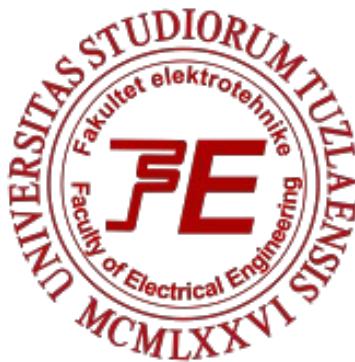


UNIVERZITET U TUZLI
FAKULTET ELEKTROTEHNIKE



PROJEKTNI ZADATAK

(Raspberry PI)

Predmet: Projektovanje sistema na čipu

Tema: Automatizirani sistem sušenja drveta

PROFESOR:

dr. sci. Lejla Banjanović-Mehmedović,
vanr. prof.

STUDENTI:

Amar Degirmendžić
Fatima Alić
Azur Jusić

TUZLA, 2024. god.

Sadržaj

1. Uvod.....	4
2. Opis problema.....	5
2.1 Opis projekta.....	5
3. Opis korištenih komponenti	6
3.1 Raspberry PI 4.....	6
3.1.1 Razvojno okruženje.....	7
3.2 DC servo motor.....	8
3.3 DHT 11 senzor temperature.....	9
3.4 Relej.....	10
3.5 Keypad 4x4.....	12
3.6 DC ventilator 12V.....	13
3.7 LCD 16x2.....	15
3.8 PTC grijajući	17
3.9 N-kanalni MOSFET.....	18
4. Algoritam upravljanja sušarom drveta.....	19
4.1 Ulazi i izlazi.....	19
4.2 Šema povezivanja.....	20
4.3 Grafički odziv.....	21
4.4 Tabelarni prikaz.....	29
4.5 Slike makete	33
5. KODOVI.....	37
6. Eksperimentalni dio.....	49
7. Zaključak.....	50
8.Literatura	51

Popis slika:

Figure 1 Sušara drveta	4
Figure 2 Raspored pinova	7
Figure 3 Servo motor	8
Figure 4 Ugaona rotacija	9
Figure 5 DHT 11 senzor	10
Figure 6 Šema releja	11
Figure 7 Relej 5V	12
Figure 8 Keypad 4x4	12
Figure 9 Šematski prikaz keypada	13
Figure 10 DC ventilator 12V	15
Figure 11 LCD 16x2	16
Figure 12 LCD 16x2 pin diagram	16
Figure 13 PTC grijac	17
Figure 14 Tipovi MOSFET-a	18
Figure 15 Šema povezivanja	20
Figure 16 Temperatura	21
Figure 17 Odziv-vlažnost	21
Figure 18 Odziv servo motora u zavisnosti od temperature i vlažnosti	23
Figure 19 Odziv ventilatora u zavisnosti od temperature i vlage	26
Figure 20 Blok dijagram dio 1	30
Figure 21 Blok dijagram dio 2	31
Figure 22 Blok dijagram dio 3	32
Figure 23 Maketa 1	33
Figure 24 Maketa 2	34
Figure 25 Maketa 3	35
Figure 26 Maketa 4	36
Figure 27 Zavisnost vlažnosti od temperature	49

Popis tabela

Tabela 1 Popis komponenti	6
Tabela 2 Ulazi i izlazi na Raspberry pi	19
Tabela 3 Ponašanje sistema	28

Popis kodova

Kod 1 Skripta za prikaz odziva vlažnosti drveta u zavisnosti od temperature.....	22
Kod 2 Odziv servo motora u zavisnosti od teperature i vlažnosti	24
Kod 3 Odziv grijaca u zavisnosti od vlažnosti drveta	24
Kod 4 Odziv releja (grijaca)	25
Kod 5 Odziv ventilatora	27
Kod 6 Upravljanje sušarom pomoću Phytona	43

1. Uvod

U savremenoj drvnoj industriji, potreba za inovativnim i inteligentnim rješenjima postaje sve izraženija kako bi se optimizirali procesi sušenja drveta. Sušare drveta imaju ključnu ulogu u postizanju kvalitetnih proizvoda, smanjenju troškova i očuvanju resursa. S razvojem tehnologije, koncept "automatizirani sistem sušenja drveta" postaje ključni faktor u postizanju ovih ciljeva. Automatizirani sistem sušenja drveta predstavlja napredno tehničko rješenje koje koristi visoko sofisticirane tehnologije kako bi optimizirala procese sušenja. Ovaj pristup omogućava precizno praćenje i upravljanje parametrima sušenja, prilagodbu procesa u stvarnom vremenu i optimizaciju potrošnje energije. U tom kontekstu, ovaj projekt istražuje primjenu najnovijih tehnologija i rješenja za stvaranje sušare drveta visoke učinkovitosti, koje će biti temeljene na korištenju Raspberry Pi tehnologije. Drvna industrija, kao važna grana industrije koja se bavi preradom drvnih materijala, suočava se s izazovima vezanim uz brzinu sušenja, kvalitetu finalnih proizvoda i optimizaciju proizvodnih procesa. U sušarama drveta, sušenje je kritičan korak u proizvodnom lancu, jer utječe na svojstva i kvalitetu gotovih proizvoda kao što su daske, gredice i furniri. U kontekstu ovog projekta, istraživanje se usmjerava na stvaranje automatiziranog sistema sušenja drveta koja kombinira hardverske i softverske komponente kako bi se optimizirali procesi sušenja. U narednim sekcijama izvještaja detaljnije će se istražiti tehnički aspekti projekta, uključujući hardverske i softverske komponente, tehnološke izazove i rezultate postignute kroz implementaciju sušare drveta temeljene na Raspberry Pi tehnologiji.



Figure 1 Sušara drveta

2. Opis problema

Sušenje daske u drvnoj industriji je jedna od svakodnevnih stvari koja se dešava u toku proizvodnog procesa. Tek odrezani trupci imaju veliki postotak vlage, te prilikom izrade finalnih proizvoda takvo drvo se ne može koristiti, iz razloga, što bi finalni proizvod prošao kroz nekontrolirani proces sušenja, što za posljedicu ima pucanje drveta i opadanje kvaliteta finalnog proizvoda. Kako živimo u Tuzlanskom kantonu, gdje je jedna od važnijih privrednih grana drvna industrija, te obrada trupaca za pripremu poluproizvoda i proizvoda, vjerujemo da je ova tema i iz tog razloga zanimljiva. Također, veliki broj sušara koristi stariju, relejnu tehniku, koja je prilično robusna, ali sa druge strane ima manji postotak tačnosti i ponovljivosti. Sušenje drveta nije šablonska stvar, jer zavisi od mnogo faktora (porijeklo i kvalitet trupaca, vremenski i klimatski uslovi, način slaganja, vrsta drveta, željena vlažnost drveta nakon procesa sušenja, te je radi toga potrebno i modifikovati same režime sušenja. Svrha inteligentne sušare drveta je upravo u tome da kontroliramo što više parametara navedenih iznad i da drvo pripremimo za narednu fazu proizvodnog procesa, vodeći računa i o energetskoj efikasnosti. Proces sušenja drveta je dugotrajan, međutim, u ovome radu, pokušat ćemo da taj proces “isforsiramo” na nekoliko minuta simulirajući neke parametre.

2.1 Opis projekta

Automatizirani sistem sušenja drveta ima za cilj razvoj sistema za kontrolu i automatizaciju procesa sušenja drveta u drvnoj industriji. Sušenje drveta je ključni korak u proizvodnom procesu koji utiče na kvalitet i performanse finalnih proizvoda od drveta, kao što su namještaj, građevinski materijali, ili druge drvne konstrukcije. Suština projekta je implementacija automatiziranog sistema koji će pratiti i kontrolisati ključne parametre procesa sušenja drveta kako bi se postigla optimalna vlažnost i kvalitet finalnog proizvoda. Za to će se koristiti senzori za praćenje temperature i vlažnosti vazduha i drveta, kao i sistem za kontrolu ventilacije i grijanja unutar sušare.

Konkretni ciljevi projekta uključuju:

- Implementaciju senzora temperature i vlažnosti (poput DHT11 senzora) za praćenje uslova u sušari.
- Razvoj sistema za kontrolu ventilatora i grijaća unutar sušare radi precizne kontrole temperature i vlažnosti.
- Integraciju mikrokontrolera (kao što je Raspberry Pi 4) za obradu podataka sa senzora i upravljanje sistema za kontrolu.
- Implementaciju korisničkog interfejsa putem LCD displeja i tastature (keypad-a) za podešavanje parametara sušenja i praćenje stanja sistema.
- Testiranje i optimizaciju sistema kako bi se postigla efikasnost u sušenju drveta i kvalitet finalnog proizvoda.
- Integracija releja pomoću kojeg kontrolisemo grijaće, te step motora pomoću kojeg otvaramo i zatvaramo sušaru radi kontrole vazduha.

Cilj ovog projekta je razviti sušaru drveta koja će koristiti senzore za praćenje vlage i temperature u realnom vremenu, te automatski prilagođavati parametre sušenja kako bi se postigla optimalna kvaliteta i brzina sušenja.

3. Opis korištenih komponenti

Komponente	Količina	Cijena
Raspberry PI 4	1x	/
DHT 11 senzor	1x	120 KM
	1x	5 KM
Relej	1x	1x
Keypad	1x	3 KM
LCD display 16x2	1x	3 KM
DC ventilator 12 V	1x	10 KM
Servo motor	2x	16 KM
Grijači	1x	4KM
N-kanalni MOSFET	2x	/
Maketa		2KM
Ukupno:		0 KM
		160 KM

Tabela 1 Popis komponenti

3.1 Raspberry PI 4

Raspberry Pi je serija malih single-board računara (SBC). Raspberry Pi je naziv serije single-board računara koje je napravila Raspberry Pi Foundation, britanska dobrovorna organizacija koja ima za cilj da edukuje ljude o računarstvu i omogući lakši pristup računarskom obrazovanju. Raspberry Pi je lansiran 2012. godine i od tada je objavljeno nekoliko iteracija i varijacija. Originalni Pi je imao jednojezgarni CPU od 700MHz i samo 256MB RAM-a, a najnoviji model ima četvorjezgarni CPU koji radi na preko 1,5GHz i 4GB RAM-a. Širom svijeta, ljudi koriste Raspberry Pi za učenje vještina programiranja, izgradnju hardverskih projekata, automatizaciju doma, implementaciju Kubernetes klastera i Edge računarstva, pa čak i da ih koriste u industrijskim aplikacijama. Raspberry Pi je vrlo jeftin računar koji pokreće Linux, ali također pruža set GPIO (unos/izlaz opće namjene) pinova, koji vam omogućavaju kontrolu elektroničkih komponenti za fizičko računanje i istraživanje Interneta stvari (IoT).

Raspberry Pi 4 je 4. generacija glavne serije Raspberry Pi računara sa jednom pločom. Razvijen od strane Raspberry Pi Tradinga i objavljen 24. juna 2019., Pi 4 je došao sa mnogim poboljšanjima u odnosu na svog prethodnika; SoC je nadograđen na Broadcom BCM2711, dva od četiri USB porta Raspberry Pi-a su nadograđena na USB 3.0, a dodane su i opcije za RAM kapacitete veće od standarda od 1 GB za prethodnu Raspberry Pi 3 seriju.

