mișcarea soarelui pe parcursul zilei, asigurând permanent panourilor solare orientarea spre soare pentru a maximiza producția de energie. Există două tipuri principale de sisteme de urmărire solară: cu o singură axă sau cu două axe. Sistemele cu o singură axă deplasează panourile solare într-o singură direcție, fie de la est la vest, fie de la nord la sud, în funcție de design. Sistemele de

urmărire solară cu două axe pot mișca panourile în ambele direcții pentru a urmări mișcarea soarelui mai precis.

Sistemele de urmărire solară pot crește semnificativ producția de energie a unui sistem de energie solară cu până la 25% în comparație cu panourile montate fix. Acest lucru se datorează faptului, că panourile solare generează cea mai mare cantitate de electricitate atunci când panourile sunt orientate direct spre soare. Prin urmărirea mișcării soarelui, sistemele de urmărire solară pot menține un unghi optim al panourilor față de soare.

1.2 TEMA PROIECTULUI

Tema proiectului meu constă în realizarea unui prototip al unui sistem inteligent pentru o seră, care va folosi o sursă de energie regenerabilă.

O seră este utilizată în principal pentru a cultiva anumite tipuri de plante pe tot parcursul anului sau plante care necesită o monitorizare continuă pentru a obține o calitate și o cantitate ridicată.

Serele solare oferă o oportunitate de integrare între lumea agricolă și cea a surselor regenerabile, unde este posibilă cultivarea și utilizarea panourilor fotovoltaice pentru a produce energie fără a ocupa terenuri. În sere temperatura și umiditatea sunt controlate prin intermediul sistemelor de ventilație, încălzire și umidificare. Toate aceste sisteme au nevoie de o cantitate mare de energie pentru a funcționa.

În acest proiect am ales să combin agricultura cu generarea de energie electrică dintr-o sursă de energie regenerabilă (panouri cu sistem Solar Tracker). Sistemul utilizează patru senzori de lumină pentru a detecta poziția soarelui pe cer și pentru a face ajustările necesare pentru panoul solar.

1.3 SCOPUL

Scopul acestei lucrări este de a reduce dependența de sursele tradiționale de energie, și de a contribui la protejarea mediului înconjurător prin reducerea emisiilor de gaze cu efect de seră și a altor poluanți.

Prin utilizarea sistemelor de energie regenerabilă, cum ar fi panourile solare fotovoltaice, sistemele de urmărire solară sau turbinele eoliene, serele pot obține energia electrică necesară pentru iluminarea, încălzirea și răcirea mediului de creștere a plantelor.

Aceste sisteme pot reduce, de asemenea costurile operaționale ale serelor și pot crește rentabilitatea producției agricole, prin eliminarea costurilor de energie și prin creșterea productivității plantelor prin asigurarea unui mediu controlat și optim pentru creșterea lor.

1.4 OBIECTIVELE URMĂRITE

- Transpunerea cunoștințelor teoretice în vederea realizării sistemului final;
- Realizarea unui sistem care să funcționeze în condiții optime;
- Creșterea eficienței panourilor solare prin urmărirea mișcării soarelui;
- Reducerea costurilor de energie electrică în sere;
- Integrarea mai multor sisteme de automatizare.

1.5 STADIUL ACTUAL

Într-o seră modernă, există multe procese care rulează simultan și interacționează între ele. Toate aceste procese sunt interconectate și se bazează pe automatizarea serelor [1]. Sistemele electrice de înaltă performanță sunt esențiale pentru fiabilitatea operațională a unei afaceri. Productivitatea recoltei depinde de echilibrul optim între toate procesele din seră, de la plantare până la recoltare, de la gestionarea apei până la realizarea unui climat ideal. Pentru ca plantele să aibă posibilitatea de a crește cât mai eficient, trebuie să se obțină și să se mențină continuu un anumit climat pe toată perioada anului [1].

Sera SMART este echipată cu un sistem de automatizare integrat care își controlează singur microclimatul care poate fi conectat la panouri solare. În plus, acest sistem vă informează pe telefonul mobil despre starea parametrilor de seră și plante, înregistrează toate valorile parametrilor într-o bază de date pentru o perioadă îndelungată. Sistemul, poate fi programat să își regleze parametrii: temperatura, lumina, umiditatea, ventilația etc. Depinde de cultura selectată pe telefonul mobil [2].



Figura 1. Sistemul SOLARINO [2]

Soluția care oferă aceste avantaje a fost dezvoltată de compania de tehnologie **Mobilecontrol**. SOLARINO (Figura 1) este un sistem integrat de control al serelor și solarelor agricole, acesta constă dintr-un dispozitiv instalat în seră și o aplicație (Figura 2) de control care poate fi utilizată pe un telefon mobil [3].

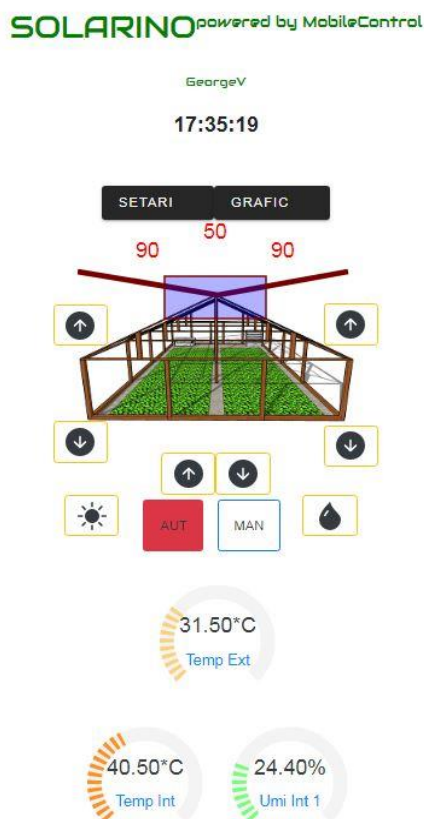


Figura 2. Aplicația de control a sistemului SOLARINO [3]

Funcționalitățile sistemului sunt:

- acționarea elementelor de aerisire, umbrire, irigare și iluminare;
- poate controla motoare pe 24V (personalizat 220V sau 380V);
- controlul ventilației se face în funcție de temperatura interioară, exterioară și viteza vântului. Deschide / închide ferestrele de ventilație la o temperatură setată de utilizator din telefonul mobil și le închide forțat la depășirea unui prag critic al vitezei vântului. Poate acționa și ventilația forțată în cazul în care temperatura este mare și viteza vântului nu permite deschiderea ferestrelor;
- acționarea umbririi se face manual sau automat în funcție de nivelul de radiație;
- acționarea irigației se face în funcție de senzorul de umiditate în sol;
- poate acționa și electrovalve de apă în procesul de irigație;
- acționarea motoarelor de deschidere ferestre se poate face și manual de pe panoul central.

În acest moment sistemul este prezentat în șase variante care diferă în funcție de dimensiunea și de echiparea acestuia cu senzori. Toate versiunile sunt cu WiFi sau cu router 3G integrat pentru comunicarea cu serverul de management. Aceste sisteme sunt autonome și pot funcționa o vreme și fără o conexiune la internet, dar conexiunea la internet este importantă pentru a putea seta parametrii și a monitoriza starea serei în timp real. Sistemul poate fi, de asemenea operat manual sau local prin intermediul unui telefon mobil.

Toate soluțiile Solarino oferă, pe lângă controlul la distanță foarte precis și o serie de instrumente analitice, grafice (Figura 3) și rapoarte foarte detaliate [1].

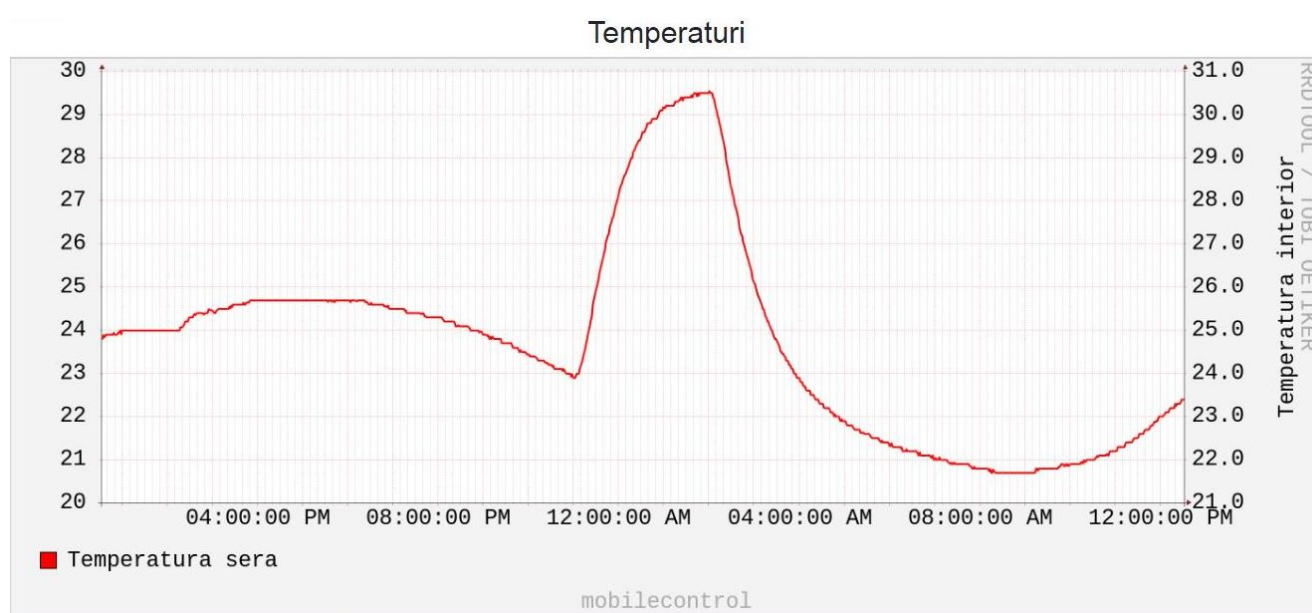


Figura 3. Grafic pentru temperatură realizat de sistemul SOLARINO [2]

Monitorizarea serelor cu BlueMonitor (Figura 4) SMART Greenhouse

Conceptul de automatizare a serelor este destul de general. Monitorizarea online a temperaturii și umidității în seră și solarii, a temperaturii și umidității solului și a luminii, precum și reglarea automată a acestor parametri în sere și solarii, reduce costurile de producție prin măsurarea corectă a alimentării cu apă și îngrășămintă și prin reducerea nerezonabilă a puterii de alimentare [4].



Figura 4. Interfața aplicației BlueMonitor SMART Greenhouse instalată pe telefon mobil, tabletă, laptop [4]

Funcționalități:

- Monitorizarea temperaturii și a umidității aerului, umidității solului, nivelurile de iluminare din seră și funcționarea instalațiilor din teren în timp real de pe orice computer, tabletă sau smartphone;
- Primirea alertelor prin e-mail sau SMS atunci când valorile normale sunt depășite, cu locația serei sau a sectorului în care s-au înregistrat valorile de alertă, pentru a corecta rapid problemele care pot afecta dezvoltarea plantelor;
- Vizualizarea rapidă a rapoartelor de analiză și a graficelor cu valori înregistrate pe diferite perioade de timp pentru a îmbunătăți planificarea producției;
- Trimiterea de comenzi de ajustare a parametrilor oricând și oriunde, de pe orice computer, tabletă sau smartphone [4].

Agrovoltaica

Agrovoltaica constă în folosirea aceleiași suprafețe de teren pentru a obține atât energie solară, cât și produse agricole. Principalele beneficii ale acestei energii este că reduce emisiile de gaze cu efect de seră din sectorul agricol, utilizarea duală a terenurilor pentru agricultură, cât și pentru energie. Studiile estimează că energia generată din panourile solare crește economia fermelor agrovoltaice cu peste 30%.

Integrarea panourilor solare cu terenul agricol poate ajuta, de asemenea, la reducerea cantității de teren necesar pentru serele solare, minimizând astfel potențialele conflicte legate de utilizarea terenului. Chiar dacă aceste sisteme agrovoltaice oferă multe beneficii, acestea necesită o planificare și gestionare atentă pentru a se asigura că panourile solare nu afectează negativ producția de culturi.

Există diferite tipuri de sisteme agrovoltaice, cum ar fi cele care utilizează module fotovoltaice montate pe stâlpi deasupra culturilor agricole, cele care utilizează panouri solare montate pe acoperișul serei, sisteme cu panouri solare mobile care se pot deplasa în funcție de poziția soarelui pentru a maximiza producția de energie solară. Aceste sisteme pot fi utile atât pentru agricultori pentru a reduce costurile de energie, cât și pentru companiile energetice care doresc să producă energie solară în zonele rurale.

Eficiența panourilor solare depinde de mai mulți factori, inclusiv nivelul de radiație solară, temperatură și umiditate. În sistemele agrovoltaice, panourile solare sunt amplasate la înălțime, astfel încât acestea pot beneficia de mai multă lumină solară, sistemul poate fi proiectat să maximizeze producția de energie solară, permițând culturilor să primească doar lumina necesară pentru a crește [5].

Avantajele și dezavantajele utilizării energiei regenerabile. Structura unui sistem fotovoltaic.

Energia solară este o sursă regenerabilă, deoarece aceasta va fi disponibilă în continuare atât timp cât soarele va exista. În timp ce reactoarele nucleare și instalațiile termice generează poluare în timpul funcționării lor, prin evacuarea deșeurilor nucleare sau producerea de fum și cenușă, panourile solare nu produc niciun fel de emisii poluante în timpul utilizării lor. Comparativ cu alte metode de producție a energiei, utilizarea panourilor solare sau a altor tehnologii bazate pe energie solară este caracterizată de absența zgomotului. Montarea panourilor solare este facilă și eficientă din punct de vedere al costurilor.

Panourile solare au nevoie de mai puțină mentenanță și monitorizare. Există și unele dezavantaje cum ar fi costul ridicat al componentelor folosite, celulele solare funcționează doar pe timpul zilei, eficiența panourilor este destul de redusă în zilele înnorate.

O celulă fotovoltaică (PV) convertește energia luminii solare în energie electrică prin intermediul efectului fotovoltaic. Celulele fotovoltaice sunt constituite din materiale semiconductoare, cum ar fi siliciul sau arsenură de galiu, care absorb lumină și generează curent electric. De obicei, celulele fotovoltaice sunt grupate în module solare, care pot fi interconectate pentru a forma un sistem fotovoltaic mai mare. Celulele fotovoltaice sunt utilizate într-o gamă largă de aplicații, cum ar fi sistemele de alimentare electrică pentru case și clădiri comerciale, serele, aparatele electronice portabile, sursele de alimentare pentru sateliți și alte domenii.

Sistemul fotovoltaic este un sistem de putere conceput pentru a furniza energie electrică prin intermediul panourilor fotovoltaice. Aceste sisteme sunt compuse din mai multe componente: panouri fotovoltaice, care sunt responsabile de transformarea energiei solare în energie electrică. Acestea sunt alcătuite dintr-un strat subțire de material semiconductor (de obicei siliciu), care este în contact cu un material conductor și cu un strat protectiv. Atunci când lumina solară atinge panourile fotovoltaice, electronii din materialul semiconductor sunt eliberați și produc curent electric. Invertorul este o altă componentă a unui sistem fotovoltaic, acesta are rolul de a transforma energia electrică produsă de panourile fotovoltaice (curent continuu c.c) în energie electrică care poate fi folosită de dispozitivele electrice de uz casnic sau de alte echipamente (curent alternativ c.a). Bateriile sunt utilizate pentru a stoca energia electrică și acestea pot fi folosite pentru a furniza energie electrică atunci când soarele nu este disponibil sau când cererea de energie electrică este mai mare decât producția de energie electrică din panouri. Sistemul de cablare și de control este folosit pentru a conecta toate componentele sistemului fotovoltaic și pentru a controla fluxul de energie electrică în întregul sistem fotovoltaic.

În general, sistemele fotovoltaice sunt considerate o opțiune ecologică și economică pentru producerea de energie electrică. Ele pot fi folosite pentru a reduce costurile energiei electrice și pentru a reduce emisiile de gaze cu efect de seră.

Un sistem de urmarire a soarelui ST este un dispozitiv care urmărește mișcarea soarelui și ajustează poziția panourilor solare astfel încât acestea să fie mereu orientate spre soare. ST-ele pot fi construite în diferite forme și dimensiuni, dar toate au în comun anumite componente de

bază cum ar fi: senzorii care detectează poziția soarelui și transmit informații către controler, motorul care este responsabil cu mutarea panourilor solare în direcția dorită. Există două tipuri principale de astfel de sisteme, sisteme de urmărire solară cu o singură axă sau cu două axe.

Sistemele cu o singură axă mișcă sau ajustează panourile solare prin rotirea în jurul unei axe. Mișcarea sa este de obicei aliniată în direcțiile nord și sud.

Un tracker cu două axe permite panourilor să se rotească pe două axe simultan, este aliniat atât orizontal, cât și vertical, adică se poate regla în toate direcțiile – nord, sud, est și vest. Acest sistem de urmărire solară cu două axe este conceput pentru a maximiza generarea de energie solară pe tot parcursul anului.

2. ARHITECTURĂ HARDWARE

Placa de dezvoltare ESP32

Senzori utilizați, specificații, integrare și testare

Componente necesare

Baterie Litiu-Ion 12V

Sistemul final

Arhitectura hardware a proiectului se bazează pe utilizarea unui microcontroller ESP32, a două servomotoare pentru solar tracker-ul cu două axe, patru senzori de lumină, panoul solar care furnizează la ieșire 12V, un controler solar pentru încărcarea bateriei de 12V, un senzor de temperatură și umiditate, senzor higrometru capacitiv pentru verificarea umidității solului, o pompă de apă cu diafragmă, LED-uri UV pentru creșterea plantelor și un display oled pentru afișarea datelor de la solar tracker și seră.

Sistemul este alimentat cu ajutorul unui panou fotovoltaic. Senzorii utilizați în proiectul meu, servomotoarele, display-ul oled, senzorul de temperatura și umiditate, senzorul higrometric sunt alimentați de la 5V, pompa de apă și ventilatorul este alimentat de la 12V, microcontroller-ul este alimentat la 5V.

În imaginea de mai jos am reprezentat schema de ansamblu a sistemului și modul în care am interfațat toate componentele. Viziunea mea a fost ca pentru fiecare intrare reprezentată de senzori, să existe o ieșire.

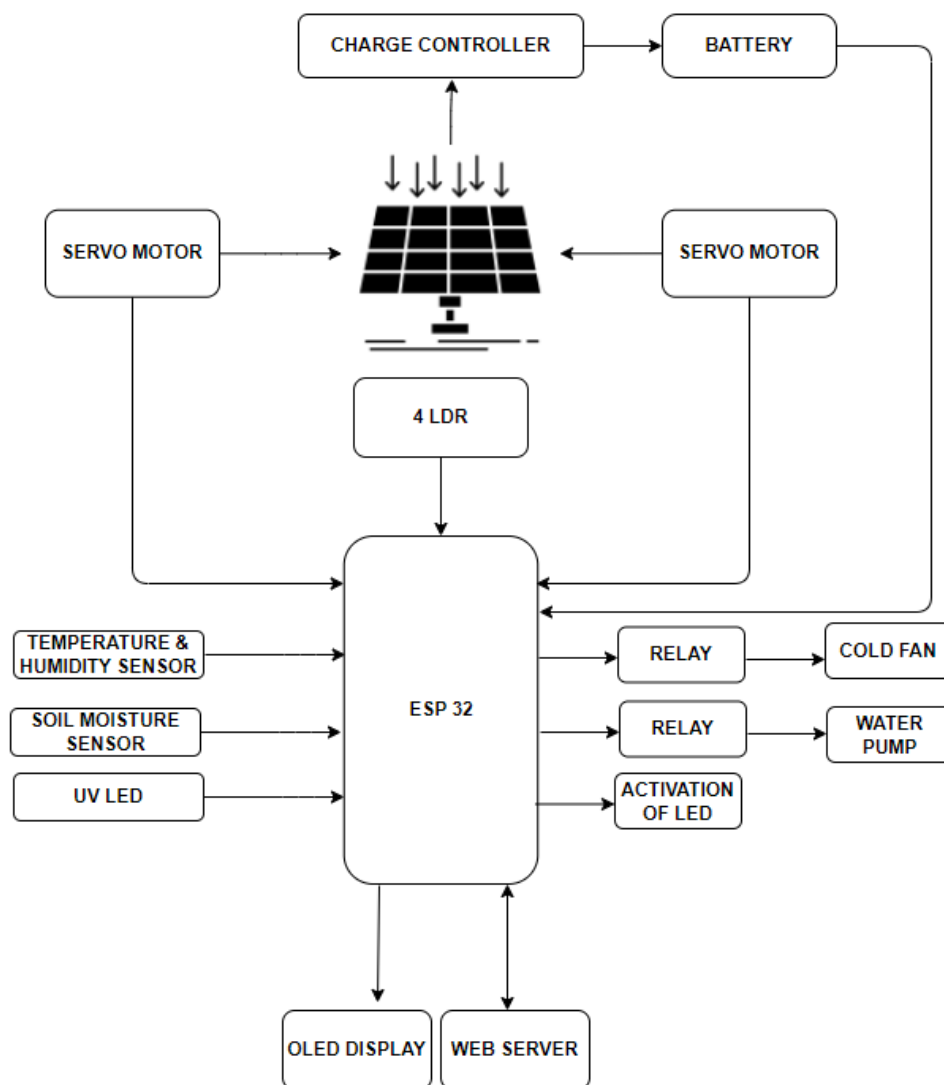


Figura 5. Schema bloc a sistemului

2.1 PLACA DE DEZVOLTARE ESP32

ESP32 este un microcontroller System on Chip (SoC) de la Espressif Systems, dezvoltorii celebrului SoC ESP8266. Este un successor al SoC ESP8266 și vine în variante single-core și dual-core ale microprocesorului Tensilica Xtensa Lx6 pe 32 de biți cu Wi-fi și Bluetooth integrat [6].

