

Fotonaponski sustav s mogućnošću dvo-osnog praćenja položaja Sunca temeljen na Arduino platformi

Muzica, Leo

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:190:939041>

Rights / Prava: [Attribution 4.0 International](#) / [Imenovanje 4.0 međunarodna](#)

Download date / Datum preuzimanja: **2025-03-15**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Engineering](#)



SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Prijediplomski sveučilišni studij elektrotehnike

Završni rad

**Fotonaponski sustav s mogućnošću dvo-osnog praćenja položaja
Sunca temeljen na Arduino platformi**

Rijeka, srpanj 2024

Leo Muzica

0069090108

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Prijediplomski sveučilišni studij elektrotehnike

Završni rad

**Fotonaponski sustav s mogućnošću dvo-osnog praćenja položaja
Sunca temeljen na Arduino platformi**

Mentor: doc. dr. sc. Ivan Volarić

Rijeka, srpanj 2024

Leo Muzica

0069090108

Rijeka, 05.03.2024.

Zavod: Zavod za automatiku i elektroniku
Predmet: Elementi automatizacije postrojenja

ZADATAK ZA ZAVRŠNI RAD

Pristupnik: **Leo Muzica (0069090108)**
Studij: Sveučilišni prijediplomski studij elektrotehnike (1030)

Zadatak: **Fotonaponski sustav s mogućnošću dvo-osnog praćenja položaja sunca temeljen na Arduini platformi / Arduino based photovoltaic system for dual-axis solar tracking**

Opis zadatka:

U sklopu završnog rada potrebno je implementirati fotonaponski sustav koji ima mogućnost dvo-osnog praćenja položaja sunca u svrhu postizanja veće učinkovitosti solarnih panela. Sustav ima dva načina rada: (1) dvo-osno praćenje položaja sunca, te (2) u fiksnom položaju. Upravljanje sustavom potrebno je ostvariti preko Arduino razvojne pločice, koja će s odgovarajućom programskom podrškom upravljati s dva servo motora, na temelju informacija dobivenih iz četiri foto-otpornika koji mjere intenzitet svjetlosti. Također, potrebno je provesti eksperimentalnu provjeru sustava u kojoj je potrebno usporediti učinkovitost između prethodno opisana dva načina rada.

Rad mora biti napisan prema Uputama za pisanja diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Zadatak uručen pristupniku: 20.03.2024.

Mentor:
doc. dr. sc. Ivan Volarić

Predsjednik povjerenstva za
završni ispit:
prof. dr. sc. Dubravko Franković

IZJAVA

Sukladno članku 8. Pravilnika o završnom radu, završnom ispitu i završetku prijediplomskih sveučilišnih studija/stručnih studija Tehničkog fakulteta Sveučilišta u Rijeci, izjavljujem da sam samostalno izradio/izradila završni rad prema zadatku preuzetom dana 20. ožujka 2024.

Rijeka, srpanj 2024.

Leo Muzica

ZAHVALA

Zahvalio bih se mentoru doc. dr. sc. Ivanu Volariću na pomoći tijekom izrade i pisanja završnog rada. Također, zahvaljujem roditeljima, sestri i prijateljima na ukazanom strpljenju i podršci tijekom cijelog obrazovanja.

SADRŽAJ

1.	UVOD	1
2.	ARDUINO	2
2.1.	Hardver	2
2.2.	Softver	4
3.	SERVO MOTOR	5
4.	FOTOOTPORNIK	6
5.	SOLARNI TRAGAČI	8
5.1.	Solarni paneli	8
5.2.	Fiksni solarni paneli	10
5.3.	Jednoosni solarni tragači	11
5.4.	Dvoosni solarni tragač	12
6.	RTC MODUL PCF85063A	13
7.	MODUL ZA ČITANJE SD KARTICE	14
8.	IZRADA KUĆIŠTA	15
8.1.	Modeliranje dijelova	15
8.2.	Priprema dijelova za printanje	16
8.3.	3D printanje	17
9.	SHEMA SPOJA	18
10.	OPIS PROGRAMSKOG KODA	20
11.	USPOREDBA UČINKOVITOSTI PRI FISKONOM TE DVOOSNOM NAČINU RADA FOTONAPONSKOG SUSTAVA	24
11.1.	Mjerenje napona	24
11.2.	Shema spoja za mjerenje napona	25
11.3.	Opis programskog koda	26
11.4.	Provođenje eksperimenta	30

10. ZAKLJUČAK.....	33
LITERATURA.....	34
POPIS SLIKA.....	36
POPIS TABLICA.....	37
SAŽETAK.....	38
ABSTRACT.....	39
DODATAK.....	40

1. UVOD

U posljednjih stotinjak godina, zbog naglog razvoja industrije, dolazi do povećanih emisija CO₂ uslijed korištenja fosilnih goriva, čime dolazi do globalnog zatopljenja. Upravo iz tog razloga u proteklih dvadesetak godina sve se više istražuje i primjenjuje tehnologija dobivanja električne energije iz obnovljivih izvora energije. Obnovljivi izvori energije su izvori energije koji se mogu iskorištavati bez ograničenja, kao što energija Sunca, vode i vjetra.

Sunčeva energija pretvara se u električnu pomoću fotonaponskih sustava. Princip rada fotonaponskog sustava temelji se na fotonaponskom efektu koji se odvija unutar solarnih ćelija. Za postizanje veće učinkovitosti fotonaponskih sustava razvijeni su solarni tragači. Ideja solarnih tragača je da se praćenjem položaja Sunca, solarni paneli pomoću mehaničkog sustava pomiču na način da se paneli uvijek izlažu zrakama sunca pod optimalnim kutom. Po konstrukciji razlikujemo jednoosne i dvoosne solarne tragače.

U ovome završnom radu pomoću Arduino razvojne platforme izrađen je fotonaponski sustav koji ima dva načina rada. U prvome načinu rada solarni panel je fiksiran te nema mogućnosti praćenja položaja Sunca. U drugome načinu rada implementiran je dvo-osni način rada što znači da će se solarni panel pomicati u skladu s promjenom položaja Sunca.

Zaključno, proveden je eksperiment u kojemu je uspoređena učinkovitost fiksno namještenog fotonaponskog sustava sa sustavom koji ima mogućnost praćenja položaja Sunca.

2. ARDUINO

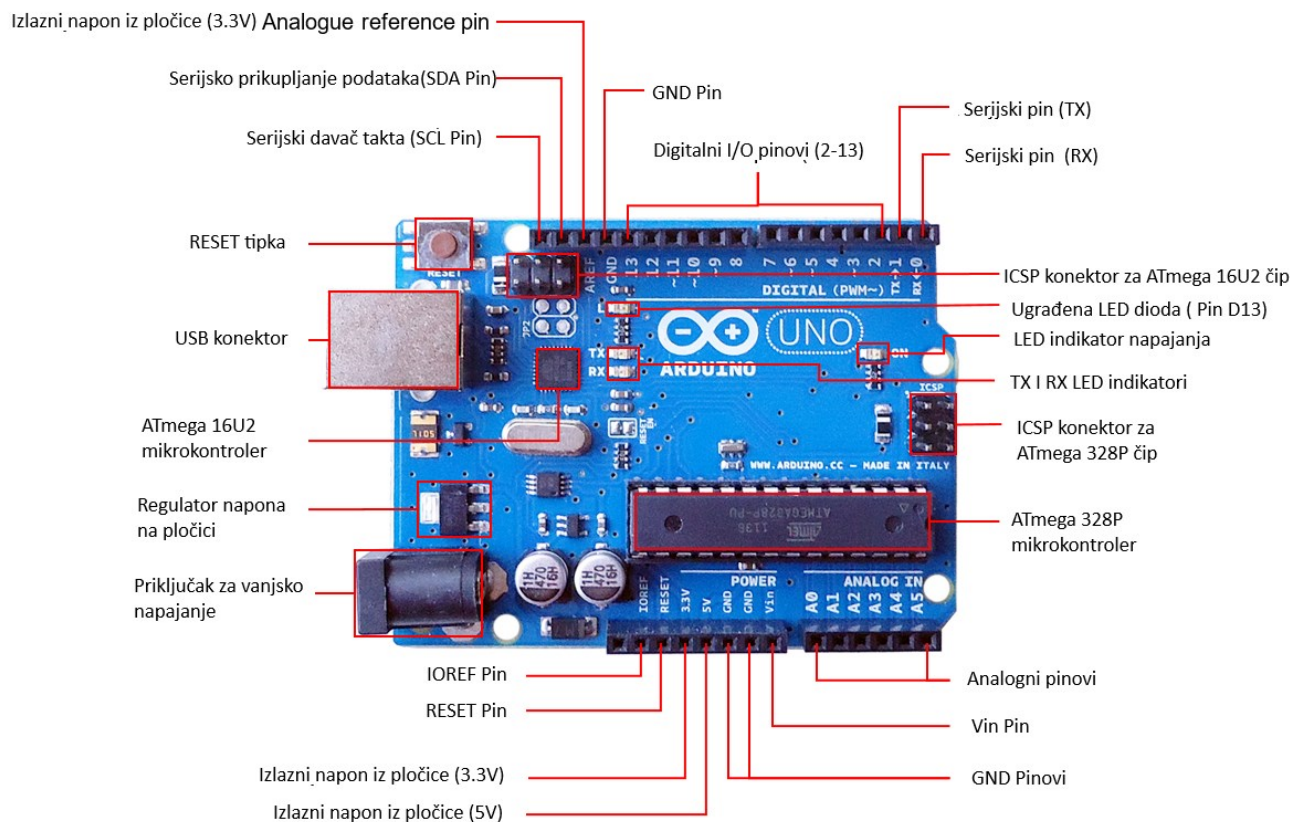
Arduino je platforma otvorenog koda koja je se koristi za izradu elektroničkih projekata. U početku se razvio kao jednostavni alat za svrhu lake izrade mnogih projekata bez prethodnog predznanja o elektronici i programiranju. Tokom godina se zbog svoje niske cijene, jednostavnosti programiranja, šarolikog izbora razvojnih pločica te dodatnih modula postao neizostavan alat pri izradi DIY projekata vezanih uz robotiku i automatizaciju.

Sama Arduino platforma se sastoji od fizičkog dijela kojeg čini razvojna pločica temeljena na mikrokontroleru te softvera Arduino IDE (eng. *Integrated Development Enviroment*) koji se koristi za programiranje mikrokontrolera. Pri izradi fotonaponskog sustava, kao i za prikupljanje podataka prilikom provođenja eksperimenta korištene su dvije Arduino UNO razvojne pločice.

2.1. Hardver

Arduino UNO temeljen je na ATmega328P mikrokontroleru. Razvojna pločica ima 14 digitalnih pinova (od kojih se 6 može koristiti kao PWM (eng. *Pulse-Width Modulation*) izlaz), 6 analognih ulaza, nekoliko pinova za napajanje pločice te nekoliko pinova koji služe za komunikaciju pločice i spojenih modula ili računala.

Digitalni pinovi 0 i 1 (Rx i Tx) služe za serijsku komunikaciju između pločice i spojenog računala te ih u pravilu nije preporučeno koristiti u ostale svrhe. Pinovi SCL (eng. *Serial Clock*) i SDA (eng. *Serial Data*) služe za I2C (eng. *Inter-Integrated Circuit*) komunikaciju između mikrokontrolera i spojenih modula. Pomoću AREF (eng. *Analog reference*) pina odabire se interni referentni napon AD (eng. *Analog-Digital*) pretvornika. Pin RESET služi za resetiranje mikrokontrolera. Pomoću IOREF (eng. *Input/Output reference*) pina konfiguriramo napon kojeg mikrokontroler očekuje na digitalnim pinovima. Kada USB (eng. *Universal Serial Bus*) konektor nije dostupan za spajanje mikrokontrolera na računalo koristi se ICSP (eng. *In circuit serial programming*) konektor. Razvojna pločica prihvaća napone napajanja u rasponu od 6 do 20V, dok je preporučeni napon napajanja u rasponu od 7 do 12V. Pomoću ugrađenih naponskih regulatora ulazni napon se stabilizira na 5 i 3.3V koji potom služe za napajanje mikrokontrolera i ostalih spojenih modula. Napajanje možemo ostvariti preko USB priključka, DC priključka namijenjenog za napajanje ili pak preko vanjskog napona napajanja preko Vin pina. Detaljan prikaz i specifikacije Arduino UNO pločice prikazani su na Slici 2.1 i Tablici 2.1.



Slika 2.1. Arduino UNO R3 pločica [3].

Tablica 2.1. Specifikacije Arduino UNO pločice [1].

Naziv pločice:	Arduino UNO R3
Mikrokontroler:	ATmega328P
USB priključak:	USB-B
Broj digitalnih I/O pinova:	14
Broj Analognih ulaznih pinova:	6
Broj pinova s PWM mogućnošću:	6
UART komunikacija	Da
I2C komunikacija	Da
SPI komunikacija	Da
Napon I/O pinova:	5V
Napon napajanja:	7-12 V
Dopuštena maksimalna struja za svaki I/O pin:	20 mA
Mogućnost napajanja:	USB, DC konektor, Vin pin
Radni takt ATmega 328P:	16 MHz
Memorija:	2KB SRAM, 32KB FLASH, 1KB EEPROM

2.2. Softver

Za programiranje mikrokontrolera možemo koristiti bilo koji IDE (eng. *Integrated Development Environment*). Najčešće se koristi sučelje Arduino IDE koje je besplatno za korištenje te se može preuzeti sa službene stranice Arduina. Arduino IDE se sastoji od tekstualnog okvira za pisanje samog koda, konzole, te alatne trake. Programski jezik kojim programiramo mikrokontroler je varijanta C++.