Specifikacija

- Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.8GHz
- 1GB, 2GB, 4GB ili 8GB LPDDR4-3200 SDRAM (zavisno od modela)
- 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE Gigabit Ethernet
- 2 USB 3.0 ports; 2 USB 2.0 ports.
- Raspberry Pi standard 40 pin GPIO header (fully backwards compatible with previous boards)
- 2 × micro-HDMI® ports (up to 4kp60 supported)
- 2-lane MIPI DSI display port
- 2-lane MIPI CSI camera port
- 4-pole stereo audio and composite video port
- H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)

- OpenGL ES 3.1, Vulkan 1.0
- Micro-SD card slot for loading operating system and data storage
- 5V DC via USB-C connector (minimum 3A*)
- 5V DC via GPIO header (minimum 3A*)
- Power over Ethernet (PoE) enabled (requires separate PoE HAT)
- Operating temperature: 0 – 50 degrees C ambient

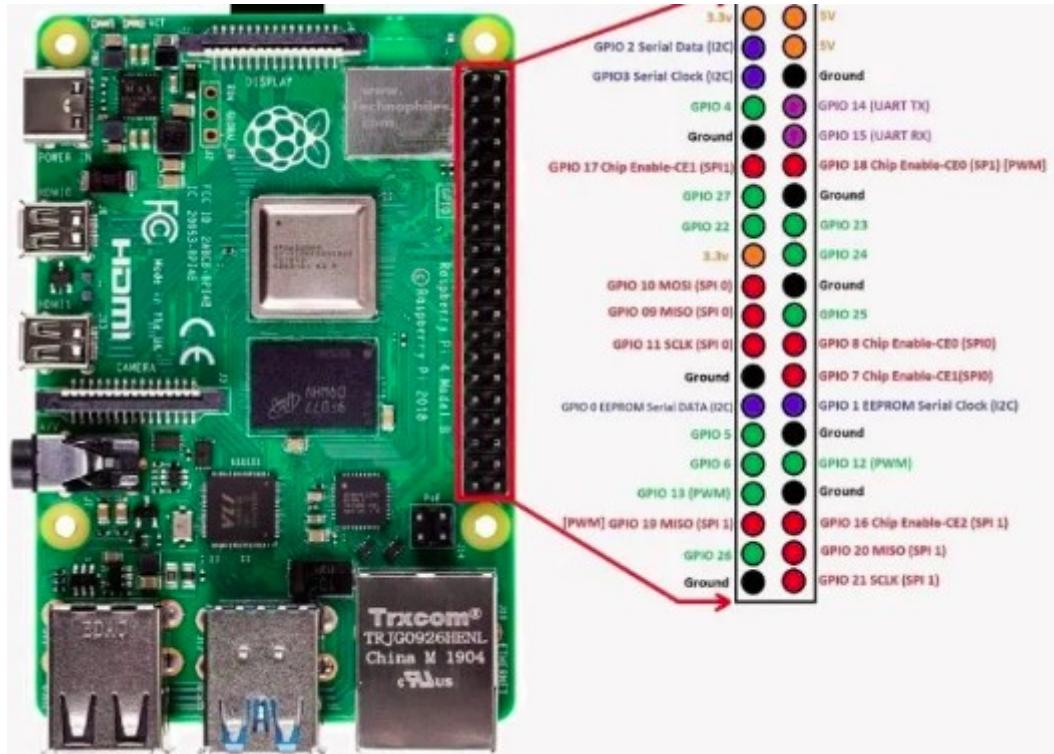


Figure 2 Raspored pinova

3.1.1 Razvojno okruženje

Thonny je Python integrisano razvojno okruženje (IDE) dizajnirano za početnike. Ime je dobio po finskoj riječi "thonny", što znači "početnik". Ima jednostavno i intuitivno sučelje koje početnicima olakšava početak sa Python programiranjem. Uključuje funkcije kao što su isticanje sintakse, automatsko uvlačenje i program za otklanjanje grešaka koji može pomoći učenicima da pronađu i poprave greške u svom kodu. Thonny također uključuje ugrađenu Python shell, koja omogućava korisnicima da testiraju male dijelove koda i odmah vide rezultate. Kada na raspberi instaliramo njegov operativni sistem dobijamo Thonny razvojno okruženje koje možemo koristiti. Sama instalacija Raspberi Pi operativnog sistema je urađena preko micro SD kartice jer raspberi ima podršku za takvu vrstu memorije.

3.2 DC servo motor

Servo motor je vrsta motora koji se može rotirati s velikom preciznošću. Obično se ovaj tip motora sastoji od kontrolnog kola koje daje povratnu informaciju o trenutnoj poziciji osovine motora, ova

povratna informacija omogućava rotaciju servo motora s velikom preciznošću. Položaj servo motora se određuje električnim impulsom, a njegovo kolo se postavlja pored motora.

Sastoji se iz tri dijela:

- Kontrolisani uređaj
- Izlazni senzor
- Sistem povratnih informacija

To je sistem zatvorene petlje gdje koristi sistem pozitivne povratne sprege za kontrolu kretanja i konačnog položaja osovine. Ovdje se uređajem upravlja povratnim signalom generiranim poređenjem izlaznog signala i referentnog ulaznog signala. Ovdje se referentni ulazni signal upoređuje sa referentnim izlaznim signalom, a treći signal proizvodi sistem povratne sprege. I ovaj treći signal djeluje kao ulazni signal za kontrolu uređaja. Ovaj signal je prisutan sve dok se generiše povratni signal ili postoji razlika između referentnog ulaznog signala i referentnog izlaznog signala. Dakle, glavni zadatak servomehanizma je da održi izlaz sistema na željenoj vrednosti uz prisustvo šuma.



Figure 3 Servo motor

Servo uređaji imaju tri žice koje izlaze iz njih, od kojih će dvije biti korištene za napajanje (pozitivna i negativna), a jedna će se koristiti za signal koji se šalje iz MCU-a. Svi servo motori rade direktno sa +5V naponom, ali moramo paziti na količinu struje koju bi motor potrošio ako planiramo koristiti više od dva servo motora(potrebno je dizajnirati odgovarajući servo štit). Servo motorom upravlja PWM (Pulse sa modulacijom) koji je osiguran kontrolnim žicama. Postoji minimalni puls, maksimalni puls i stopa ponavljanja. Servo motor se može okrenuti za 90 stepeni iz bilo kojeg smjera iz svog neutralnog položaja. Servo motor očekuje da vidi impuls svakih 20 milisekundi (ms) i dužina impulsa će odrediti koliko se motor okreće. Na primjer, impuls od 1,5 ms će natjerati motor da se okreće u položaj od 90° , ako je impuls kraći od 1,5 ms osovina se pomiče na 0° , a ako je duži od 1,5 ms onda će servo okrenuti na 180° . Servo motor radi na principu PWM (Pulse width modulation), što znači da se njegov ugao rotacije kontroliše trajanjem primjenjenog impulsa na njegov kontrolni PIN. U osnovi servo motor se sastoji od DC motora koji je kontroliran promjenjivim otpornikom (potenciometrom) i nekim zupčanicima. Velika brzina DC motora se pretvara u obrtni moment pomoću zupčanika.

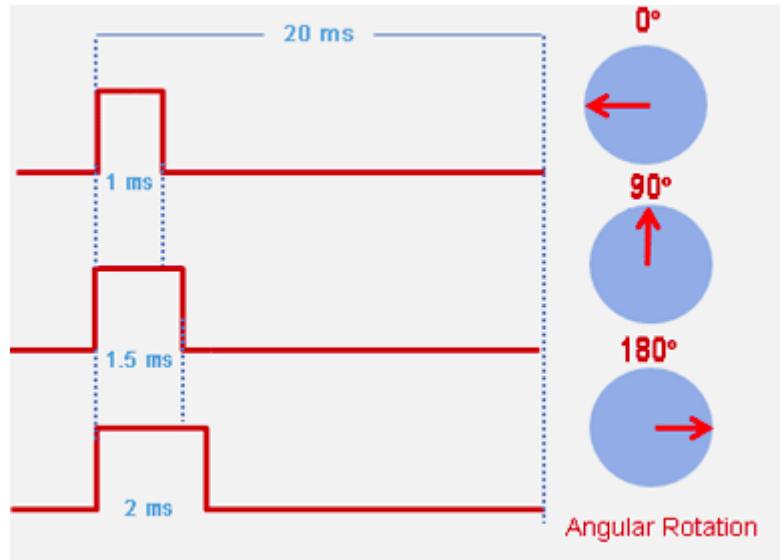


Figure 4 Ugaona rotacija

3.3 DHT 11 senzor temperature

DHT11 je često korišteni senzor temperature i vlage koji dolazi sa posebnim NTC-om za mjerjenje temperature i 8-bitnim mikrokontrolerom za izlaz vrijednosti temperature i vlage kao serijskih podataka.

Specifikacije:

- Radni napon: 3.5V do 5.5V
- Radna struja: 0.3mA (mjerjenje) 60uA (standby)
- Izlaz: Serijski podaci
- Opseg temperature: 0°C do 50°C
- Opseg vlage: 20% do 90%
- Rezolucija: Temperatura i vlažnost su oba 16-bitna
- Točnost: $\pm 1^{\circ}\text{C}$ i $\pm 1\%$

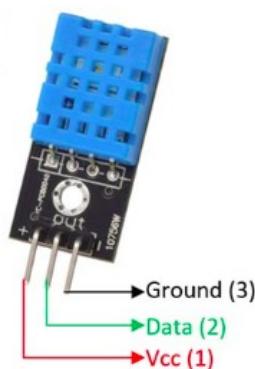


Figure 5 DHT 11 senzor

DHT11 senzor sastoji se od kapacitivnog senzora vlage i termistora za senzor temperature. Kondenzator za senzor vlažnosti ima dvije elektrode sa supstratom za zadržavanje vlage kao dielektrikom

između njih. Promjena vrijednosti kapacitivnosti događa se s promjenom razine vlažnosti. IC mjeri, obrađuje te promijenjene vrijednosti otpora i mijenja ih u digitalni oblik. Za mjerjenje temperature ovaj senzor koristi termistor s negativnim temperaturnim koeficijentom, koji uzrokuje smanjenje vrijednosti otpora s porastom temperature. Kako bi se dobila veća vrijednost otpora čak i za najmanju promjenu temperature, ovaj senzor se obično sastoji od poluvodičke keramike ili polimera. Temperaturni raspon DHT11 je od 0 do 50 stupnjeva Celzija s točnošću od 2 stupnja. Raspon vlažnosti ovog senzora je od 20 do 80% s 5% točnosti. Brzina uzorkovanja ovog senzora je 1Hz, tj. daje jedno očitanje za svaku sekundu. DHT11 je malih dimenzija s radnim naponom od 3 do 5 volti. Maksimalna struja koja se koristi tijekom mjerjenja je 2,5 mA.

3.4 Relej

Relejni modul od 5 V je jednokanalni ili višekanalni relejni modul koji radi s niskom razinom okidačkog napona od 5 V DC. Ulagani napon može biti iz bilo kojeg mikrokontrolera ili logičkog čipa koji daje digitalni signal. Kao i većina drugih releja, 5V relejni modul je električni elektromagnetski prekidač koji se može koristiti za uključivanje ili isključivanje strujnog kruga. Sastoji se od dva dijela: samog releja i upravljačkog modula. Relej sadrži zavojnicu koja stvara magnetsko polje, armaturu koja se pomiče kako bi dovršila ili odspojila strujni krug i kontakte koji se otvaraju i zatvaraju za rad prekidača opterećenja. Relejni upravljački modul je sučelje ili dio relejnog modula s kojim korisnik komunicira. Sadrži ulazne terminale za spajanje na mikrokontroler, kao i izlazne terminale za spajanje na opterećenje. Upravljački modul također sadrži LED indikatore za napajanje i status te druge uređaje poput zaštitne diode, tranzistora, otpornika i drugih poluvodičkih uređaja potrebnih za njegov rad.

- Normalni napon: 5V DC
- Normalna struja: 70mA
- Maksimalna struja opterećenja: 10A/250V AC, 10A/30V DC
- Maksimalni napon prekidača: 250V AC, 30V DC
- Vrijeme rada: ≤ 10ms
- Vrijeme otpuštanja: ≤ 5ms

Relejni moduli od 5 V sastoje se od spojnih točaka ili pinova i nekoliko glavnih komponenti, kao što su diode, tranzistori, otpornici i sam relej. Oni čine krug koji kontrolira relej.

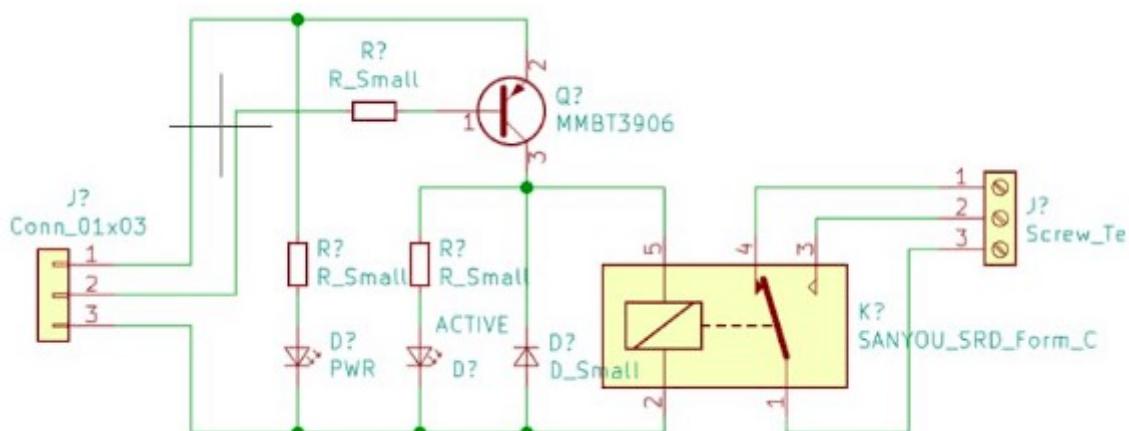


Figure 6 Šema releja

Pinout 5V relejnog modula sastoji se od priključaka na ulaznoj strani gdje prima signal okidača i izlazne strane gdje kontrolira opterećenje. Ulazna strana, kao što je prikazano na gornjem dijagramu strujnog kruga relejnog modula, ima 3 ili 4 priključka:

- VCC – ovo je priključak za napajanje. Napaja 5 V DC modulu i obično je spojen na pozitivni priključak napajanja.
- GND – ovo je uzemljenje. Spaja se na negativni terminal napajanja.
- IN1, IN2 – ovo su ulazi na koje se primjenjuje signal okidača. IN1 je za jednokanalni relejni modul, dok je IN2 za dvokanalni relejni modul. IN (ulazni) pin je spojen na izlaz mikrokontrolera, senzora ili logičkog uređaja.