ESP32 are un procesor ceva mai puternic față de Raspberry Pi Pico, placa de dezvoltare cu care am început acest proiect, dar am renunțat datorită faptului că acest ESP32 are mult mai mulți pini programabili și o memorie mult mai mare față de Raspberry Pi Pico.

Am ales acest microcontroller pentru proiectarea solar tracker-ului și a serei inteligente, datorită caracteristicilor sale funcționale, prețul accesibil și dimensiunii avantajoase.

Caracteristici funcționale:

- Microprocesor LX6 dual-core, pe 32 de biți, cu frecvența de ceas de până la 240MHz
- 520 KB de SRAM, 448 KB de ROM și 16 KB de RTC SRAM
- Conectivitate Wi-fi 802.11b/g/n cu viteze de până la 150 Mb/s
- Classic Bluetooth v4.2
- 34 GPIO programabile
- Până la 18 canale de SAR ADC de 12 biți și 2 canale de DAC de 8 biți
- Interfață serial SPI Master/Slave
- Motor PWM și până la 16 canale de LED PWM
- Pornire securizată și criptare flash

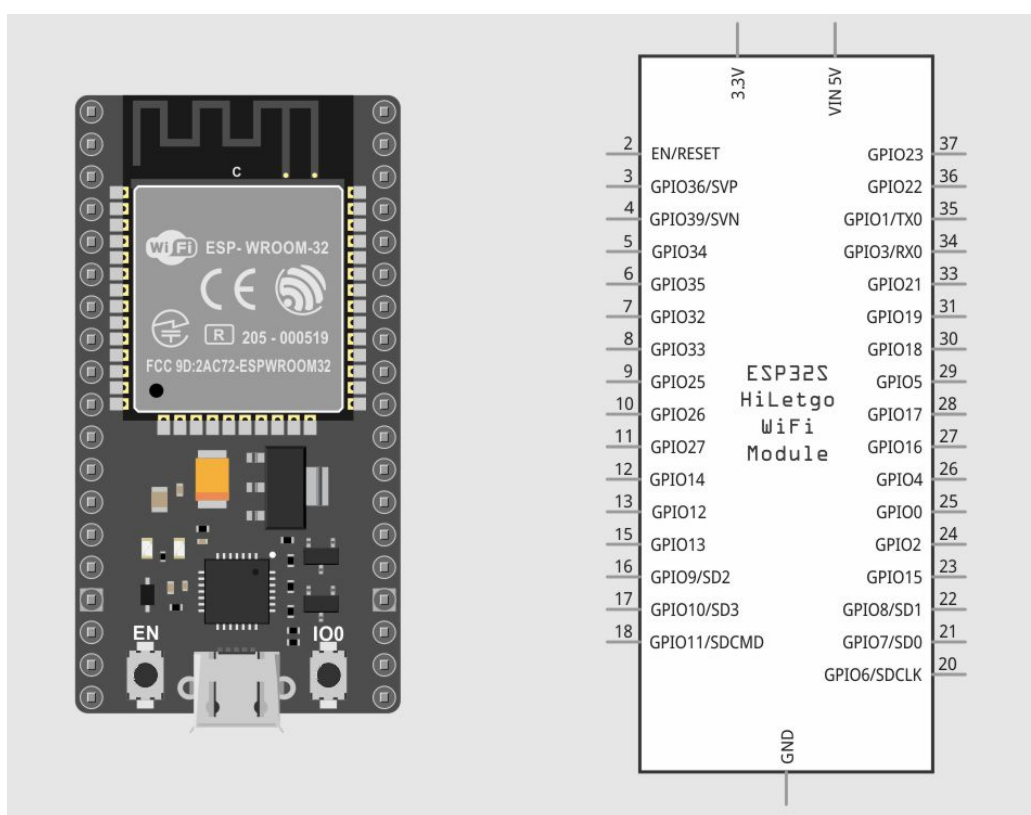


Figura 6. Pini ESP32 [6]

2.2 SENZORII UTILIZAȚI, SPECIFICAȚII, INTEGRARE ȘI TESTARE

Pentru realizarea acestui proiect, a fost necesară folosirea unor senzori care au rolul de a urmări mișcarea soarelui, pentru a capta puterea maximă de către celulele fotovoltaice (PV) ale panoului utilizat, în momentul iradierii uniforme sau parțiale ale soarelui. A mai fost necesară și folosirea unor senzori digitali care asigură conversia mărimii de interes în mărimi măsurabile și conversia acestora în semnal digital, care apoi este transmis prin protocolul SPI (Serial Peripheral Interface) către microcontroller pentru acționarea echipamentelor de reglaj a climatului dintr-o seră, în vederea obținerii unor condiții optime de dezvoltare a plantelor. Protocolul utilizat este un protocol de comunicație serial sincronă, utilizat frecvent în interconectarea dispozitivelor electronice. Datele achiziționate de la senzori vor fi afișate pe un display oled și preluate ulterior pe o pagină web.

2.2.1 Senzorul de lumina LDR

Senzorul de lumina LDR (Light Dependent Resistor) este un tip de rezistență care poate suferi modificări ale rezistenței atunci când se confruntă cu modificări ale recepției luminii. LDR este cunoscut sub multe denumiri: fotorezistor, fotoconductor, celulă fotoconductivă sau doar fotocelulă. Valoarea rezistenței LDR-ului depinde de cantitatea de lumină primită de senzorul de lumină LDR înșuși.

De obicei, senzorul este fabricat din sulfură de cadmiu, care este un material semiconductor a cărui rezistență variază în funcție de cantitatea de lumină (raze) care cade pe el. Senzorul și simbolul LDR poate fi văzut în figura de mai jos [7].

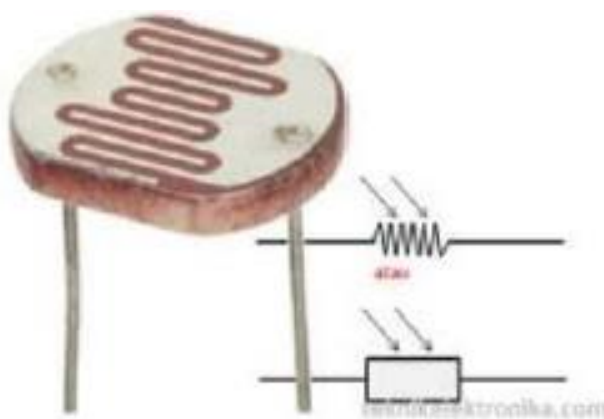


Figura 7. Senzorul și simbolurile LDR [7]

Caracteristici senzor:

- Tensiune maximă: 150 VDC
- Consum maxim de energie: 100mW

- Temperatură: -30°C - $+70^{\circ}\text{C}$
- Valoare spectrală: 540nm
- Rezistența la lumina (10 Lux) (k Ω): 10-20

Principiu de funcționare:

În lipsa luminii, rezistența LDR este de aproximativ 10M Ω , iar sub iluminare aceasta este 1k Ω sau mai puțin. Când lumina este puternică, LDR poate avea o rezistență mică de ordinul ... Ω [7].

2.2.2 Senzorul de umiditate și temperatură DHT11

Senzorul de umiditate și temperatură DHT11 (Figura 8) este un senzor tipic, simplu de instalat, care necesită puțină întreținere, este compact și are un preț rezonabil. Senzorul DHT11 constă dintr-un element capacitiv de detectare a umidității și un termistor pentru detectarea temperaturii [8]. Senzorul este de mici dimensiuni, ceea ce îl face ușor de integrat în diverse proiecte.



Figura 8. Senzorul de umiditate și temperatură DHT11 [8]

Caracteristici senzor:

- Tensiune de alimentare: 3.3V sau 5V
- Umiditate măsurată: 20% - 95%
- Acuratețe umiditate: $\pm 5\%$

- Temperatură măsurată: 0°C - 50°C
- Acuratețe pentru temperatura: $\pm 2^{\circ}\text{C}$

2.2.3 Senzorul higrometric capacitiv de umiditate a solului SEN0193

Senzorul de umiditate SEN0193 (Figura 9) poate măsura umiditatea din sol. Este un senzor bun pentru monitorizarea unei grădini urbane sau a nivelului de apă al unei plante. Acest senzor de umiditate a solului este capacitiv, iar atunci când solul este uscat, capacitatea electrozilor este mică, solul având o conductivitate electrică scăzută. Pe măsură ce conținutul de apă din sol crește, capacitatea electrozilor crește, solul devenind un mediu bun pentru transferul de încărcături electrice [9].

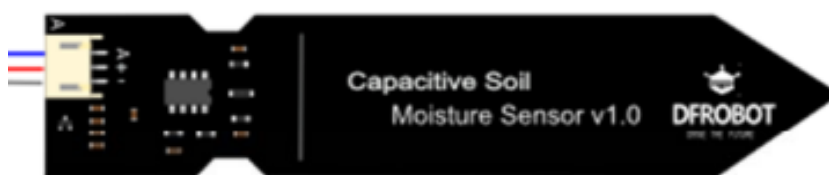


Figura 9. Senzorul higrometric capacitiv de umiditate SEN0193 [9]

Caracteristici senzor:

- Tensiune de alimentare : 3.3V sau 5V
- Tensiune de ieșire: 0-3V
- Curent: 5mA
- Dimensiuni: 98x23x4mm

2.3 COMPONENTE NECESARE

2.3.1 Servomotoare MG996R

MG996R (Figura 10) este un servomotor cu angrenaje metalice cu un cupliu maxim de blocare de 11Kg/cm. La fel ca și la alte servomotoare, motorul se rotește de la 0 la 180 de grade pe baza ciclului de lucru al unde PWM furnizate pinului său de semnal [10].



Figura 10. Servomotorul MG996R 10]

Caracteristicile servomotorului:

- Tensiune de funcționare: +5V
- Curent: 2,5A
- Cuplu maxim de oprire: 11Kg/cm
- Rotație: 0°-180°
- Greutatea motorului: 55g

2.3.2 Panou fotovoltaic cu celule policristaline

Energia solară fotovoltaică se obține prin conversia directă a radiației solare în energie electrică. Siliciul este cel mai comun material utilizat în fabricarea celulelor solare, datorită abundenței și durabilității ridicate. Cele mai utilizate celule de siliciu sunt monocristaline și policristaline, deși au fost dezvoltate și utilizate alte tehnologii [11].

Celulele de siliciu monocristalin sunt produse din bare cilindrice de siliciu monocristalin în cuptoare speciale, eficiența lor în transformarea luminii solare în energie electrică este de aproximativ 20%, având totuși costuri de producție ridicate. Celulele de siliciu policristalin sunt realizate din blocuri de siliciu obținute prin topirea unor porțiuni de siliciu pur în forme speciale, având o eficiență de transformare a razelor solare în energie electrică de aproximativ 18% [11].

Principalii factori care afectează performanța unui panou fotovoltaic sunt intensitatea radiației solare și temperatura celulei, așa cum se arată în Figurile 11 și 12 [12].

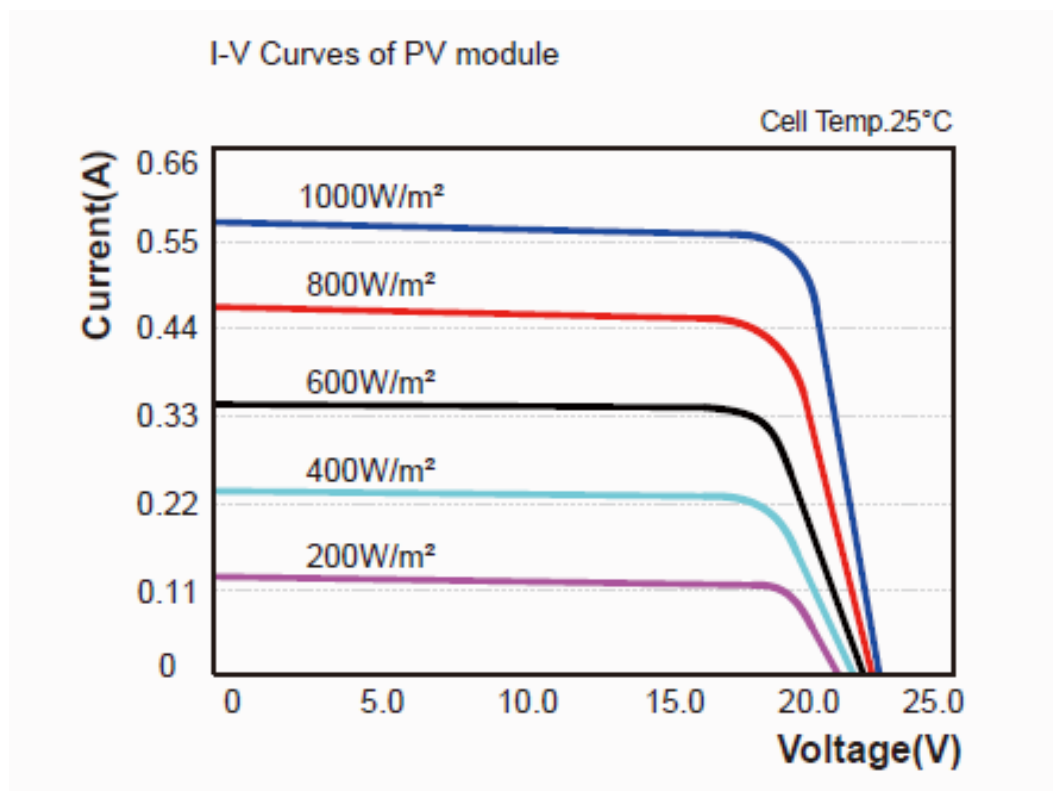


Figura 11. Curbele de iradiere ale panoului utilizat [12]

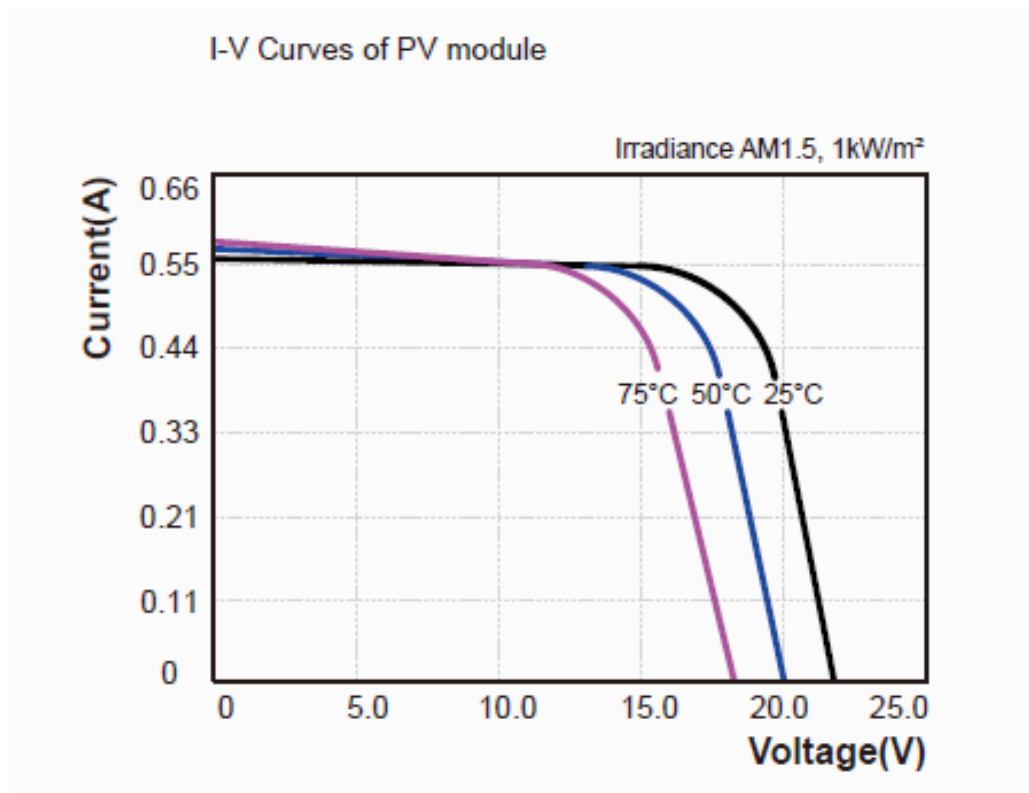


Figura 12. Caracteristicile panoului în funcție de temperatură [12]

În acest proiect am ales să folosesc un panou fotovoltaic de 10W de dimensiuni mici pentru tracker-ul care va alimenta tot sistemul. Panoul utilizat este format din 36 de celule policristaline.



Figura 13. Panou fotovoltaic 10W [12]

Caracteristicile modului fotovoltaic:

- Putere maximă: 10W
- Tensiune maximă: 18.2V
- Curent maxim: 0.55A
- Tensiunea de circuit deschis: 22.6V
- Curent de scurtcircuit: 0.59A
- Temperatură de funcționare: -40°C - 85°C
- Dimensiunea panoului: 354x251x17mm

2.3.3 Controller încărcare solară

Un controller solar este un dispozitiv electronic care este utilizat pentru a gestiona și controla încărcarea, respectiv descărcarea bateriilor solare.

În proiectul meu am ales să folosesc un modul care folosește una dintre cele două metode principale (metoda MPPT și metoda PWM) de încărcare a bateriei cu ajutorul unui panou solar și anume metoda PWM. Acest controler PWM acționează precum un întrerupător electronic între panoul solar și baterie: Comutatorul este ON atunci când bateria este descărcată, panoul transmite toată energia către baterie, comutatorul intră în modul PWM atunci când bateria este aproape încărcată și menține tensiunea bateriei la tensiunea de absorbție, comutatorul este OFF atunci când bateria este încărcată complet, în timp ce tensiunea bateriei scade.

Controllerele PWM cu încărcare solară sunt echipate cu protecții pentru a evita daunele la baterie sau la sistemul solar, pot include protecții împotriva supratensiunii, protecție la scurtcircuit și la descărcarea excesivă a bateriei.



Figura 14. Modul control încărcare solară PWM

Caracteristicile modului de încărcare solară:

- Tensiune baterie: 12-24V
- Curent încărcare maxim: 30A
- Curent descărcare: 10A
- Putere maximă: 240W
- Ecran LCD

2.3.4 Modul Display Oled SSD1306

Modulul cu display oled este o componentă electronica folosită frecvent pentru a afișa diferite informații în proiecte educaționale sau industriale, motivele fiind prețul accesibil, disponibilitatea și interfața prietenoasă cu utilizatorul.

Ecranul OLED SSD1306 este un driver CMOS cu un singur cip, cu un controler pentru sistemul de afișare grafică cu diodă emițătoare de lumina organică cu matrice în puncte. Acesta are un contrast ridicat, cu o rezoluție de 128x64 pixeli [13].

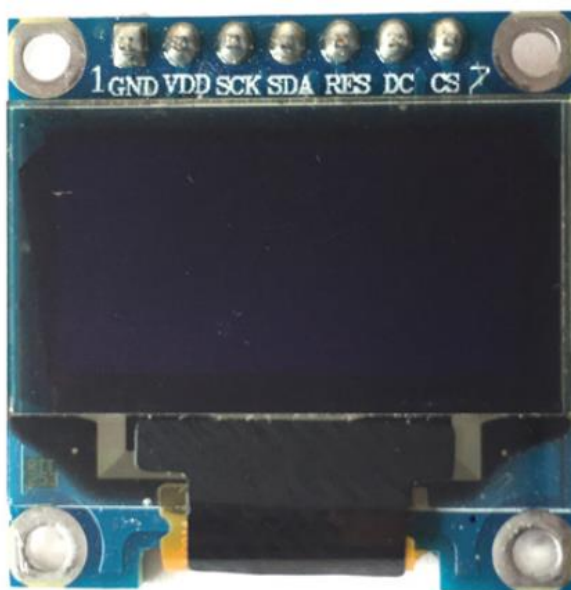


Figura 15. Display Oled SSD1306 [13]

Caracteristicile display-ului:

- Rezoluție: 128x64
- Tensiune alimentare: 3.3V-5V
- Temperatură funcționare -30°C-70°C
- Dimensiune: 29 x 27.5 x 4.3 mm
- Consum redus: 0.08W

2.3.5 Modul releu 2 canale comandat pe 5V

Un modul releu cu 2 canale (Figura 16) este o componentă esențială a sistemului hardware a acestui proiect. Acest modul oferă posibilitatea de a controla două circuite electrice separate folosind semnale digitale, fiecare canal al modului poate fi utilizat pentru a activa sau dezactiva un circuit în funcție de necesități.

Modulul releu cu 2 canale este compact și ușor de integrat într-un sistem mai mare, de asemenea, beneficiază de o construcție robustă și durabilă, ceea ce îl face potrivit pentru utilizare în diverse aplicații industriale, automatizări etc.



Figura 16. Modul releu 2 canale comandat pe 5V [14]

Caracteristicile modului:

- Tensiunea de alimentare necesară pentru funcționarea optocuploarelor este de 5V
- Fiecare releu primește comandă separat prin intermediul unui optocuplor
- Curent maxim: 10A
- Curent necesar pentru control: 5mA

2.3.6 Pompă apa/aer cu diafragmă, 6-12V, R385

Produsul din Figura 17 este o pompă de apa/aer cu diafragmă, cu durată lungă de viață, zgomot redus și fără interferențe electromagnetice [15]. Pompa R385 este proiectată să funcționeze la o tensiune de alimentare cuprinsă între 6V și 12V, în cazul proiectului pompa este alimentată la o baterie de 12V. R285 este o pompă de apă cu diafragmă, ceea ce înseamnă că utilizează diafragma pentru a crea presiune și a deplasa apa [15]. Tensiunea utilizată determină dacă debitul crește sau scade.