Svaki programski kod mora sadržavati minimalno dvije funkcije: `void setup()` te `void loop()`. Na samom početku programa, izvan te dvije funkcije, potrebno je uključiti sve korištene biblioteke, te definirati sve globalne varijable. Funkcija `void setup()` poziva se samo jednom, na samom početku rada pločice ili nakon njenog reseta. Ona se uobičajeno koristi za konfiguriranje ulaznih i izlaznih pinova, inicijalizaciju varijabli, definiranje početnih stanja spojenih pinova te sličnih inicijalizacija. Funkcija `void loop()` se poziva nakon što se funkcija `void setup()` izvrši, te se taj dio koda neprestano izvršava dokle god je mikrokontroler priključen na izvor napajanja. Iako ove dvije funkcije čine jezgru programskog koda, također možemo napisati kod podijeliti u zasebne funkcije koje možemo jednostavno pozvati u funkciji `loop()`. U nastavku je izlistan primjer kod kojeg pomoću potenciometra upravljamo pomakom servo motora.

```
#include <Servo.h>      //Pozvana biblioteka
Servo myservo;        // Definiran objekt klase Servo

int potpin = A0;      // analogni pin  na koji je spojen potenciometar
int val;             // Varijabla u koju se sprema ocitanje sa analognog ulaza
const int ledPin = 13; // definiran pin na koji je spojena LED dioda
const int servoPin = 9; // definiran pin na koji je spojen servo motor

void setup() {        // Dio koda koji se samo jednom izvršava
  myservo.attach(servoPin); // Konfiguracija korištenih pinova
  pinMode(ledPin, OUTPUT);
}
void loop() {         // Dio koda koji se neprestano izvršava
  val = analogRead(potpin); // Ocitava vrijednost sa potenciometra -->
<0, 1023>
  val = map(val, 0, 1023, 0, 180); // sakliramo raspon analognog ulaza u kutni
pomak motora
  myservo.write(val); // Pomoćemo motor za određeni pomak
potenciometra
  delay(500); // čekaj 500 ms pa izvrši ponovno
}
```

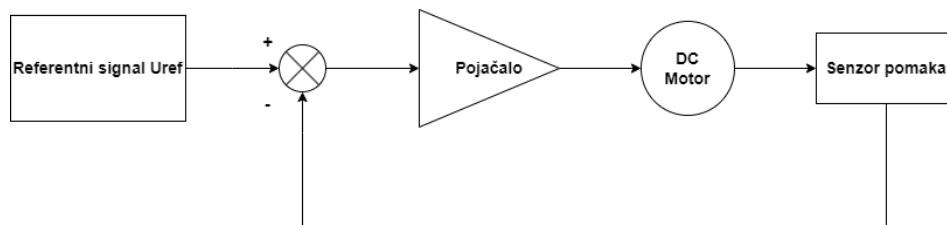
3. SERVO MOTOR

Servo motor je posebna vrsta motora koji se koristi za ostvarivanje preciznog pomaka. Sastoji se od običnog istosmjernog motora čiji je položaj osovine upravljan povratnom vezom. Svaki servo motor se sastoji od istosmjernog motora, upravljačkog kruga te mehanizma povratne veze. Na osovini istosmjernog motora spojen je sustav reduktora čija je uloga smanjenje izlazne brzine vrtnje te povećanja momenta motora. Uloga upravljačkog kruga je da upravlja kutnim pomakom motora, dok je uloga povratne veze kontrola konačnog položaja. Na slici 3.1 prikazan je servo motor korišten u završnom radu.



Slika 3.1. MG995 Servo motor [4].

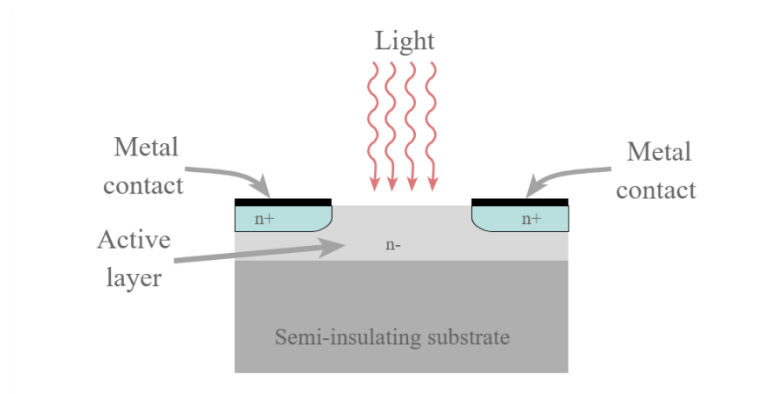
Blokovska shema servo motora dana je na slici 3.2. Referentni signal kojeg generiramo pomoću mikrokontrolera predstavlja željeni pomak za koji se pomiče osovina motora. Referentni signal realizira se pomoću PWM modulacije što znači da se kutni pomak regulira pomoću trajanja impulsa. Pomak servo motora mjeri se pomoću senzora koji kutni pomak pretvara u mjereni signal koji se preko povratne veze dovodi na sumator. Izlaz iz sumatora predstavlja tzv. signal pogreške koji se definira kao razlika između referentnog i mjenog signala na izlazu. Ukoliko je signal greške različit od nule ostvareni pomak motora nije jednak zadanom, a kada je signal greške jednak nuli, motor se prestaje okretati, te je ostvaren željeni pomak.



Slika 3.2. Blokovska shema servo motora.

4. FOTOOTPORNIK

Fotootpornik ili LDR (eng. *Light-dependent resistor*) je poluvodička komponenta čiji se otpor smanjuje porastom osvijetljena. Kada se fotootpornik osvijetli izvorom svjetlosti, elektroni iz valentnog pojasa poluvodičkog materijala apsorbiraju energiju emitiranih fotona. Kada poprimе dovoljno energije, prelaze u vodljivi pojas čime se smanjuje otpor fotootpornika. Poprečni presjek i izgled fotootpornika dani su na slikama 4.1 te 4.2.



Slika 4.1. Poprečni presjek fotootpornika [6].

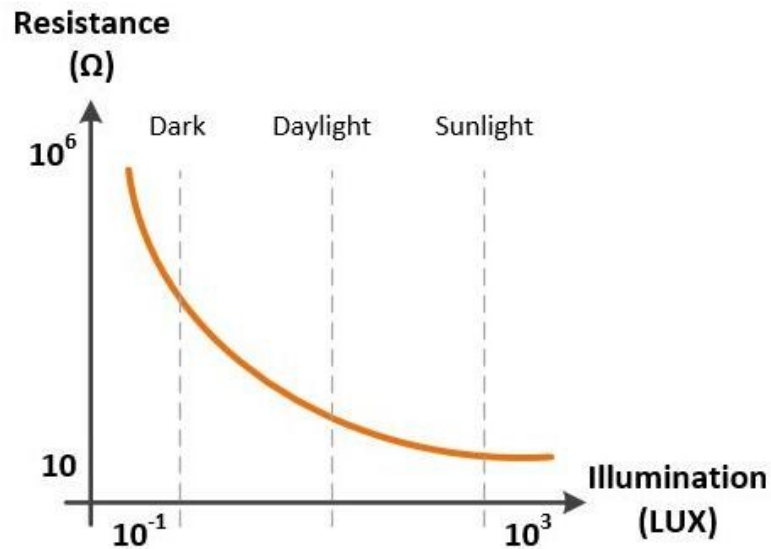


Slika 4.2. Fotootpornik [7].

Ovisno o poluvodičkom materijalu od kojeg su napravljeni, fotootpornici se dijele u dvije skupine: intrinzične i ekstrinzične fotootpornike. Intrinzični fotootpornici su izrađeni od intrinzičnih poluvodičkih materijala poput silicija ili germanija. Takvi materijali su pravilne strukture te u vodljivom pojasu nemaju slobodnih elektrona, dok im se valentni pojas sastoji isključivo od šupljina. Kako bi došlo do prelaska elektrona iz valentnog u vodljivi pojas, potrebno ih je pobuditi s većom količinom energije. Stoga se intrinzični poluvodiči koriste pri detekciji svijetla visokih frekvencija. S druge strane, ekstrinzični fotootpornici proizvode se tako da se intrinzični poluvodič dopira sa primjesama. Rezultat toga je poluvodički materijal s viškom elektrona u valentnom pojasu. Upravo zbog toga potrebno je manje emitiranih fotona nego kod intrinzičnih fotootpornika kako bi elektroni prešli iz valentnog u vodljivi pojas. Ekstrinzični poluvodiči se koriste pri detekciji svijetla nižih frekvencija.

Na slici 4.3 prikazana je karakteristika fotootpornika. Dva glavna parametra fotootpornika su: tzv. mračni otpor (eng. *Dark resistance*) te svijetli otpor (eng. *Light resistance*). Mračni otpor definira

se kao otpor kojeg će komponenta imati 10 sekundi nakon što se ukloni izvor svjetla intenziteta 10 Lux. Mračni otpor predstavlja otpor fotootpornika kada se nalazi u potpunom mraku, te može iznositi i do nekoliko desetaka MΩ. S druge strane, tzv. svijetli otpor definira se kao otpor kojeg će komponenta imati nakon što se izloži izvoru svjetla intenziteta 400 – 600 Lux u trajanju od dva sata te nakon toga intenzitetu svjetla od 10 Lux.



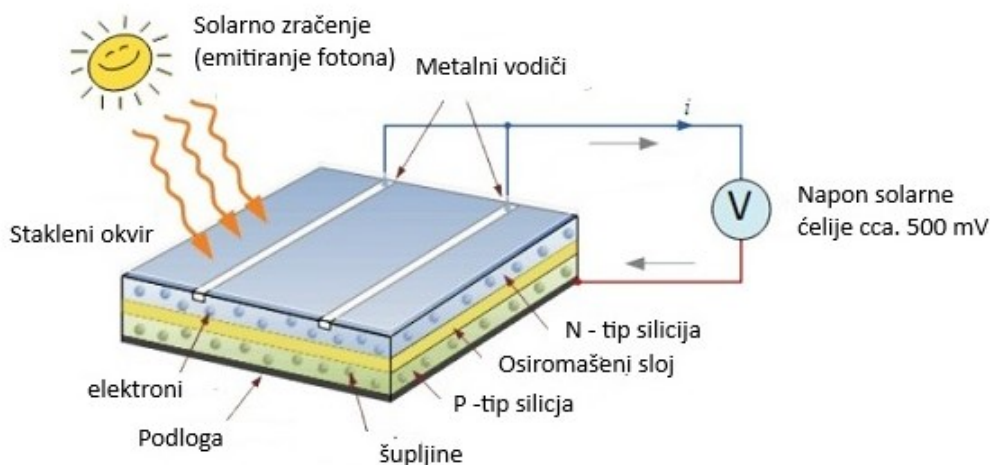
Slika 4.3. Karakteristika fotootpornika [8].

Zbog svojstava da im se otpor mijenja promjenom svjetlosti, te nelinearne karakteristike, fotootpornici se najčešće koriste za detekciju svjetla ili za mjerenje intenziteta zračenja svjetla. U ovome završnom radu fotootpornici su korišteni kako bi se pratila promjena položaja Sunca.

5. SOLARNI TRAGAČI

5.1. Solarni paneli

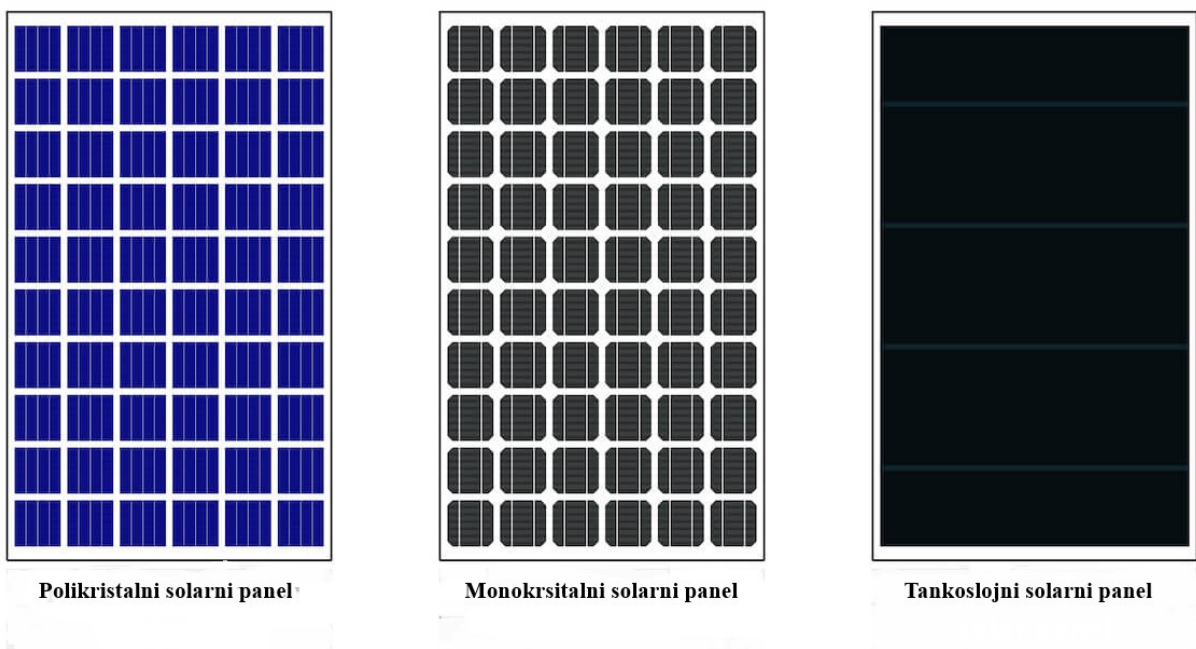
Uloga solarnih panela je pretvorba solarne u električnu energiju. Solarni panel se sastoji od više manjih fotonaponskih ćelija. Fotonaponska ćelija je izrađena od poluvodičkog materijala kao što je silicij. Princip rada fotonaponskih ćelija temelji se na fotonaponskom efektu, prikazan na slici 5.1. Tanka poluvodička pločica je obrađena na način kako bi se unutar pločice stvorilo električno polje. Sunčeva svjetlost koja pada na površinu panela emitira energiju koju prenose fotoni. Kada foton udari u površinu fotonaponske ćelije dolazi do izbijanja elektrona iz valentnog u vodljivi pojas. Kada se tako slobodni elektroni nađu u električnom polju, usmjerenom se kreću od nižeg prema višem potencijalu, što se manifestira kao protok struje. Svaka solarna ćelija, neovisno o veličini može proizvesti najveći napon od 500 mV , dok iznos struje ovisi o veličini kao i o konstrukciji solarne ćelije. Kako bi se postigao veći izlazni napon i snaga u praksi se više solarnih ćelija spaja serijski.