Izlazna strana relejnog modula ima tri veze:

- NO (Normalno otvoren) – ovo je spoj opterećenja kada je relej uključen. Kada je relej isključen, NO održava otvorenu vezu s COM.
- COM (uobičajeno) – Veza relejnog modula označena s "COM" je uobičajena veza za NO i NC (normalno zatvorene) pinove.
- NC (Normalno zatvoreno) – ovo je priključak opterećenja. Spaja se na COM terminal prema zadanim postavkama ili kada je relej ISKLJUČEN.



Figure 7 Relej 5V

3.5 Keypad 4x4

Matrična tastatura 4×4 je ulazni uređaj, obično se koristi za pružanje ulazne vrijednosti u projektu. Ima ukupno 16 tipki, što znači da može pružiti 16 ulaznih vrijednosti. Najzanimljivije je što je koristi samo 8 GPIO pinova mikrokontrolera. Tipke na keypad-u raspoređeni su u redove i kolone. Keypad 3×4 ima 4 reda i 3 kolone, a keypad 4×4 ima 4 reda i 4 kolone :



Figure 8 Keypad 4x4

Ispod svake tipke nalazi se membranski prekidač. Svaki prekidač u nizu povezan je s ostalim prekidačima u nizu vodljivim tragom ispod podloge. Svaki prekidač u koloni povezan je na isti način - jedna strana prekidača povezana je sa svim ostalim prekidačima u toj koloni vodljivim tragom. Pritisakom na tipku zatvara se prekidač između traga kolone i reda, dopuštajući protok struje između pina kolone i pina reda. Shema tipkovnice 4X4 prikazuje kako su redovi i stupci povezani:

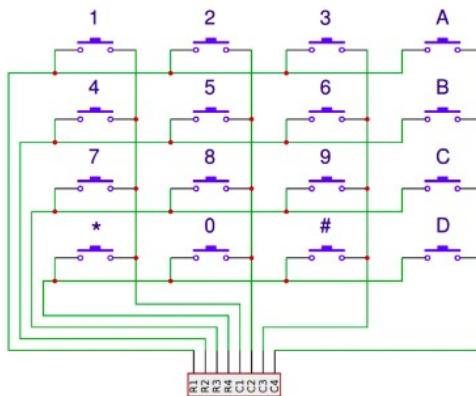


Figure 9 Šematski prikaz keypada

- Otpor kontakta od 500; Otpor izolacije 100 M; Ključna radna sila 150-200 N
- Vrijeme odskoka 1 (ms); Životni vijek od 100 milijuna (puta); Radna temperatura 60
- Oznaka strujnog kruga: 35 V (DC), 100 mA, 1 W; Otpor kontakta: $10 \Omega \sim 500 \Omega$
- Otpor izolacije: $100 \text{ M}\Omega$ 100 V; Dielektrična čvrstoća: 250 VRms ($50 \sim 60 \text{ Hz}$ 1 min);
Treperenje od električnog udara: <5 ms; Životni vijek: taktilni tip: \geq milijun puta
- Radni tlak: Osjećaj dodira: $170 \sim 397 \text{ g}$ ($6 \sim 14 \text{ oz}$); Put prekidača: tip dodira: $0,6 \sim 1,5 \text{ mm}$
- Vibracija: 20G, max. (10 200 Hz, Mil-SLD-202 M204. Uvjet B)

3.6 DC ventilator 12V

Jednostavnim rječnikom, DC ventilator je ventilator za hlađenje koji pretvara električnu energiju u elektromagnetsku energiju putem istosmjernog napona i elektromagnetske indukcije, a zatim pretvara elektromagnetsku energiju u mehaničku energiju, te na kraju u kinetičku energiju, tako da se

lopatice ventilatora okreću . DC ventilator se uglavnom sastoji od četiri dijela: rotor, stator, motor i vanjski okvir. kako slijedi:

- 1. Sastav istosmjernog motora: sastoji se od rotora s permanentnim magnetom, višestepenskog statora namota, senzora položaja i elektroničkog komutacijskog upravljačkog kola.
- 2. Sastav rotora: Sastoji se od ljske motora, trajne magnetske trake, jezgre osovine i lopatica ventilatora.
- 3. Dio statora: emajlirana žica, plastificirani silikonski čelični lim, ležaj, detekcija Hallovog senzora, pogonska sklopna ploča i osovina.

Osnovne komponente DC ventilatora su stator i prijenos. Iz amperovog pravila desne ruke znamo da provodnik prolazi strujom i da se oko njega stvara magnetsko polje. Ako se ovaj provodnik postavi u drugo fiksno magnetsko polje, proizvest će privlačenje ili odbijanje, što će uzrokovati pomicanje objekta. Unutar lopatice ventilatora DC ventilatora pričvršćen je gumeni magnet ispunjen magnetizmom koji okružuje lim od silikonskog čelika, a dio jezgre osovine je namotan s dva seta zavojnica, a komponenta Hallovog senzora koristi se kao sinkroni uređaj za otkrivanje kontrolirati skup sklopova. Kolo čini da dva skupa zavojnica omotanih oko jezgre osovine rade naizmjenično, tako da silikonski čelični lim proizvodi različite magnetske polove, a ovaj magnetski pol i gumeni magnet stvaraju silu odbijanja. Kada je odbojna sila veća od statičkog trenja ventilatora, lopatice ventilatora će se okretati prirodno. Zahvaljujući signalu sinkronizacije koji daju komponente Hallovog senzora, lopatice ventilatora mogu nastaviti raditi, a smjer vrtnje određen je Flemingovim pravilom desne ruke. Ovo je princip rada DC ventilatora.

Postoje tri vrste ventilatora:

- 2-pin ventilatori imaju samo + i - pinove na koje se povezuje napon od 12 V i uzemljenje.
- 3-pin ventilator je najčešće korišten DC ventilator koji ima pin za dovod napajanja, GND pin i signalni pin koji obavještava sustav da li je ventilator uključen ili isključen. Računala prate ovaj signalni pin i, ako signalizira da ventilator ne radi, sustav prijavljuje grešku jer ne može raditi ako nije ispravan ventilator koji hlađi CPU.
- 4-pin ventilator je sličan DC 3-pin ventilatoru, ali ima dodatni PWM pin na koji se dovode PWM impulsi. Ti PWM impulsi kontroliraju brzinu okretanja ventilatora. Ovaj tip ventilatora troši više energije od običnog 3-pin ventilatora.

U našem projektu koristimo samo DC ventilatore koji su prethodno služili kao ventilatori za hlađenje CPU-a (centralne procesorske jedinice). Projekt sušenja drveta sadrži dva PWM-ventilatora. Jedan se nalazi fizički unutar susare i služi za cirkulaciju zraka u samoj susari, dok se drugi nalazi izvan susare, odvojen izolacijskom roletnom, te uvlači hladan zrak iz okoline i ubacuje ga u susaru.S obzirom na to da naši PWM ventilatori imaju neispravan PWM pin koji ne utječe na brzinu okretanja motora dovodenjem PWM impulsa, morali smo pronaći drugi način kontroliranja brzine ventilatora.

- Jedna od mogućnosti bila je dovodjenje otpornosti na pin izvornog napona. Promjenom otpornosti povećava se ili smanjuje struja koja protiče kroz ventilator, a time i brzina okretanja samog ventilatora.
- Druga mogućnost bila je korištenje manjeg izvora napona umjesto 12 V, npr. niskog napona od 11 ili 10 V. Međutim, to bi nam onemogučilo promjenu brzine okretanja jer bi brzina okretanja za taj napon bila fiksna. Ipak, morali smo paziti na minimalni prag vrijednosti ulaznog napona ispod kojeg se ventilator gasi.
- Treća mogućnost bila je korištenje Mosfet tranzistora. Ova mogućnost nam je bila najatraktivnija pa smo je iskoristili.

Našem projektu smo dodali dva N-mosfet tranzistora. Oni služe kao prekidači koji uključuju i isključuju ventilator, a brzinu uključivanja i isključivanja mosfeta kontrolira brzinu okretanja ventilatora. Na Gate mosfet tranzistora dovodimo impulz PWM s mikrokontrolera. Dovođenjem PWM signala na gate može se postići efekat promjene brzine ventilatora u zavisnosti koju vrijednost duty cycle-a pošaljemo. Naš projekat sadrži dva ventilatora, od kojih prvi ima početnu brzinu 10% maksimalne brzine a drugi ima početnu brzinu od 0%. Promjena njihove brzine ovisi o temperaturi i vlažnosti. Kada je temperatura manja od 26 stepeni ventilatori rade sa zadanim početnim vrijednostima. Kada je temperatura između 26 i 28 stepeni oba ventilatora povećavaju svoju brzinu za 2% u iterativnom procesu tokom vremena. Kada je temperatura između 28 i 31 stepen ventilatori povećavaju svoju brzinu kroz svaku iteraciju za 5%. Postavljanjem duty-a u kodu određujemo kakav signal šaljemo na gate tranzistora. Kada je vlažnost drveta manja od željene vlažnosti ventilatori se postavljaju da rade na 100% jer će tada maketa biti otvorena radi ispuštanja toplog vazduha. Source mosfeta je spojen na GND našeg punjača od 12 V, a Drain mosfeta je spojen na minus pol ventilatora. Impulsi poslani na Gate kada su na visokom naponu uključuju mosfet, tj. provodi se struja između Source-a i Drain-a, dok kada su na niskom naponu gase mosfet, pa nema protoka struje. Ovaj proces uključivanja i isključivanja mosfeta obavlja se na jako visokoj frekvenciji od 25 kHz, jer je za normalan rad ventilatora potrebna frekvencija od 25 kHz i više.



Figure 10 DC ventilator 12V

3.7 LCD 16x2

Izraz LCD označava tekući kristalni zaslon. To je jedna vrsta električkog modula za prikaz koji se koristi u različitim aplikacijama poput različitih krugova i uređaja kao što su mobilni telefoni, kalkulatori, računala, televizori itd. Glavne prednosti korištenja ovog modula su niska cijena, jednostavno programiranje, animacije i nema ograničenja za prikazivanje prilagođenih znakova, posebnih znakova pa čak i animacija, itd.



Figure 11 LCD 16x2

LCD 16x2 sastoji se od 16 pinova prikazanih na sljedećoj slici:

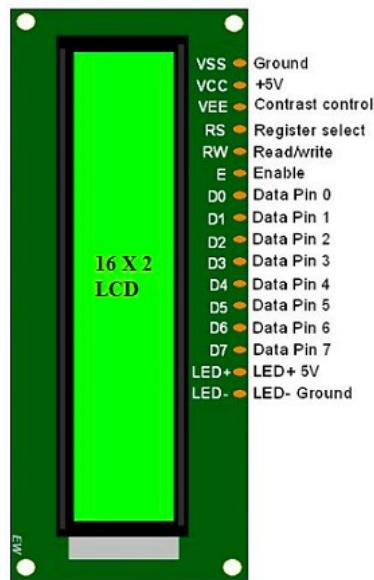


Figure 12 LCD 16x2 pin diagram

- Pin1 (Ground/Source Pin): Ovo je GND pin zaslona, koristi se za povezivanje GND terminala mikrokontrolera ili izvora napajanja.
- Pin2 (VCC/Source Pin): Ovo je pin za napajanje napona zaslona, koristi se za povezivanje napajanja izvora napajanja.

- Pin3 (V0/VEE/Control Pin): Ovaj pin regulira kontrast zaslona, koristi se za povezivanje promjenjivog POT-a koji može isporučivati 0 do 5V.
- Pin4 (Register Select/Control Pin): Ovaj pin prebacuje između registra naredbi ili podataka, koristi se za povezivanje pinova mikrokontrolera i dobiva vrijednost 0 ili 1 (0 = način podataka, 1 = način naredbi).
- Pin5 (Read/Write/Control Pin): Ovaj pin prebacuje zaslon između operacija čitanja ili pisanja i povezan je s pinom mikrokontrolera kako bi dobio vrijednost 0 ili 1 (0 = operacija pisanja, 1 = operacija čitanja).
- Pin 6 (Enable/Control Pin): Ovaj pin treba držati visokim za izvođenje operacije čitanja/pisanja i povezan je s mikrokontrolerom te ga konstantno drži na visokoj razini.
- Pinovi 7-14 (Pinovi za podatke): Ovi pinovi koriste se za slanje podataka na zaslon. Ovi pinovi su povezani u dvonitni način rada, kao što su 4-nitni način rada i 8-nitni način rada. U 4-nitnom načinu rada, samo četiri pina su povezana s mikrokontrolerom kao 0 do 3, dok se u 8-nitnom načinu rada povezuje 8 pinova s mikrokontrolerom kao 0 do 7.
- Pin15 (+ve pin za LED): Ovaj pin je povezan s +5V.
- Pin 16 (-ve pin za LED): Ovaj pin je povezan s GND

3.8 PTC grijaci

PTC je akronim za pozitivan temperaturni koeficijent. Ovi grijaci su grijace ploče izrađene od keramičkih materijala. Istraživanja su pokazala da toplina proizvedena ovom tehnologijom energetski je učinkovita, snažna, stabilna i trajna, što je čini posebno primjenjivom u kompaktnim prostorima poput kućanskih aparata. PTC grijaci rade u otvorenoj petlji (open-loop), što znači da im nije potrebna povratna kontrola niti vanjska dijagnostika. To ih čini mnogo sigurnijim jer postoji nizak rizik od pregrijavanja. Za uređaje koji zahtijevaju sigurno, jednolično i snažno grijanje, PTC grijaci su pouzdana opcija.