Figura 17. Pompă apa/aer cu diafragmă, 6-12V, R385 [15]

Caracteristici pompă:

- Tensiune de alimentare: 6-12V
- Tensiune recomandată: 9V, 1A

- Putere funcționare: 6W
- Debit aproximativ: 1.5-2L/Min
- Diferența de nivel maximă aspirare: 2m înălțime
- Diferența de nivel maximă ieșire: 3m înălțime

2.3.7 LED UV (Ultraviolete)

Am ales să folosesc LED-uri UV în proiectul meu, deoarece lumina UV ajută la creșterea plantelor, expunerea la anumite spectre de lumină UV poate influența forma și mărimea plantelor, lumina UV putând fi folosită în scopuri de control al creșterii plantelor.

Radiația ultravioletă este împărțită în trei domenii de lungimi de undă. Razele UV-A au domeniul lungimilor de undă cuprins între 315-400nm, având lungime de undă cea mai mare dintre domeniile UV, razele UV-B au domeniul lungimilor de undă cuprins între 280-315nm și razele UV-C domeniul cuprins între 180-280nm (având lungimile de undă cele mai mici), fiind și cele mai dăunătoare [17].

Lumina UV poate influența compoziția chimică și nutrițională a plantelor, putând crește conținutul anumitor substanțe, cum ar fi antioxidanții care pot avea efecte benefice asupra sănătății umane și potențialul plantelor ca alimente [16].



Figura 18. LED UV 395-400nm [18]

Caracteristici LED:

- Diametru LED: 3mm
- Tensiune de alimentare: 3.2V

- Putere 40mW
- Curent: 20mA

2.4 BATERIE LITIU-ION 12V

Stocarea energiei solare se poate face cu ajutorul unei baterii, ceea ce am ales și eu în acest proiect, și anume o baterie litiu-ion de 12V.

Bateria într-un sistem fotovoltaic are rolul de a stoca excesul de energie produs de panourile solare în timpul zilei pentru a putea fi utilizată ulterior în perioadele de consum crescut sau chiar atunci când radiația solară este scăzută. Bateriile pot oferi o anumită protecție împotriva fluctuațiilor de tensiune într-un sistem fotovoltaic, pot menține o anumită tensiune constantă și o stabilitate în sistem, ceea ce poate preveni deteriorarea echipamentelor sensibile la variațiile de tensiune. Acestea mai pot contribui și la prelungirea de viață a unui sistem fotovoltaic, prin reducerea solicitării asupra panourilor solare și inversarea fluxului de energie ceea ce poate reduce uzura și crește durabilitatea sistemului [19].

În general utilizarea bateriilor sistemelor fotovoltaice poate aduce multiple beneficii, cum ar fi autonomia energetică, economii de costuri, protecție împotriva întreruperilor de alimentare și optimizarea consumului de energie. Cu toate acestea, alegerea unei baterii corespunzătoare unui sistem fotovoltaic necesită o planificare atentă și o evaluare a nevoilor specifice ale sistemului fotovoltaic în cauză [19].

Pentru stocarea energiei am folosit o baterie de 12V realizată din 3 acumulatori legați în serie (3S). La această baterie am adăugat și un BMS pentru a monitoriza, gestiona și proteja bateria. Rolul principal al unui BMS este de a monitoriza constant parametrii critici ai bateriei, cum ar fi tensiunea, curentul, temperatura și nivelul de încărcare/descărcare al bateriei, ce permite funcționarea în limitele de siguranță și optimizează performanța bateriei. BMS-ul permite comunicarea și coordonarea între baterie și alte componente ale sistemului, în cazul acestui proiect fiind panoul fotovoltaic și controler-ul de încărcare solară.

În Figura 19 este prezentată conectarea BMS-ului la baterie, conform specificațiilor producătorului.

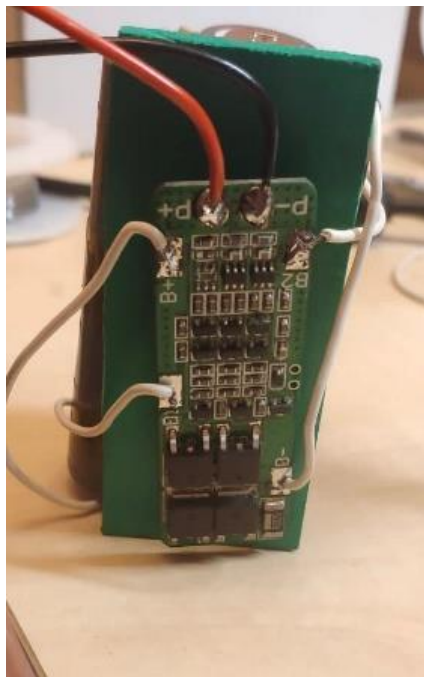


Figura 19. Conectarea BMS-ului la baterie

2.5 SISTEMUL FINAL

2.5.1 Schema electrică a sistemului

Figura 20 ilustrează schema electrică a sistemului în care sunt conectați toți senzorii analogici la porturile ADC ai microcontrollerului, iar în funcție de acestea să fie acționate servomotoarele pentru mișcarea panoului solar, cele două ventilatoare, pompa de apă și ledurile UV. Datele achiziționate din seră vor fi afișate pe ecranul oled și transmise către o pagină web prin intermediul modului Wi-fi pe care îl are placa de dezvoltare ESP32.

Pentru partea de tracker am folosit 4 senzori LDR. Fiecare senzor LDR are 2 pini, unul dintre ei va fi conectat la 3.3V, iar celălalt pin va fi legat în serie la o rezistență de 10k Ω . Acele rezistențe de la fiecare senzor se conectează la GND. Acești senzori de lumină controlează cele două servomotoare pentru a plasa panoul solar în soare, pentru a-i crește eficiența, prin intermediul unei măsurători diferențiale. Servo-motoarele au 3 pini, unul de alimentare (conectat la 5V), unul de împământare și unul pentru transmiterea semnalelor de control, pe care la unul dintre servomotoare (cel orizontal) l-am conectat la pinul GPIO12, iar pe celălalt servomotor (cel vertical) l-am conectat la pinul GPIO13.

Pentru măsurarea temperaturii și a umidității am folosit un senzor DHT11 pe care l-am conectat la pinul GPIO19 al microcontrollerului. Ceilalți 2 pini ai senzorului i-am conectat la 3.3V și la GND.

Senzorul de umiditate al solului are tot 3 pini, unul de alimentare, unul de împământare și unul pentru transmiterea semnalului analogic pe care l-am conectat la pinul GPIO32.

Contribuția cea mai valoroasă la funcționarea sistemului este adusă de modulul cu două relee controlate. Acest modul funcționează ca un întrerupător. În schema din Figura 20, se poate observa că fiecare relee are trei contacte. Pentru realizarea sistemului am utilizat primul relee pentru a controla pompa. Pinul 4 reprezintă contactul în mod normal închis, pinul 5 reprezintă contactul de mijloc, iar pinul 6 reprezintă contactul în mod normal deschis. Deoarece bobina este întotdeauna sub tensiune, contactul 4 (în mod normal închis) și contactul 6 (în mod normal deschis) sunt întotdeauna în contact. De ce se numește "normal"? Deoarece acestea reprezintă stările "normale" ale unui relee atunci când nu primește tensiune. În această stare, un contact este în mod normal deschis, iar celălalt este în mod normal închis. Cu toate acestea, când se aplică tensiune, stările releului se inversează. Celălalt relee din modul controlează ventilatoarele.

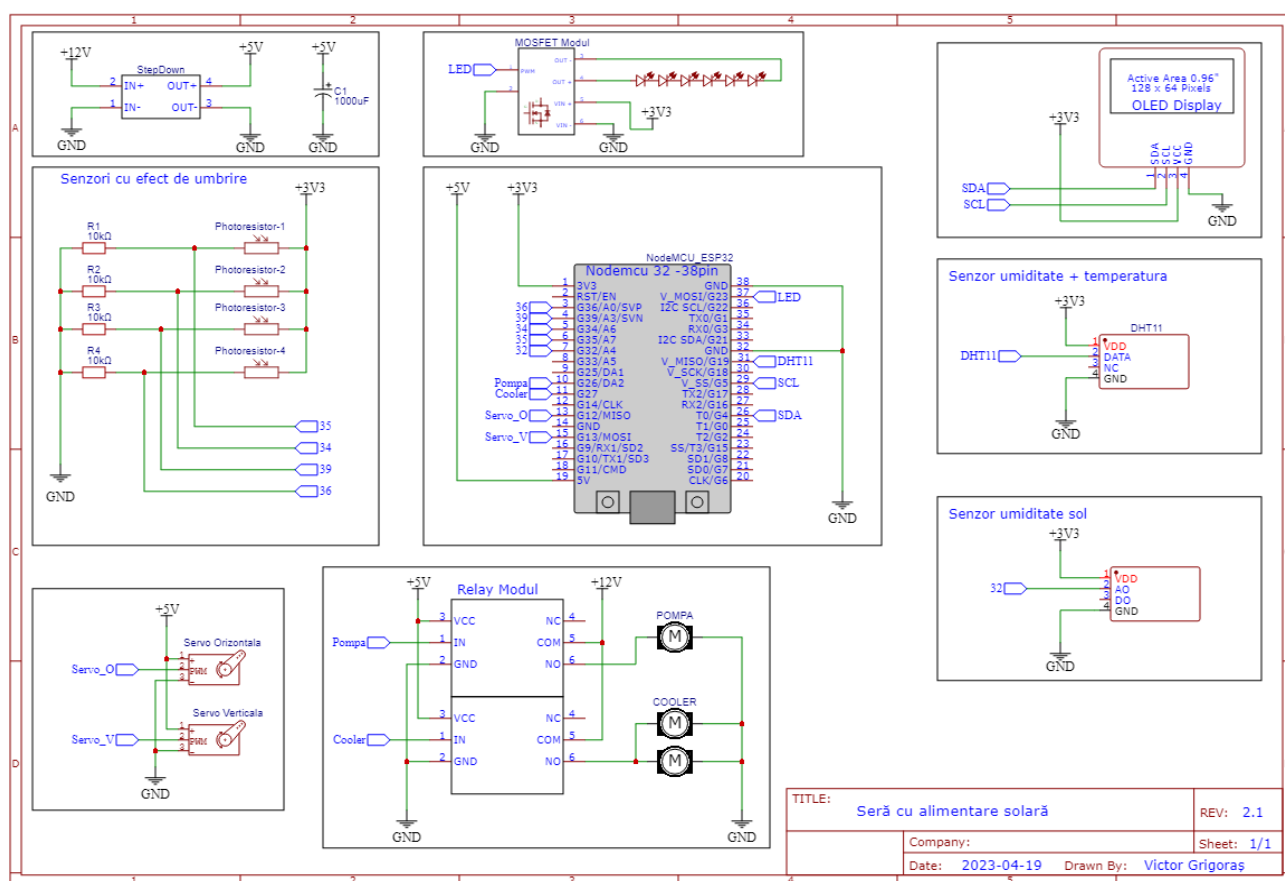


Figura 20. Schema electrică a sistemului

2.5.2 Testarea componentelor

Testarea componentelor reprezintă, în principiu, verificarea acestora pentru a afla dacă funcționează în parametrii normali.

Am testat fiecare componentă pe rând, deoarece unele se verifică doar prin conectarea la o sursă de tensiune, iar altele au nevoie și de câteva funcții scrise într-un limbaj de programare și care trebuie încărcate pe microcontroller.

Am început cu testarea LDR-ului. Pentru testarea LDR-ului am folosit un multimetru pentru a măsura rezistența sa în condiții de lumina scăzută, iar atunci când lumina scade rezistența LDR-ului a crescut. Apoi am continuat să măsoar rezistența LDR-ului în condiții de lumina puternică, atunci rezistența a scăzut.

Tabelul 1. Tensiunea de ieșire a celor patru senzori LDR

Nr. Crt.	LDR_ss(stânga sus)	LDR_sj(stânga jos)	LDR_ds(dreapta sus)	LDR_dj(dreapta jos)
1	0.69	0.91	1.11	0.94
2	0.92	1.19	1.14	1.17
3	1.76	2.12	1.89	1.57
4	2.14	2.53	2.71	3.21
5	2.37	2.38	2.75	2.39
6	2.76	2.76	2.86	2.65
7	2.43	2.43	2.78	2.12
8	2.5	2.56	2.94	2.74
9	3.53	3.57	3.23	3.47
10	2.77	2.74	2.71	2.60
11	2.59	2.56	3.13	2.31
12	2.63	2.76	3.17	2.57
13	3.73	3.78	3.21	3.81
14	2.74	2.70	3.26	2.72
15	2.66	2.63	2.93	2.80

Tabelul 1 prezintă tensiunea de ieșire a celor patru senzori LDR în timp real, tensiunea de ieșire nu este stabilă din cauza diverselor intensități scăzute în care au fost efectuate aceste citiri pentru o zi însorită. Se poate observa că tensiunile de ieșire înregistrate pentru fiecare senzor LDR variază în timp și nu sunt constante, acest lucru poate fi atribuit intensității diferite a luminii la care au fost expuși senzorii în timpul citirii lor, dar poate fi influențat și de alți factori, precum umbrirea sau reflexiile de la obiectele din jur.

Pentru a verifica servomotoarele dacă sunt funcționale am conectat firele de semnal, putere și masă ale servomotoarelor la placa de dezvoltare, respectând pinii corespunzători ai microcontrollerului. După testarea servomotoarelor am început realizarea sistemului de urmărire solară. Pe o placă de prototipare am conectat 4 LDR și 4 rezistențe de 10kΩ sub forma unui divizor de tensiune (Figura 21). Apoi am printat la o imprimantă 3D o cruce care să despartă fiecare LDR. LDR-urile simt cantitatea de lumină solară care cade asupra lor.

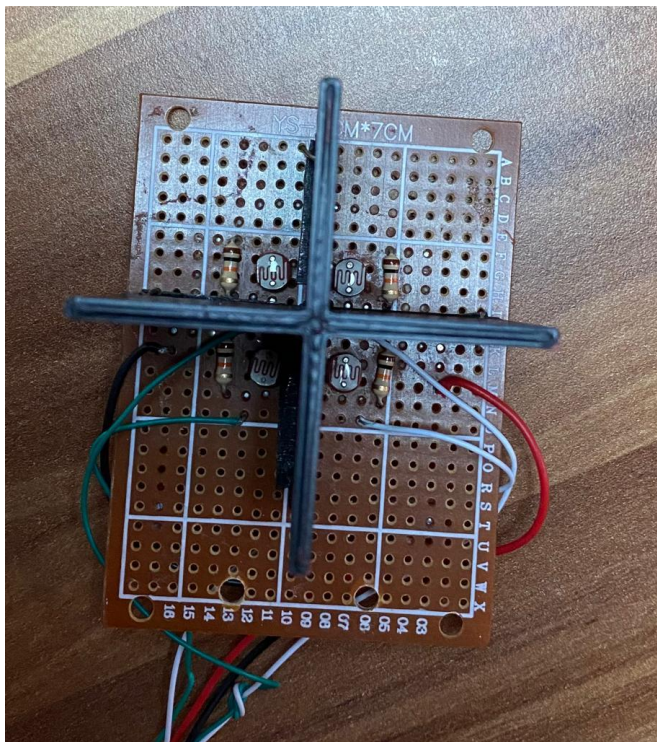


Figura 21. Placă de prototipare cu cei 4 LDR

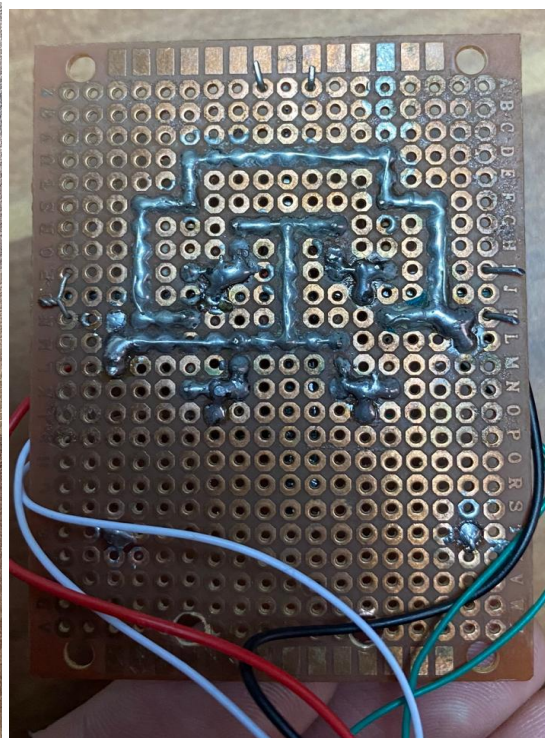


Figura 22. Placă de prototipare pe a 2-a față

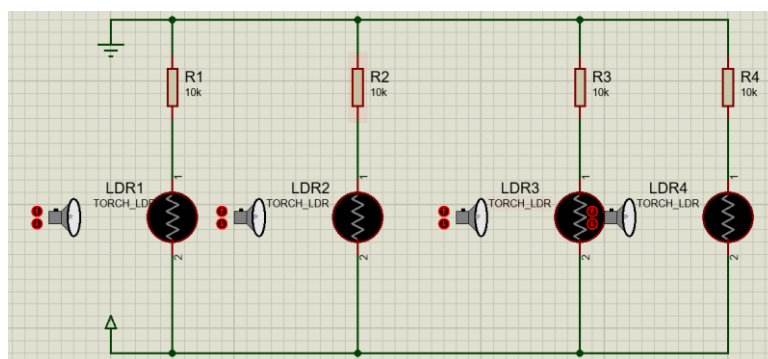


Figura 23. Schema circuitului celor 4 LDR

Pentru a verifica pompa de apă R385 dacă funcționează corect, am introdus-o într-un recipient cu apă, care a acoperit pompa complet, așa cum cer foile de catalog, am alimentat-o cu o tensiune de 12V și a pornit. Am constatat că pompa are suficientă putere pentru a muta apa dintr-un recipient în alt vas.. Pentru acest experiment, am conectat un furtun doar la orificiul de evacuare a apei.

LED-urile UV le-am legat în serie și cu ajutorul unui tranzistor le-am alimentat, LED-urile se pot aprinde sau stinge de pe pagina web folosind un slider.

Pentru testarea senzorilor utilizați și a display-ului oled, a fost nevoie de conectarea lor la întregul sistem și de scrierea programului care citește valorile măsurate de senzori și afișate pe display-ul oled. Modul în care s-au desfășurat aceste teste, precum și corectitudinea datelor afișate, vor fi elaborate într-un capitol ulterior din partea software.

După ce toate componentele au fost conectate și testate, am continuat cu montarea întregului sistem pe o placă din lemn, ulterior realizând o "mini seră" (Figura 24) din plexiglas, iar apoi a urmat partea de programare a sistemului și realizarea a unei pagini web pentru a vizualiza datele din interiorul serei.