Slika 5.1. Princip rada solarne ćelije [9].

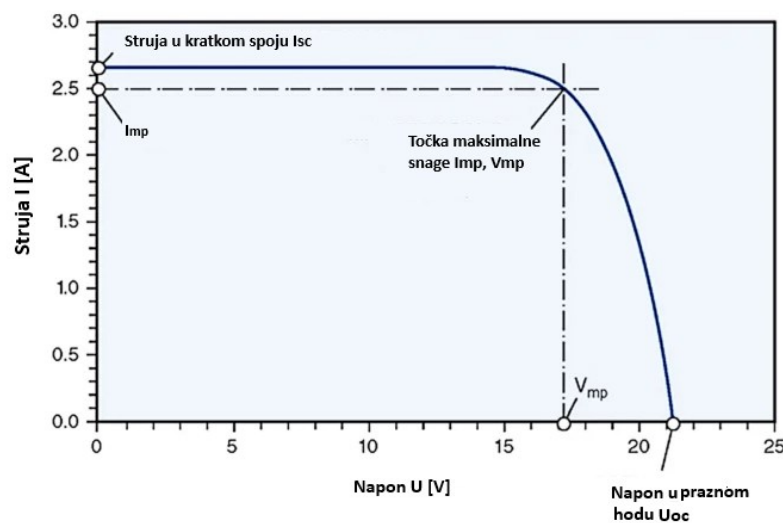
Ovisno o izvedbi, razlikujemo monokristalnu, polikristalnu te tankoslojnu izvedbu. Kod monokristalne izvedbe, silicij se oblikuje u šipke, reže na pločice, te se slaže u karakterističan uzorak monokristalnih solarnih panela. Učinkovitost monokristalnih solarnih panela se kreće između 15% i 20%. S druge strane, solarni paneli s polikristalnom izvedbom u pravilu imaju nešto nižu učinkovitost koja se kreće između 13% i 16%, no njihova prednost je niža cijena. Polikristalni solarni panel se također proizvode od silicija. Razlika je u tome što se umjesto korištenja jednog kristala silicija, u

postupku proizvodnje koristi više fragmenata silicija. Kao rezultat korištenja više fragmenata silicija, polikristalni solarni paneli imaju nešto nižu učinkovitost od monokristalnih solarnih panela koja se kreće između 3% i 16%. U novije vrijeme došlo je do razvoja tankoslojne izvedbe solarnih panela. Tankoslojni solarni paneli su tanji i fleksibilniji. U usporedbi s monokristalnim i polikristalnim solarnim panelima, tankoslojni solarni paneli u pravilu imaju nižu učinkovitost i snagu. Zbog niže učinkovitosti obično se ne koriste za potrebe kućanstva, međutim zbog male težine i fleksibilnosti koriste se u većim projektima poput komercijalnih zgrada. Navedene izvedbe solarnih panela prikazane su na slici 5.2. Izbor izvedbe solarnih panela ovisi o zahtjevima koje moramo zadovoljiti. Ako se traži najniža cijena, najbolji izbor su polikristalni solarni paneli, no ako se traži veća učinkovitost najbolji izbor su monokristalni solarni paneli.



Slika 5.2. Vrste solarnih panela [10].

Na slici 5.3 prikazana je U-I karakteristika solarnog panela. Napon u praznom hodu U_{OC} je maksimalni napon na stezaljkama panela kada nije spojen na trošilo. Struja u kratkom spoju I_{SC} je maksimalna struja koju ćelija može dati kada je kratko spojena. U točki kratkog spoja, izlazni napon, pa time i izlazna snaga su jednake nuli. Struja kratkog spoja je vrlo važan parametar pri projektiranju fotonaponskih sustava jer nam govori koliku maksimalnu struju sustav treba podnijeti prije nego što se strujna zaštita uklopi. Točka maksimalne snage se naziva radnom točkom solarnog panela.



Slika 5.3. U-I Karakteristika solarnog panela [11].

5.2 Fiksni solarni paneli

Fiksni solarni paneli montiraju se u određenom položaju. Kako bi fiksni solarni panel postigao maksimalnu učinkovitost, mora biti postavljen u optimalni položaj. Postoje dva čimbenika koji utječu na učinkovitost solarnih panela a to su: orijentacija i kut solarnog panela.

Orijentacija solarnog panela je zapravo strana svijeta prema kojoj je panel okrenut. Kako bi postigli maksimalnu učinkovitost solarnog panela, poželjna je orijentacija prema pravom jugu. Solarni paneli

koji su montirani prema pravom istoku ili zapadu, proizvoditi će oko 20 % manje električne energije od panela koji su postavljeni prema pravom jugu.

S druge strane, optimalni nagib solarnog panela određuje se prema zemljopisnoj širini lokacije na kojoj je solarni panel montiran. Optimalni nagib računa se na način da tokom ljetnih mjeseci zemljopisnoj širini oduzmemo 15° , dok tokom zimskih mjeseci zemljopisnoj širini na kojoj je nalazi panel dodamo 15° . Dobiveni kut je tada kut pri kojem će se u teoriji ostvariti maksimalna učinkovitost solarnog panela. Kako bi se izbjeglo mijenjanje nagiba panela, u praksi se za nagib solarnog panela uzima proizvoljni kut između 30° i 45° .

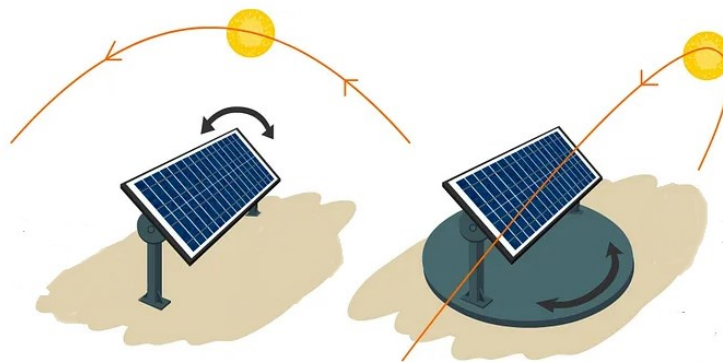
Najčešća primjena fiksnih solarnih panela je ugradnja na krovove kuće za generiranje električne energije u kućanstvu. Primjer fiksno postavljenog panela prikazan je na slici 5.4.



Slika 5.4. Fiksno pozicioniran solarni panel na krovu kuće [12].

5.3. Jednoosni solarni tragači

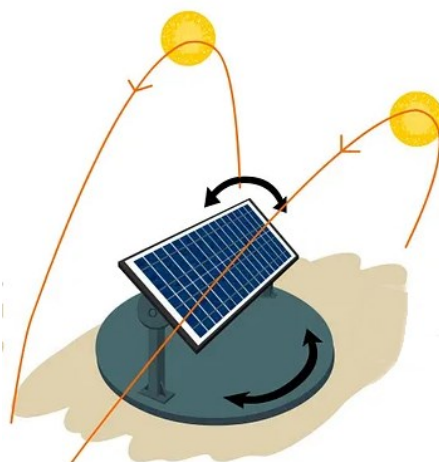
Jednoosni solarni tragači imaju jedan stupanj slobode gibanja, što znači da solarni panel ima mogućnost rotiranja samo po jednoj osi. Pri ovakvoj izvedbi najčešći slučaj je da se solarni panel postavi pod neki nagib te onda cijeli sustav rotira s obzirom na podlogu. Takva izvedba naziva se vodoravni jednoosni tragač (eng. *Horizontal single axis tracker - HSAT*). Princip rada jednoosnog solarnog tragača prikazan je na slici 5.5.



Slika 5.5. Princip rada jednoosnog solarnog tragača [13].

5.4. Dvoosni solarni tragač

Dvoosni solarni tragači imaju dva stupnja slobode, što znači da solarni panel ima mogućnost rotiranja po dvije osi. S obzirom na to da ovakvi tragači prate položaj Sunca u vertikalnom i horizontalnom smjeru, implementacijom ovog sustava postiže se maksimalna moguća učinkovitost solarnih panela, kao što je eksperimentalno pokazano i u ovom završnom radu, usporedbom učinkovitosti solarnih panela u fiksnom položaju te pri implementiranom dvoosnom režimu rada. Princip rada dvoosnog solarnog tragača prikazan je na slici 5.6.



Slika 5.6. Princip rada dvoosnog solarnog tragača [13].

6. RTC MODUL PCF85063A

RTC (eng. *Real-time clock*) je modul koji se koristi za apsolutno praćenje vremena, što mikrokontroler nije u mogućnosti kada izgubi napajanje, stoga, ovakvi moduli imaju bateriju koja osigurava kontinuiranu informaciju o točnom vremenu. Postoji više varijanti RTC modula, no u ovome završnom radu korišten je PCF85063A modul. Korišteni modul dizajniran je za iznimno malu snagu što omogućuje dugotrajan rad. Osim praćenja vremena modul također ima ugrađenu alarm funkciju koja se koristi kako bi se modul pobudio iz stanja mirovanja. Modul je korišten kako bi pri mjerenju napona solarnog panela pratio vrijeme, te svakih 10 minuta spremio izmjerenu vrijednost napona na SD karticu. U nastavku je izlistan primjer korištenja RTC modula za praćenje trenutnog vremena.

```
#include <LiquidCrystal.h>           // Biblioteka za upravljanje LCD zaslonom
#include "PCF85063A-SOLDERED.h"     // Biblioteka za upravljanje RTC modulom
PCF85063A rtc;
LiquidCrystal lcd(1, 2, 4, 5, 6, 7); // LCD pin konfiguracija (rs, enable, d4,
d5, d6, d7)
int minute = 0;
int sekunde = 0;
int sat = 0;
String vrijeme = "";
String datum = "";

void setup(){
  Wire.begin();
  Serial.begin(9600);
  rtc.setTime(7, 0, 0);           // Postavljanje početnog vremena
  rtc.setDate(6, 17, 6, 2024);   // Postavljanje početnog datuma
  rtc.begin();                  // Inicijalizacija rtc modula
  lcd.begin(16,2);              // Incijalizacija LCD modula
}

void loop(){
  sekunde = rtc.getSecond();     // Ocitaj vrijeme s modula
  minute = rtc.getMinute();
  sat = getHour();
  datum = String(rtc.getMonth()) + "/" + String(rtc.getDay()) + "/" +
String(rtc.getYear());
  vrijeme = String(rtc.getHour()) + ":" + String(rtc.getMinute()) + ":" +
String(rtc.getSecond()) + ".";
  Serial.println(vrijeme);
  //Ispiši trenutni datum i vrijeme na LCD zaslon
  lcd.setCursor(0,0);
  lcd.print("Vrijeme: ");
  lcd.print(vrijeme);
  lcd.setCursor(0,1);
  lcd.print("Datum: ");
  lcd.print(datum);
  delay(1000);
}
```