Figure 13 PTC grijac

PTC grijaci izrađeni su od naprednih provodnih keramičkih materijala. Imaju pozitivnu promjenu otpora kada su izloženi povećanju temperature. Kako se element izlaže većoj toplini, keramika ograničava prolaz struje kroz komponentu. Suprotno tome, kad je izložen nižoj temperaturi, povećava se prolaz struje. To je način na koji PTC grijac može stvarati, održavati i samopraćenje topline. Istraživanja podupiru učinkovitost PTC tehnologija za grijanje u različitim aplikacijama. Izvrsno se pokazuju i omogućavaju proizvođačima aparata da odrede određene temperaturne granice unutar kojih grijac treba funkcionirati.

PTC grijaci se u našem projektu koriste u kombinaciji sa relejom koji upravlja grijacima na način da ih pali i gasi po potrebi. U našem projektu susara drveta koristili smo PTC grijace čiji otpor raste sa

povećanjem temperature, dok se smanjuje sa padom temperature, što rezultira povećanom strujom kroz pločicu. Kroz dugotrajno testiranje naše susare u različitim temperaturnim uvjetima i s različitim kodovima, morali smo eksperimentalno prilagoditi naš kod za različita godišnja doba. Takoder, primijetili smo da drvo reagira drugačije na različite temperature sušenja, što smo odražavali u našem kodu kroz konstante eksperimentalno određene za dati raspon temperatura koji koristimo u našem projektu, a koji se kreće samo do 31°C.

3.9 N-kanalni MOSFET

N-kanalni MOSFET je tip MOSFET-a u kojem se kanal MOSFET-a sastoji od većine elektrona kao nosilaca struje. Kada je MOSFET aktiviran i uključen, većina struje koja teče su elektroni koji se kreću kroz kanal. Postoje 2 tipa N-kanalnih MOSFET-ova, MOSFET-ovi obogaćenog i tipa MOSFET-ovi tipa osiromašenja. Terminali GATE-a su napravljeni od P materijala. Ovisno o količini i vrsti napona (negativan ili pozitivan) određuje kako će tranzistor raditi da li se uključuje ili isključuje.

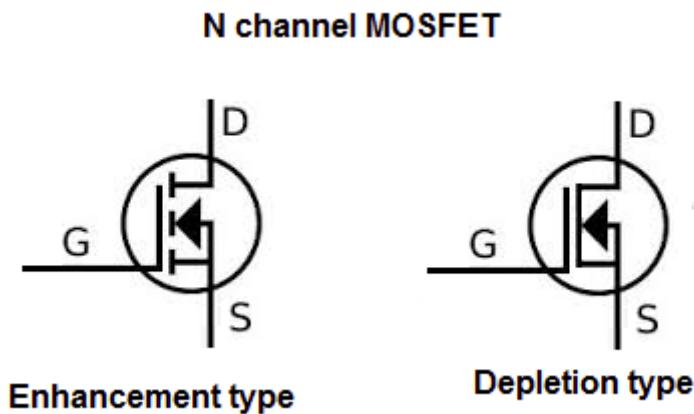


Figure 14 Tipovi MOSFET-a

Uključivanje obogaćenog MOSFET-a

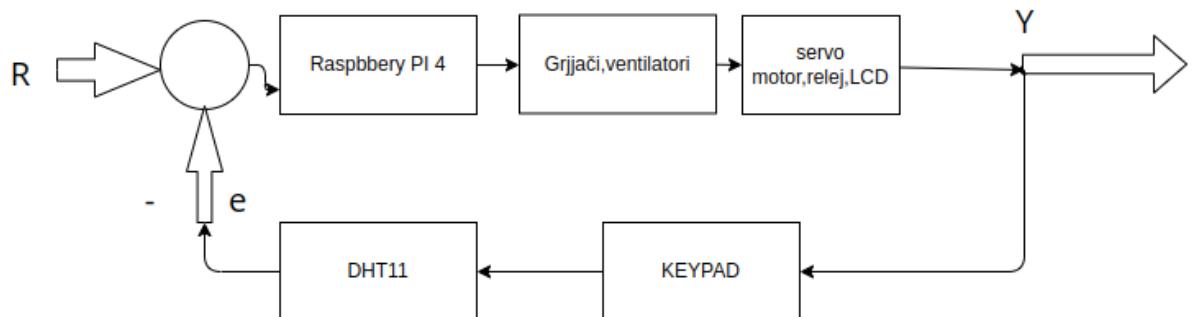
Da biste uključili MOSFET tipa N-Channel Enhancement, primijenite dovoljan pozitivan napon VDD na odvod tranzistora i dovoljan pozitivan napon na gejt tranzistora. Ovo će omogućiti struji da teče kroz kanal odvod-izvor. Dakle, sa dovoljnim pozitivnim naponom, VDD, i dovoljnim pozitivnim naponom primjenjenim na gejt, MOSFET tipa N-Channel Enhancement je potpuno funkcionalan i u 'ON' operaciji.

Isključivanje MOSFET-a

Da biste isključili N-kanalni MOSFET za poboljšanje, možete poduzeti 2 koraka. Možete ili prekinuti pozitivni napon prednapona, VDD, koji napaja odvod. Ili možete isključiti pozitivni napon koji ide do gejta tranzistora.

4. Algoritam upravljanja sušarom drveta

Prvo je ručno potrebno unutar makete ubaciti drvo koje je potrebno osušiti. Na lcd displeju nam se prikaže poruka da ubacimo drvo tek kada odaberemo jednu od opcija za sušenje drveta. Nakon ubacivanja drveta u sušaru kreće proces sušenja koji se odvija u beskonačnoj petlji. DHT 11 senzor unutar makete zabilježi trenutnu temperaturu i vlažnost makete prije ubacivanja samog drveta. Te vrijednosti se pohrane u variable kako bi se koristile za kasnije poređenje. DHT 11 senzor nakon ubacivanja drveta počinje da mjeri vlažnost i temperaturu samog drveta na način da je drvo pribijeno uz sam senzor. Kada je drvo ubačeno u zavisnosti od vlažnosti makete i vlažnosti samog drveta uključuju se grijaci i ventilatori unutar makete kako bi se regulirala vlažnost drveta na neku optimalnu vrijednost. Grijaci se aktiviraju pomoću releja kojeg upravlja Raspberi PI 4. Raspberi također upravlja i sa ostalim komponentama (ventilatorima, senzorom temperature, tastaturom, displejom, step motorom). Kada je drvo ubačeno aktivira se i servo motor koji zatvara maketu. Unutar beskonačne petlje uključuju se ili isključuju grijaci, ventilatori, step motor u zavisnosti da li su temperatura i vlažnost drveta zadovoljavajući. Pomoću raspberi-a kontrolišemo i prilagođavamo brzinu samih ventilatora u maketi radi bržeg ili sporijeg sušenja. Kada vlažnost drveta poprimi neku optimalnu vrijednost tada će dio tog procesa krenuti ka zavšetku na način da se petlja završi te kreće proces u kojem se otvara i rashlađuje maketa, gase grijaci. Rashlađivanje makete je određeno kratkim vremenskim periodom kako bi se ponovo stekli uslovi za sušenje sljedećeg drveta. Kada je proces završen lcd displej nas o tome obavijesti te kaže da drvo izvadimo. Imamo određeni vremenski period da drvo uklonimo iz makete. Nakon što je drvo uklonjeno na lcd displeju nam se ponovo nude opcije sušenja kako bi ponovo mogli pristupiti procesu sušenja.



Slika 1 Blok dijagram upravljanja sušarom

4.1 Ulazi i izlazi

Ulazi:

- Senzor temperature i vlage DHT11 —> mjeri temperaturu i vlažnost vazduha u okolini i vlažnost i temperaturu samog drveta.
- Tasteri na tastaturi (keypad) —> Koristi se za unos komandi ili vrijednosti u sistem.

Izlazi:

- LCD displej —> Prikazuje informacije o trenutnoj temperaturi i vlažnosti, kao i druge informacije koje želite prikazati korisniku.

- DC ventilatori (12V) —> Koriste se za kontrolu temperature unutar sistema.
- PTC grijajući —> Koriste se za povećanje temperature unutar sistema.
- Relej —> Kontroliše napajanje PTC grijajuća u skladu sa potrebama sistema.
- Servo motor —> otvara i zatvara maketu.

Raspberry PI 4	
Ulazi	Izlazi
DHT11 senzor temperature i vlažnosti	DC ventilator 12V x2 Grijajući x2
Keypad	Servo motor, relej
	LCD displej

Tabela 2 Ulazi i izlazi na Raspberry pi

4.2 Šema povezivanja

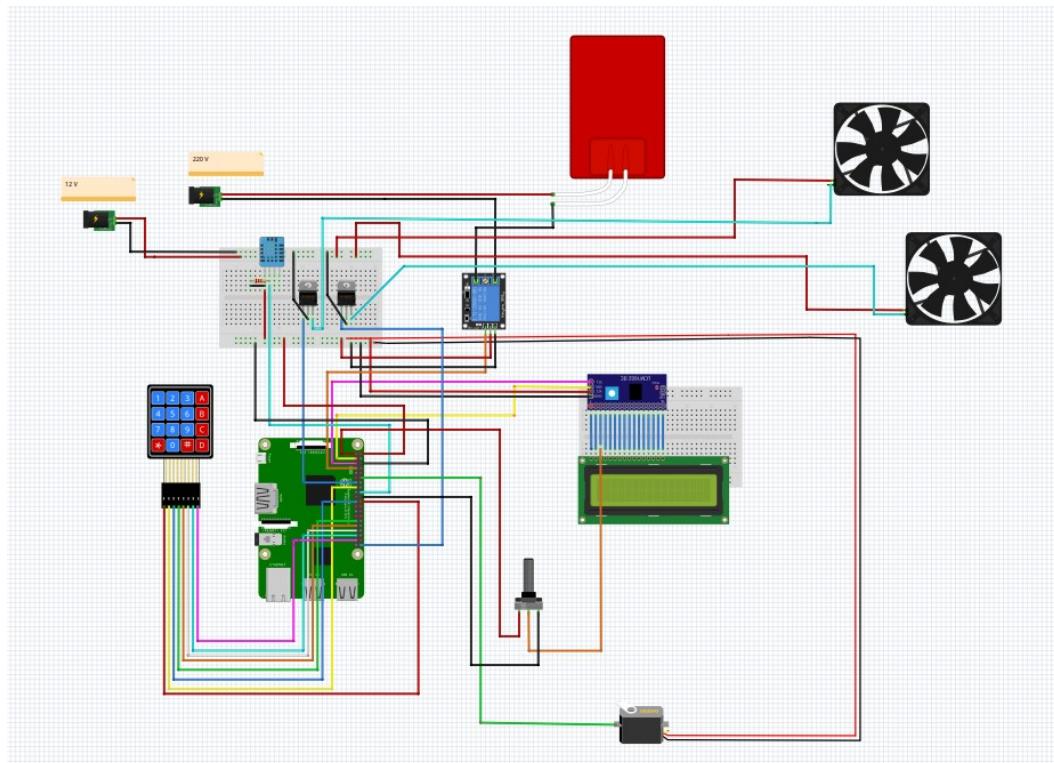


Figure 15 Šema povezivanja

4.3 Grafički odziv

Sa grafičkog prikaza vidimo da će se povećanjem temperature u našoj maketi pomoći grijajuća i ventilatora koji raspršuju toplotu vlažnost drveta smanjivati tokom vremena. Kada drvo dosegne određenu vlažnost dobijamo obavijest na našem lcd displeju da je drvo suho. Zatim ponovo imamo

mogućnost sušenja drugog drveta. Informacije o temperaturi i vlažnosti su prikupljene tokom samog procesa sušenja i isporučene u tekstualni fajl, zatim su te vrijednosti korištene u kodu ispod kako bi se dobio odziv.