Figura 24. Construcție seră

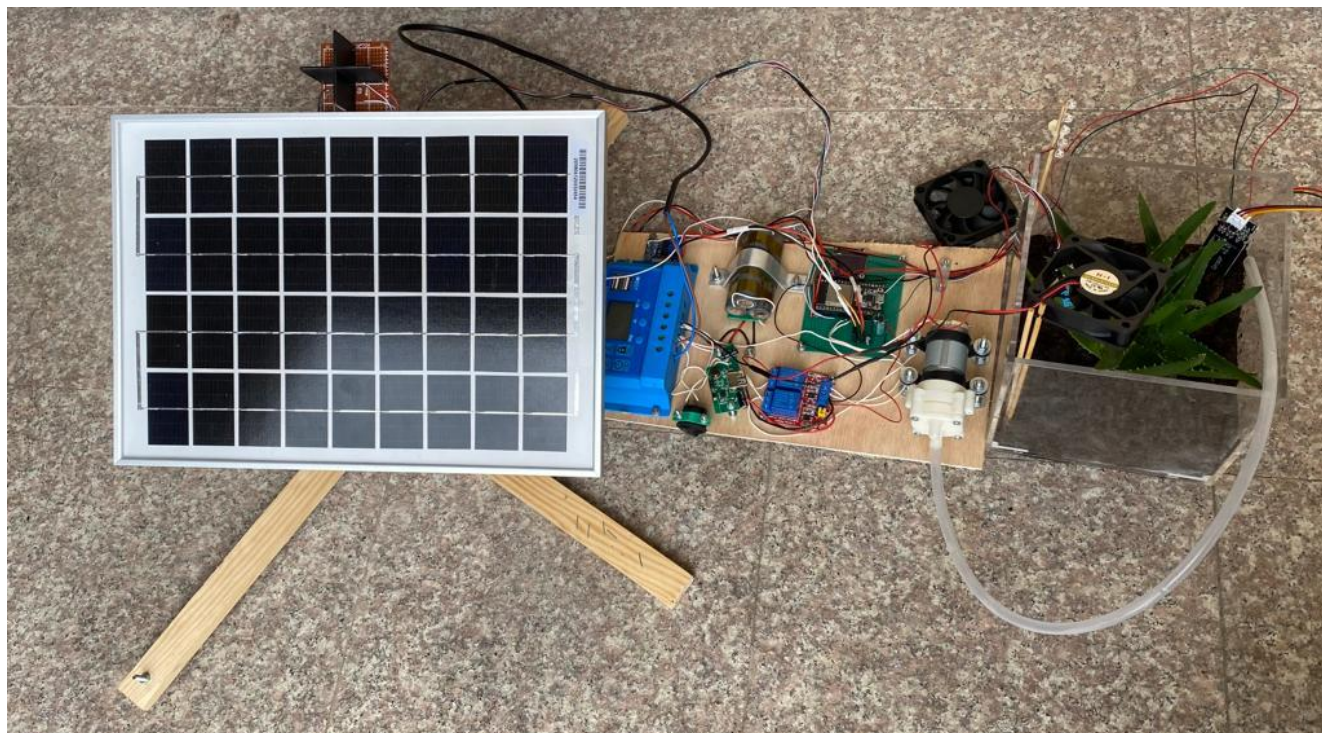


Figura 25. Sistemul final (1)

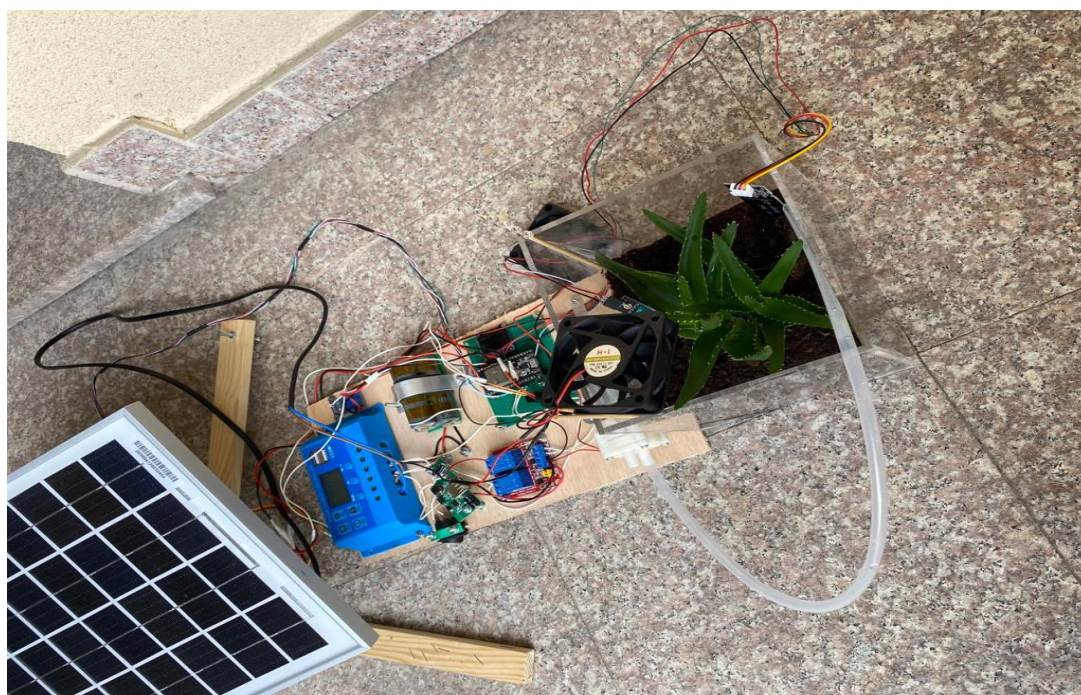


Figura 26. Sistemul final (2)

3. ARHITECTURĂ SOFTWARE

Programare senzori

Pagină Web pentru monitorizarea serei

Arhitectura software este reprezentată atât de programarea propriu-zisă a sistemului pentru a interacționa cu senzorii utilizați, cât și de pagina Web destinată acestuia. Pentru interacțiunea dintre componentele hardware am folosit software-ul Arduino IDE.

3.1 PROGRAMARE SENZORI

Programarea senzorilor presupune configurarea acestora în vederea obținerii unor valori măsurabile reale. Fiecare senzor analogic este configurat într-un mod specific lui.

3.1.1 Configurare senzor de lumina LDR

În programarea senzorilor de lumină LDR se efectuează citirea valorilor fiecărui senzor conectat la pini specificați ai microcontroller-ului apoi se calculează nivelul de lumină mediu prin adunarea valorilor citite de la toți cei patru senzori, iar acest nivel mediu de lumină este stocat în variabila *'sunLevel'*. Microcontroller-ul ESP32 are un convertor analog-digital cu o rezoluție de 12 biți, ceea ce înseamnă că poate măsura valori analogice în 4095 de nivele distincte de ADC, adică acesta poate să facă citirea tensiunii pe baza a 4095 de praguri. Nivelul mediu de lumină care este stocat în variabila *'sunLevel'* se redimensionează cu ajutorul funcției *'map(sunLevel, 0, 4095, 0, 100)'*, din intervalul citit de senzori (0-4095) în intervalul specificat (0-100), pentru a reprezenta un nivel de lumină mai ușor de utilizat. Apoi a urmat definirea variabilelor *'dtime'* și *'toleranta'* pentru a specifica diferența de timp și toleranța utilizate în algoritmul de urmărire a soarelui. Cu ajutorul formulelor de mai jos am calculat diferențele de lumină între perechile de senzori.

```
// citire senzori ldr:
int ss = analogRead(ldr_ss); // stânga sus
int ds = analogRead(ldr_ds); // dreapta sus
int sj = analogRead(ldr_sj); // stânga jos
int dj = analogRead(ldr_dj); // dreapta jos
sunLevel = int((ss+ds+sj+dj)/4); // media celor 4 LDR
sunLevel = map(sunLevel, 0, 4095, 0, 100); // funcția map
int dtime = 20; int toleranta = 200; // dtime=diferența de timp
//mediile valorilor senzorilor LDR
```

```

int valoareMedieSus      = (ss + ds) / 2; // valoare medie sus
int valoareMedieJos      = (sj + dj) / 2; // valoarea medie jos
int valoareMedieStanga   = (ss + sj) / 2; // valoarea medie stânga
int valoareMedieDreapta  = (ds + dj) / 2; // valoarea medie dreapta
int difVerticala = valoareMedieSus - valoareMedieJos; // diferența între sus și jos
int difOrizontala = valoareMedieStanga - valoareMedieDreapta; // diferența între stânga și dreapta

```

Codul 1. Citirea senzorilor LDR și mediile valorilor senzorilor

Pentru a ajusta servo-motoarele se verifică dacă diferențele de lumină în plan vertical și orizontal se află în toleranța specificată, dacă diferența nu se încadrează în toleranță, se ajustează poziția servo-motoarelor pentru a urmări sursa de lumină. Pentru ajustarea pe verticală, se compară mediile valorilor senzorilor superiori și inferiori. Dacă valoarea medie superioară este mai mare decât cea inferioară, se incrementează valoarea servo-motorului vertical pentru a muta servo-motorul în sus, iar dacă valoarea medie superioară este mai mică decât cea inferioară, valoarea se decrementează pentru a muta servo-motorul în jos. Aceste valori sunt limitate în intervalul *'limitaInferioaraServov'* și *'limitaSuperioaraServov'*, pentru a evita mișcările necontrolate ale servo-motorului.

Pentru ajustarea poziției pe orizontală, se compară mediile valorilor senzorilor din partea stângă, respectiv partea dreaptă, iar dacă valoarea medie a senzorilor din partea stângă este mai mare decât cea a celor din partea dreaptă, valoarea servo-motorului se decrementează pentru a se muta spre stânga. Dacă valoarea medie a senzorilor de pe partea stângă este mai mică de media senzorilor de pe partea dreapta, valoarea servo-motorului se incrementează pentru a se muta șa dreapta, iar dacă mediile valorilor senzorilor sunt egale, se va introduce o întârziere de 5 secunde pentru a aștepta o schimbare în lumină.

```

if (-1*toleranta > difVerticala || difVerticala > toleranta) { // daca nu se afla in toleranta se
face mutarea pe verticala
    if (valoareMedieSus > valoareMedieJos) {
        servov = ++servov;
        if (servov > limitaSuperioaraServov) {
            servov = limitaSuperioaraServov;
        }
    } else if (valoareMedieSus < valoareMedieJos) {
        servov = --servov;
        if (servov < limitaInferioaraServov) {
            servov = limitaInferioaraServov;
        }
    }
    vertical.write(servov);
}

if (-1*toleranta > difOrizontala || difOrizontala > toleranta) { // daca nu se afla in
toleranta
    if (valoareMedieStanga > valoareMedieDreapta) {
        servoh = --servoh;
        if (servoh < limitaInferioaraServoh) {
            servoh = limitaInferioaraServoh;
        }
    } else if (valoareMedieStanga < valoareMedieDreapta) {
        servoh = ++servoh;
        if (servoh > limitaSuperioaraServoh) {

```

```

    servoh = limitaSuperioaraServoh;
}
} else if (valoareMedieStanga = valoareMedieDreapta) {
    delay(5000);
}

horizontal.write(servoh);
}

```

Codul 2. Ajustarea servo-motoarelor în funcție de lumina solară

Senzorul de lumina LDR este de fapt o rezistență care se modifică în funcție de cantitatea de lumină care ajunge pe suprafața acesteia. Dacă intensitatea luminii este mare, atunci rezistența senzorului scade, deci LDR este un conductor bun al curentului electric. În cazul în care intensitatea luminoasă este mică, rezistența senzorului crește, deci LDR devine un conductor slab. Pentru a determina cantitatea de lumină exactă care ajunge pe LDR, am realizat o schema electrică (Figura 27) în care am legat în serie, cu senzorul de lumină, o rezistență de 10kΩ. Astfel punctul de legătură dintre aceștia va fi conectat la un pin ADC al microcontroller-ului, formându-se astfel un divizor de tensiune. Tensiunea care cade pe rezistența de 10kΩ va fi tensiunea aplicată pinului ADC, care este direct proporțională cu numărul de lușci.

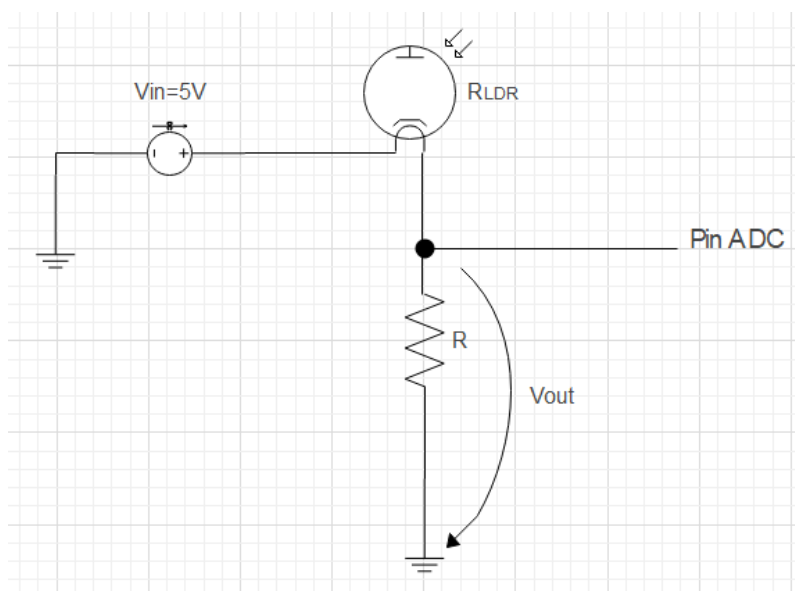


Figura 27. Schema electrică pentru RLDR

Formula divizorului de tensiune este:

$$V_{out} = V_{in} * \frac{R}{R + R_{LDR}} \quad (1)$$

Din aceasta rezultă ca R_{LDR} este:

$$R_{LDR} = \frac{(V_{in} - V_{out}) * R}{V_{out}} \quad (2)$$

Folosind un luxmetru am realizat mai multe măsurători pe baza schemei de mai sus (Figura 27). Am plasat luxmetrul în apropierea senzorului LDR şi am măsurat numărul de lucşi, căderea de tensiune pe senzorul LDR cu multimetru şi valoarea tensiunii aplicate pe pinul ADC. Valorile măsurate atât ziua cât şi noaptea, le-am trecut într-un tabel (Tabelul 2) şi am realizat două grafice (Figurile 28 şi 29) cu acestea pentru a reuşi să determin o formulă de calcul a numărului de lucşi, ce este prezentată în [20].

$$LUX = (1.25 * 10^7) * R_{LDR}^{-1.4059} \quad [20] \quad (3)$$

Tabelul 2. Valorile măsurate pentru a calcula numărul de lucşi

Tensiune	ADC	Lux
0.41	1023	2063
0.44	1011	2041
0.57	995	1018
0.72	972	522
0.95	894	391
0.99	888	271
1.2	870	257
1.41	827	241
1.5	787	158
1.99	636	72
2.02	664	64
2.45	528	31
2.55	538	37
2.6	525	26
2.75	490	25
2.76	483	21
3.01	393	19
3.74	253	15
4.23	113	14
4.86	126	0.44

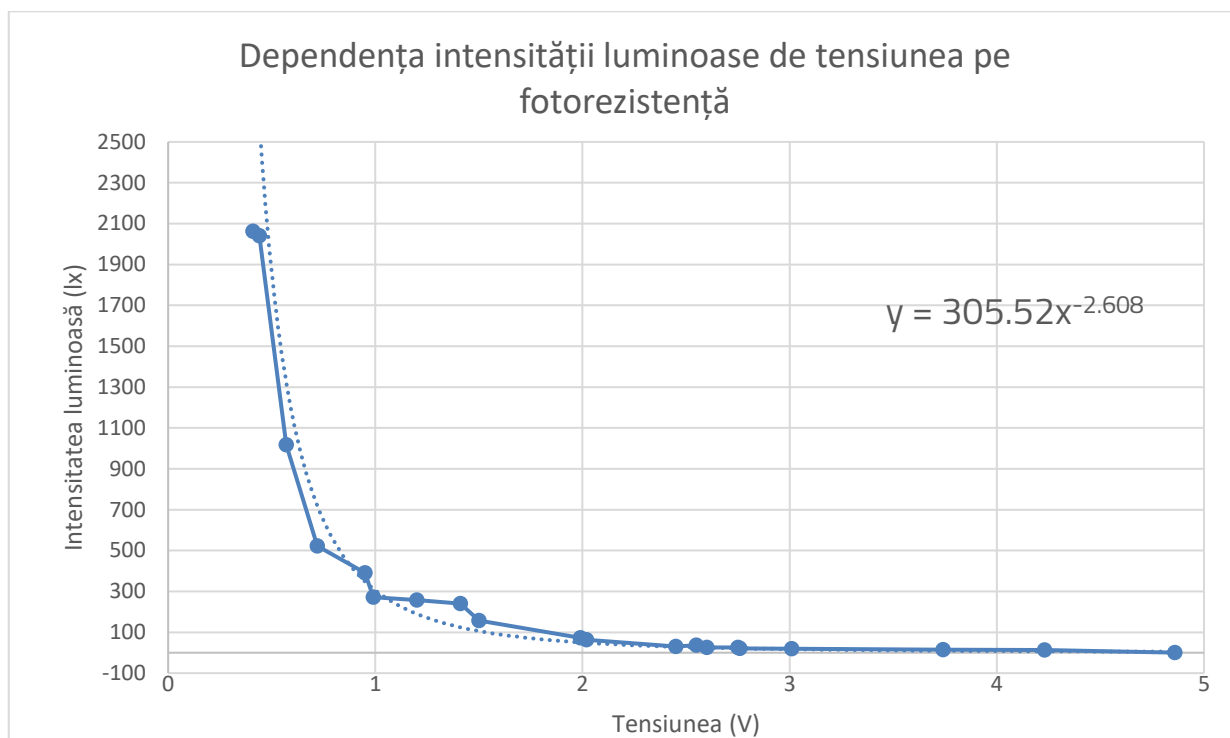


Figura 28. Dependența intensității luminoase de tensiunea aplicată pe LDR

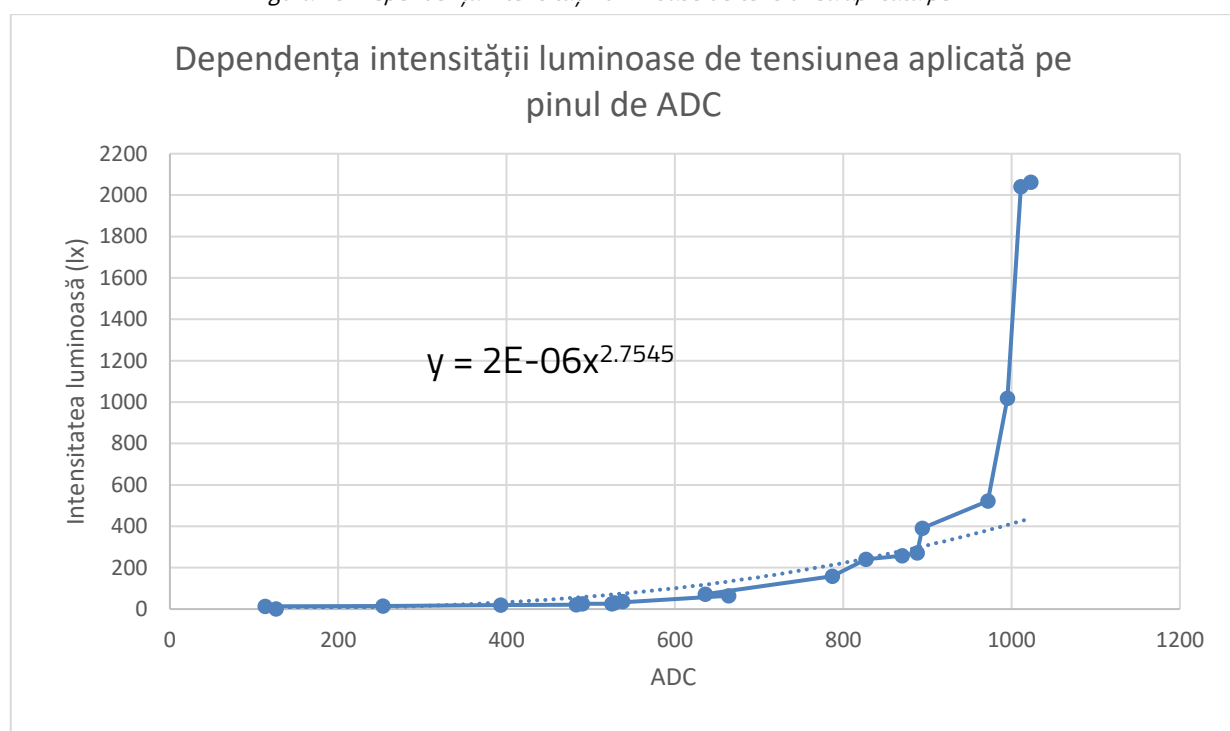


Figura 29. Dependența intensității luminoase de tensiunea aplicată pe pinul de ADC

În urma măsurărilor și a ecuațiilor determinate din grafic, am testat câteva dintre valorile din tabel pentru a verifica corectitudinea acestora. Din păcate valorile erau diferite. Din acest motiv, am încercat să găesc o formulă care să îmi verifice valorile din acest tabel. Spre exemplu, pentru valoarea 528 din tabel, am realizat următoarele calcule:

$$V_{in} - V_{out} = 5 - \frac{528 * 5}{1023} = 2.4193 \text{ V}$$

$$R_{LDR} = \frac{(V_{in} - V_{out}) * R}{V_{out}} = \frac{2.4193 * 10K\Omega}{2.5806} = 9374.95 \Omega$$

$$LUX = (1.25 * 10^7) * R_{LDR}^{-1.4059} = 12500000 * 9374.95^{-1.4059} = 32.56 \text{ lx}$$

În tabel este trecută valoarea 31, pe care am măsurat-o, iar din calcul am obținut aproximativ 32 lx.

3.1.2 Configurare senzor de umiditate a solului

Senzorul de umiditate a solului este tot un senzor analogic care trebuie conectat la un pin ADC al microcontroller-ului ESP32. Acesta măsoară valori în mai multe intervale: între 0-300 solul este uscat, 300-700 este umed, iar peste 700 până la 950 umiditatea devine prea mare.

Aceste valori analogice le-am determinat în procente, deoarece umiditatea solului reprezintă cantitatea de apă din sol, care în majoritatea cazurilor reprezintă un procentaj.

Am considerat valoarea de 950 ca fiind echivalentul lui 100%.

Astfel cu regula de 3 simplă am realizat toate determinările în procente ale intervalelor, și tot cu aceasta, am calculat și cantitatea de apă din sol, folosind o funcție de citire a canalului ADC '*analogRead(SensorPin)*'.

➤ Pentru 300:

$$x = \frac{300 * 100}{950} = 31.57\%$$

➤ Pentru 700:

$$x = \frac{700 * 100}{950} = 73.68\%$$

Pentru acest sistem, am considerat că trebuie aleasă o valoare medie pentru a acționa pompa de apă. Pragul de 300 ar fi prea mic, iar solul ar fi prea uscat, iar cel de 700 ar fi mare, iar solul prea umed. De aceea am calculat cu aceeași relație și pentru valoarea 500.

➤ Pentru 500:

$$x = \frac{500 * 100}{950} = 52.63\%$$

3.1.3 Programarea propriu zisă în vederea acționării sistemului

Senzorul DHT11, care poate citi și primi date despre temperatură și umiditatea aerului este amplasat în seră. Acesta este conectat la pinul GPIO19 al microcontroller-ului, pentru programarea sa am avut nevoie de o bibliotecă adițională pentru a putea citi datele.

După ce am calibrat toți senzorii, am conectat toate componentele, senzorii, ventilatoarele, LED-urile UV, pompa de apă pentru a vedea dacă funcționează corect. Am afișat toate datele citite pe display-ul oled.