7. MODUL ZA ČITANJE SD KARTICE

Modul za čitanje SD kartice najčešće se primjenjuje u projektima gdje je potrebno spremati podatke sa senzora, što je slučaj i u ovom završnom radu, gdje je korišten za spremanje vrijednosti mjenog napona u određenim vremenskim trenucima. Čitač SD kartice za komunikaciju s mikrokontrolerom koristi SPI komunikaciju (eng. *Serial peripheral Interface*). Modul ima ukupno 5 pinova: GND, VCC, MISO (eng. *Master In Slave Out*), MOSI (eng. *Master Out Slave In*), SCK (eng. *Serial Clock*) te CS (eng. *Chip Select*). GND i VCC pinovi koriste se za napajanje modula koji može raditi na 3,3 ili 5V. Pin CS koristi se kako bi se omogućila komunikacija između modula i mikrokontrolera, SCK pin koristi se za sinkronizaciju radnog takta modula s radnim taktom mikrokontrolera, dok MISO i MOSI pinovi služe za razmjenu podataka između modula i mikrokontrolera. U nastavku je izlistan primjer otvaranja i uređivanja datoteke na SD kartici pomoću Arduino UNO pločice.

```
#include <SPI.h>      // Biblioteka za korištenje SPI komunikacije
#include <SD.h>       // Biblioteka za korištenje SD modula

File myFile;         // Kreiranje objekta klase File
const int CS_pin = 10; // Chip select pin na mikrokontroleru

void setup() {

    Serial.begin(9600); // Inicijalizacija Serijske komunikacije
    Serial.print("Initializing SD card...");

    if (!SD.begin(CS_pin)) { // Ako CS pin nije povezan
        Serial.println("Neuspješno! "); // Korištenje modula onemogućeno
    }
    Serial.println("Uspješno!");
    myFile = SD.open("test.txt", FILE_WRITE); // Otvori datoteku

    if (myFile) { // Ako je datoteka dostupna izvrši p
        Serial.print("Tresetno otvaranje: ");
        Serial.println("Brojevi 1, 2, 3");
        myFile.println("Brojevi 1, 2, 3");
        myFile.close(); // Zatvori datoteku
        Serial.println("Spremljeno!");
    } else {
        Serial.println("Greška pri otvaranju test.txt");
    }
    delay(1000);
}

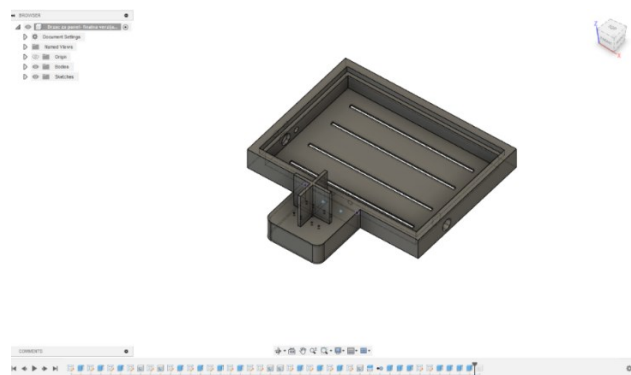
void loop() {

}
```

8. IZRADA KUĆIŠTA

8.1. Modeliranje dijelova

Kućište sustava je izrađeno pomoću 3D printanih dijelova koji su modelirani u programskom alatu Fusion 360. Fusion 360 je programski alat za komercijalno računalno potpomognuti dizajn CAD (eng. *Computer aided design*), računalno potpomognutu proizvodnju CAM (eng. *Computer aided manufactory*), računalno potpomognuto inženjerstvo CAE (eng. *Computer aided engineering*) te dizajni tiskanih pločica PCB (eng. *Printed circuit boards*) kojeg je razvio Autodesk. Sam program podržava razne mogućnosti poput 3D modeliranja te upravljanja proizvodnim procesima poput tokarenja, glodanja, strojne obrade i dr. Na slici 8.1 prikazan je model držača solarnog panela, dok je na slici 8.2 prikazana maketa solarnog tragača sastavljena od modeliranih dijelova.



Slika 8.1. Držač solarnog panela modeliran u Fusion 360.

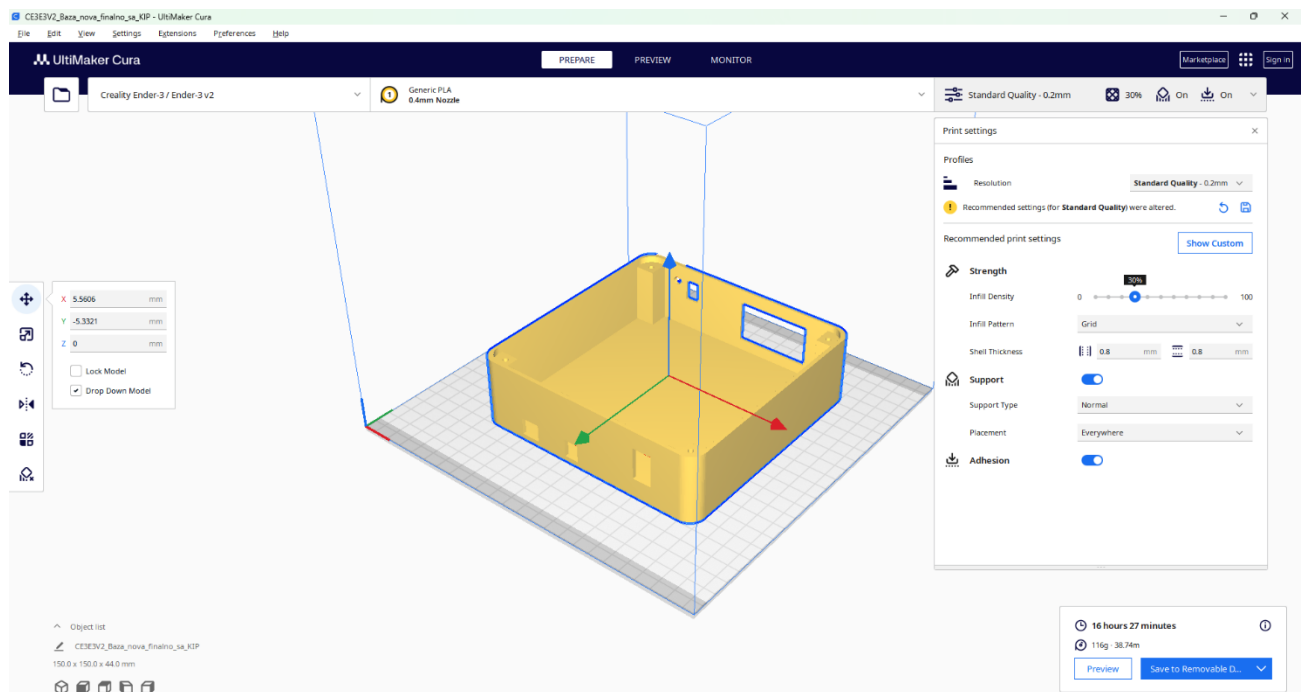


Slika 8.2. Maketa solarnog tragača.

8.2. Priprema dijelova za printanje

Prije samog printanja izrađenih modela, potrebno ih je pripremiti u programu za rezanje 3D ispisa. Program za rezanje 3D ispisa u osnovi djeluje kao posrednik između 3D modela i printera. Djeluje na način da se model kojeg želimo printati razdvoji na više slojeva, koje pisač kasnije printa jednog po jednog. Potom se takav model pretvara u G-kod koji 3D pisač može interpretirati. Jedan od najčešće korištenih programa za rezanje je Ultimaker Cura.

Ultimaker Cura je program otvorenog koda koji osim raščlanjivanja po slojevima ima mogućnost podešavanja postavka 3D printera kao što su brzina printanja, kvalitete printanja, postotak popune modela, mogućnost odabira opcije potpore (za printanje visokih predmeta), mogućnost odabira opcije adhezije (između printanog modela i podloge printa se dodatni sloj kako bi se izbjeglo lijepljenje isprintanog predmeta za podlogu) te mnoge druge. Sučelje programa prikazano je na slici 8.3. Ono što čini Curu vrlo popularnom je činjenica da je zamišljena kao univerzalni alat koji podržava više brendova 3D pisača, a ne samo jednog.



Slika 8.3 Izgled sučelja programa Ultimaker Cura.

8.3. 3D printanje

Za potrebe ovog završnog rada korišten je Ender 3 pisač. Pisač koristi FDM (eng. *Fused Deposition Modeling*) tehnologiju. Princip rada FDM tehnologije temelji se na dovođenju filameta do mlaznice koja je zagrijana na temperaturu pri kojoj se filament omekša te se nanosi na prethodno zagrijanu podlogu. Jedna od karakteristika ovog pisača je mogućnost pomicanja mlaznice po X i Z osi, dok se podloga pomiče samo po Y osi. Radni prostor je dimenzija 220x220x250mm što je dovoljno za većinu rekreativnih projekata. Prilikom printanja korišten je PLA (eng. *Polyactic Acid*) filament koji se zbog pristupačne cijene te zadovoljavajuće kvalitete najčešće koristi. Osim PLA filameta, Ender 3 printer podržava printanje i pomoću ABS (eng. *Acrylonitrile Butadiene Styrene*) filamenata, ugljičnih vlakana i dr. Tokom samog printanja proces se može nadzirati preko upravljačkog ekrana pomoću kojeg se može postaviti pozicija radne podloge i mlaznice, temperatura radne podloge, temperatura mlaznice, itd. Printer Ender 3 prikazan je na slici 8.4, a na slici 8.5 sastavljeno kućište.



Slika 8.5 Ender 3 3D printer [14].



Slika 8.4 Sastavljeno kućište.

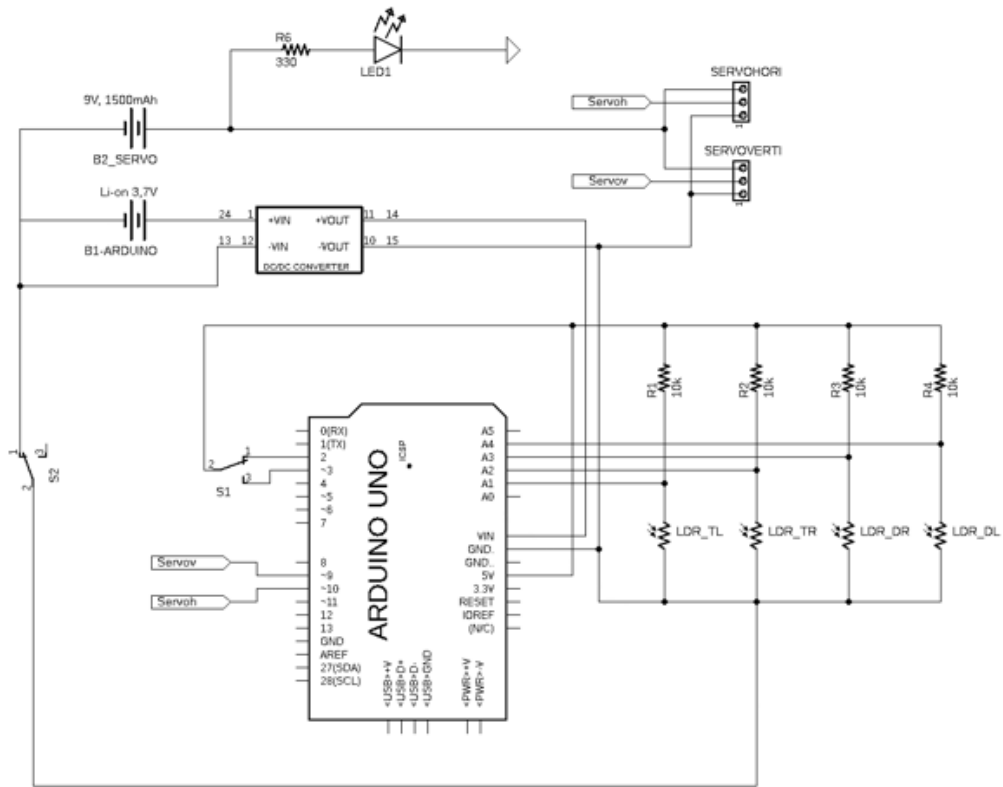
9. SHEMA SPOJA

Schema spajanja komponenti izrađena je u programskom paketu Eagle CAD. Eagle CAD je EDA programski paket (eng. *Electronic design automation*) koji služi za jednostavnu izradu PCB pločica. Shema spoja i fizička izvedba sustava prikazane su na slikama 9.1 i 9.2.

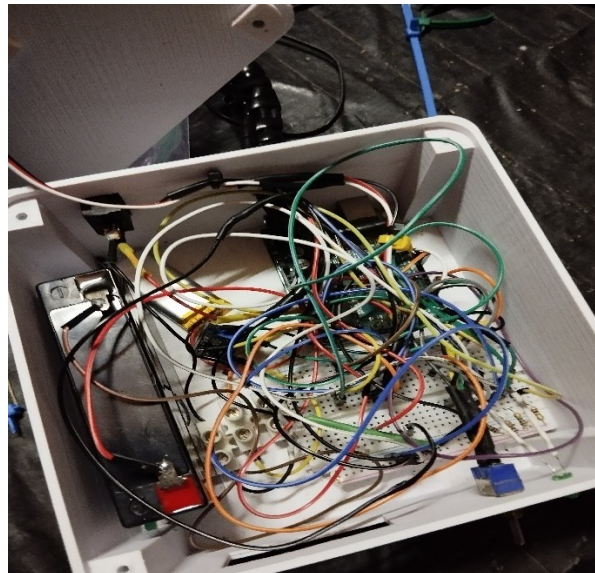
Cijeli sklop napaja se pomoću dva neovisna izvora napajanja. Kao napajanje B1-ARDUINO korištena je Li-ion baterija koja je spojena na pretvarač napona čija je uloga podizanja izlaznog napona na 7V što odgovara radnome naponu Arduina kada se napaja preko Vin pina. Kao izvor napona B2_SERVO korišten je olovni akumulator koji služi za napajane oba servo motora, kao i LED indikatora. Servo motori SERVOHORI te SERVOVERTI služe za pomicanje solarnog panela po horizontalnoj i vertikalnoj osi. Upravljački pinovi oba motora moraju biti spojeni na Arduino pinove koji imaju mogućnost PWM modulacije. Fotootpornici LDR_TL, LDR_TR, LDR_DR te LDR_DL služe za praćenje položaja Sunca. Svaki od fotootpornika je spojen kao djelilo napona s otpornikom od 10kΩ, gdje je srednji izvod spojen na jedan od analognih pinova mikrokontrolera. Nadalje, sklopka S2 služi za uključivanje cijelog sustava, dok sklopka S1 služi za odabir načina rada sustava. Kada se sklopka S1 nalazi u položaju 1, sustav se nalazi u fiksnom načinu rada, tj. solarni panel postaviti će se u fiksni položaj te neće imati mogućnost praćenja položaja Sunca. S druge strane, kada se sklopka S1 nalazi u položaju 2, omogućeno je dvo-osnog praćenja položaja Sunca. Popis komponenti korištenih pri izradi sustava te njihova uloga dani su u tablici 9.1

Tablica 9.1. Popis i uloga korištenih komponentni.