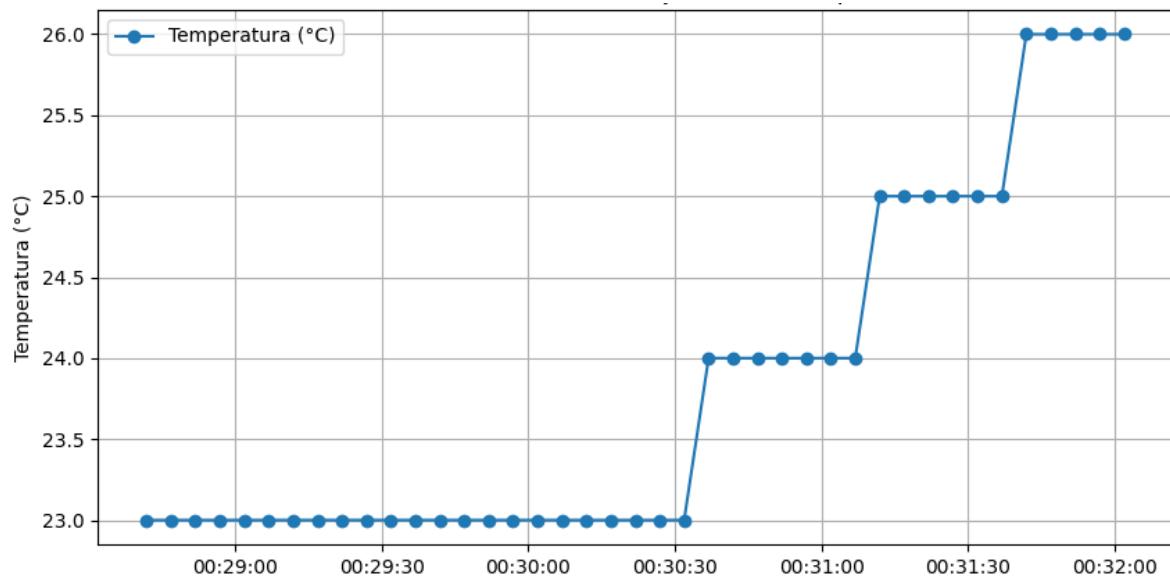


Figure 16 Temperatura

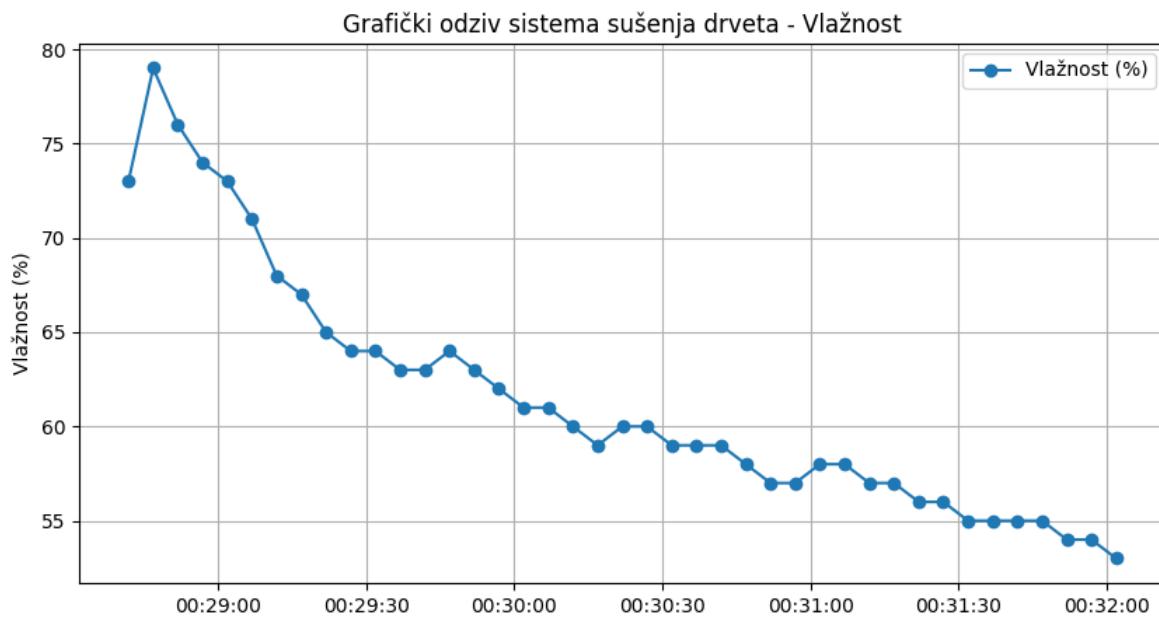


Figure 17 Odziv-vlažnost

```
import matplotlib.pyplot as plt
import datetime

# Podaci o temperaturi i vlažnosti drveta tokom vremena
start_time = datetime.datetime.now()

temperatura =
```

Kod 1 Skripta za prikaz odziva vlažnosti drveta u zavisnosti od temperature

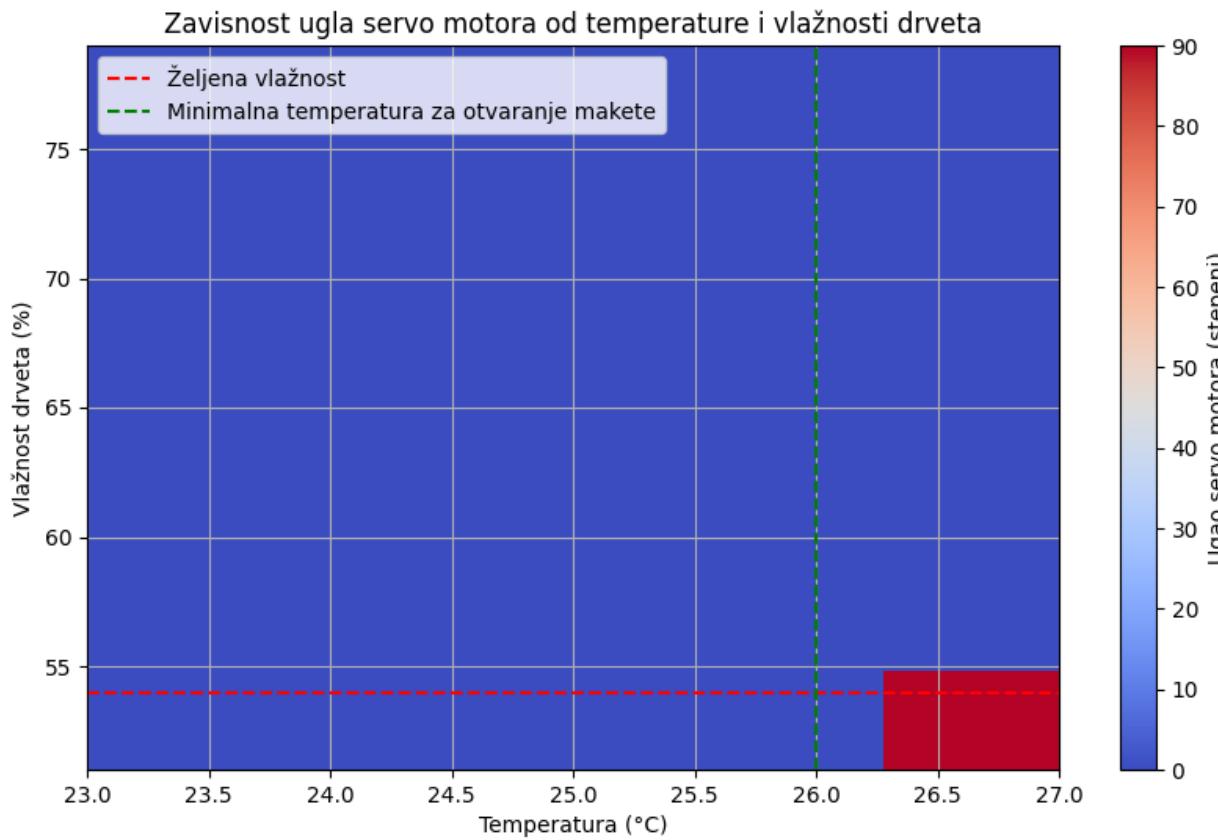


Figure 18 Odziv servo motora u zavisnosti od temperature i vlažnosti

Na ovom grafiku možemo vidjeti da će ugao servo motora biti 90 stepeni , to jeste da će maketa biti otvorena u slučaju kada je temperatura unutar makete veća od 26 stepeni i ako je vlažnost drveta manja od željene vlažnosti. U drugom slučaju (označen plavom bojom) ugao će biti 0 stepeni to jeste maketa ce biti zatvorena kada je vlažnost drveta veća od željene vlažnosti.

```
import matplotlib.pyplot as plt
import numpy as np

# Vlažnost drveta, temperatura i odgovarajući uglovi servo motora
vlažnost_drveta =
[73,79,76,74,73,71,68,67,65,64,64,63,63,64,63,62,61,61,60,59,60,60,59,59,59,58,57,57,58,58,57,
57,56,56,55,55,55,55,54,54,53,52,52,51]

temperatura =
[23,23,23,23,23,23,23,23,23,23,23,23,23,23,23,23,23,23,23,23,23,23,23,23,24,24,24,24,24,24,25,
25,25,25,25,26,26,26,26,26,26,27,27]

zeljena_vlaznost = 54
granicna_temp=26
```

```

uglovi_servo_motora = np.zeros((len(vlažnost_drveta), len(temperatura))) # Inicijalizacija matrice uglova servo motora

for i, temp in enumerate(temperatura):
    for j, v in enumerate(vlažnost_drveta):
        if temp >= 26 and v <= zeljena_vlaznost:
            uglovi_servo_motora[j, i] = 90 # Ugao kada je temperatura veća od 27 i vlažnost drveta veća od željene vlažnosti
        if temp < 26:
            uglovi_servo_motora[j, i] = 0 # Ugao kada je temperatura manja ili jednaka 27 i vlažnost drveta manja ili jednaka željenoj vlažnosti

# Prikaz heatmap grafikona
plt.figure(figsize=(10, 6))

plt.imshow(uglovi_servo_motora, cmap='coolwarm', extent=[min(temperatura), max(temperatura), min(vlažnost_drveta), max(vlažnost_drveta)], aspect='auto')
plt.colorbar(label='Ugao servo motora (stepeni)')
plt.xlabel('Temperatura (°C)')
plt.ylabel('Vlažnost drveta (%)')
plt.title('Zavisnost ugla servo motora od temperature i vlažnosti drveta')
plt.grid(True)

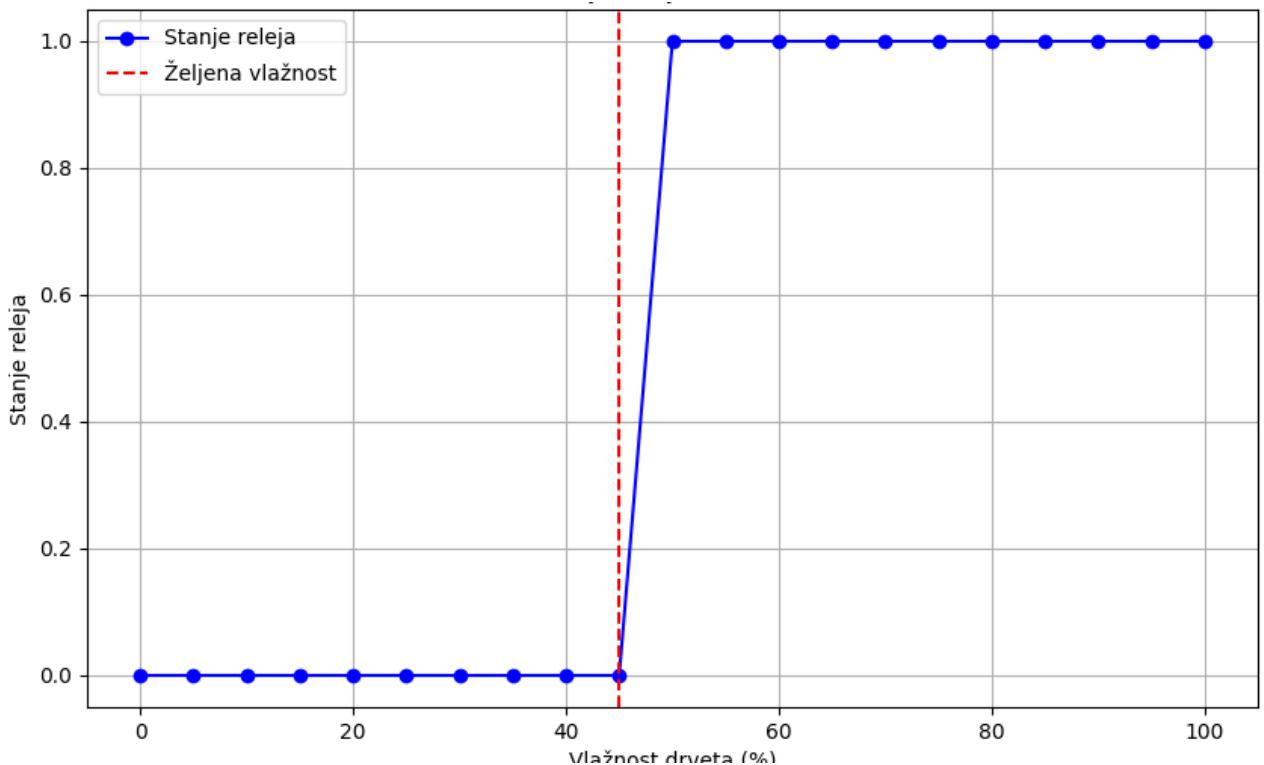
# Dodavanje oznake za željenu vlažnost
plt.axhline(y=zeljena_vlaznost, color='r', linestyle='--', label='Željena vlažnost')

# Dodavanje oznake za temperaturu
plt.axvline(x=26, color='g', linestyle='--', label='Minimalna temperatura za otvaranje makete')

plt.legend() # Dodavanje legende
plt.show()