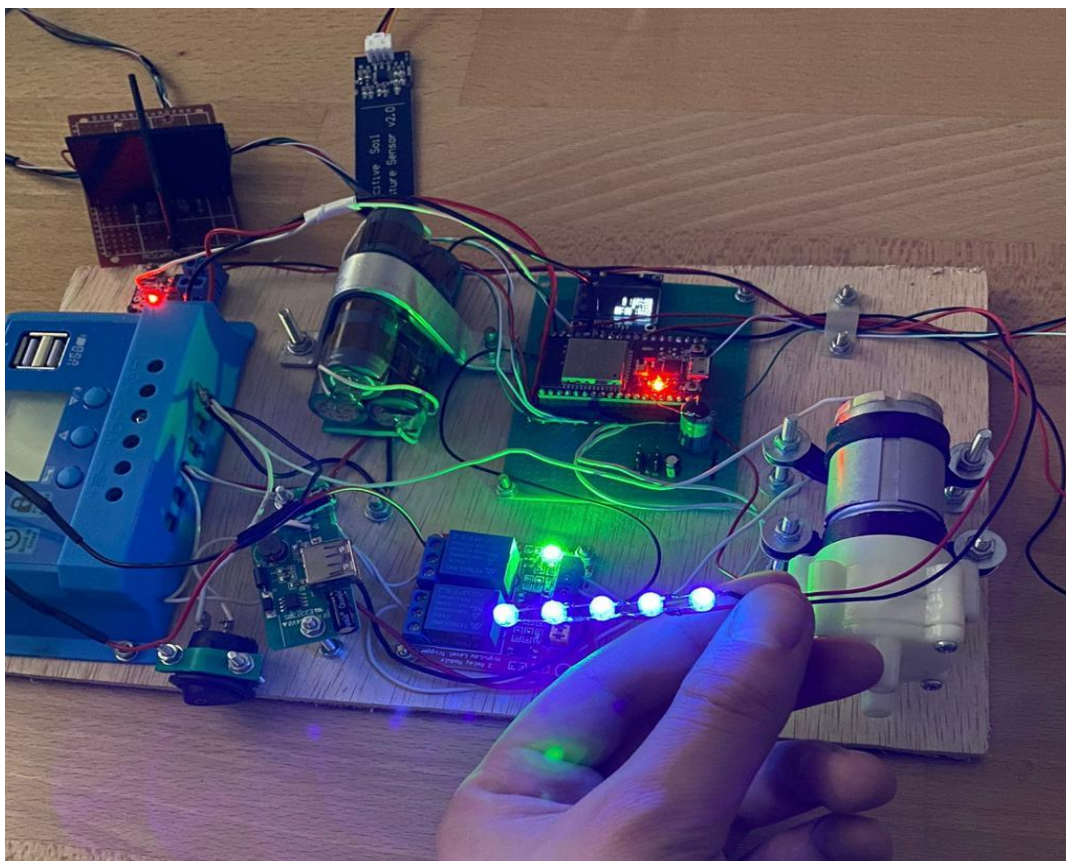


Figura 30. Testarea senzorilor

Utilizând informațiile furnizate de senzori, echipamentele sunt programate astfel încât să ia decizii și să acționeze în momentul în care sunt îndeplinite anumite condiții în cod, acest lucru asigurând funcționarea optimă a sistemului.

Datele achiziționate se trimit către pagina web folosind protocolul HTTP pentru comunicarea între server și client.

3.2 PAGINA WEB PENTRU MONITORIZAREA SEREI

3.2.1 Realizarea paginii web

Scopul paginii web este de a oferi o interfață utilizatorului pentru afișarea datelor și controlul unui sistem de monitorizare al unei sere alimentate cu energie regenerabilă.

Pagina web a fost proiectată pentru a permite vizualizarea unor informații relevante, cum ar fi temperatura și umiditatea aerului, nivelul de umiditate al solului, nivelul de lumină solară, starea ventilatoarelor și a pompei și alte date similare.

Pentru transmiterea datelor către pagina web, am folosit protocolul HTTP care este un protocol de comunicare utilizat pentru transmiterea de informații pe internet, acesta definește modul în care clientul și server-ul comunică și schimbă date între ei [21].



Figura 31. Pagina Web a sistemului

În Figura 31 de mai sus se poate observa interfața paginii web, a serei cu alimentare solară, această interfață permite monitorizarea anumitor parametri, cum ar fi: temperatura și umiditatea aerului, umiditatea solului, unghiul servomotoarelor folosite la tracker, valoarea soarelui în procente (media celor 4 LDR, ESP32 are 4095 de nivele de ADC apoi am folosit funcția MAP pentru a scoate o valoare în procente), lumina UV care se poate acționa din slider, ventilația și pompa care pornesc în funcție de temperatura aerului, respectiv umiditatea solului. Aceste date pot fi actualizate și afișate în timp real pe pagina web, se pot introduce valorile minime și maxime pentru umiditatea solului și temperatură, aceste valori pot fi ajustate prin intermediul câmpurilor de introducere. Trimiterea datelor către server se realizează prin apăsarea butonului "Confirm". Pagina web realizată are o interfață grafică cu un aspect plăcut prin utilizarea de pictograme (iconițe) pentru a ilustra informațiile prezentate.

În codul de mai jos este reprezentat un script JavaScript care efectuează acțiuni pe pagina web și comunică cu server-ul pentru a trimite și primi date. Scriptul definește o funcție numită *'sendData()'*, această funcție este apelată atunci când se dă submit unui formular sau se face o anumită acțiune pe pagină. În interiorul acestei funcții se obțin valorile pentru câmpurile (*'waterMin'*, *'waterMax'*, *'coolerMin'*, *'coolerMax'*, *'slider'*) din pagina HTML și sunt stocate în variabile corespunzătoare. Apoi se creează un obiect *'XMLHttpRequest'* pentru a iniția o cerere HTTP către server, se construiește un șir de parametri care conține valorile de mai sus și sunt codificate într-un format adecvat pentru a fi transmise în cadrul cererii GET către un URL. În funcția de tratare a evenimentelor, se verifică starea cererii și codul de stare HTTP, dacă starea este 4 (cererea a fost finalizată) și codul de stare este 200 (cererea a fost realizată cu succes), atunci se afișează un mesaj de alertă cu răspunsul primit de la server și se actualizează conținutul elementului HTML cu id-ul *'uvLight'* în funcție de valoarea selectată pentru controlul glisant (slider).

```
function sendData() {
    var wmin = document.getElementById('waterMin').value;
    var wmax = document.getElementById('waterMax').value;
    var cmin = document.getElementById('coolerMin').value;
    var cmax = document.getElementById('coolerMax').value;
    var statusUv = document.getElementById('slider').value;
    var http = new XMLHttpRequest();
    var params = 'wmin='+wmin+'&wmax='+wmax+'&cmin='+cmin+'&cmax='+cmax+'&statusUv='+statusUv;
    var url = 'set?' + params;
    http.open('GET', url, true);
    //Trimite informațiile de header corespunzătoare împreună cu cererea
    http.setRequestHeader('Content-type', 'application/x-www-form-urlencoded');
    http.onreadystatechange = function() {///Apelează o funcție atunci când starea se schimbă
        if(http.readyState == 4 && http.status == 200) {
```

```

alert(http.responseText);
var chk = document.getElementById('slider').value;
document.getElementById('uvLight').innerHTML = (chk == 0 ? 'Nu' : 'Da');
}
}

```

Codul 3. Scriptul JavaScript folosit pentru pagina web.

În momentul pornirii sistemului pe ecranul oled vor apărea informații legate de rețeaua WiFi la care se dorește conectarea. Modulul WiFi a fost setat într-un mod staționar, sistemul încercând să se conecteze la rețeaua WiFi specificată. După conectarea cu succes la rețeaua respectivă, pe ecranul OLED se va afișa adresa ip pentru a accesa pagina, toate dispozitivele care sunt conectate la aceeași rețea vor putea accesa această adresă. În configurarea serverului web se stabilesc rutele și funcțiile de tratare a cererilor HTTP pentru serverul web. Se vor configura și evenimentele SSE (Server-Sent Events), care permit comunicarea bidirecțională între server și client, se trimit mesaje către client atunci când acesta se conectează și sursa evenimentelor.

Dacă există un client web conectat, se vor actualiza informațiile pe ecranul OLED cu valorile curente de temperatură și umiditate a aerului, umiditatea solului etc. Se va verifica dacă umiditatea solului se încadrează în intervalul specificat (*waterMin* și *waterMax*) și în funcție de rezultat se va activa sau dezactiva pompa de apă, același lucru se va întâmpla și pentru temperatura aerului, se va verifica dacă temperatura se încadrează în intervalul specificat (*coolMin* și *coolMax*) și în funcție de rezultat se activează sau dezactivează sistemul de răcire.

4. SIMULARE SIMULINK

Un panou fotovoltaic este format din mai multe celule fotovoltaice în conexiuni paralele și în serie, conexiunea paralelă este responsabilă pentru creșterea curentului în matrice, în timp ce conexiunile în serie sunt responsabile pentru creșterea tensiunii modulului. Energia obținută dintr-un sistem fotovoltaic depinde de tensiunea produsă în modulul fotovoltaic, de temperatura celulei și de iradierea solară.

În această simulare este implementată programarea sistemului fotovoltaic, tehnica MPPT de perturbare și observare, convertorul buck și sistemul de încărcare a bateriei.

În Figura 32 de mai jos este reprezentată schema bloc a sistemului realizat în Simulink.

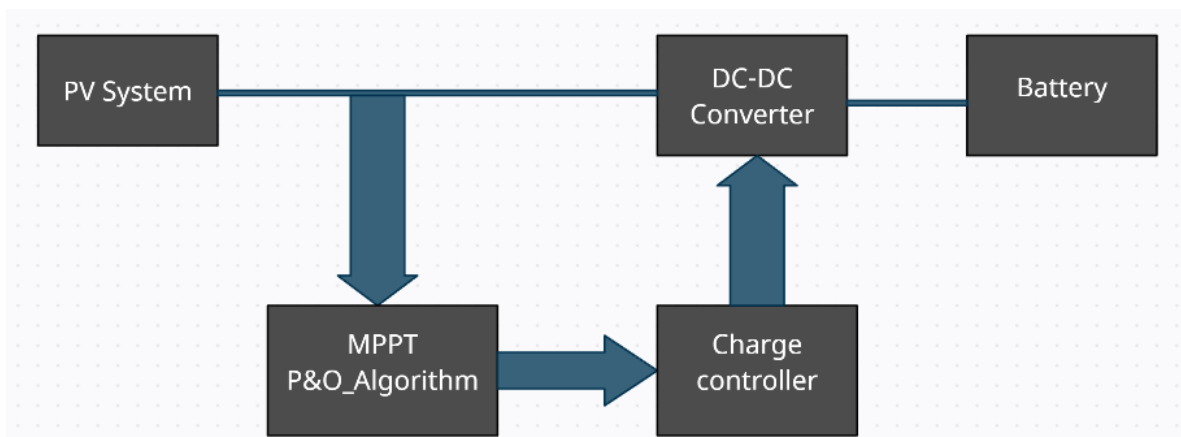


Figura 32. Schema bloc a sistemului de încărcare Simulink

4.1 MATLAB SIMULINK

Matlab Simulink este un mediu de simulare și proiectare bazat pe modele pentru sisteme dinamice și integrate. Simulink, este dezvoltat de MathWorks, este un instrument de limbaj de programare grafică cu flux de date pentru modelarea, simularea și analiza sistemelor dinamice. Acesta permite să se adauge algoritmi Matlab în modele, precum și să se exporte rezultatele simulării în Matlab.

4.2 MODUL FOTOVOLTAIC

Blocul PV Array implementează o serie de module fotovoltaice (PV). Matricea este construită din șiruri conectate în paralele, fiecare șir constând din module conectate în serie [22]. Blocul PV Array este un model cu cinci parametri care utilizează o sursă de curent (IL), o diodă, rezistență serie (R_s) și o rezistență de shunt (R_{sh}) pentru a reprezenta caracteristicile IV dependente de iradiere și temperatură ale modulelor [22].

Circuitul echivalent al unui panou solar ideal este prezentat în Figura 33.

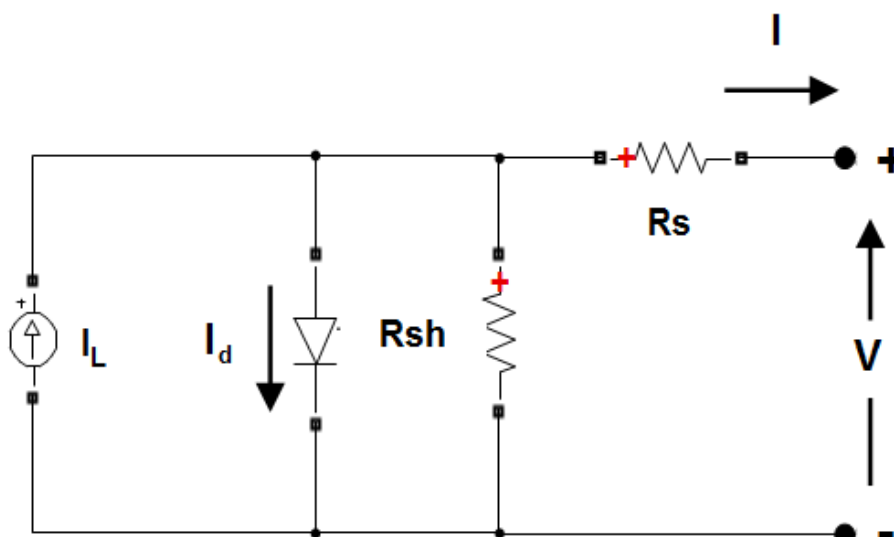


Figura 33. Circuitul echivalent al unui sistem fotovoltaic [22]

The screenshot shows the 'Block Parameters: PV Array' dialog box. It contains the following sections and parameters:

- Parameters:**
 - Advanced tab selected.
 - Array data:
 - Parallel strings: 1
 - Series-connected modules per string: 1
 - Module data:
 - Module: User-defined
 - Maximum Power (W): 275.1192
 - Cells per module (Ncell): 60
 - Open circuit voltage Voc (V): 38.58
 - Short-circuit current Isc (A): 8.99
 - Voltage at maximum power point Vmp (V): 32.52
 - Current at maximum power point Imp (A): 8.46
 - Temperature coefficient of Voc (%/deg.C): -0.36099
 - Temperature coefficient of Isc (%/deg.C): 0.102
- Display I-V and P-V characteristics of ...**
 - array @ 25 deg.C & specified irradiances
 - Irradiances (W/m2): [1000 500 100]
 - Plot button
- Model parameters**
 - Light-generated current I_L (A): 9.0739
 - Diode saturation current I_0 (A): 3.3652e-10
 - Diode ideality factor: 1.0433
 - Shunt resistance R_{sh} (ohms): 150.2986
 - Series resistance R_s (ohms): 0.13915

Buttons at the bottom: OK, Cancel, Help, Apply.

Figura 34. Parametri panoului fotovoltaic Simulink

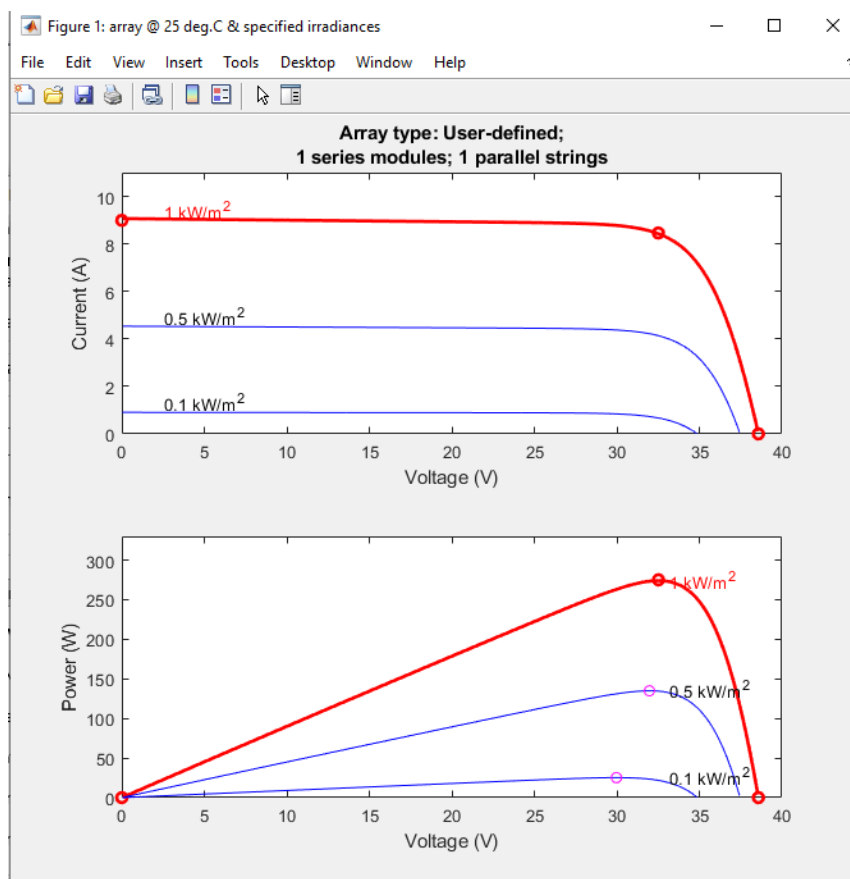


Figura 35. Caracteristica I-V și P-V pentru temperatură constantă

4.3 CONVERTORUL BUCK

Un convertor buck este un dispozitiv electronic care convertește o tensiune de intrare mai mare într-o tensiune de ieșire mai mică. Acesta este o formă de convertor DC-DC, ceea ce înseamnă că convertește o tensiune de curent continuu (DC), într-o altă tensiune de curent continuu [23]. Acest circuit funcționează în două stări: atunci când comutatorul este în conducție, dioda este izolată, iar atunci când dioda este în conducție comutatorul este deschis. Raportul dintre timpul în care comutatorul este în conducție și perioada de comutare a circuitului se numește raport de sarcină și este notat cu simbolul D . Pentru circuitul convertorului buck, această valoare reprezintă, de asemenea, coeficientul de transformare al circuitului. În figura 30 sunt reprezentate componentele utilizate pentru realizarea convertorului buck.

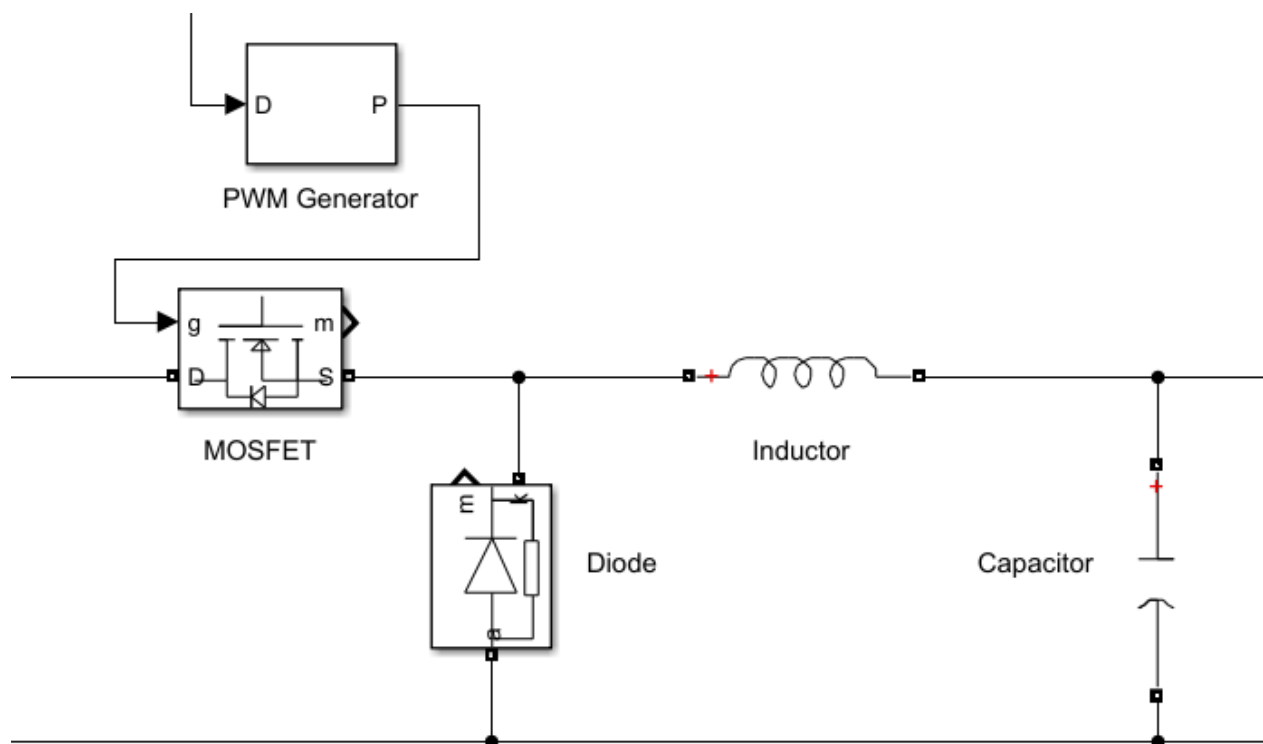


Figura 36. Convertorul Buck Simulink

4.4 ALGORITM P&O MPPT (PERTURBARE ȘI OBERVARE)

În Figura 37 este prezentată schema algoritmului P&O. Algoritmul P&O este o tehnică populară de urmărire a punctului de putere maximă (MPPT) utilizată în sistemele fotovoltaice. Este un algoritm simplu și ușor de implementat, care poate fi utilizat în sistemele fotovoltaice autonome atât autonome, cât și cele conectate la rețea [23]

Algoritmul P&O funcționează prin perturbare continuă a punctului de funcționare al sistemului fotovoltaic și observarea schimbării corespunzătoare a puterii de ieșire. Algoritmul compară puterea de ieșire a punctului de funcționare curent cu cea a punctului de funcționare anterior, dacă puterea de ieșire a crescut, algoritmul continuă să perturbe în aceeași direcție până când puterea de ieșire începe să scadă. În acest moment, algoritmul schimbă direcția și începe să perturbe în direcția opusă, acest proces este repetat până când se ajunge la punctul de putere maximă.