Komponente	Uloga
Li-ion 3.7V 500mAh	Izvor napona za Arduino UNO
Olovni akumulator 9V, 1500mAh	Izvor napona za Servo motor i fotootpornike
Istosmjerni pretvarač SX1308	Podizanje izlaznog napona Li-ion baterije na radni napon Arduino pločice
Servo motor SG90M	Pomicanje solarnog panela po vertikalnoj osi
Servo motor Tower PRO MG995	Pomicanje solarnog panela po horizontalnoj osi
4 x VT-93N1, 36kΩ, 80mW	Praćenje položaja Sunca
Arduino UNO	Upravljanje servo motorima



Slika 9.1. Shema spoja.



Slika 9.2. Povezane komponente.

10. OPIS PROGRAMSKOG KODA

Na samom početku koda u program je uključena biblioteka `Servo.h` koja se koristi za upravljanje servo motorima. Potom su instancirani objekti `servohori` te `servoverti` kojima upravljamo motorima zaduženima za horizontalni i vertikalni pomak solarnog panela. Za svaki servo motor pojedinačno definirani su maksimalni i minimalni pomak kojeg je moguće ostvariti kako bi se izbjeglo neželjeno ponašanje sustava, pinove `servoh_pin` i `servov_pin` na koje su spojeni upravljački pinovi, te varijable koje će se koristiti za praćenje pomaka motora. Nakon što su definirane sve varijable vezane uz upravljanje motorima, potrebno je definirati i analogne pinove na koje su povezani fotootpornici. Završno definirani su pinovi na koje su spojeni kontakti sklopke S1 koji služe za odabir režima rada sklopa. Za definiranje pinova korišten je `const int` tip varijabli kako bi se naglasilo da se vrijednost tih varijabli ne može mijenjati, za razliku od `int` tipa čija se vrijednost može mijenjati.

Nakon deklaracije svih korištenih globalnih varijabli definirana je `void setup()` funkcija koja se izvršava samo jednom. Unutar funkcije, pomoću naredbe `Serial.begin()` pokrenuta je serijska komunikacija koja je korištena pri praćenju rada programa. Naredbom `pinMode()` konfiguriran je način rada svakog korištenog digitalnog pina kao ulaznog ili izlaznog, dok su analogni pinovi uvijek ulazni. Pinovi na koje su spojeni kontakti sklopke S1 definirani su kao ulazi sa internim *pull-up* otpornikom. Servo motorima upravljački pin je definiran metodom `attach()`, te su zatim postavljeni u početni položaj metodom `write()` s eksperimentalno određenom pozicijom.

Zatim je definirana funkcija `void loop()` koja se neprestano izvršava. Na samom početku funkcije pomoću funkcije `digitalRead()` očitana je položaj sklopke koja služi za odabir načina rada sustava, na temelju kojih je definirana vrijednost varijable `nacin_rada` koja kasnije služi kao argument `switch case` naredbe. Naredba `Switch case` omogućuje kontrolu izvršavanja programskog koda na način da uspoređuje vrijednost argumenata s definiranim vrijednostima, te ovisno o položaju sklopke izvršava se jedna od dvije funkcije: `fiks_nacin()` ili `dvoosni_nacin()`. U prvoj funkciji sustav se postavlja u unaprijed fiksno definiranu poziciju, dok je u drugom slučaju omogućeno dvoosno praćenje položaja Sunca.

U prvom slučaju, kada je odabran fiksni način rada, servo motori se postavljaju u unaprijed definirani položaj. Kada se pozove funkcija `fiks_nacin()` solarni panel se vertikalno postavlja pod

kutom od 45°, a horizontalno paralelno s kućištem. Pomaci od 135 za vertikalni servo motor i 58 za horizontalni su određeni eksperimentalno.

U drugom slučaju, kada je odabran dvoosni način rada, solarni panel uvijek prati položaj Sunca. Na samom početku, u varijable `servoh` i `servov` pomoću metode `read()` spremljene su trenutne pozicije servo motora. Potom su definirane varijable: `gore_l`, `gore_d`, `dole_l` te `dole_d` u koje se spremaju naponi na pripadajućim fotootpornicima pomoću funkcije `analogRead()`, a koje služe za praćenje položaja Sunca, te varijable `srednje_gore`, `srednje_dole`, `srednje_lijevo` te `srednje_desno` u kojima se sprema prosječna vrijednost između skupa od dva fotootpornika. Npr, u varijablu `srednje_gore` sprema se prosječan pad napona na gornjim fotootpornicima (varijable `gore_d` i `gore_l`), dok se u varijablu `srednje_desno` sprema izračunati prosjek pada napona na desno postavljenim fotootpornicima (varijable `gore_d` i `dole_d`). Pomak servo motora dobiva se usporedbom prosječne vrijednosti pada napona na skupu fotootpornika. Za upravljanje vertikalnim servo motorom uspoređuju se varijable `srednje_gore` te `srednje_dole`: za slučaj kada je varijabla `srednje_gore` veća od `srednje_dole` servo motor se pomoće prema gore, ako ne prema dolje. Ako u bilo kojem trenutku servo motor dosegne maksimalno tj. minimalno definirani pomak daljnji pomak je onemogućen. Slično upravljanju pomakom vertikalnog servo motora, za upravljanje pomakom horizontalnog servo motora uspoređuju se prosječne vrijednosti lijevo i desno postavljenih fotootpornika: za slučaj kada je varijabla `srednje_lijevo` veća od varijable `srednje_desno`, horizontalni servo motor pomiče se desno, inače u lijevo. Kao što je bio slučaj i kod vertikalnog servo motora, ako u bilo kojem trenutku horizontalni servo motor dosegne minimalni tj. maksimalni pomak, daljnji pomak biti će onemogućen. Navedeni postupak dvoosnog praćenja izvršava se sve dok sva četiri fotootpornika nisu podjedanko osvijetljena. U nastavku je izlistan programski kod izrađenog fotonaponskog sustava.

```
#include <Servo.h> // Biblioteka za upravljanje Servo motorima
Servo servohori; // Kreiranje objekta servohori
const int servoh_pin = 10;
int servoh = 0;
int servohLimitHigh = 160; // Definiranje graničnih položaja
int servohLimitLow = 20;
Servo servoverti; // Kreiranje objekta servoverti
const int servov_pin = 9;
int servov = 0;
int servovLimitHigh = 160; // Definiranje graničnih položaja
int servovLimitLow = 0;
const int LDR_DR = A3; // Definiranje pinova na koji su spojeni
const int LDR_DL = A4; // na koje su spojeni fotootpornici
const int LDR_TR = A2;
```

```

const int LDR_TL = A1;
const int pol_1 = 4; // Pinovi sklopke kojima određujemo
const int pol_2 = 3; // način rada sustava

void setup () // Izvršava se samo jednom
{
    Serial.begin(9600); // Inicijalizacija serijske komunikacije

    servoverti.attach(servov_pin); // Opredjeljuje pinove
    servohori.attach(servoh_pin);

    servohori.write(58); // Nulta pozicija s obizrom na horizontalu
    servoverti.write(135); // Nulta pozicija s obizrom na vertikalnu

    pinMode(pol_1, INPUT_PULLUP); // Definiranje opisa pina
    pinMode(pol_2, INPUT_PULLUP);

    delay(500); // Čekaj 500 ms
}

void loop(){
// Odabir načina rada
int stanje_1 = digitalRead(pol_1);
int stanje_2 = digitalRead(pol_2);
int način_rada = 1;

if((stanje_1 == LOW) && (stanje_2 == HIGH)){
    način_rada = 1;
}else if((stanje_1 == HIGH) && (stanje_2 == LOW)){
    način_rada = 2;
}
// Switch case --> realizacija odabira načina rada
switch(način_rada){
    case 1:
        fiksna_način(); // Fiksna način rada
        break;
    case 2:
        dvoosni_način(); // Dvoosni način rada
        break;

    delay(5); // Čekaj 5 ms
}

void fiksna_način(){
    servoverti.write(135);
    servohori.write(58); // Odgovara nultom položaju s obzirom na kućište
}

void dvoosni_način(){
    servoh = servohori.read(); // Očitani prijašnji pomaci servo motora
    servov = servoverti.read();
    int gore_l = analogRead(LDR_TL); // Očitane vrijednosti otpora fotootpornika
    int gore_d = analogRead(LDR_TR);
    int dole_l = analogRead(LDR_DL);
    int dole_d = analogRead(LDR_DR);
    int srednje_gore = (gore_l + gore_d) / 2;
    int srednje_dole = (dole_l + dole_d) / 2; // skupa fotoopronika
    int srednje_lijevo = (gore_l + dole_l) / 2;
    int srednje_desno = (gore_d + dole_d) / 2;
}

```

```

if (srednje_gore < srednje_dole) // Upravljanje vertikalnim servo motorom
{
  servoverti.write(servov +1);
  if (servov > servovLimitHigh) // Ako je pomak veći od vertikalnog
  {
    servov = servovLimitHigh; // onemogući daljni pomak
  }
  delay(10);
} else if (srednje_dole < srednje_gore)
{
  servoverti.write(servov -1);
  if (servov < servovLimitLow)
  {
    servov = servovLimitLow;
  }
  delay(10);
} else
{
  servoverti.write(servov);
}
if (srednje_lijevo > srednje_desno) // Upravljanje horizontalnim servom
{
  servohori.write(servoh -1);
  if (servoh > servohLimitHigh)
  {
    servoh = servohLimitHigh;
  }
  delay(10);
} else if (srednje_desno > srednje_lijevo)
{
  servohori.write(servoh +1);
  if (servoh < servohLimitLow) // Ako je pomak servo motora manji od
minimalnog // pomaka onemogući daljni pomak
  {
    servoh = servohLimitLow;
  }
  delay(10);
}
else
{
  servohori.write(servoh);
}
delay(50);
}

```

11. USPOREDBA UČINKOVITOSTI PRI FISKNOM TE DVOOSOONOM NAČINU RADA FOTONAPONSKOG SUSTAVA

11.1. Mjerenje napona

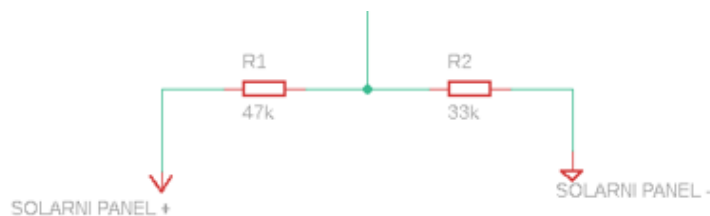
U eksperimentalnom dijelu završnog rada uspoređena je učinkovitost solarnog panela kada se nalazi u fiksnom položaju s učinkovitosti kada je omogućeno dvoosno praćenje položaja Sunca. Eksperiment je proveden mjerenjem izlaznog napona u periodu od 14 sati.

Napon solarnog panela mjeren je pomoću analognog pina A0 Arduino UNO pločice. S obzirom na to da je nazivni napon korištenog solarnog panela 12V, a maksimalni dozvoljeni napon koji se može spojiti na analogni pin Arduina 5V, potrebno je smanjiti napon analognog pina ispod 5V pomoću naponskog dijelila. Uloga naponskog dijelila je da ulazni napon u rasponu 0-12V, skalira na raspon 0-5V koji se može mjeriti analognim pinom. Proračun naponskog dijelila vrši se tako da prije svega ustanovimo maksimalni napon kojeg ćemo mjeriti, u našem slučaju to je 5V te nakon toga namjestimo omjer otpornika R_1 i R_2 prema sljedećem izrazu, na način da pri ulaznom naponu od 12V, izlazni napon iznosi 5V.

$$V_{IN} = V_{OUT} * \frac{R_2}{R_1 + R_2}, \quad (11.1)$$

pri čemu je V_{IN} maksimalni napon na analognom pinu (5V), dok je V_{OUT} maksimalni napon solarnog panela (12V).