```

Kod 2 Odziv servo motora u zavisnosti od teperature i vlažnosti



Kod 3 Odziv grijaca u zavisnosti od vlažnosti drveta

Sa ovog grafika možemo zaključiti da će relaj biti u stanju logičke jedinice (5V) u slučaju kada je vlažnost drveta veća od željene vlažnosti. Tada će grijaci biti uključeni pomoću relaja. U drugom slučaju grijaci će biti isključeni a relaj u stanju logičke nule.

```
import matplotlib.pyplot as plt

# Vlažnost drveta i odgovarajuće stanje releja
vlažnost_drveta = range(0, 101, 5) # Vlažnost od 0 do 100%, korak 5
stanje_releja = []
zeljena_vlaznost = 45 # Željena vlažnost drveta

for v in vlažnost_drveta:
    if v <= zeljena_vlaznost:
        stanje_releja.append(0)
    else:
        stanje_releja.append(1)

# Plotovanje zavisnosti stanja releja od vlažnosti drveta
plt.figure(figsize=(10, 6))
```

```

plt.plot(vlažnost_drveta, stanje_releja, marker='o', linestyle='-', color='blue', label='Stanje releja')
plt.axvline(x=zeljena_vlaznost, color='r', linestyle='--', label='Željena vlažnost')
plt.xlabel('Vlažnost drveta (%)')
plt.ylabel('Stanje releja')
plt.title('Zavisnost stanja releja od vlažnosti drveta')
plt.legend()
plt.grid(True)
plt.show()

```

Kod 4 Odziv releja (grijača)

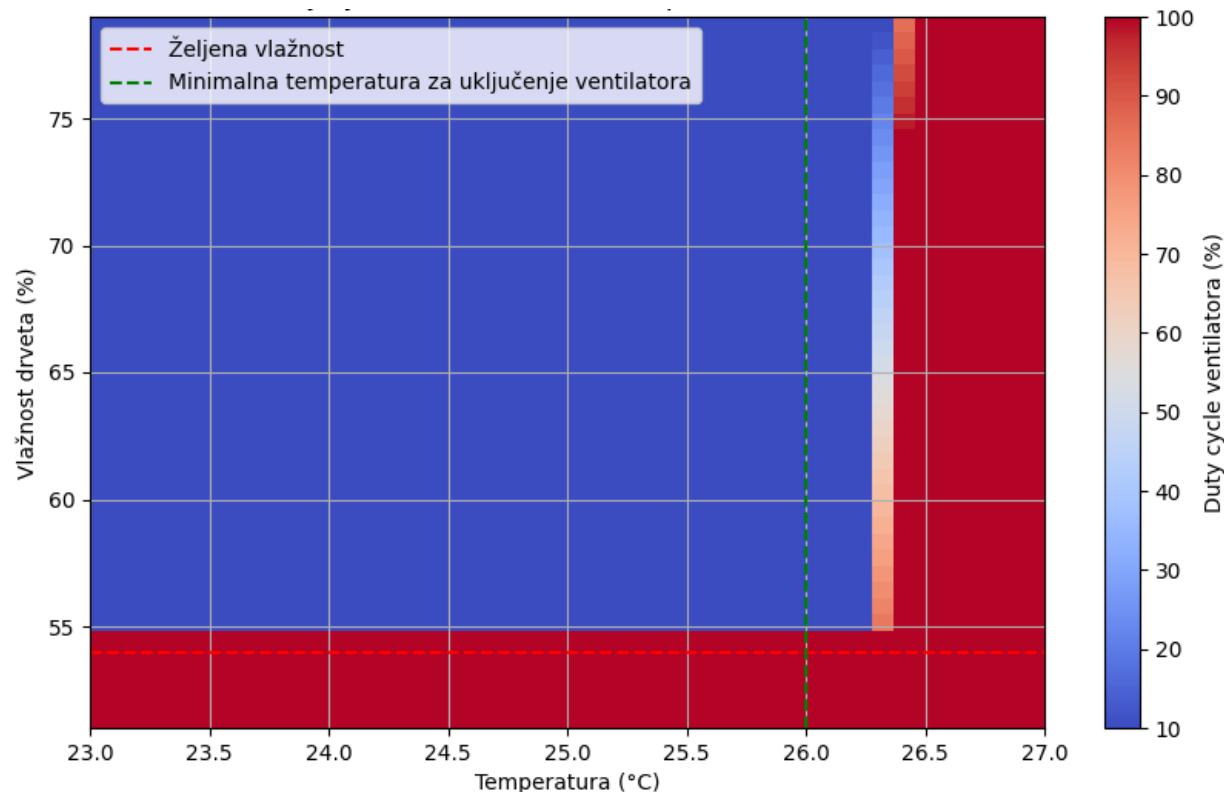


Figure 19 Odziv ventilatora u zavisnosti od temperature i vlage

Sa grafika možemo vidjeti ukoliko je vlažnost drveta manja ili jednaka željenoj vlažnosti da će brzina ventilatora biti 100% jer će tada krenuti proces hlađenja makete gdje će maketa biti otvorena, a ukoliko je vlažnost drveta veća od željene vrijednosti onda će brzina odnosno duty cycle ventilatora da se mijenja od 0 do 100 pri zatvorenoj maketi, to jeste povećavat će svoju brzinu u zavisnosti od temperature i vlažnosti.

```

import matplotlib.pyplot as plt
import numpy as np

# Vlažnost drveta, temperatura i odgovarajući duty cycle za ventilator
vlažnost_drveta = [73, 79, 76, 74, 73, 71, 68, 67, 65, 64, 64, 63, 63, 64, 63, 62, 61, 61, 60, 59, 60,

```



```

plt.ylabel('Vlažnost drveta (%)')
plt.title('Zavisnost duty cycle-a ventilatora od temperature i vlažnosti drveta')
plt.grid(True)

# Dodavanje oznake za željenu vlažnost
zeljena_vlaznost = 54
plt.axhline(y=zeljena_vlaznost, color='r', linestyle='--', label='Željena vlažnost')

# Dodavanje oznake za minimalnu temperaturu za uključenje ventilatora
granicna_temp = 26
plt.axvline(x=granicna_temp, color='g', linestyle='--', label='Minimalna temperatura za uključenje
ventilatora')

plt.legend() # Dodavanje legende
plt.show()

```

Kod 5 Odziv ventilatora

4.4 Tabelarni prikaz

Lozinka is-pravna	Sušenje _1	Sušenje _2	Vлага drveta	Ventilator 1	Ventilator 2	Grijači	Servo motor	Opis
0	0	0	/	0	0	0	zatvoren-0	Lozinka ponovo
1	1	0	zadovoljena	0	0	0	Otvaranje makete radi ispuštanja zraka(1)	1-Osušeno 2. Unos lozinke Novi proces
1	1	0	Nije zadovoljena	1	1	1	0	Nastavlja se proces sušenja dok ne bude

									zadovoljeno
1	0	1	zadovoljena	0	0	0	1		1-Osušeno 2. Unos lozinke Novi proces
1	0	1	Nije zadovoljeno	1	1	1	0		Nastavlja se proces sušenja dok ne bude zadovoljeno