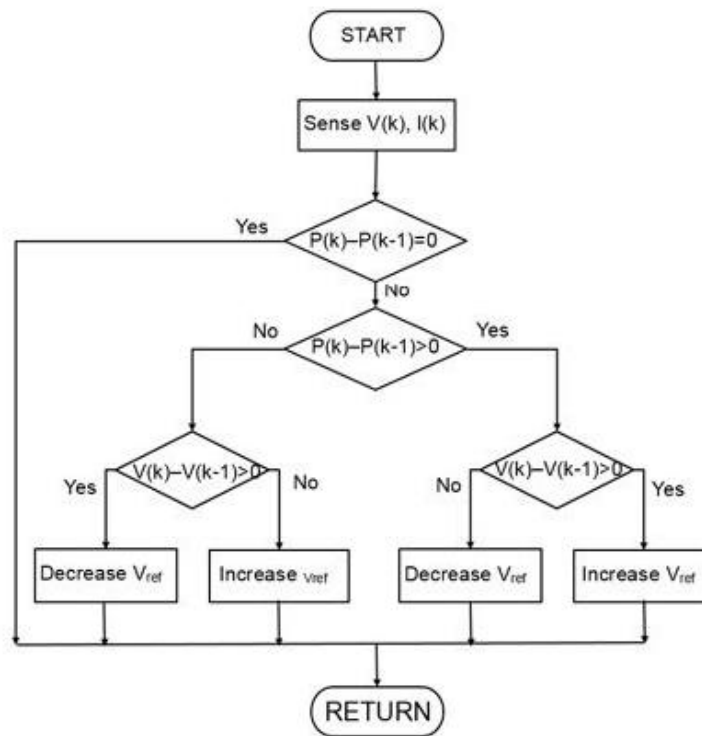


Figura 37. Algoritmul de bază P&O MPPT [23]

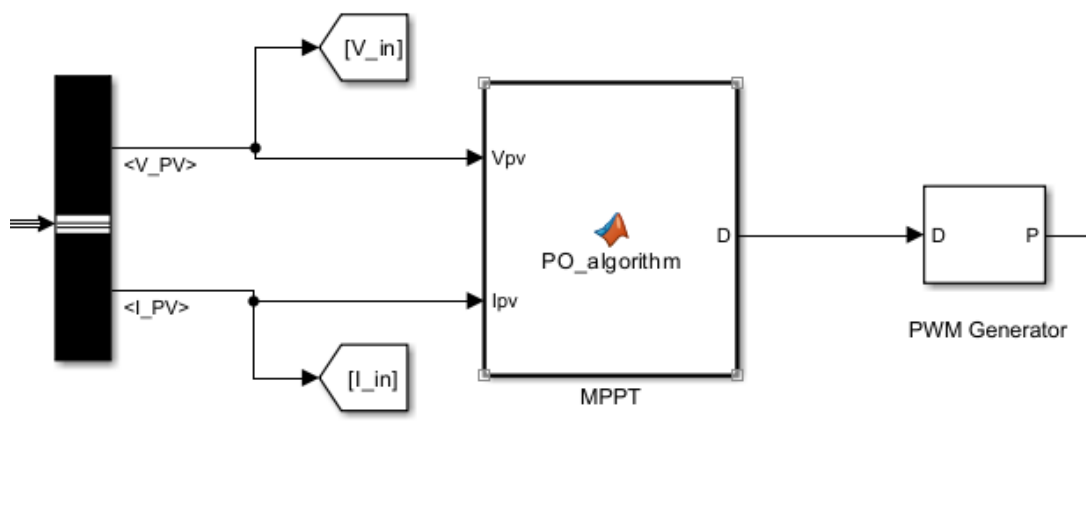
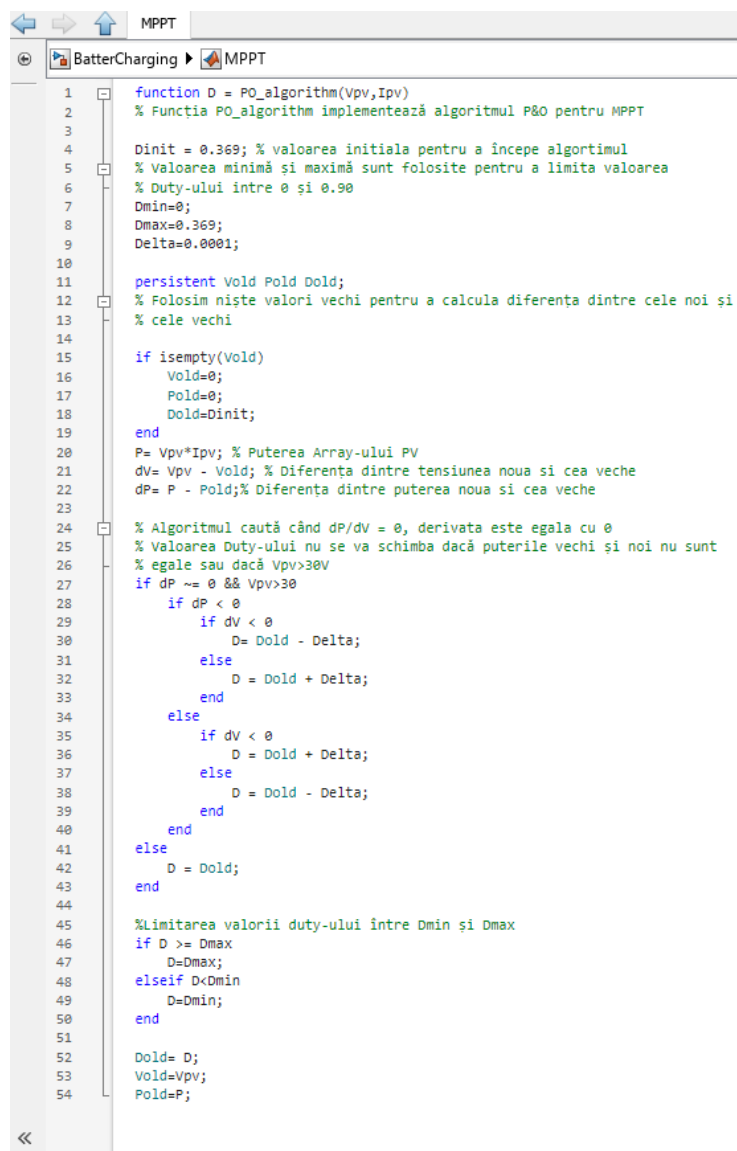


Figura 38. P&O MPPT algorithm Simulink



```

1 function D = PO_algorithm(Vpv,Ipv)
2 % Funcția PO_algorithm implementează algoritmul P&O pentru MPPT
3
4 Dinit = 0.369; % valoarea initiala pentru a începe algoritmul
5 % Valoarea minimă și maximă sunt folosite pentru a limita valoarea
6 % Duty-ului între 0 și 0.90
7 Dmin=0;
8 Dmax=0.369;
9 Delta=0.0001;
10
11 persistent Vold Pold Dold;
12 % Folosim niște valori vechi pentru a calcula diferența dintre cele noi și
13 % cele vechi
14
15 if isempty(Vold)
16     Vold=0;
17     Pold=0;
18     Dold=Dinit;
19 end
20 P= Vpv*Ipv; % Puterea Array-ului PV
21 dV= Vpv - Vold; % Diferența dintre tensiunea noua și cea veche
22 dP= P - Pold; % Diferența dintre puterea noua și cea veche
23
24 % Algoritmul caută când dP/dV = 0, derivata este egala cu 0
25 % Valoarea Duty-ului nu se va schimba dacă puterile vechi și noi nu sunt
26 % egale sau dacă Vpv>30V
27 if dP ~= 0 && Vpv>30
28     if dP < 0
29         if dV < 0
30             D = Dold - Delta;
31         else
32             D = Dold + Delta;
33         end
34     else
35         if dV < 0
36             D = Dold + Delta;
37         else
38             D = Dold - Delta;
39         end
40     end
41 else
42     D = Dold;
43 end
44
45 %Limitarea valorii duty-ului între Dmin și Dmax
46 if D >= Dmax
47     D=Dmax;
48 elseif D<Dmin
49     D=Dmin;
50 end
51
52 Dold= D;
53 Vold=Vpv;
54 Pold=P;

```

Figura 39. Programarea MPPT-ului în Matlab

4.5 SEMNALUL DE RADIAȚIE

În Simulink se poate utiliza un generator de semnal pentru a crea diferite cazuri de testare pentru a observa comportamentul unui model. În cadrul sistemului meu, am aplicat un semnal variabil, cum ar fi o radiație solară care variază în timpul zilei, folosind un generator de semnal. Am folosit acest semnal pentru a urmări punctul de putere maximă al panoului implementând algoritmul P&O pe care l-am prezentat mai sus. Scopul algoritmului este de a calcula puterea în funcție de valorile curentului și tensiunii furnizate de la intrarea panoului și de a compara valorile anterioare ale puterii, pentru a obține eficiență maximă.

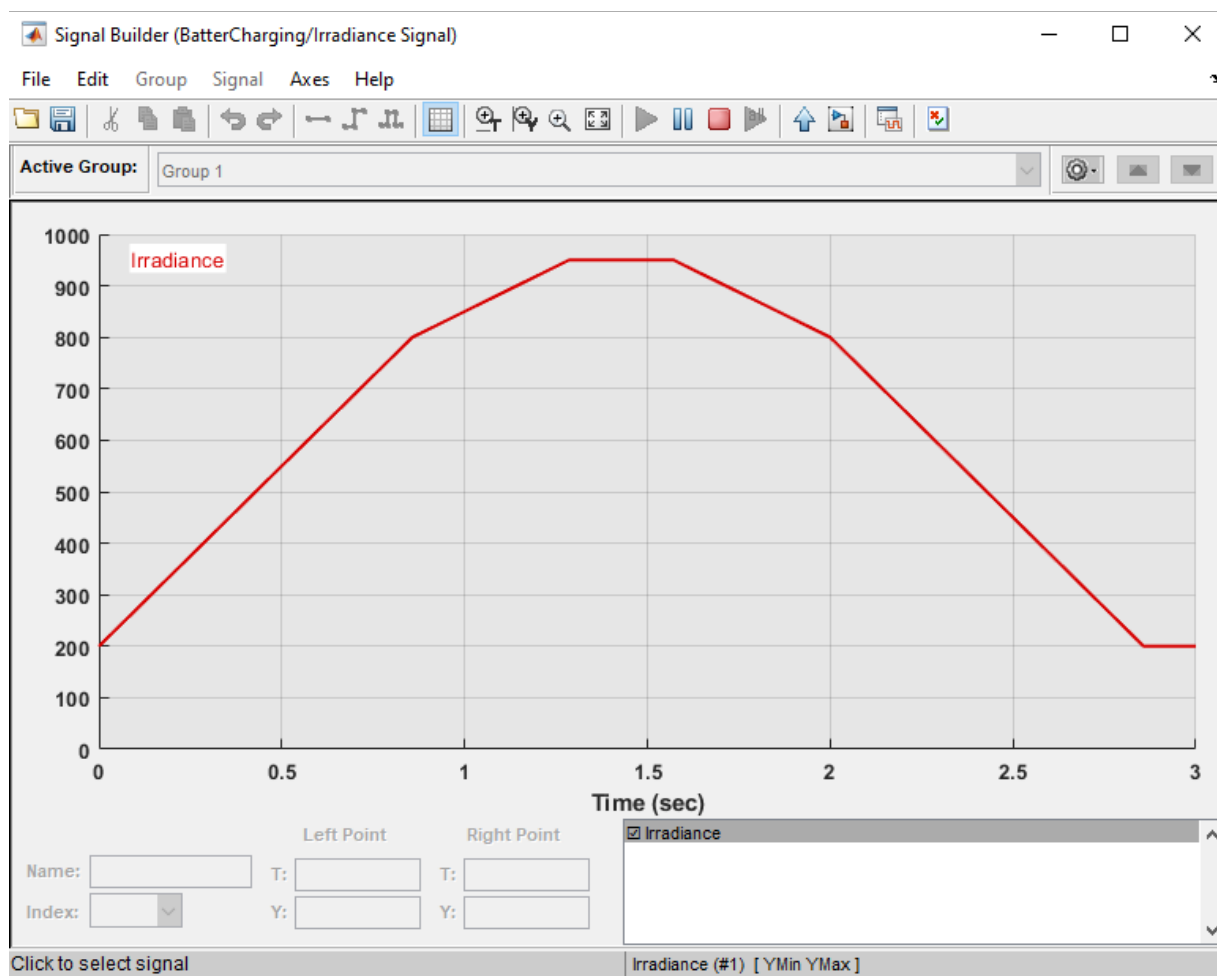


Figura 40. Graficul semnalului de radiație

4.6 SISTEMULUI DE ÎNCĂRCARE AL BATERIEI

Sistemul acesta are scopul de a simula un modul de încărcare al bateriei cu ajutorul unui panou fotovoltaic, pentru a urmări punctul de putere maximă. La intrarea panoului a fost aplicat un semnal variabil, cum ar fi radiația solară care variază pe timpul zilei. Am utilizat algoritmul P&O, care calculează puterea cu valoarea tensiunii primite de la intrarea panoului și compară valorile anterioare ale puterii pentru a asigura urmărirea punctului de putere maximă (MPPT). Dacă diferența este 0, algoritmul se întoarce și efectuează din nou calculul, dacă diferența nu este 0, se verifică valoarea de intrare a panoului, iar tensiunea de referință crește sau scade în funcție de modificarea tensiunii de intrare.

Am folosit funcția MATLAB în Simulink și am calculat eficiența la pornirea simulării. În acest sistem raportul de putere de ieșire/intrare este de aproximativ 70%. Pentru încărcarea inițială a bateriei am ales 20%, iar bateria se încarcă în timp în funcție de intensitatea radiației care vine de la intrare.

În ceea ce privește analiza statisticilor bateriei, semnalul albastru reprezintă curentul, care este negativ deoarece bateria folosită este în curs de încărcare. Tensiunea bateriei utilizate este de 12V și pare să fie stabilă, dar dacă mărim și ne uităm mai atent se poate vedea că bateria se încarcă. Acest sistem continuă să funcționeze în funcție de intensitatea radiației care vine de la intrare.

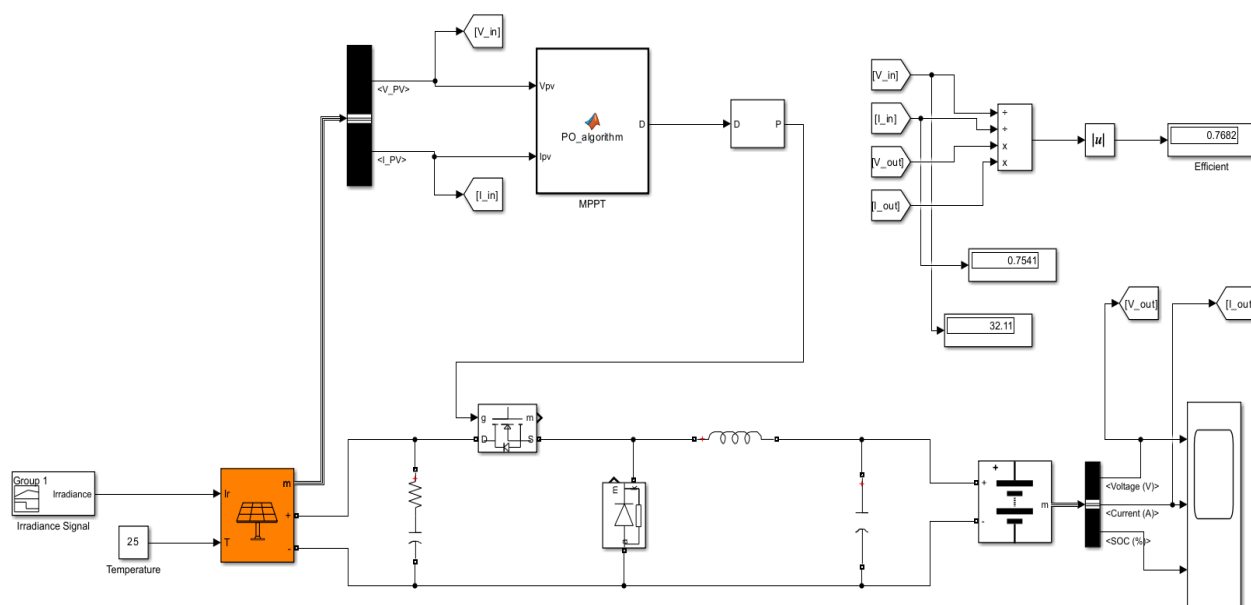


Figura 41. Simularea sistemului de încărcare al bateriei

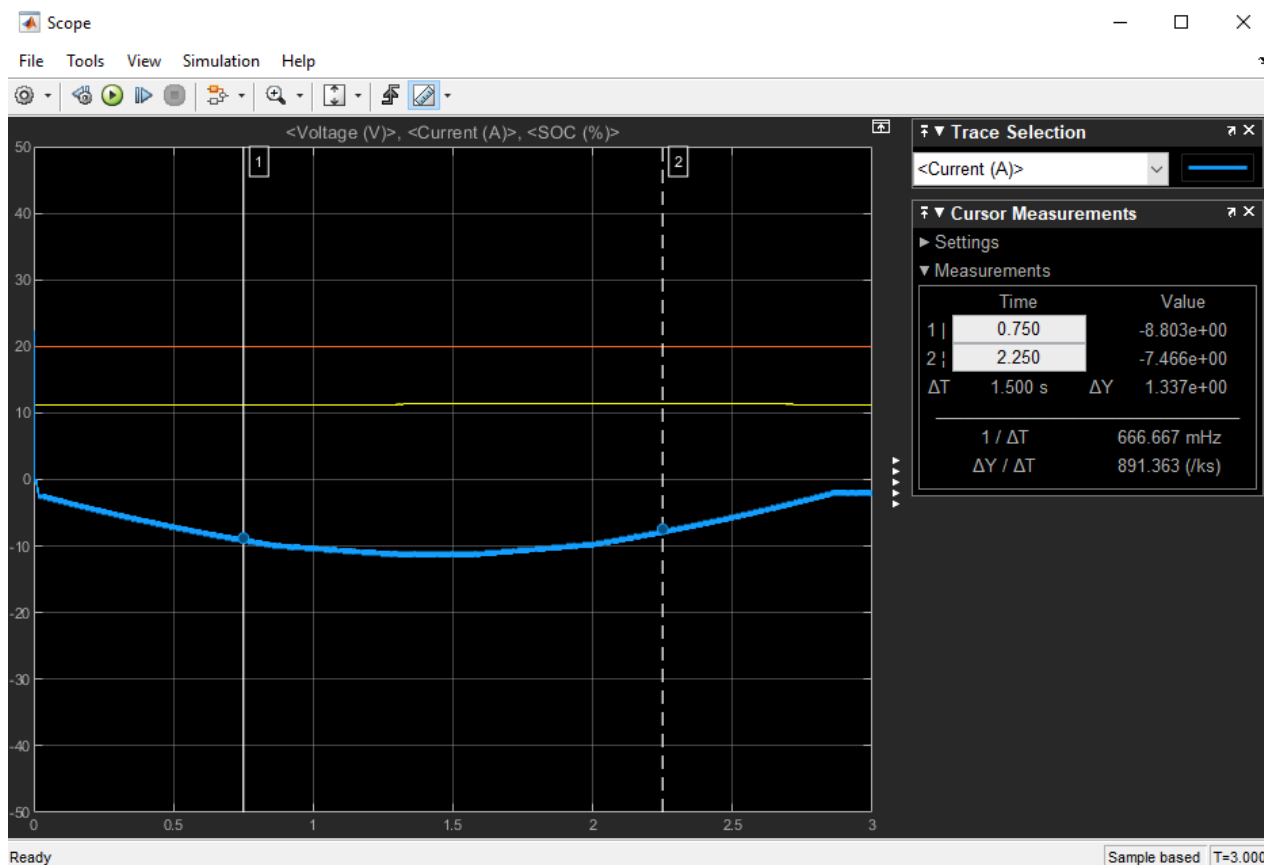


Figura 42. Graficul de încărcare al bateriei

În această simulare se obține o eficiență de aproximativ 70%. Această eficiență se obține din raportul putere de ieșire/putere de intrare. În Figura 37 este reprezentat calculul eficienței.