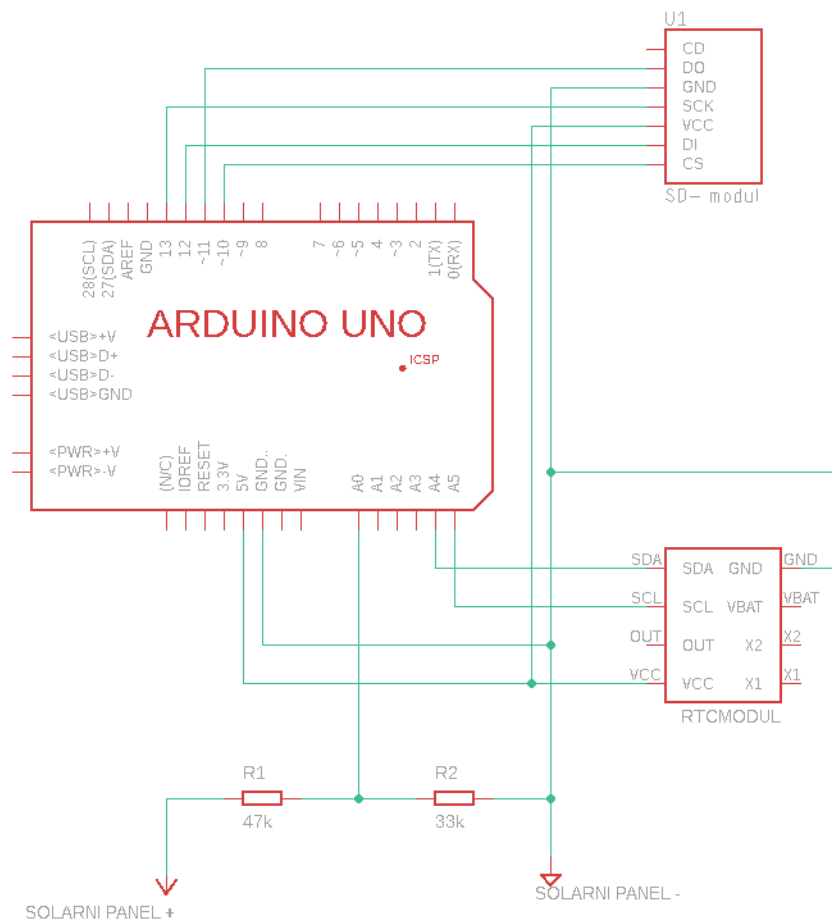
Odabrane su sljedeće vrijednosti otpornika: $R_1 = 47k\Omega$ te $R_2 = 33k\Omega$. Mjereni napon se potom dobije na način da se izmjereni napon pomnoži s recipročnim omjerom otpornika iz (11.1).



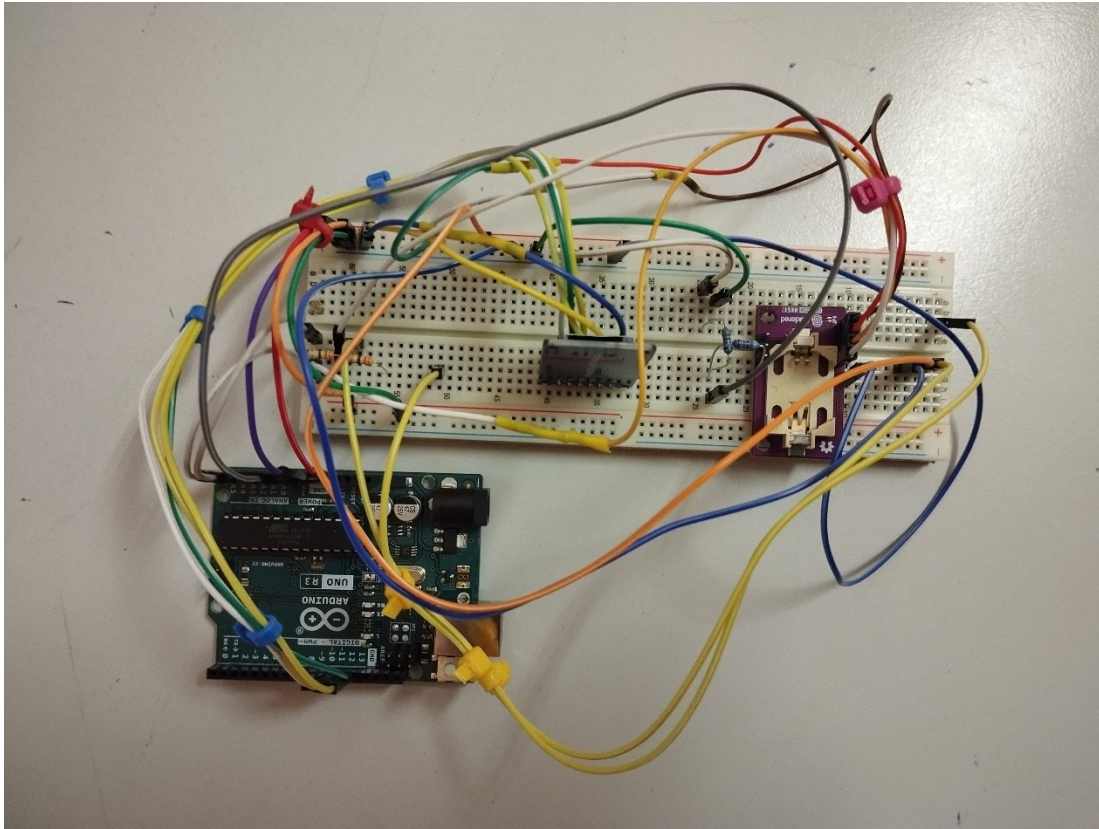
Slika 11.1. Naponsko dijelilo.

11.2. Shema spoja za mjerenje napona

Na slici 11.2. prikazana je shema spoja korištenog u eksperimentalnom dijelu pri mjerenju. Pinovi SDA i SCL RTC modula spojeni su na SDA i SCL pinove Arduina tj. na analogne pinove A4 i A5. Pinovi čitača SD kartice DI (MOSI pin), DO (MISO pin), SCK, te CS spojeni su na pinove Arduina koji su unaprijed definirani. Način povezivanja navedenih modula objašnjen je u 6 i 7. poglavlju. Povezani sklop prikazan je na slici 11.3.



Slika 11.2. Shema spoja korištenog pri mjerenju napona solarnog panela.



Slika 11.3. Sklop korišten za mjerenje napona solarnog panela.

11.3. Opis programskog koda

Na samom početku u program su uključene biblioteke `SD.h` i `SPI.h` koje služe za komunikaciju s modulom za čitanje SD kartice, te biblioteku `PCF85063A-SOLDERED.h` koja se koristi za upravljanje RTC modulom. Potom su deklarirani objekti `rtc` i `dataFile` koji služe za korištenje gornjih biblioteka. Nakon toga su deklarirane varijable koje se koriste u ostatku programa. Varijable `minute`, `sekunde` i `sat` koriste se za praćenje vremena koje generira RTC modul, dok varijable `vrijeme` i `datum` služe za formatiranje navedenih podataka u tekstualni oblik. Nadalje, u varijablu `ocitanje_senzora` sprema se analogna vrijednost napona kojeg mikrokontroler mjeri, dok se varijabla `vout` koristi kako bi se ta analogna vrijednost pretvorila u stvarnu vrijednost napona. Varijable `R1` te `R2` predstavljaju stvarne izmjerene vrijednosti otpornika naponskog dijelila, dok varijabla `arduino_5v` predstavlja stvarnu izmjerenu vrijednost napona na 5v pinu Arduina. Pri mjerenju napona preporučeno je koristiti izmjerene vrijednosti otpora i napona umjesto idealnih kako

bi se dobilo što preciznije mjerenje. Završno, deklarirane su varijable struja i snaga u koje se spremaju izračunate vrijednosti struje i snage.

Na početku `void setup()` funkcije pokrenuta je serijska komunikacija naredbom `Serial.begin()` koja je korištena za praćenje mjerenja preko *Serial monitor*-a. Potom, pomoću metoda `setTime()` i `setDate()` definirano je početno vrijeme i datum kao referencu RTC modula. Završno, slijedi provjera ispravnosti rada čitača SD kartice: ako je modul neispravan, onemogućeno je pisanje podataka na SD karticu. Potom je pomoću metode `open()` otvorena datoteka pod nazivnom `dvoosno.csv` odnosno `fiksno.csv` u slučaju mjerenja kada je solarni panel u fiksnom položaju.

Funkcija `ocitanje_napona()` koristi se za mjerenja napona. Za mjerenje napona korišteno je naponsko dijelilo koje je opisano u potpoglavlju 11.1. U varijablu `ocitanje_senzora` sprema se analogna vrijednost napona kojeg Arduino mjeri. Kako bi dobili stvarnu vrijednost napona potrebno je digitalnu vrijednost pretvoriti u analognu za što je potrebno poznavati maksimalnu vrijednost napona na analognom pinu koja je spremljena u varijablu `arduino_5v`. S obzirom na to da AD pretvarač ima razlučivost od 10 bita, maksimalna digitalna vrijednost koja se može izmjeriti jest 1023. Poznavajući ovaj podatak, napon kojeg Arduino mjeri dobiva se tako da očitanoj vrijednosti podijelimo s maksimalnim mogućim očitanjem te pomnožimo s varijablom `arduino_5v`. Kada smo odredili napon kojeg mikrokontroler mjeri, napon solarnog panela dobijemo prema relaciji (11.1). Funkcija `vrijeme_uzorka()` vrlo je jednostavna te se koristi za praćenje stvarnog vremena kojeg generira RTC modul. Osim praćenja vremena funkcija ima ulogu određivanja intervala u kojem će se uzimati podaci mjerenja. U funkciji `loop()` uspoređuju se vrijednosti varijabli minute i sekunde kako bi se mjerenja ponavljala svakih deset minuta.

Na samom početku `void loop()` funkcije pozvane su prethodno opisane funkcije: `vrijeme_uzorak()` te `ocitanje_napona()`. Korištenjem ohmovog zakona izračunata je struja i snaga koju panel daje, te su dobivene vrijednosti ispisane na *Serial monitor*. Završno, svakih deset minuta izmjereni podaci se zapisuju na SD karticu. U nastavku je izlistan kod korišten pri eksperimentalnom mjerenju napona solarnog panela.

```

// Deklarirane korištene biblioteke
#include <SD.h>
#include <SPI.h>
#include <PCF85063A-SOLDERED.h>
#include <Wire.h>

//RTC modul
PCF85063A rtc;
int minute = 0;
int sekunde = 0;
int sat = 0;
String vrijeme = "";
String datum = "";

//SD kartica
File dataFile;
const int CS = 10;    // Chip select na Arduino ( D10)

// Ocitanje napona
const int pin_senzora = 0; // Pin koji mjeri napon s djelitelja
float vout = 0.0;
float vin = 0.0;
float R1 = 4380.00; // Izmjerena vrijednost otpornika R1
float R2 = 3053.00; // Izmjerena vrijednost otpornika R2
float arduino_5v = 5.017; // Izmjerena vrijednost napona na 5V pinu Arduina
float ocitanje_senzora = 0;
float struja = 0;
float snaga = 0;

void setup() {

    pinMode(CS, OUTPUT);
    Serial.begin(9600);

    // RTC modul za praćenje vremenad
    // Inicijalizacija RTC modual

    Wire.begin();

    // Postavljanje datuma i vremena
    rtc.setTime(6,50,0);
    rtc.setDate(6,17,6,2024);
    rtc.begin();

    //Inicijalizacija SD kartice
    Serial.print("Inicijalizacija SD kartice...");
    if(!SD.begin(CS)){
        Serial.println("Neuspješno!");
        while(1);
    }
    Serial.print("Uspješno!");

    dataFile = SD.open("dvoosno.csv", FILE_WRITE);
    if(!dataFile){
        Serial.println("Greška pri otvaranju datoteke");
        return;
    }
}