Tabela 3 Ponašanje sistema

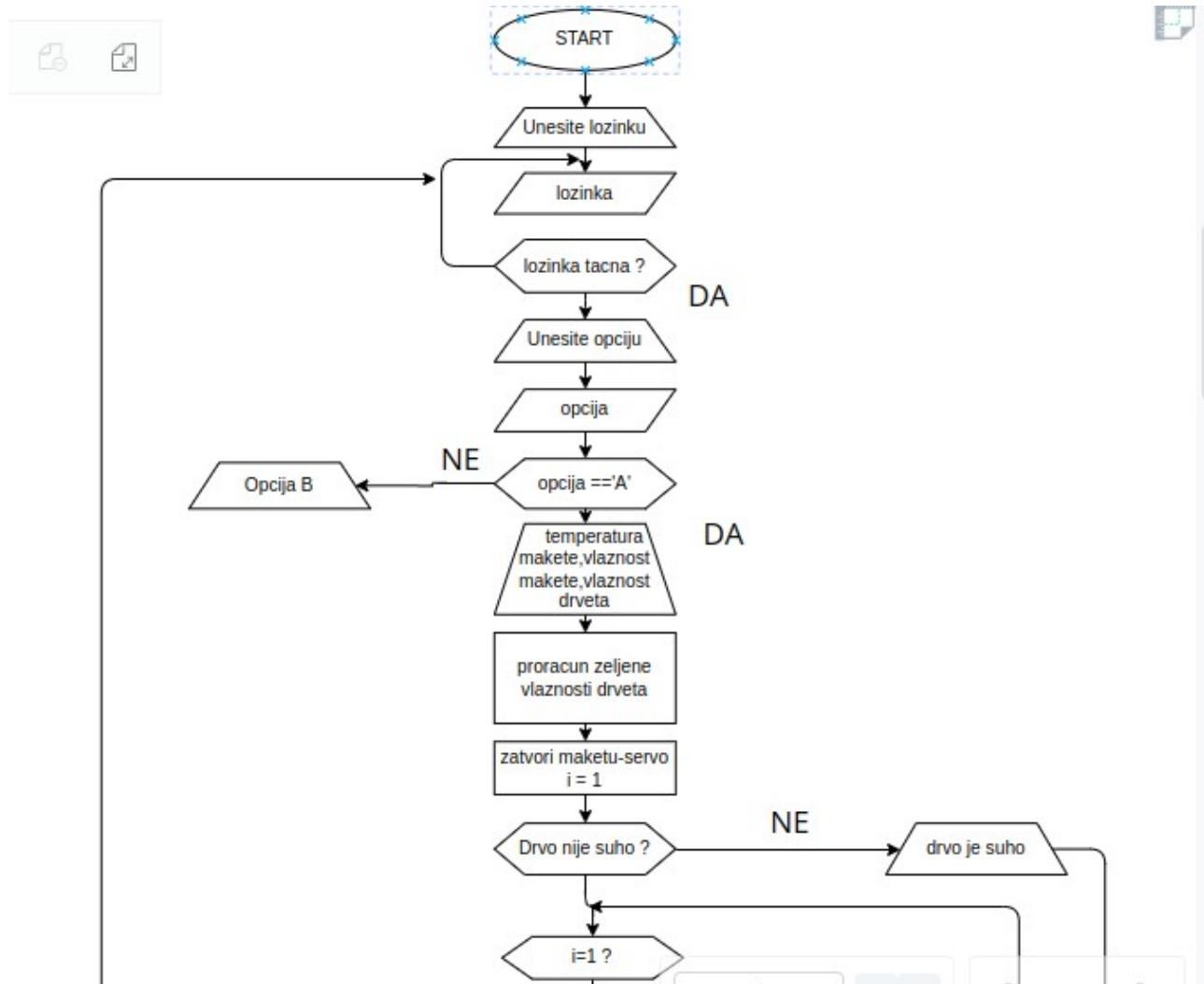


Figure 20 Blok dijagram dio 1

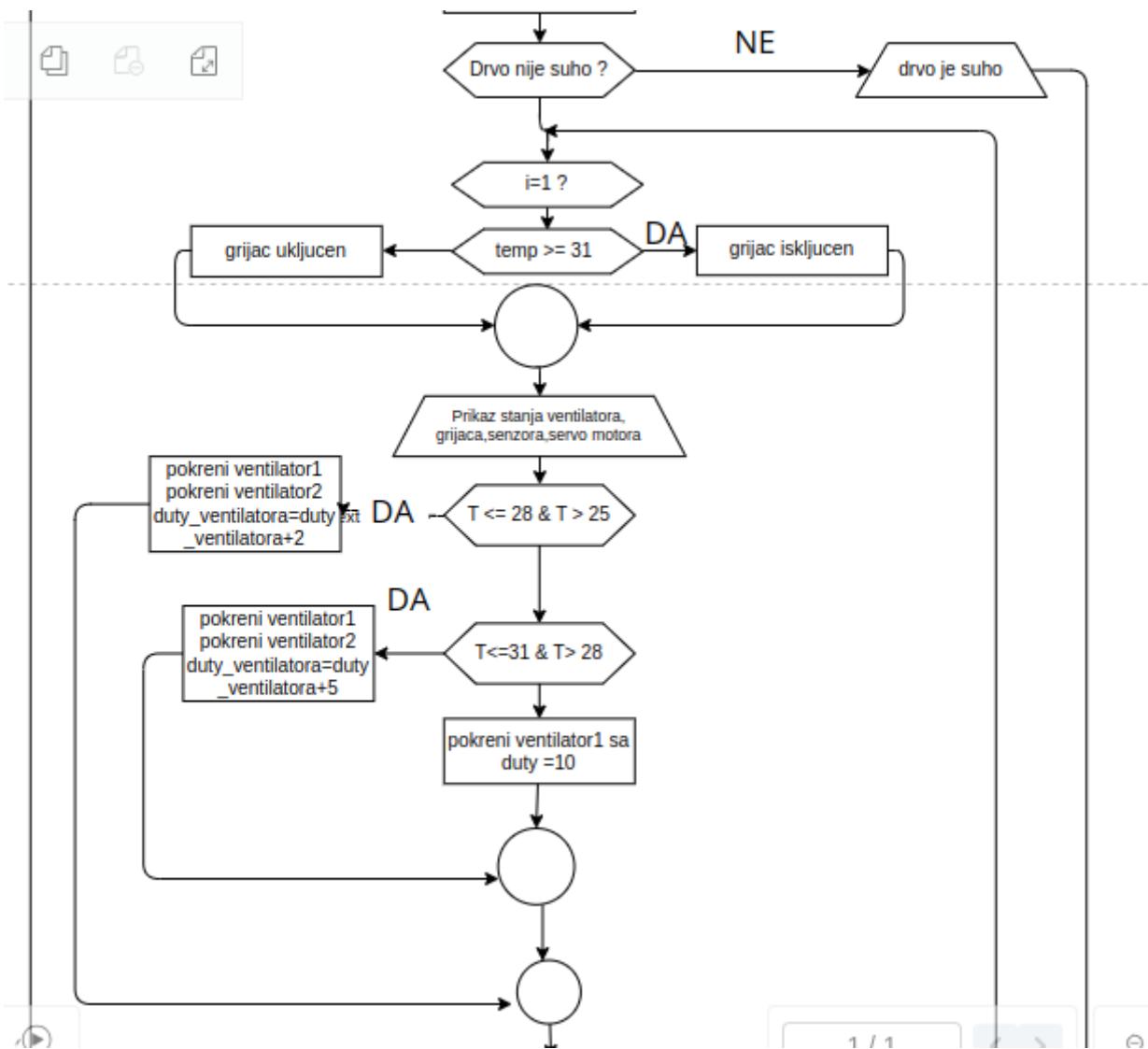


Figure 21 Blok dijagram dio 2

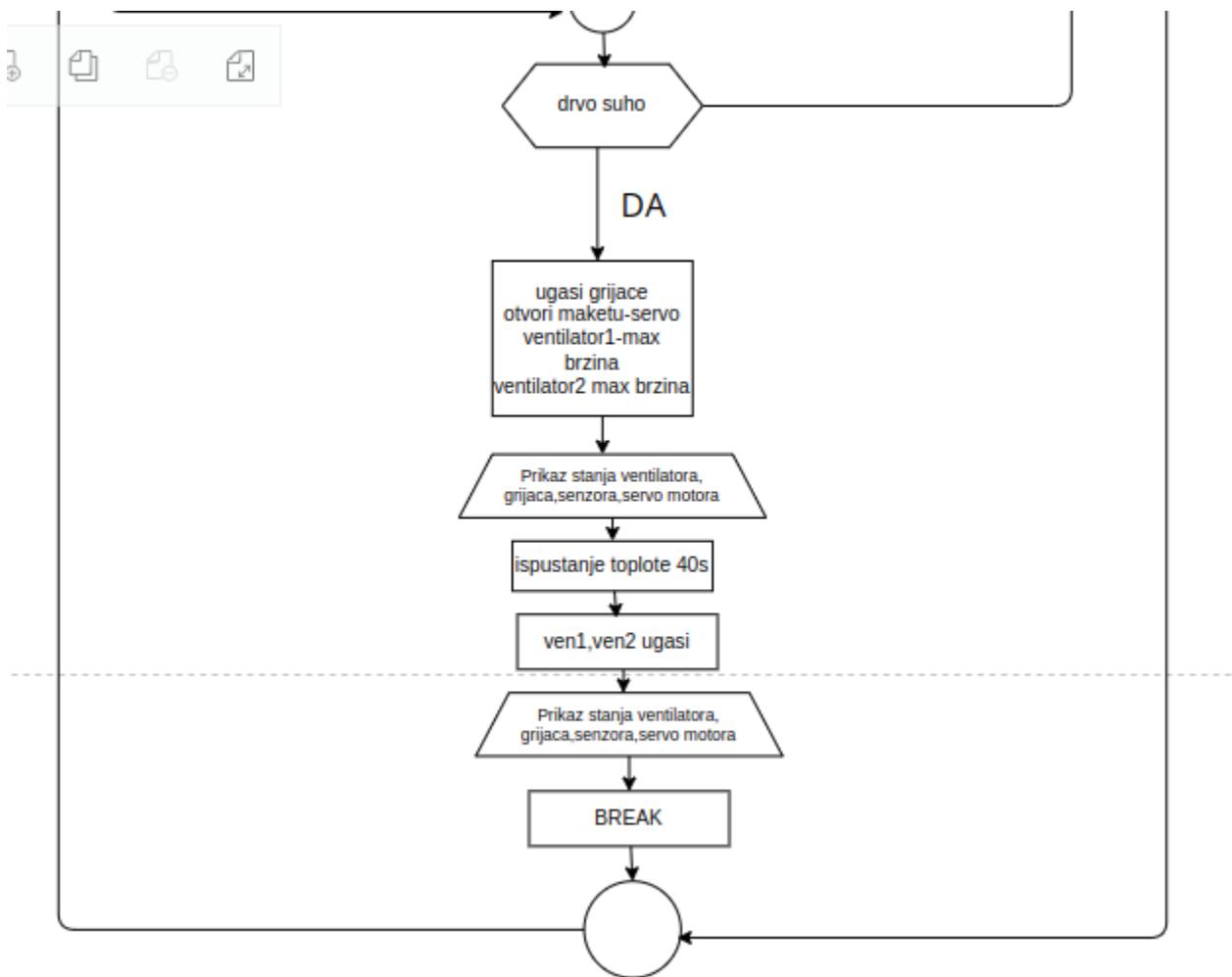


Figure 22 Blok dijagram dio 3

4.5 Slike makete



Figure 23 Maketa 1



Figure 24 Maketa 2



Figure 25 Maketa 3

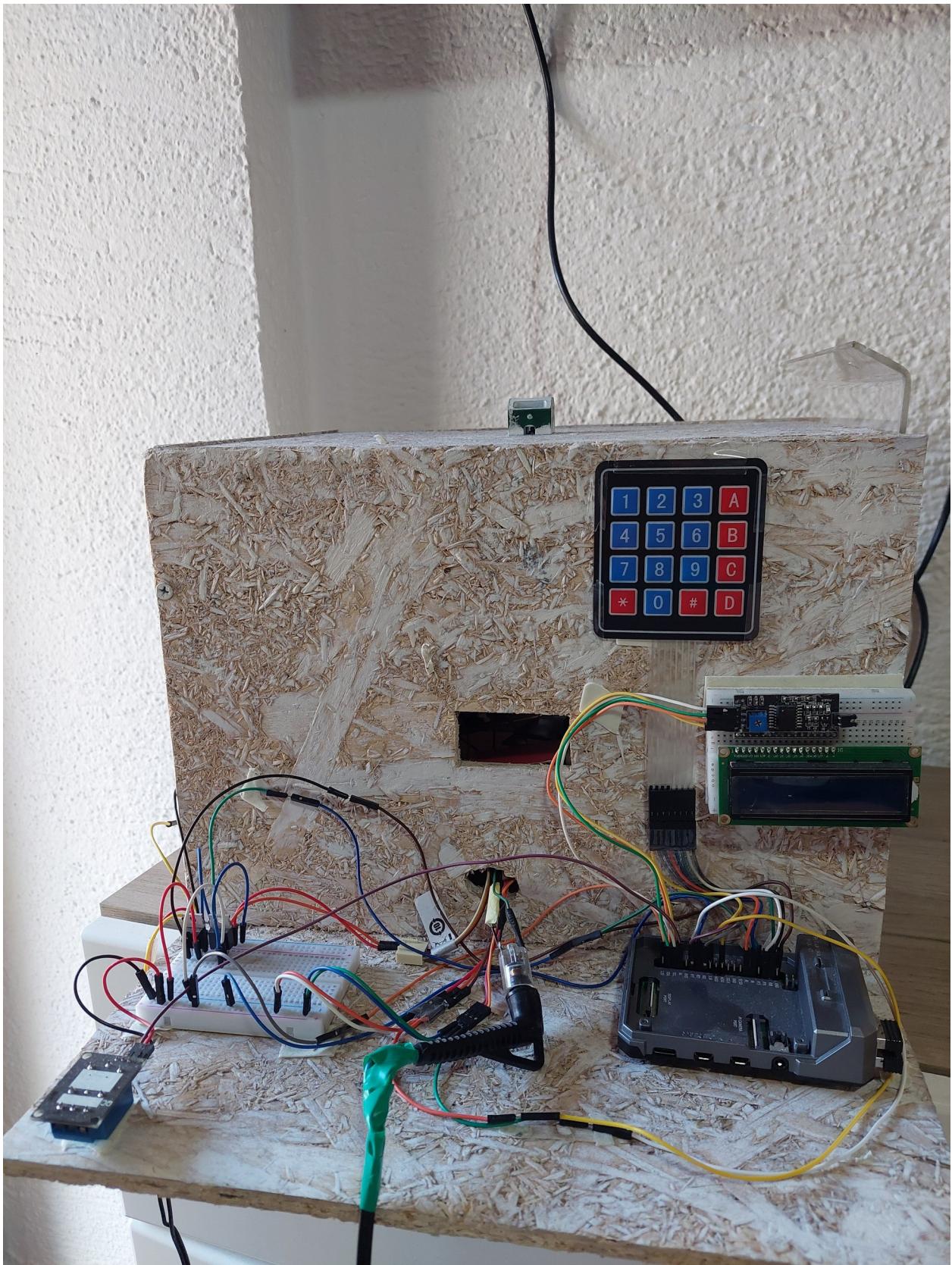


Figure 26 Maketa 4

5. KODOVI

```
import time
import I2C_LCD_driver
import RPi.GPIO as GPIO
import sys
import Adafruit_DHT
from pad4pi import rpi_gpio
from gpiozero import Servo

temperatura = []
vlaznost = []

KEYPAD = [
    [1, 2, 3, 'A'],
    [4, 5, 6, 'B'],
    [7, 8, 9, 'C'],
    ['*', 0, '#', 'D']
]

factory = rpi_gpio.KeypadFactory()

# these GPIO pins are connected to the keypad
# change these according to your connections!
L1 = 25
L2 = 22
L3 = 27
L4 = 5

C1 = 6
```

```

C2 = 13
C3 = 19
C4 = 26

# Initialize the GPIO pins

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)

GPIO.setup(L1, GPIO.OUT)
GPIO.setup(L2, GPIO.OUT)
GPIO.setup(L3, GPIO.OUT)
GPIO.setup(L4, GPIO.OUT)

# Make sure to configure the input pins to use the internal pull-down resistors

GPIO.setup(C1, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(C2, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(C3, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(C4, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

ROW_PINS = [L1, L2, L3, L4]
COL_PINS = [C1, C2, C3, C4]

keypad = factory.create_keypad(keypad=KEYPAD, row_pins=ROW_PINS,
col_pins=COL_PINS)

GPIO.setmode(GPIO.BCM)

def readLine(line, characters):
    GPIO.output(line, GPIO.HIGH)

```

```

if(GPIO.input(C1) == 1):
    print(characters[0])
if(GPIO.input(C2) == 1):
    print(characters[1])
if(GPIO.input(C3) == 1):
    print(characters[2])
if(GPIO.input(C4) == 1):
    print(characters[3])
GPIO.output(line, GPIO.LOW)

# Inicijalizacija LCD ekrana
lcd=I2C_LCD_driver.lcd()

# Funkcija za prikaz poruke na LCD ekranu
def prikazi_poruku(poruka,red=1):
    lcd.lcd_clear()
    lcd.lcd_display_string(poruka,red)

# Inicijalizacija promenljive za unos šifre
unesena_sifra = ""
ispravna_sifra = "333"

#servo=AngularServo(18,min_pulse_width=0.0006,max_pulse_width=0.0023)
servo_pin=18
GPIO.setup(servo_pin,GPIO.OUT)
servo1= GPIO.PWM(servo_pin,50)
servo1.start(0)

```

```
fan_pin=27
```

```
GPIO.setup(fan_pin,GPIO.OUT)
#GPIO.output(fan_pin, GPIO.HIGH)
fan_pwm=GPIO.PWM(fan_pin,25000)
fan_pwm.start(0)
```

```
fan_pin2=21
```

```
GPIO.setup(fan_pin2,GPIO.OUT)
#GPIO.output(fan_pin, GPIO.HIGH)
fan_pwm2=GPIO.PWM(fan_pin2,25000)
fan_pwm2.start(0)
```

```
#REleJ
```

```
GPIO.setup(4,GPIO.OUT)
GPIO.output(4,GPIO.LOW)
```

```
# Beskonačna petlja za nastavak programa
```

```
while True:
```

```

prikazi_poruku("Unesite sifru1:")

# Glavna petlja za unos šifre
while len(unesena_sifra) != 3: # Korisnik treba uneti tačno 3 broja
    key = keypad.getKey()
    if key:
        unesena_sifra += str(key)
        prikazi_poruku("Unesite sif:" + "*" * len(unesena_sifra))
        time.sleep(0.2)

if unesena_sifra == ispravna_sifra:
    prikazi_poruku("Sifra je tacna. ")
    unesena_sifra = ""

# Display the available options to the user
prikazi_poruku("Izaberite opciju:")
time.sleep(1.0) # Sleep for 1 second

prikazi_poruku("A: Kvalitetno")
lcd.lcd_display_string("B:Drvo za ogrev",2)

time.sleep(1.0) # Sleep for 1 second
choice = None

# Wait for user input (A or B)
while choice not in ('A', 'B'):
    key = keypad.getKey()
    if key and key in ('A', 'B'):
        choice = key
        if choice == 'A':

            #servo1.ChangeDutyCycle(9)
            servo1.ChangeDutyCycle(12)