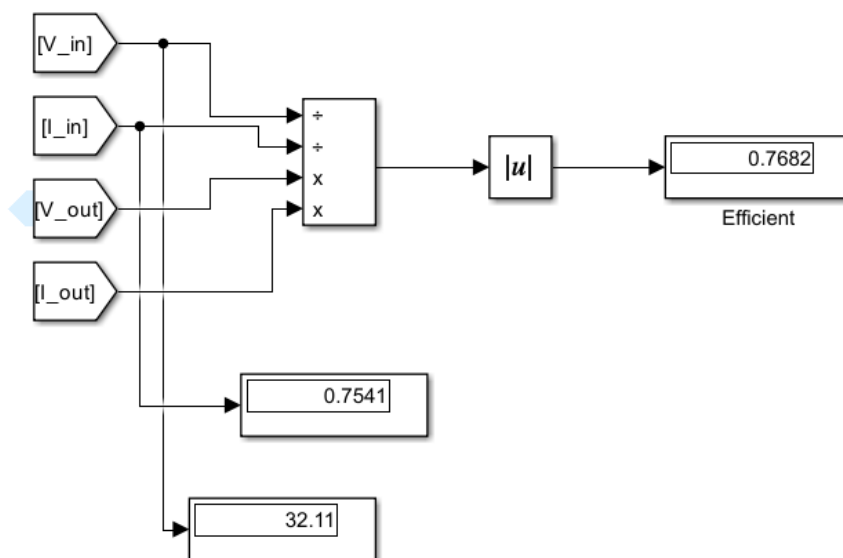


Figura 43. Calculul eficienței

4.7 CONCLUZIILE ACESTEI SIMULĂRI

Scopul acestui sistem este de a simula un modul de încărcare al unei baterii cu ajutorul unui panou fotovoltaic. La intrarea panoului nostru se aplică un semnal variabil, cum ar fi radiația care se modifică constant pe parcursul unei zile. Scopul principal este de a urmări punctul de putere maximă prin aplicarea acestui semnal. Există diferite metode pentru această urmărire, dare metoda pe care am ales-o este algoritmul P&O.

Scopul algoritmului P&O este de a calcula puterea folosind valoarea tensiunii preluate la intrarea în panou. În acest algoritm valoarea puterii de la intrare este comparată cu valoare puterii anterioare, iar apoi se calculează diferența. Dacă diferența este 0, algoritmul se întoarce la început și efectuează din nou calculul. Dacă diferența nu este 0, valoarea de intrare a panoului este controlată.

În timpul simulării modelului s-au obținut trei paramteri unde se observă că bateria se încarcă și menține o tensiune constantă.

5 REZULTATE EXPERIMENTALE

Scopul acestui experiment este de a evalua performanța și eficiența unui sistem de urmărire solară în comparație cu un sistem fix, în ceea ce privește producția de energie solară. Sistemul de urmărire solară este proiectat pentru a muta panoul solar în direcția optimă în funcție de mișcarea soarelui pe parcursul zilei, în timp ce sistemul fix are panoul solar într-o poziție statică.

Pentru acest experiment am preluat datele de tensiune și curent pe parcursul unei zile cu ajutorul unui multimetru atât pentru sistemul fix, cât și pentru sistemul de urmărire solară.

Rezultatele experimentale ale sistemului fotovoltaic fix și ale sistemul de urmărire solară sunt prezentate în Tabelul 3 și Tabelul 4, iar reprezentările grafice pot fi văzute în Figura 44.

Tabelul 3. Valoarea curentului și a tensiunii sistemului fotovoltaic fix la diferite intervale de timp

Ora	Tensiune (V)	Curent (A)	Putere (W)
7:00 AM	12.45	0.37	4.847
8:00 AM	13.87	0.41	5.6867
9:00 AM	14	0.45	6.3
10:00 AM	14.7	0.44	6.468
11:00 AM	15.1	0.48	7.248
12:00 PM	15.13	0.5	7.565
1:00 PM	15	0.49	7.35
2:00 PM	14.95	0.44	6.578
3:00 PM	14.93	0.41	6.1213
4:00 PM	14.91	0.4	5.964
5:00 PM	14.9	0.38	5.662
6:00 PM	13.5	0.37	4.995

Tabelul 4. Valoarea curentului și a tensiunii sistemului fotovoltaic de urmărire solară la diferite intervale de timp

Ora	Tensiune (V)	Curent (A)	Putere (W)
7:00 AM	15	0.41	6.15
8:00 AM	16	0.45	7.2
9:00 AM	16	0.5	8
10:00 AM	15.8	0.53	8.374
11:00 AM	15.7	0.56	8.792
12:00 PM	15.7	0.57	8.949
1:00 PM	15.6	0.57	8.892
2:00 PM	15.6	0.56	8.736
3:00 PM	15.5	0.54	8.37
4:00 PM	15.35	0.53	8.1355
5:00 PM	15.32	0.47	7.2004
6:00 PM	15.31	0.39	5.9709

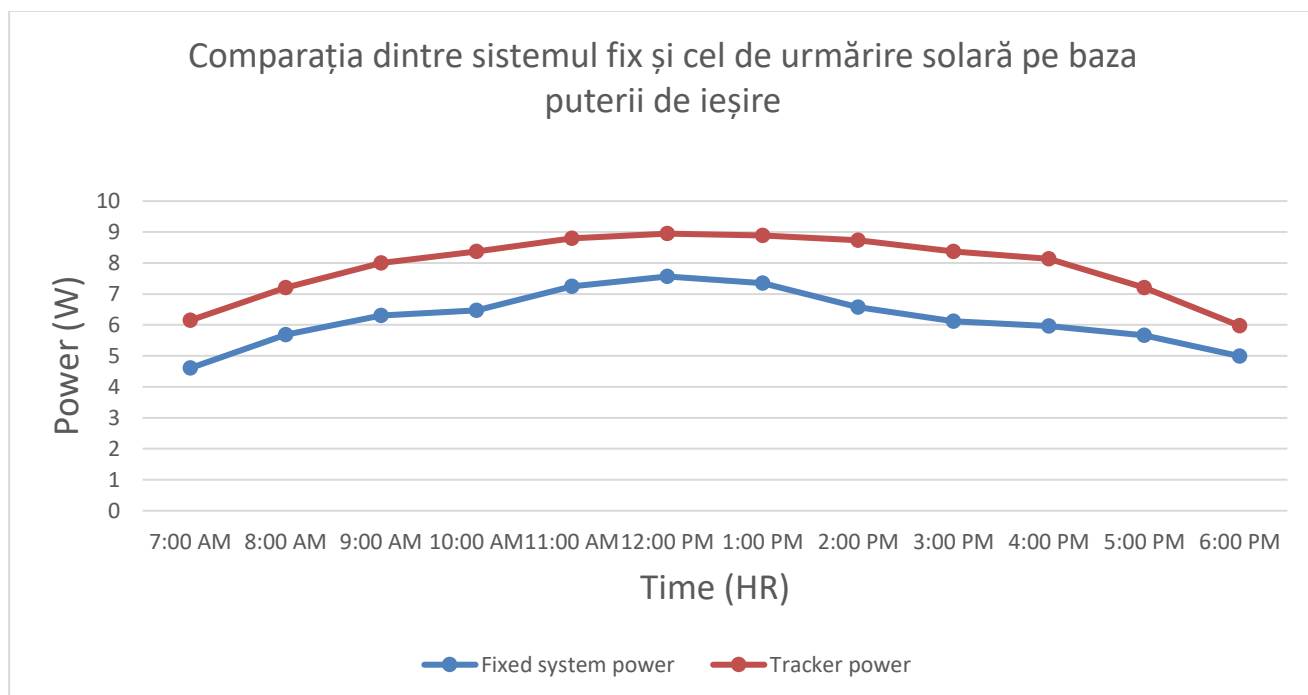


Figura 44. Comparația dintre sistemul fix și cel de urmărire solară

Analizând datele măsurate, se poate observa cum producția de energie electrică variază în funcție de momentul zilei, se pot remarca creșteri de producție în anumite intervale orare ale zilei. Câștigul de putere obținut de sistemul de urmărire solară cu două axe față de sistemul fix este prezentat în Tabelul 5. Puterea maximă de ieșire calculată a sistemului fix este de 7,56W, pe când puterea de ieșire a sistemului de urmărire solară este de 8,94W.

Tabelul 5. Câștigul de putere obținut de sistemul de urmărire solară

Ora	Putere (W) Sistemul fix	Putere (W) Sistemul de urmărire	Câștigul în procente (%)
7:00 AM	4.847	6.15	26.88
8:00 AM	5.6867	7.2	26.61
9:00 AM	6.3	8	26.98
10:00 AM	6.468	8.374	29.47
11:00 AM	7.248	8.792	21.30
12:00 PM	7.565	8.949	18.29
1:00 PM	7.35	8.892	20.98
2:00 PM	6.578	8.736	32.81
3:00 PM	6.1213	8.37	36.74
4:00 PM	5.964	8.1355	36.41
5:00 PM	5.662	7.2004	27.17
6:00 PM	4.995	5.9709	19.54

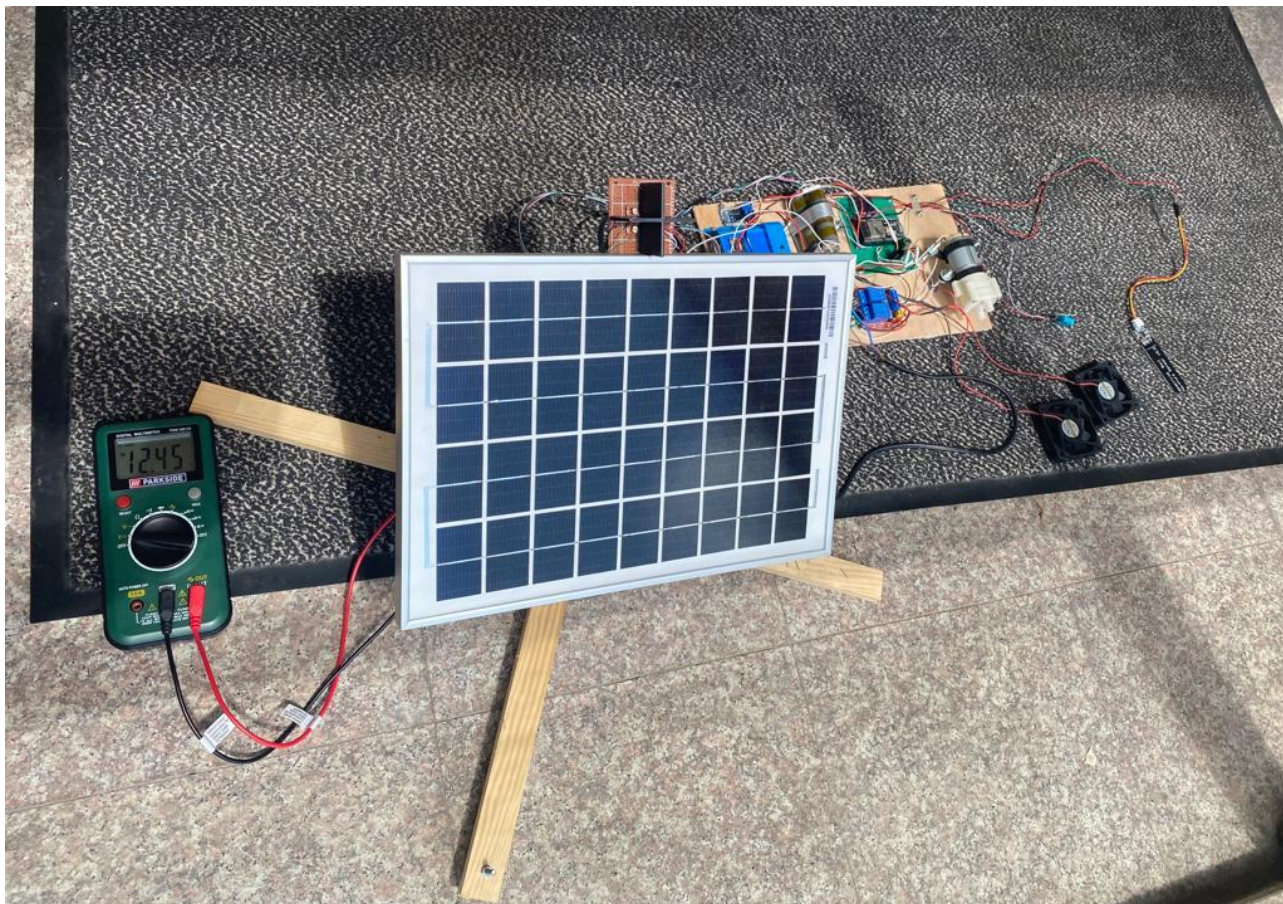


Figura 45. Imagine din timpul experimentului

În Figura 46 se poate observa interfața grafică a serei cu alimentare solară, unde în acel moment temperatura și umiditatea solului din interiorul serei erau sub pragurile setate de mine pentru a fi pornit sistemul de răcire, respectiv sistemul de irigare.



Figura 46. Sistemele de răcire și irigare oprite

Am crescut temperatura cu ajutorul unui feon, temperatura ajungând să fie între pragurile setate, sistemul de ventilație a pornit (Figura 47).

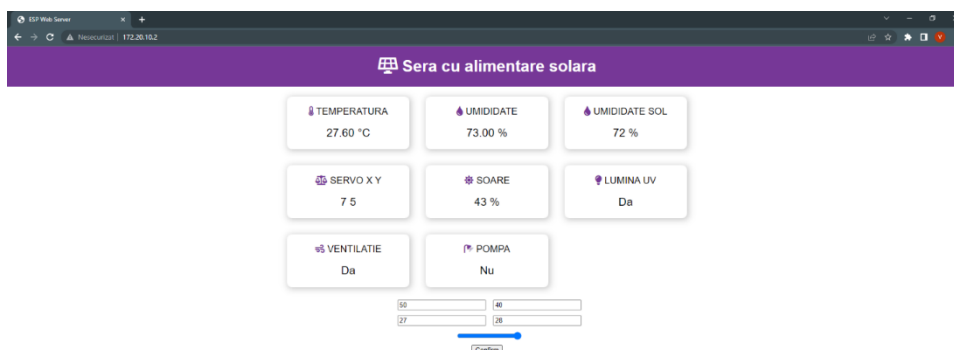


Figura 47 Sistemul de răcire pornit

Am așteptat până când umiditatea solului a scăzut sub pragul maxim setat, iar apoi pompa de apă a pornit și a început să ude planta (Figura 48).

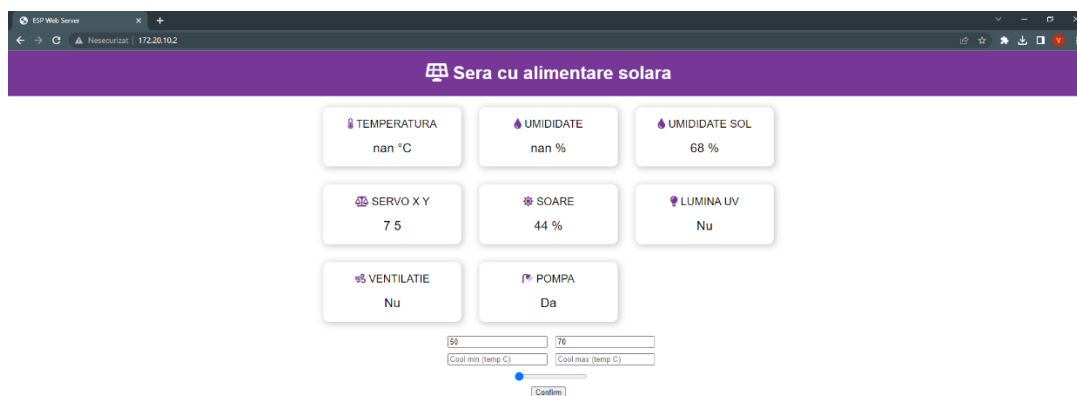


Figura 48. Sistemul de irigare pornit

6 CONCLUZII

Având la bază un bagaj de cunoștințe, mi-am dorit ca prin intermediul acestui proiect să scot în evidență informațiile dobândite pentru a realiza o lucrare de care să fiu mulțumit.

Pentru realizarea proiectului, am urmat mai multe etape, pe care le-am evidențiat și explicat în paginile anterioare ale acestei documentații.

Am tratat mai multe aspecte din punct de vedere științific, precum implementarea unor condiții optime în care o plantă se poate dezvolta, dar și din punct de vedere tehnic, cum ar fi programarea unui microcontroller pentru a controla funcționalitatea sistemului de urmărire solară. Acesta a fost responsabil de detectarea și urmărirea mișcării soarelui, astfel încât panoul solar să fie poziționat în permanență pentru a obține eficiența maximă de captare a energiei solare. Am utilizat tehnici de programare și algoritmi pentru a asigura un tracking precis și eficient al poziției soarelui.

Prin integrarea sistemului de urmărire solară într-o seră inteligentă, am creat o soluție inovatoare care optimizează utilizarea energiei solare pentru alimentarea serelor și controlul mediului de creștere al plantelor.

Prin intermediul acestui proiect, am realizat și lucruri ce țin de mecanică, precum construirea sistemului de urmărire solară și construirea unei versiuni în miniatură a unei sere, cu tot ce implică acest lucru, de la cumpărarea materialelor, până la construirea propriu zisă.

Am avut numeroase beneficii lucrând la acest proiect, mi-am atribuit atât cunoștințe teoretice, cât și practice, care țin atât de software, cât și de hardware.

Dezvoltări ulterioare

- Îmbunătățirea eficienței sistemului de urmărire solară utilizând algoritmi și senzori avansați;
- Controlarea mai multor senzori: de pH, CO₂;
- Acționarea unor ferestre pentru seră;
- Acționarea unor sisteme de umbrire și de închidere a ferestrelor în funcție de temperatură și viteza vântului.

7 BIBLIOGRAFIE

- [1] <https://www.lumeasatului.ro/articole-revista/agrotehnica/6911-solarino-solutie-de-automatizare-smart-a-serelor-agricole.html>
- [2] <https://mobilecontrol.ro/main/?product=automatizare-smart-pentru-sere-agricole>
- [3] <https://express.adobe.com/page/ZzVRwilNwtaK/>
- [4] <https://www.bluemonitor.ro/sere-si-solari/>
- [5] Al Mamun, M. A., Dargusch, P., Wadley, D., Zulkarnain, N. A., & Aziz, A. A. (2022). A review of research on agrivoltaic systems. *Renewable and Sustainable Energy Reviews*, 161, 112351.
- [6] https://www.mouser.com/datasheet/2/891/esp-wroom-32_datasheet_en-1223836.pdf
- [7] Putri, M., & Aryza, S. (2018). Design of security tools using sensor Light Dependent Resistor (LDR) through mobile phone. *Int. J. Innov. Res. in Comp. and Comm. Eng.*, 4(10), 168-173.
- [8] <https://datasheetspdf.com/pdf/785590/D-Robotics/DHT11/1>
- [9] https://media.digikey.com/pdf/data%20sheets/dfrobot%20pdfs/sen0193_web.pdf
- [10] <https://datasheetspdf.com/pdf/942981/ETC/MG996R/1>
- [11] Nogueira, C. E. C., Bedin, J., Niedzialkoski, R. K., de Souza, S. N. M., & das Neves, J. C. M. (2015). Performance of monocrystalline and polycrystalline solar panels in a water pumping system in Brazil. *Renewable and Sustainable Energy Reviews*, 51, 1610-1616
- [12] <https://www.tme.eu/Document/f73597b9cc5801bdd87f2781fa4ee352/CL-SM10P.pdf>
- [13] <https://datasheetspdf.com/pdf/798762/SolomonSystech/SSD1306/1>
- [14] <https://www.sigmanortec.ro/modul-releu-cu-optocuplor-nivel-high-sau-low-selectabil-2-canale-5v>
- [15] <https://www.sigmanortec.ro/Pompa-Apa-Aer-cu-diafragma-6-12V-R385-p190556176>
- [16] Yadav, A., Singh, D., Lingwan, M., Yadukrishnan, P., Masakapalli, S. K., & Datta, S. (2020). Light signaling and UV-B-mediated plant growth regulation. *Journal of Integrative Plant Biology*, 62(9), 1270-1292.
- [17] SAPUTRA, H., SUMITRA, I. D., HIRAWAN, D., LESMANA, R., & SOEGOTO, E. S. (2023). SMART URBAN FARMING APPLICATION: UV LIGHT IN HYDROPONIC INSTALLATIONS. *Journal of Engineering Science and Technology*, 18(2), 1007-1018.

- [18] <https://www.sigmanortec.ro/LED-5mm-UV-395-400nm-p125799722>
- [19] Eldahab, Y. E. A., Saad, N. H., & Zekry, A. (2016). Enhancing the design of battery charging controllers for photovoltaic systems. *Renewable and Sustainable Energy Reviews*, 58, 646-655.
- [20] <https://www.allaboutcircuits.com/projects/design-a-luxmeter-using-a-light-dependent-resistor/>
- [21] Singrodia, V., Mitra, A., & Paul, S. (2019, January). A review on web scrapping and its applications. In 2019 international conference on computer communication and informatics (ICCCI) (pp. 1-6). IEEE.
- [22] <https://www.mathworks.com/help/sps/powersys/ref/pvarray.html>
- [23] <https://www.mathworks.com/solutions/electrification/mppt-algorithm.html>
- [24] Cruz, C., Baptista Lema, M., del Toro García, X., & Roncero-Sánchez, P. (2016). Energy storage integration with renewable energies: The case of concentration photovoltaic systems. *Environment, Energy and Climate Change II: Energies from New Resources and the Climate Change*, 73-94
- [25] Ahmed, J., & Salam, Z. (2015). An improved perturb and observe (P&O) maximum power point tracking (MPPT) algorithm for higher efficiency. *Applied Energy*, 150, 97-108
- [26] Gaikwad, N. A., Sreekar, V. S., Sathe, T., & Mangrulkar, C. (2023). Review on Recent Developments in Agrovoltaic System. *Renewable Resources and Energy Management*, 359-369

REZUMAT

Proiectul pe care l-am realizat are ca obiectiv principal dezvoltarea unei plante în condiții normale, în zonele rurale, unde oamenii își construiesc sere pentru a-și ușura munca. Un sistem care controlează și monitorizează parametrii dintr-o seră, în primul rând presupune muncă de documentare despre plante și în ce condiții se dezvoltă acestea.

Pe de altă parte sunt necesari senzori care să funcționeze în condițiile dintr-o seră și să poată oferi valori precise. Am interfațat acești senzori cu un microcontroller pe care l-am programat să afișeze toate valorile achiziționate pe un display și pe o pagină web.