```

```

    dataFile.println("Datum: , Vrijeme: , Napon [V]:, Struja [mA]:, Snaga
[mW]: ");
    dataFile.flush();
}

//Mjeri napon solarnog panela
float ocitanje_napona(){
    ocitanje_senzora = analogRead(pin_senzora);
    vout = (ocitanje_senzora*arduino_5v)/1024;
    vin = vout / (R2/(R1+R2));
    Serial.print("Ocitani napon: ");
    Serial.println(vin);
}

void vrijeme_uzorak(){

    sekunde = rtc.getSeconds();
    minute = rtc.getMinutes();
    sat = rtc.getHours();
    datum = String(rtc.getMonth()) + "/" + String(rtc.getDay()) + "/" +
String(rtc.getYear());
    vrijeme = String(rtc.getHours()) + ":" + String(rtc.getMinutes()) + ":" +
String(rtc.getSeconds()) + ".";

    Serial.println(vrijeme);
}

void loop() {

    vrijeme_uzorak(); // Kreiranje vremnea
    ocitanje_napona(); // Mjerni napon sa panela
    struja = (vin / (R1+R2))*1000; // Izračun struje koju panel daje u mA
    snaga = vin * struja; // Izračun snage koju panel daje u mW
    Serial.print("Mjereni napon: ");
    Serial.println(vin);
    Serial.print("Mjereni struja: ");
    Serial.println(struja);
    Serial.print("Mjereni snaga: ");
    Serial.println(snaga);

    if((minute % 10 == 0) || (minute == 0) && (sekunde == 0)){

        Serial.println("||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||");
        dataFile.print(datum);
        dataFile.print(", ");
        dataFile.print(vrijeme);
        dataFile.print(", ");
        dataFile.print(vin);
        dataFile.print(", ");
        dataFile.print(struja);
        dataFile.print(", ");
        dataFile.println(snaga);
        dataFile.flush();
        Serial.println("Spremljeno!");
    }
    delay(1000);
}

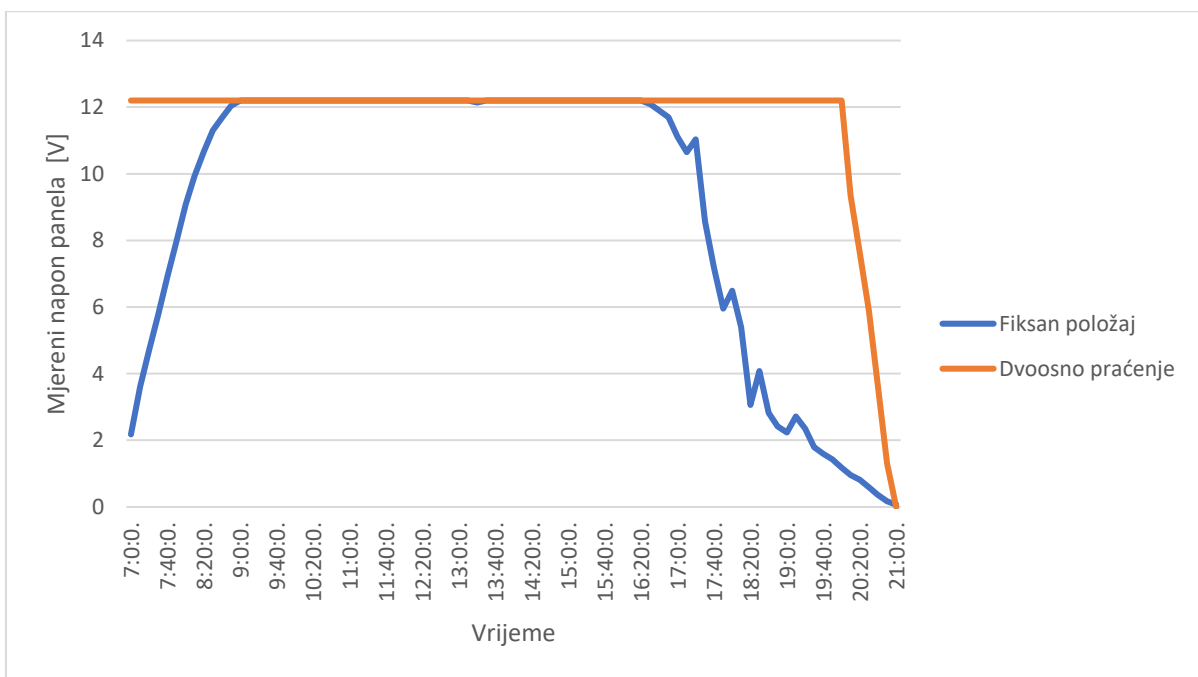
```

11.4. Provođenje eksperimenta

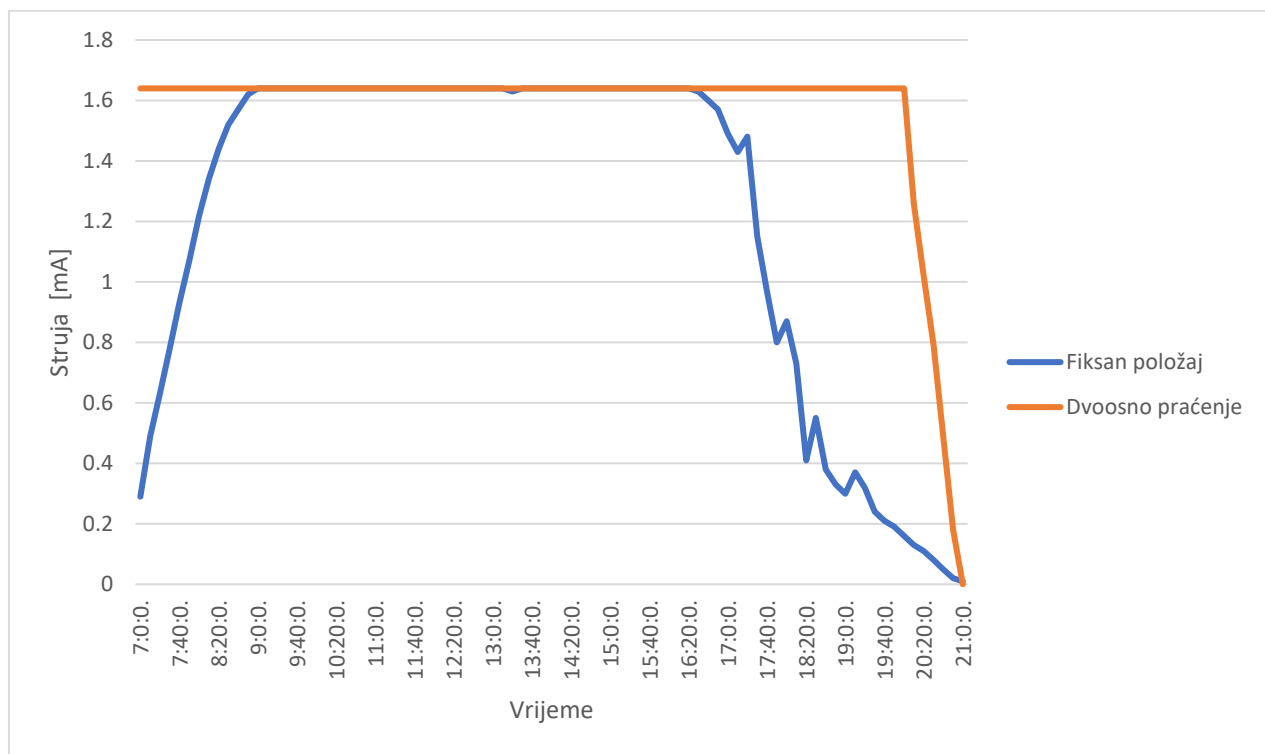
Eksperimentalni dio proveden je u periodu od 14 sati, točnije između sedam ujutro i devet navečer, dakle tokom cijelog dana. Tokom dana bilo je relativno sunčano uz vrlo malo naoblake. Na slici 11.4 prikazan je graf ovisnosti mjerenog napona za slučaj fiksnog položaja panela te omogućenog dvoosnog praćenja položaja Sunca. U fiksnom načinu rada, solarni panel bio je orijentiran prema jugu pod nagibom od 45° .

Sa grafa možemo primijetiti da se najveća razlika u mjerenom naponu javila u periodu između sedam i osam sati ujutro te između pet i devet sati navečer, dok u ostatku dana nema značajnije razlike u mjerenom naponu. Razlog tome je što pri izlasku Sunca u fiksnom položaju rada solarni panel je orijentiran prema jugu, zbog čega je panel slabo osvijetljen. Kako se Sunce diže na nebu, veći dio sunčevih zraka upada na panel, čime se generira veći mjereni napon. Kako se Sunce opet počinje spuštati sve manji dio sunčevih zraka pada na površinu panela te ponovno dolazi do pada napona.

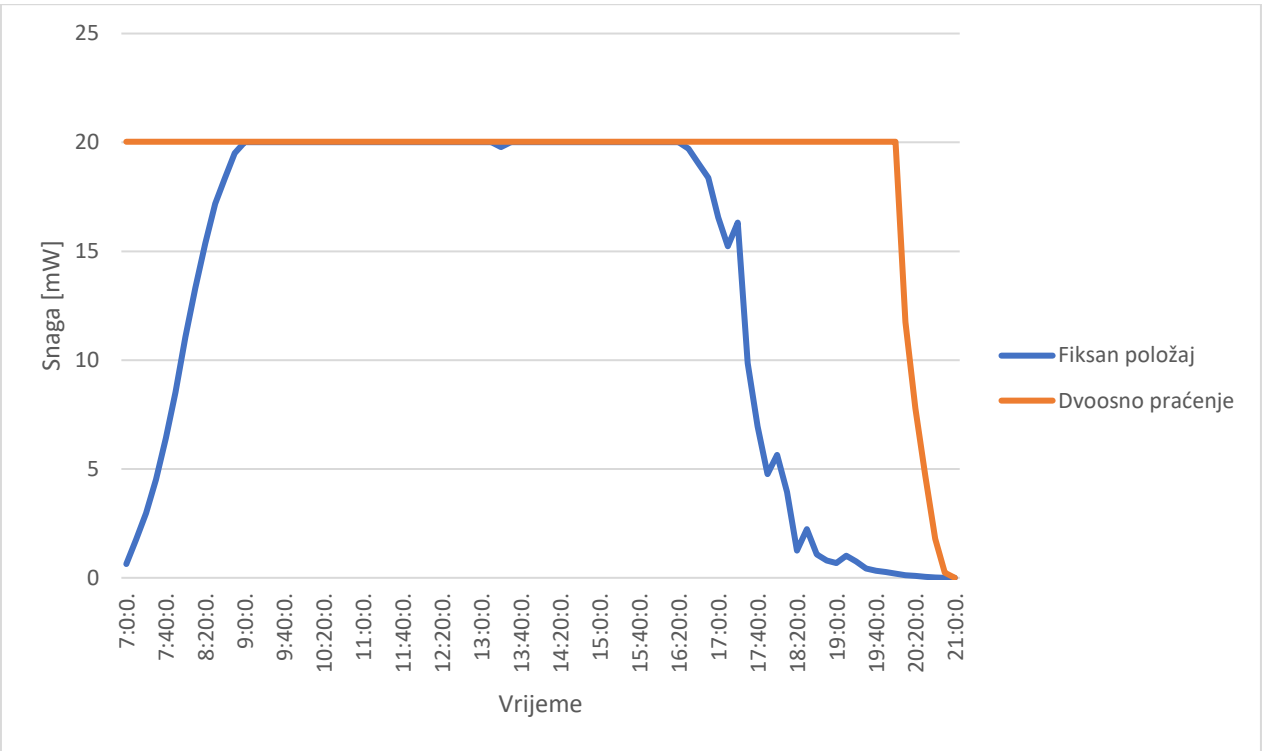
Kada je omogućeno dvoosno praćenje položaja Sunca, možemo primijetiti da se već pri samom izlasku Sunca panel orijentira prema Suncu, te kako se mijenja položaj Sunca na nebu, mijenja se i položaj solarnog panela zbog čega solarni panel kroz većinu dana generira nazivni napon. Tek kada Sunce počinje zalaziti, oko osam sati navečer, pada i generirani napon na panelu. Na slikama 11.5 i 11.6 također je dana i usporedba izračunatih struja i snaga pri fiksnom položaju rada te pri dvoosnom praćenju položaja Sunca. Na temelju izmjerenih podataka možemo zaključiti da se implementacijom dvoosnog praćenja položaja Sunca postiže 24% veća učinkovitost nego kada je solarni panel u fiksnom položaju.



Slika 11.4. Usporedba mjerenih napona pri dvoosnom praćenju položaja Sunca te fiksnom položaju solarnog panela.



Slika 11.5. Usporedba izračunatih struja pri dvoosnom praćenju položaja Sunca te fiksnom položaju solarnog panela.



Slika 11.6. Usporedba izračunatih snaga pri dvoosnom praćenju položaja Sunca te fiksnom položaju solarnog panela.

10. ZAKLJUČAK

U današnje vrijeme dolazi do sve veće potrebe za proizvodnjom električne energije. S obzirom na ograničenost fosilnih goriva, koji još uvijek služe kao primarni izvori energije, kao i njihovog utjecaja na ekosustav, u zadnjih 50-tak godina dolazi do razvoja obnovljivih izvora energije. Obnovljivi izvori energije služe kao alternativa fosilnim gorivima za proizvodnju električne energije. Trenutno najpopularniji obnovljivi izvor energije je uz, energiju vjetra, upravo energija Sunca. Korištenje solarnih panela je vrlo praktičan način proizvodnje električne energije kako za sitne potrošače (kućanstva, punjenje automobila) tako i za proizvodnju električne energije za veće potrošače (gradovi, općine itd.). Najveća prednost solarnih panela je što za svoj rad koriste čisti izvor energije koji je stalno dostupan. S druge strane, mana korištenja solarnih panela je što količina proizvedene električne energije ovisi o jačini sunčevog zračenja, što znači da kada je solarni panel postavljen u fiksnom položaju najveću učinkovitost imati će u zadanom vremenskom intervalu, a ne tokom čitavog dana. Upravo iz tog razloga, razvojem solarnih tragača moguće je znatno povećati količinu proizvedene električne energije dobivene iz Sunčevog zračenja. Iako korištenje solarne energije još uvijek nije do kraja istraženo smatram da u bliskoj budućnosti barem dijelom moglo zamijeniti fosilna goriva te tako smanjiti ljudski utjecaj na onečišćenje okoliša.