```

```

time.sleep(0.5)
servo1.ChangeDutyCycle(0)
time.sleep(0.3)

prikazi_poruku("Kvalitetno (A)")

n=4
for i in range(n):
    humidity_prostora,temp_prostora= Adafruit_DHT.read_retry(11, 24)
    time.sleep(1.4)

    humidity_str1 = f"Humidity: {humidity_prostora:.2f}%""
    temperature_str1 = f"Temp: {temp_prostora:.2f}°C"

    lcd.lcd_display_string("Temp prostora", 2)
    lcd.lcd_display_string("Vlaznost makete", 1)
    time.sleep(1.5)

    lcd.lcd_display_string(temperature_str1, 2)
    lcd.lcd_display_string(humidity_str1, 1)
    time.sleep(2)

    prikazi_poruku("Ubacite drvo")
    time.sleep(13)

broj_ocitanja = 15

for i in range(broj_ocitanja):
    humidity_pocetna, temperature_pocetna = Adafruit_DHT.read_retry(11,
24)
    time.sleep(2.4)

```

```

if(humidity_pocetna > 90):
    zeljena_humidity = humidity_pocetna - 37
elif(humidity_pocetna > 85 and humidity_pocetna <=90):
    zeljena_humidity = humidity_pocetna - 33
elif(humidity_pocetna > 80 and humidity_pocetna <=85):
    zeljena_humidity = humidity_pocetna - 29
elif(humidity_pocetna > 75 and humidity_pocetna <=80):
    zeljena_humidity = humidity_pocetna - 25
elif(humidity_pocetna <= 75 and humidity_pocetna > 70):
    zeljena_humidity = humidity_pocetna - 20
elif(humidity_pocetna <= 70 and humidity_pocetna > 64):
    zeljena_humidity = humidity_pocetna - 17
else:
    zeljena_humidity = humidity_pocetna - 15

```

```

humidity_str2 = f"Humidity: {humidity_pocetna:.2f}%"  

temperature_str2 = f"Temp: {temperature_pocetna:.2f}°C"

```

```

lcd.lcd_display_string("Temp drveta", 2)  

lcd.lcd_display_string("Vlaznost drveta", 1)  

time.sleep(1.5)

```

```

lcd.lcd_display_string(temperature_str2, 2)  

lcd.lcd_display_string(humidity_str2, 1)

```

```

if(humidity_pocetna > humidity_prostora + 5):
    duty=10
    duty2=0

```

```

while True:

    humidity, temperature = Adafruit_DHT.read_retry(11, 24)
    flag=True
    stanjeGrijaca = 0
    if temperature >= 31:
        GPIO.output(4,GPIO.LOW)
        stanjeGrijaca = 0
    if (temperature < 31 ):
        GPIO.output(4,GPIO.HIGH)
        stanjeGrijaca = 1
        #fan_pwm.ChangeDutyCycle(20)
        #fan_pwm2.ChangeDutyCycle(15)

    time.sleep(0.2)
    vlaznost.append(humidity)
    temperatura.append(temperature)

    str_1 = f"H:{int(humidity)}% T:{int(temperature)}°C S:0 "
    str_2 = f"V:{duty}% G:{stanjeGrijaca} V2:{duty2}% "
    lcd.lcd_clear()
    lcd.lcd_display_string(str_2, 2)
    lcd.lcd_display_string(str_1, 1)

if temperature <= 28 and temperature > 25 :
    if(duty >= 100 or duty2 >= 100 ):
        fan_pwm.ChangeDutyCycle(100)
        fan_pwm2.ChangeDutyCycle(100)
        time.sleep(5)

```

```
else :  
    print(duty)  
    fan_pwm.ChangeDutyCycle(duty)  
    fan_pwm2.ChangeDutyCycle(duty2)  
    time.sleep(5)  
    duty = duty + 2  
    duty2=duty2 + 2  
    print(duty)
```

```
elif temperature <= 31 and temperature > 28 :
```

```
    if(duty >= 100 or duty2 >=100):  
        fan_pwm.ChangeDutyCycle(100)  
        fan_pwm2.ChangeDutyCycle(100)  
        time.sleep(5)
```

```
else :
```

```
    print(duty)  
    fan_pwm.ChangeDutyCycle(duty)  
    fan_pwm2.ChangeDutyCycle(duty)  
    time.sleep(5)  
    duty = duty + 5  
    duty2=duty2 + 5  
    print(duty)
```

```
else:
```

```
    fan_pwm.ChangeDutyCycle(duty)  
    #fan_pwm2.ChangeDutyCycle(10)  
    #GPIO.output(4,GPIO.HIGH)  
    #flag=True
```

```

if humidity <= zeljena_humidity and temp_prostora <26:
    if temperature >= 27:
        break
    if humidity <= zeljena_humidity and temp_prostora >= 26:
        if temperature >= temp_prostora+1 :
            break
        duty = 100
        duty2 = 100
        humidity, temperature = Adafruit_DHT.read_retry(11, 24)
        time.sleep(1)
        GPIO.output(4,GPIO.LOW)
        stanjeGrijaca = 0
        servo1.ChangeDutyCycle(6)
        time.sleep(0.5)
        servo1.ChangeDutyCycle(0)
        time.sleep(0.3)
        fan_pwm.ChangeDutyCycle(duty)
        fan_pwm2.ChangeDutyCycle(duty2)

        str_1 = f"H:{int(humidity)}% T:{int(temperature)}°C S:1 "
        str_2 = f"V:{duty}% G:{stanjeGrijaca} V2:{duty2}% "
        lcd.lcd_clear()
        lcd.lcd_display_string(str_2, 2)
        lcd.lcd_display_string(str_1, 1)
        time.sleep(3)
        lcd.lcd_clear()
        lcd.lcd_display_string("Ispustanje ", 1)
        lcd.lcd_display_string("Toplote", 2)

```

```

time.sleep(40)

humidity, temperature = Adafruit_DHT.read_retry(11, 24)

lcd.lcd_clear()
duty = 0
duty2 = 0
fan_pwm.ChangeDutyCycle(duty)
fan_pwm2.ChangeDutyCycle(duty2)

str_1 = f"H:{int(humidity)}% T:{int(temperature)}°C S:1 "
str_2 = f"V:{duty}% G:{stanjeGrijaca} V2:{duty2}% "

lcd.lcd_display_string(str_2, 2)
lcd.lcd_display_string(str_1, 1)
time.sleep(3)

with open ('podaci.txt' , 'w') as file:
    for temp,hum in zip (temperatura,vlaznost):
        file.write(f"{temp},{hum}\n")

else:
    prikazi_poruku("Drvo je suho",1)
    time.sleep(2.8)

elif choice == 'B':
    prikazi_poruku("Drvo za ogrev (B)")
    time.sleep(1.8)

else:
    prikazi_poruku("Sifra je netacna.")
    unesena_sifra = ""

```

```
time.sleep(0.6) # Sleep for 0.6 seconds  
#servo1.stop()  
#GPIO.cleanup()
```

Kod 6 Upravljanje sušarom pomoću Phythona

6. Eksperimentalni dio

Kroz duži vremenski period i u različitim sezonama testirali smo našu susaru. Na početku, proces sušenja kvalitetnog drveta trajao je oko 30 minuta. Ovu logiku smo detaljno ispitivali, posebno zbog potrebe za eksperimentalnim praćenjem vlage u prostoriji, koja zavisi od promjena temperature i stanja drveta, odnosno da li je drvo suho ili vlažno. Rezultate smo bilježili u tablicu koristeći Arduino pločicu i MATLAB program. Dobiveni grafikon je prikazan ispod:

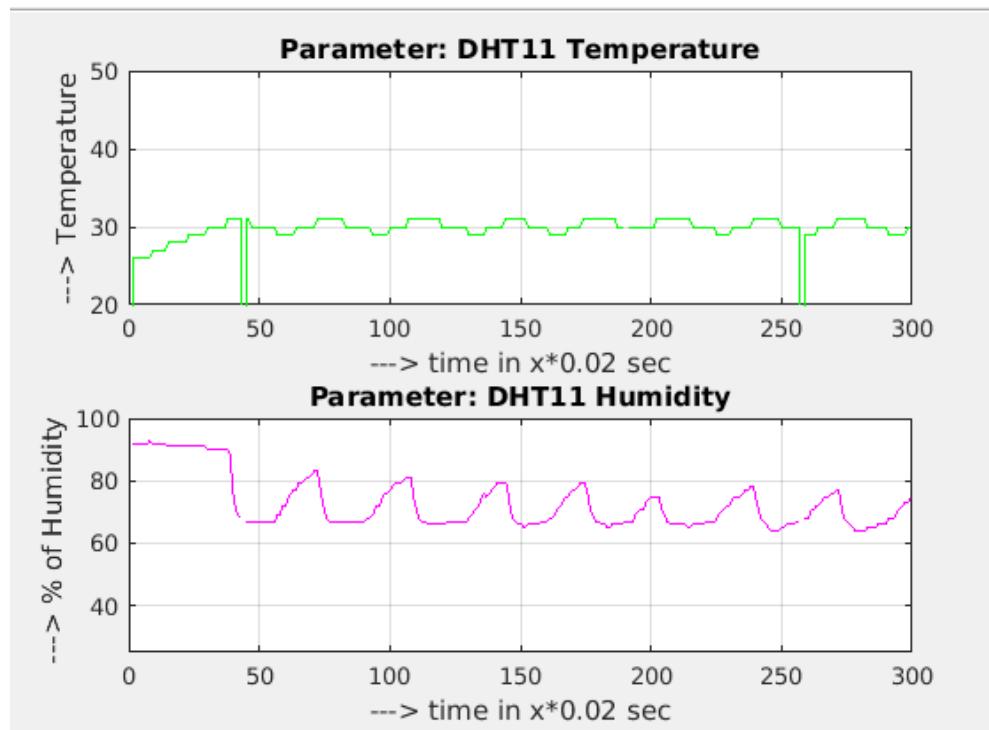


Figure 27 Zavisnost vlažnosti od temperature

Na osnovu rezultata dobivenih ovakvim načinom sušenja, odredili smo konstante za prilagodbu novog koda unutar ograničenog raspona temperatura. Cilj nam je bio postići što efikasniji i brži proces sušenja, s obzirom na to da je prvotni proces trajanja od 30 minuta bio predugo za prezentacijski prikaz.

7. Zaključak

U suštini, projekat "automatizirani sistem sušenja drveta" pruža praktično rešenje za poboljšanje procesa sušenja drveta kroz pametnu tehnologiju. Integracija senzora temperature, mikrokontrolera, i korisničkog interfejsa omogućava efikasno praćenje i kontrolu uslova sušenja. Dodatak DC ventilatora i PTC grejača poboljšava preciznost sistema, čineći ga efikasnim i energetski štedljivim. Projekat pruža osnovu za modernizaciju tradicionalnih industrijskih procesa, omogućavajući daljinsko praćenje i prilagodljivost za različite primene. Iako nije revolucionaran izum, "Sušara Drveta" naglašava potencijal tehnologije u unapređenju svakodnevnih operacija i održivosti. U zaključku, ovaj projekat predstavlja korak ka efikasnijem korišćenju resursa udrvnoj industriji, s naglaskom na tehnološkim rešenjima koja mogu doneti realne prednosti u poboljšanju tradicionalnih procesa sušenja drveta.

8.Literatura

- [1]. Predavanja iz predmeta „Projektovanje sistema na čipu“ dr.sc. Lejla Banjanović-Mehmedović, vanr.prof.
- [2]. <https://circuitdigest.com/article/servo-motor-working-and-basics>
- [3]. <https://www.geya.net/5v-relay-module-how-it-works-and-application/>
- [4]. <https://www.circuitbasics.com/how-to-set-up-a-keypad-on-an-arduino/>
- [5]. <https://www.leipole.net/article/dc-fan-and-its-working-principle.html>
- [6]. http://www.pavouk.org/hw/fan/en_fan4wire.html
- [7]. <https://www.elprocus.com/lcd-16x2-pin-configuration-and-its-working/>
- [8]. https://en.wikipedia.org/wiki/Raspberry_Pi_4
- [9]. <http://www.learningaboutelectronics.com/Articles/N-Channel-MOSFETs>
- [10]. https://www.amazon.com/Matrix-Membrane-Switch-Keyboard-Arduino/dp/B07THCLGCZ/ref=as_li_ss_til?dchild=1&keywords=arduino+4x4+keypad&qid=1594879483&sr=8-3&linkCode=ll1&tag=circbasi-20&linkId=1b716d684cdf99e20ad316478193b748&language=en_US