Pe lângă funcționalitatea de monitorizare și control a serei, obiectivul proiectului este de a reduce dependența de sursele tradiționale de energie și de a contribui la protejarea mediului înconjurător, prin utilizarea unui sistem de energie regenerabilă (Solar Tracker), care urmărește mișcarea soarelui și ajustează panoul solar pentru a obține o producție de energie cât mai mare. Prin utilizarea acestui sistem, se reduc emisiile de gaze cu efect de seră și a altor poluanți.

Proiectul se concentrează pe dezvoltarea unei soluții complete și integrate, care să faciliteze creșterea plantelor în sere, să ofere control și monitorizare eficientă a parametrilor de mediu și să contribuie la protejarea mediului prin utilizarea energiei regenerabile.

ABSTRACT

The main goal of my project is to develop a plant in normal conditions, in rural areas, where people build greenhouses to make their work easier. A system that controls and monitors the parameters of a greenhouse, primarily involves documentation of plants and the conditions under which they grow.

On the other hand, sensors are needed that work under the conditions in a greenhouse and can provide accurate values. We interfaced these sensors with a microcontroller that we programmed to display all the acquired values on a display and a web page.

In addition to greenhouse monitoring and control functionality, the project's goal is to reduce dependence on traditional energy sources and help protect the environment by using a renewable energy system (Solar Tracker) that tracks the movement of the sun and adjusts the solar panel to achieve the highest possible energy output. By using this system, emissions of greenhouse gases and other pollutants are reduced.

The project focuses on developing a complete and integrated solution that facilitates plant growth in greenhouses, provides effective control and monitoring of environmental parameters and contributes to environmental protection through the use of renewable energy.

ANEXE

```

#include <ESP32Servo.h>           // Bibliotecă pentru controlul servo-motoarelor
#include <SPI.h>                  // Bibliotecă pentru comunicare prin SPI
#include <Wire.h>                 // '-' pentru comunicarea prin I2C
#include <Adafruit_GFX.h>         // '-' pentru controlul afișajului OLED
#include <Adafruit_SSD1306.h>     // '-' pentru controlul afișajului OLED
#include "DHTesp.h"              // '-' pentru senzorul DHT11
#include <WiFi.h>                 // '-' pentru gestionarea conexiunii WiFi
#include <AsyncTCP.h>             // '-' pentru crearea unui server web
#include <ESPAsyncWebServer.h>    // '-' pentru crearea unui server web
#include <EEPROM.h>              // '-' pentru lucrul cu memoria EEPROM a ESP32

const char* ssid = "neosera23";   // SSID-ul rețelei WiFi la care ESP32 se va conecta
const char* password = "nanotech"; // Parola rețelei WiFi

// Crearea unui server web asincron pe portul 80
AsyncWebServer server(80);

// Crearea unui eveniment de tip Event Source care poate fi utilizat pentru a trimite actualizări în
// timp real către client
AsyncEventSource events("/events");

// Variabile de temporizare
unsigned long lastTime = 0;
unsigned long timerDelay = 3000;
#define DHTPIN 19 // Pinul utilizat pentru senzorul de temperatură DHT11
#define REL_1 26 // Pinul utilizat pentru pompă
#define REL_2 27 // Pinul utilizat pentru ventilator
#define SCREEN_WIDTH 128 // Lățimea afișajului OLED
#define SCREEN_HEIGHT 64 // Înălțimea afișajului OLED
// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
#define OLED_RESET -1 // Reset pin # (or -1 if sharing Arduino reset pin)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
DHTesp dht;
const int AirValue = 2910; // Valoarea citită de senzor când solul este uscat (în aer)
const int WaterValue = 980; // Valoarea citită de senzor când solul este complet umed (100%
// umiditate)
const int SensorPin = 32; // Pinul utilizat pentru citirea valorii de umiditate a solului
const int ledPin = 23; // Pinul utilizat pentru LED UV
int soilMoistureValue = 0; // Stocază valoarea citită de senzorul higrometric
int soilmoisturepercent=0; // Stocază procentul de umiditate
float temp = 0; // Stocază valoarea temperaturii aerului
float hum = 0; // Stocază valoarea umidității aerului
int checkh = 0;
int checkv = 0;
Servo Orizontal; // Orizontal servo
int servoh = 180; // Inițializează unghiul la 180 de grade
int limitaSuperioaraServoh = 175;
int limitaInferioaraServoh = 5;
Servo Vertical; // Vertical servo
int servov = 45; // Inițializează unghiul la 45 de grade
int limitaSuperioaraServov = 60;
int limitaInferioaraServov = 5;
int waterMin=0, waterMax=0; // Variabile inițializate cu valoarea 0
int coolMin=0, coolMax=0;
int statusUv = 0;
bool hasClientWeb = false;
// Conexiunile pinilor LDRR
int ldr_ss = 36; //LDR stânga sus
int ldr_ds = 39; //LDR dreapta sus
int ldr_sj = 34; //LDR stânga jos
int ldr_dj = 35; //LDR dreapta jos
int sunLevel = 0;
char buff[128];
//const char index_html[] PROGMEM = R"rawliteral(
<!DOCTYPE HTML><html>
<head>
  <title>ESP Web Server</title>
  <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-fnmOCqbTlWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr"
crossorigin="anonymous">
  <link rel="icon" href="data:,">
  <style>

```

```

html {font-family: Arial; display: inline-block; text-align: center;}
p { font-size: 1.2rem;}
body { margin: 0;}
.topnav { overflow: hidden; background-color: #763797; color: white; font-size: 1rem; }
.content { padding: 20px; }
.card { border-radius: 10px; background-color: white; box-shadow: 2px 2px 12px 1px
rgba(140,140,140,.5); }
.cards { max-width: 800px; margin: 0 auto; display: grid; grid-gap: 2rem; grid-template-columns:
repeat(auto-fit, minmax(200px, 1fr)); margin-bottom: 25px; }
.reading { font-size: 1.4rem; }
.card i {color: #763797}
input {margin-bottom: 10px; margin-left: 10px;}
</style>
</head>
<body>
  <div class="topnav">
    <h1><i class="fas fa-solar-panel"></i> Sera cu alimentare solara</h1>
  </div>
  <div class="content">
    <div class="cards">
      <div class="card">
        <p><i class="fas fa-thermometer-half"></i> TEMPERATURA</p><p><span class="reading"><span
id="temp"></span> &deg;C</span></p>
      </div>
      <div class="card">
        <p><i class="fas fa-tint"></i> UMIDIDATE AER</p><p><span class="reading"><span
id="hum"></span> &percnt;</span></p>
      </div>
      <div class="card">
        <p><i class="fas fa-tint"></i> UMIDIDATE SOL</p><p><span class="reading"><span
id="soil"></span> &percnt;</span></p>
      </div>
      <div class="card">
        <p><i class="fas fa-balance-scale"></i> SERVO X Y</p><p><span class="reading"><span
id="servoh"></span> <span id="servov"></span></span></p>
      </div>
      <div class="card">
        <p><i class="fas fa-sun"></i> SOARE </p><p><span class="reading"><span id="sunLevel"></span>
&percnt;</span></p>
      </div>
      <div class="card">
        <p><i class="fas fa-lightbulb"></i> LUMINA UV </p><p><span class="reading"><span
id="uvLight">Nu</span></span></p>
      </div>
      <div class="card">
        <p><i class="fas fa-wind"></i> VENTILATIE </p><p><span class="reading"><span
id="coolerStatus">Nu</span></span></p>
      </div>
      <div class="card">
        <p><i class="fas fa-shower"></i> POMPA </p><p><span class="reading"><span
id="pumpStatus">Nu</span></span></p>
      </div>
      <div>
        <input id="waterMin" value="%waterMin%" placeholder="Apa min (soil hum.)"> <input id="waterMax"
value="%waterMax%" placeholder="Apa max (soil hum.)">
        <br>
        <input id="coolerMin" value="%coolMin%" placeholder="Cool min (temp C)"> <input id="coolerMax"
value="%coolMax%" placeholder="Cool max (temp C)">
        <br>
        <input id="slider" type="range" min="0" max="1" value="%sliderVal%" class="slider">
        <br>
        <button onclick="sendData()">Confirm</button>
      </div>
    </div>
  </div>
</script>
function sendData() {
  var wmin = document.getElementById('waterMin').value;
  var wmax = document.getElementById('waterMax').value;
  var cmin = document.getElementById('coolerMin').value;
  var cmax = document.getElementById('coolerMax').value;
  var statusUv = document.getElementById('slider').value;
  var http = new XMLHttpRequest();
  var params = 'wmin='+wmin+'&wmax='+wmax+'&cmin='+cmin+'&cmax='+cmax+'&statusUv='+statusUv;
  var url = 'set?'+params;
  http.open('GET', url, true);
  //Trimite informațiile de header corespunzătoare împreună cu cererea
  http.setRequestHeader('Content-type', 'application/x-www-form-urlencoded');
  http.onreadystatechange = function() {///Apelează o funcție atunci când starea se schimbă
    if(http.readyState == 4 && http.status == 200) {

```

```

        alert(http.responseText);
        var chk = document.getElementById('slider').value;
        document.getElementById('uvLight').innerHTML = (chk == 0 ? 'Nu' : 'Da');
    }
    http.send();
}
if (!!window.EventSource) {
    var source = new EventSource('/events');

    source.addEventListener('open', function(e) {
        console.log("Events Connected");
    }, false);
    source.addEventListener('error', function(e) {
        if (e.target.readyState != EventSource.OPEN) {
            console.log("Events Disconnected");
        }
    }, false);

    source.addEventListener('message', function(e) {
        console.log("message", e.data);
    }, false);

    source.addEventListener('temp', function(e) {
        console.log("temperature", e.data);
        document.getElementById("temp").innerHTML = e.data;
    }, false);

    source.addEventListener('hum', function(e) {
        console.log("humidity", e.data);
        document.getElementById("hum").innerHTML = e.data;
    }, false);
    source.addEventListener('soil', function(e) {
        console.log("soil", e.data);
        document.getElementById("soil").innerHTML = e.data;
    }, false);
    source.addEventListener('servoh', function(e) {
        console.log("servoh", e.data);
        document.getElementById("servoh").innerHTML = e.data;
    }, false);
    source.addEventListener('servov', function(e) {
        console.log("servov", e.data);
        document.getElementById("servov").innerHTML = e.data;
    }, false);
    source.addEventListener('sunLevel', function(e) {
        console.log("sunLevel", e.data);
        document.getElementById("sunLevel").innerHTML = e.data;
    }, false);
    source.addEventListener('coolerStatus', function(e) {
        console.log("coolerStatus", e.data);
        document.getElementById("coolerStatus").innerHTML = e.data;
    }, false);
    source.addEventListener('pumpStatus', function(e) {
        console.log("pumpStatus", e.data);
        document.getElementById("pumpStatus").innerHTML = e.data;
    }, false);
}
var chk = document.getElementById('slider').value;
document.getElementById('uvLight').innerHTML = (chk == 0 ? 'Nu' : 'Da');
</script>
</body>
</html>rawliteral";

void notFound(AsyncWebServerRequest *request) {
    request->send(404, "text/plain", "Not found");
}

String processor(const String& var){
    //Serial.println(var);
    if(var == "waterMin"){
        return String(waterMin);
    }
    else if(var == "waterMax"){
        return String(waterMax);
    }
    else if(var == "coolMin"){
        return String(coolMin);
    }
    else if(var == "coolMax"){

```

```
    return String(coolMax);
}
else if(var == "sliderVal") {
    return String(statusUv);
}
return String();
}

void setup(){
    Serial.begin(9600);
    pinMode(ledPin, OUTPUT);
    digitalWrite(ledPin, LOW);
    if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
        Serial.println(F("SSD1306 allocation failed"));
        for(;;); // Dacă inițializarea eșuează programul intră într-o buclă infinită
    }
    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, password);

    display.clearDisplay();
    display.setTextSize(1);          // Setarea dimensiunii textului
    display.setTextColor(WHITE);     // Setarea culorii textului

    while (WiFi.status() != WL_CONNECTED) {
        display.clearDisplay();
        display.setCursor(0,0);
        sprintf(buff, "SSID: %s", ssid);
        display.println(buff);
        sprintf(buff, "PASS: %s", password);
        display.println(buff);
        display.display();
        delay(1000);
    }
    display.clearDisplay();
    display.setCursor(0,0);
    sprintf(buff, "http://%s", WiFi.localIP().toString());
    Serial.println(WiFi.localIP());
    display.println(buff);
    display.display();

    EEPROM.begin(10);
    waterMin = EEPROM.read(0);
    waterMax = EEPROM.read(1);
    coolMin = EEPROM.read(2);
    coolMax = EEPROM.read(3);
    statusUv = EEPROM.read(4);
    servoh = EEPROM.read(5);
    servov = EEPROM.read(6);
    // Handle Web Server
    server.on("/", HTTP_GET, [] (AsyncWebServerRequest *request){
        hasClientWeb = true;
        request->send_P(200, "text/html", index_html, processor);
    });

    events.onConnect([] (AsyncEventSourceClient *client){
        if(client->lastId()){
            Serial.printf("Client reconnected! Last message ID that it got is: %u\n", client->lastId());
        }
        client->send("hello!", NULL, millis(), 10000);
    });
    server.on("/set", HTTP_GET, [] (AsyncWebServerRequest *request) {
        if(request->hasParam("wmin") && request->hasParam("wmax") && request->hasParam("cmin") &&
request->hasParam("cmax")) {
            waterMin = request->getParam("wmin")->value().toInt();
            waterMax = request->getParam("wmax")->value().toInt();
            coolMin = request->getParam("cmin")->value().toInt();
            coolMax = request->getParam("cmax")->value().toInt();
            statusUv = request->getParam("statusUv")->value().toInt();
            EEPROM.write(0, waterMin);
            EEPROM.write(1, waterMax);
            EEPROM.write(2, coolMin);
            EEPROM.write(3, coolMax);
            if(statusUv != 0 && statusUv != 1)
                statusUv = 0;
            EEPROM.write(4, statusUv);
            EEPROM.commit();
        }
        request->send(200, "text/html", "Setari confirmate");
    });
};
```

```

server.addHandler(&events);
server.begin();
pinMode(REL_1, OUTPUT); // Setarea pinului REL_1 ca ieșire (pompa)
pinMode(REL_2, OUTPUT); // Setarea pinului REL_2 ca ieșire (ventilatorul)
digitalWrite(REL_1, LOW);
digitalWrite(REL_2, LOW);
byte resultValue = 0;
// Inițializează senzorul DHT11
dht.setup(DHTPIN, DHTesp::DHT11);
Serial.println("DHT initiated");
ESP32PWM::allocateTimer(0);
ESP32PWM::allocateTimer(1);
ESP32PWM::allocateTimer(2);
ESP32PWM::allocateTimer(3);
if(servoh > limitaSuperioaraServoh)
    servoh = limitaSuperioaraServoh;
else if(servoh < limitaInferioaraServoh)
    servoh = limitaInferioaraServoh;

if(servov > limitaSuperioaraServov)
    servov = limitaSuperioaraServov;
else if(servov < limitaInferioaraServov)
    servov = limitaInferioaraServov;
horizontal.attach(12);
vertical.attach(13);
horizontal.write(servoh);
vertical.write(servov);
checkh = servoh;
checkv = servov;
}
/*int counterh = 0;
int counterv = 0;*/
int oldv = 0;
int oldh = 0;
bool coolerStat = false; // setam initial valoarea fals
bool pumpStat = false;   // _"
void loop() {
    soilMoistureValue = analogRead(SensorPin); // citirea valorii analogice de la senzorul de
umiditate
    soilmoisturepercent = map(soilMoistureValue, AirValue, WaterValue, 0, 100);
    if(soilmoisturepercent > 100) soilmoisturepercent = 100;
    else if(soilmoisturepercent < 0) soilmoisturepercent = 0;

    //Serial.println(soilMoistureValue); // utilizat pentru a calibra senzorul de umiditate sol la
uscat si 100% la apa

    TempAndHumidity newValues = dht.getTempAndHumidity();
    if (!isnan(!newValues.temperature) && !isnan(!newValues.humidity)) {
        Serial.println("here: ");
        Serial.print(newValues.temperature);
        Serial.print(" ");
        Serial.print(newValues.humidity);
        temp = newValues.temperature;
        hum = newValues.humidity;
    } else {
        temp = 0;
        hum = 0;
    }
}
if ((millis() - lastTime) > timerDelay) {
    if(hasClientWeb) {
        display.clearDisplay();
        display.setTextSize(1);
        display.setTextColor(WHITE);
        display.setCursor(0, 1);
        sprintf(buff, "Temp: %s", String(temp));
        display.println(buff);
        sprintf(buff, "Umid: %s", String(hum));
        display.println(buff);
        sprintf(buff, "Sol: %d", soilmoisturepercent);
        display.println(buff);
        sprintf(buff, "ServoX: %d", servoh);
        display.println(buff);
        sprintf(buff, "ServoY: %d", servov);
        display.println(buff);
        display.display();
        if(statusUv == 1) {
            digitalWrite(ledPin, HIGH);
        } else {

```

```

        digitalWrite(ledPin, LOW);
    }
}
//Serial.println(statusUv)
if(soilmoisturepercent >= waterMin && soilmoisturepercent <= waterMax) {
    digitalWrite(REL_1, HIGH);
    pumpStat = true;
} else {
    digitalWrite(REL_1, LOW);
    pumpStat = false;
}
if(temp >= coolMin && temp <= coolMax) {
    digitalWrite(REL_2, HIGH);
    coolerStat = true;
} else {
    digitalWrite(REL_2, LOW);
    coolerStat = false;
}
events.send("ping",NULL,millis());
events.send(String(temp).c_str(),"temp",millis());
events.send(String(hum).c_str(),"hum",millis());
events.send(String(soilmoisturepercent).c_str(), "soil", millis());
events.send(String(checkh).c_str(), "servoh", millis());
events.send(String(checkv).c_str(), "servov", millis());
events.send(String(sunLevel).c_str(), "sunLevel", millis());
if(coolerStat)
    events.send("Da", "coolerStatus", millis());
else
    events.send("Nu", "coolerStatus", millis());
if(pumpStat)
    events.send("Da", "pumpStatus", millis());
else
    events.send("Nu", "pumpStatus", millis());
lastTime = millis();
}
// citire senzori ldr:
int ss = analogRead(ldr_ss); // stânga sus
int ds = analogRead(ldr_ds); // dreapta sus
int sj = analogRead(ldr_sj); // stânga jos
int dj = analogRead(ldr_dj); // dreapta jos
sunLevel = int((ss+ds+sj+dj)/4); // media celor 4 LDR
sunLevel = map(sunLevel, 0, 4095, 0, 100); // funcția map
int dtime = 20; int toleranta = 200; // dtime=diferența de timp, toleranta=tooleranța
//mediile valorilor senzorilor LDR
int valoareMedieSus = (ss + ds) / 2; // valoare medie sus
int valoareMedieJos = (sj + dj) / 2; // valoarea medie jos
int valoareMedieStanga = (ss + sj) / 2; // valoarea medie stânga
int valoareMedieDreapta = (ds + dj) / 2; // valoarea medie dreapta
int difVerticala = valoareMedieSus - valoareMedieJos; // diferența între sus și jos
int difOrizontala = valoareMedieStanga - valoareMedieDreapta; // diferența între stânga și dreapta
if (-1*toleranta > difVerticala || difVerticala > toleranta) { // daca nu se afla in toleranta se
face mutarea pe verticala
    if (valoareMedieSus > valoareMedieJos) {
        servov = ++servov;
        if (servov > limitaSuperioaraServov) {
            servov = limitaSuperioaraServov;
        }
    } else if (valoareMedieSus < valoareMedieJos) {
        servov = --servov;
        if (servov < limitaInferioaraServov) {
            servov = limitaInferioaraServov;
        }
    }
}
vertical.write(servov);

// diferiti alogritmi incercati pentru optimizare:
/*if(abs(servov - oldv) > 10) {
    vertical.write(servov);
    oldv = servov;
}*/
/*
if(abs(servov - checkv) > 3) {
    checkv = servov;
    EEPROM.write(6, checkv);
    EEPROM.commit();
    vertical.write(checkv);
}
if(servov != checkv) {

```

```

        if(++counterv >= 2) {
            EEPROM.write(6, checkv);
            EEPROM.commit();
            vertical.write(checkv);
            checkv = servoh;
            counterv = 0;
        }
    }*/

}
if (-1*toleranta > difOrizontala || difOrizontala > toleranta) { // daca nu se afla toleranta se
face mutarea pe orizontala
    if (valoareMedieStanga > valoareMedieDreapta) {
        servoh = --servoh;
        if (servoh < limitaInferioaraServoh) {
            servoh = limitaInferioaraServoh;
        }
    } else if (valoareMedieStanga < valoareMedieDreapta) {
        servoh = ++servoh;
        if (servoh > limitaSuperioaraServoh) {
            servoh = limitaSuperioaraServoh;
        }
    } else if (valoareMedieStanga = valoareMedieDreapta) {
        delay(5000);
    }
    /*if(abs(servoh - oldh) > 10) {
        horizontal.write(servoh);
        oldh = servoh;
    }*/
    horizontal.write(servoh);
    /*
    if(abs(servoh - checkh) > 3) {
        checkh = servoh;
        EEPROM.write(5, checkh);
        EEPROM.commit();
        horizontal.write(checkh);
    }
    if(servoh != checkh) {
        if(++counterh >= 2) {
            EEPROM.write(5, checkh);
            EEPROM.commit();
            horizontal.write(checkh);
            checkh = servoh;
            counterh = 0;
        }
    }*/
}
delay(dtime);
}

```