LITERATURA

- [1] docs.arduino.cc: Hardware, s interneta:
<https://docs.arduino.cc/hardware/uno-rev3/> (20.5.2024)
- [2] bdavison.napier.ac.uk : Tutorials, Arduino, s interneta:
<https://bdavison.napier.ac.uk/iot/Tutorials/Arduino/> (20.5.2024)
- [3] bdavison.napier.ac.uk : Arduino UNO, s interneta:
<https://bdavison.napier.ac.uk/iot/Notes/microprocessors/arduino/> (20.5.2024)
- [4] makerselectronics.com, s interneta
<https://makerselectronics.com/product/servo-motor-half-metal-gear-mg995-towerpro-180-degree>
(22.5.2024)
- [5] Apoorve : What is Servo Motor? – Understanding the basics of Servo Motor Working, s interneta
<https://circuitdigest.com/article/servo-motor-working-and-basics> (22.5.2024)
- [6] Electronics-notes.com : Light Dependent Resistor LDR: Photoresistor, s interneta:
https://www.electronics-notes.com/articles/electronic_components/resistors/light-dependent-resistor-ldr.php (23.5.2024)
- [7] docs.onion.io : Reading a Photoresistor, s interneta:
<https://docs.onion.io/omega2-arduino-dock-starter-kit/arduino-kit-reading-a-photoresistor.html> (23.5.2024)
- [8] Louise : Photoresistor: Basics and Arduino Tutorial, s interneta:
<https://www.jakelectronics.com/blog/photoresistor-basics-and-arduino-tutorial> (23.5.2024)
- [9] MS Chaudhry: Solar Cell Construction & Working Principle, s interneta:
https://electricalengineering123.com/solar-cell-construction-working-principle/#google_vignette
(24.5.2024)
- [10] aerisoul.com: The Different Types of Solar Modules, s interneta:
<https://aerisoul.com/three-main-types-solar-modules/> (25.5.2024)
- [11] electrical4u.com: Characteristics of a Solar Cell and Parameters of Solar Cell, s interneta:
<https://www.electrical4u.com/characteristics-and-parameters-of-a-solar-cell/> (25.5.2024)
- [12] Tibi Puiu : What are the pros and cons of solar energy? Here's everything you need to know
<https://www.zmescience.com/feature-post/technology-articles/sustainability/renewable-energy/solar-panels-pros-and-cons-056654/> (25.5.2024)
- [13] tycorun.com : Solar tracking system – A comprehensive analysis of its principle and functions, s interneta:
<https://www.tycorun.com/blogs/news/solar-tracking-system> (28.5.2024)

[14] creality.com, s interneta:

<https://www.creality.com/products/ender-3-3d-printer> (28.5.2024)

[15] Catherine Lee : What is a solar tracker and is it worth the investment, s interneta:

<https://www.solarreviews.com/blog/are-solar-axis-trackers-worth-the-additional-investment> (30.5.2024)

[16] Paula E. Logan i dr. : Performance comparison of fixed, single, and dual axis tracking systems for small photovoltaic systems with measured direct beam fraction

<https://ases.org/wp-content/uploads/2021/11/Performance-Comparison-of-Fixed-Single-and-Dual-Axis-Tracking-Systems-For-Small-Photovoltaic-Systems-with-Measured-Direct-Beam-Fraction-.pdf> (1.6.2024)

[17] Rustus Eke i dr. : Performance comparison of a double-axis sun tracking versus fixed PV system :

https://www.researchgate.net/publication/236969054_Performance_comparison_of_a_double-axis_sun_tracking_versus_fixed_PV_system (4.6.2024)

POPIS SLIKA

Slika 2.1. Arduino UNO R3 pločica [3].	3
Slika 3.1. MG995 Servo motor [4].	5
Slika 3.2. Blokovska shema servo motora.	5
Slika 4.1. Poprečni presjek fotootpornika [6].	6
Slika 4.2. Fotootpornik [7].	6
Slika 4.3. Karakteristika fotootpornika [8].	7
Slika 5.1. Princip rada solarne ćelije [9].	8
Slika 5.2. Vrste solarnih panela [10].	9
Slika 5.3. U-I Karakteristika solarnog panela [11].	10
Slika 8.1. Držač solarnog panela modeliran u Fusion 360.	15
Slika 8.2. Maketa solarnog tragača.	15
Slika 8.3 Izlged sučelja programa Ultimaker Cura.	16
Slika 8.4 Sastavljeno kućište.	17
Slika 8.5 Ender 3 3D printer [14].	17
Slika 9.1. Shema spoja.	19
Slika 9.2. Povezane komponente.	19
Slika 11.1. Naponsko dijelilo.	24
Slika 11.2. Shema spoja korištenog pri mjerenju napona solarnog panela.	25
Slika 11.3. Sklop korišten za mjerenje napona solarnog panela.	26
Slika 11.4. Usporedba mjerenih napona pri dvoosnom praćenju položaja Sunca te fiksnom položaju solarnog panela.	31
Slika 11.5. Usporedba izračunatih struja pri dvoosnom praćenju položaja Sunca te fiksnom položaju solarnog panela.	31
Slika 11.6. Usporedba izračunatih snaga pri dvoosnom praćenju položaja Sunca te fiksnom položaju solarnog panela.	32

POPIS TABLICA

Tablica 2.1. Specifikacije Arduino UNO pločice [1].....	3
Tablica 9.1. Popis i uloga korištenih komponentni.	18

SAŽETAK

U sklopu ovog završnog rada izrađen je solarni tragač s mogućnošću dvoosnog praćenja položaja Sunca. Prvo su opisane sve komponente korištene pri izradi sustava. Nakon toga opisan je proces izrade samog tragača, tj. opisan je postupak izrade kućišta, shema spajanja komponenti te opis programskog koda sklopa. Završno, eksperimentom je uspoređena učinkovitost fiksnog fotonaponskog sustava sa sustavom koji ima mogućnost dvoosnog praćenja položaja Sunca, te je na temelju dobivenih mjerenja ustanovljena 24% veća učinkovitost sustava koji ima mogućnost praćenja položaja Sunca.

Ključne riječi: Arduino, solarni tragač, fotonaponski sustav, solarni panel

ABSTRACT

In this bachelors thesis, Arduino based tracker has been developed. First, all components used in the system have been described. After that, the process of building the tracker has been described. In terms of making of the housing, connection diagram of the system, and the development software. Finally, through the experiment, the efficiency of the dual-axis solar tracker has been compared with the efficiency of the fixed photovoltaic system, and the measured data indicates 24% greater efficiency of the dual-axis solar tracker.

Keywords: Arduino, dual axis solar tracker, photovoltaic system

DODATAK

Programski kod fotonaponskog sustava:

```
#include <Servo.h> // Biblioteka za upravljanje Servo motorima
Servo servohori; // Kreiranje objekta servohori
const int servoh_pin = 10;
int servoh = 0;
int servohLimitHigh = 160; // Definiranje graničnih položaja
int servohLimitLow = 20;
Servo servoverti; // Kreiranje objekta servoverti
const int servov_pin = 9;
int servov = 0;
int servovLimitHigh = 160; // Definiranje graničnih položaja
int servovLimitLow = 0;
const int LDR_DR = A2; // Definiranje pinova na koji su spojeni
const int LDR_DL = A4; // na koje su spojeni fotootpornici
const int LDR_TR = A1;
const int LDR_TL = A3;
const int pol_1 = 4; // Pinovi sklopke kojima određujemo
const int pol_2 = 3; // način rada sustava

void setup () // Izvršava se samo jednom
{
  Serial.begin(9600); // Inicijalizacija serijske komunikacije
  servoverti.attach(servov_pin); // Opredjeljuje pinove
  servohori.attach(servoh_pin);
  servohori.write(58); // Nulta pozicija s obizrom na horizontalu
  servoverti.write(135); // Nulta pozicija s obizrom na vertikalnu
  pinMode(pol_1, INPUT_PULLUP); // Definiranje opisa pina
  pinMode(pol_2, INPUT_PULLUP);

  delay(500); // Čekaj 500 ms
}

void loop(){
  int stanje_1 = digitalRead(pol_1);
  int stanje_2 = digitalRead(pol_2);
  int nacin_rada = 1;
  if((stanje_1 == LOW) && (stanje_2 == HIGH)){
    nacin_rada = 1;
  }else if((stanje_1 == HIGH) && (stanje_2 == LOW)){
    nacin_rada = 2;
  }
  // Switch case --> realizacija odabira načina rada
  switch(nacin_rada){

    case 1:
      fiksni_nacin(); // Fiksni način rada
      break;
    case 2:
      dvoosni_nacin(); // Dvoosni način rada
      break;
  }
  delay(5); // Čekaj 5 ms
}

void fiksni_nacin(){
```



```

servoverti.write(135); // Odgovara pomaku od 45 stupenjva s obizorm na
vertikalnu
servohori.write(58); // Odgovara nultom položaju s obzirom na kućište
}
void dvoosni_nacin(){
servoh = servohori.read(); // Ocitani prijašnji pomaci servo motora
servov = servoverti.read();
int gore_l = analogRead(LDR_TL); // Ocitane vrijednosti otpora fotootpornika
int gore_d = analogRead(LDR_TR);
int dole_l = analogRead(LDR_DL);
int dole_d = analogRead(LDR_DR);
int srednje_gore = (gore_l + gore_d) / 2;
int srednje_dole = (dole_l + dole_d) / 2; // skupa fotoopronika
int srednje_lijevo = (gore_l + dole_l) / 2;
int srednje_desno = (gore_d + dole_d) / 2;

if (srednje_gore < srednje_dole) // Upravljanje vertikalnim servo motorom
{
servoverti.write(servov +1);
if (servov > servovLimitHigh) // Ako je pomak veći od vertikalnog
{
servov = servovLimitHigh; // onemogućiti daljni pomak
}
delay(10);
} else if (srednje_dole < srednje_gore)
{
servoverti.write(servov -1);
if (servov < servovLimitLow)
{
servov = servovLimitLow;
}
delay(10);
} else
{
servoverti.write(servov);
}
if (srednje_lijevo > srednje_desno) // Upravljanje horizontalnim servom
{
servohori.write(servoh -1);
if (servoh > servohLimitHigh)
{
servoh = servohLimitHigh;
}
delay(10);
} else if (srednje_desno > srednje_lijevo)
{
servohori.write(servoh +1);
if (servoh < servohLimitLow)
{
servoh = servohLimitLow; // pomaka onemogućiti daljni pomak
}
delay(10);
}
else
{
servohori.write(servoh);
}
delay(50);
}
}

```

Programski kod za eksperimentalno mjerenje napona solarnog panela:

```
#include <SD.h>
#include <SPI.h>
#include <PCF85063A-SOLDERED.h>
#include <Wire.h>

PCF85063A rtc;
int minute = 0;
int sekunde = 0;
int sat = 0;
String vrijeme = "";
String datum = "";

File dataFile;
const int CS = 10; // Chip select na Arduinu ( D10)
const int pin_senzora = 0; // Pin koji mjeri napon s djelitelja

float vout = 0.0;
float vin = 0.0;
float R1 = 4380.00; // Izmjerena vrijednost otpornika R1
float R2 = 3053.00; // Izmjerena vrijednost otpornika R2
float arduino_5v = 5.017; // Izmjerena vrijednost napona na 5V pinu Arduina
float ocitanje_senzora = 0;
float mjereni_napon = 0;
float struja = 0;
float snaga = 0;

void setup() {
  pinMode(CS, OUTPUT);
  Serial.begin(9600);

  Wire.begin();

  rtc.setTime(6,50,0);
  rtc.setDate(6,17,6,2024);
  rtc.begin();

  Serial.print("Inicijalizacija SD kartice...");
  if(!SD.begin(CS)){
    Serial.println("Neuspješno!");
    while(1);
  }
  Serial.print("Uspješno!");

  dataFile = SD.open("dvoosno.csv", FILE_WRITE);
  if(!dataFile){
    Serial.println("Greška pri otvaranju datoteke");
    return;
  }
  dataFile.println("Datum: , Vrijeme: , Napon [V]:, Struja [mA]:, Snaga [mW]: ");
  dataFile.flush();
}

//Mjeri napon solarnog panela
float ocitanje_napona(){
  ocitanje_senzora = analogRead(pin_senzora);
  vout = (ocitanje_senzora*arduino_5v)/1024;
```

```

    vin = vout / (R2/(R1+R2));
    Serial.print("Ocitanje napon: ");
    Serial.println(vin);
}

void vrijeme_uzorak(){
    sekunde = rtc.getSecond();
    minute = rtc.getMinute();
    sat = rtc.getHour();
    datum = String(rtc.getMonth()) + "/" + String(rtc.getDay()) + "/" +
String(rtc.getYear());
    vrijeme = String(rtc.getHour()) + ":" + String(rtc.getMinute()) + ":" +
String(rtc.getSecond()) + ".";

    Serial.println(vrijeme);
}

void loop() {
    vrijeme_uzorak(); // Kreiranje vremnea
    ocitanje_napona(); // Mjerni napon sa panela
    struja = (vin / (R1+R2))*1000; // Izračun struje koju panel daje u mA
    snaga = vin * struja; // Izračun snage koju panel daje u mW
    Serial.print("Mjereni napon: ");
    Serial.println(vin);
    Serial.print("Mjereni struja: ");
    Serial.println(struja);
    Serial.print("Mjereni snaga: ");
    Serial.println(snaga);

    if((minute % 10 == 0) || (minute == 0) && (sekunde == 0)){

        Serial.println("||||||||||||||||||||||||||||||||||||||||||||||||||||||||");
        dataFile.print(datum);
        dataFile.print(", ");
        dataFile.print(vrijeme);
        dataFile.print(", ");
        dataFile.print(vin);
        dataFile.print(", ");
        dataFile.print(struja);
        dataFile.print(", ");
        dataFile.println(snaga);
        dataFile.flush();
        Serial.println("Spremljeno!");
    }
    delay(1000);
}

```