



**UNIVERZITET CRNE GORE ELEKTROTEHNIČKI
FAKULTET, PODGORICA**

Stefan Raičević

**PRILAGOĐENJE KUĆNIH UREĐAJA ZA HLAĐENJE
POTREBAMA SKLADIŠTENJA VINA**

MASTER RAD

Podgorica, 2025. godine

PODACI I INFORMACIJE O MAGISTRANDU

Ime i prezime: Stefan Raičević

Datum i mjesto rođenja: 25.05.1998. godine, Podgorica

Naziv završenog osnovnog studijskog programa i godina završetka studija: Energetika i automatika, 2023. godine

INFORMACIJE O MASTER RADU

Naziv master studija: Automatika i industrijska elektrotehnika

Naslov rada: Prilagođenje kućnih uređaja za hlađenje potrebama skladištenja vina

Fakultet/Akademija na kojem je rad odbranjen: Elektrotehnički fakultet, Podgorica

UDK, OCJENA I ODBRANA MASTER RADA

Datum prijave master rada: 30.12.2024.

Datum sjednice Vijeća na kojoj je prihvaćena tema: 19.05.2025.

Mentor: Prof. dr Radovan Stojanović

Komisija za ocjenu/odbranu rada:

Prof. dr Milena Erceg

Prof. dr Radovan Stojanović

Prof. dr Vesna Maraš

Lektor:

Datum odbrane: _____

Datum promocije: _____

Izjava o autorstvu

Potpisani: Stefan Raičević
Broj indeksa/upisa: 7/23

Izjavljujem

da je master rad pod naslovom: **Prilagođenje kućnih uređaja za hlađenje potrebama skladištenja vina**

- rezultat sopstvenog istraživačkog rada,
- da predloženi master rad ni u cjelini ni u djelovima nije bio predložen za dobijanje bilo koje diplome prema studijskim programima drugih ustanova visokog obrazovanja,
- da su rezultati korektno navedeni, i
- da nijesam povrijedio/la autorska i druga prava intelektualne svojine koja pripadaju trećim licima.

U Podgorici
11.07.2025.

Potpis magistranda



Zahvalnica

Posebnu zahvalnost dugujem svom mentoru prof. dr Radovanu Stojanoviću, kao i Jovanu Đurkoviću, koji su svojim znanjem i posvećenošću dali veliki doprinos izradi ovog master rada.

Sažetak

U master radu se prezentuje povoljno, inovativno i održivo rješenje prilagođenja kućnih uređaja hlađenja potrebama skladištenja vina. Budući da živimo u eri brzog tehnološkog razvoja, svakodnevni uređaji doživljavaju značajne transformacije, što se posebno ogleda i modifikaciji konvencionalnih rashladnih sistema u pametne hladnjake za vino. Ovaj rad istražuje put od tradicionalnog kućnog rashladnog uređaja do inteligentnog, senzorski baziranog uređaja, koji omogućava praćenje i adaptivnu kontrolu temperature.

Kod poljoprivrednih proizvoda kao što je vino ističe se problem njegovog očuvanja, a time i održavanja kvaliteta, u zavisnosti od temperature. Profesionalni rashladni sistemi su često preskupi za male i srednje vinarije, što stvara potrebu za pristupačnim alternativama. Kao rješenje, predlaže se modifikacija postojećih rashladnih uređaja korišćenjem savremenih tehnologija iz oblasti senzorike i Internet of Things (IoT). Razvijen je prototip uređaja u formi produžnog kabla, koji, u kombinaciji sa temperaturnim senzorom i bang-bang kontrolom, omogućava održavanje mikroklimu u bilo kom standardnom rashladnom uređaju. Zahvaljujući svojoj povoljnoj cijeni i jednostavnoj upotrebi, ovaj uređaj je pristupačan širokom spektru korisnika, uključujući male proizvođače, ali i ljubitelje vina.

Tehnička realizacija sistema uključuje dva mikrokontrolera, senzor temperature, rele, AC-DC pretvarač, kao i korisnički interfejs sa LCD ekranom i tasterima. Softversko rješenje razvijeno je u Arduino IDE okruženju. Eksperimentalna analiza pokazuje da sistem efikasno održava zadatu temperaturu. Cjenovna konkurentnost u odnosu na profesionalne sisteme i dodatne stavke ovog pristupa u aspektima održivosti – produženjem radnog vijeka postojećih uređaja i smanjenjem elektronskog otpada, ističu ekonomske i ekološke prednosti ovog prototipa.

Ključne riječi: moderne tehnologije, vino, upravljanje temperaturom, IoT, ekonomija, racionalno korišćenje resursa, ekologija.

Abstract

This Master's thesis presents an affordable, innovative and sustainable solution for adapting household cooling appliances to the specific requirements of wine storage. As we live in an era of rapid technological development, everyday devices are undergoing significant transformations – particularly in the modification of conventional refrigeration systems into smart wine coolers. This thesis demonstrates the transition from a traditional refrigerator to an intelligent, sensor-based device that empowers real-time monitoring and adaptive temperature control.

For agricultural products such as wine, temperature plays a crucial role in preservation and quality maintenance. Professional cooling systems are often too expensive for small and medium-sized wineries, creating the need for affordable alternatives. As a solution, the thesis proposes the modification of existing refrigeration units with the help of advances in sensor technology and the Internet of Things (IoT). The prototype, which is practically designed as an extension lead, in combination with a temperature sensor and bang-bang control, performs the maintenance of a stable microclimate inside any standard refrigerator. Due to its low cost and ease of use, the device would be both helpful and affordable to a wide range of users, including small producers or just wine enthusiasts.

The system's technical implementation includes two microcontrollers, a temperature sensor, relay, AC-DC converter, as well as a user interface with an LCD display and control buttons. The software solution is designed in the Arduino IDE environment. Experimental analysis demonstrates that the system effectively maintains the desired temperature. Its cost-effectiveness compared to professional systems, along with the additional sustainability benefits—such as extending the lifespan of existing devices and reducing electronic waste—highlight the economic and ecological advantages of this prototype.

Keywords: modern technologies, wine, temperature control, IoT, economy, efficient use of resources, ecology.

Sadržaj

<i>Zahvalnica</i>	6
Sažetak	7
Abstract.....	8
1. Uvod i problem	10
2. Pregled literature i postojećih rješenja	15
3. Metod	19
3.1. Sažetak metodologije	19
3.2 Hardver	20
3.2.1 Senzor	21
3.2.1.1. Korišćenje komercijalnog senzora	22
3.2.2 Predajnik	23
3.2.1.1. Predajnik na bazi osmобitnog RISK procesora	24
3.2.2.3. Predajnik na bazi ESP32	26
3.2.3. Modul napajanja	27
3.2.2. Prijemnik	28
3.2.2.1. Prijemnik na bazi ATmega328P	31
3.2.2.2. Prijemnik na bazi ESP32	32
3.2.2.3. Upotreba automatskog osigurača	33
3.2.2.4. Upotreba ispravljača	33
3.2.2.5. Aktuator	34
3.2.2.6. LCD ekran	35
3.2.2.7. Ostali hardverski elementi	35
3.2.2.8. Korisnički interfejs	36
3.2.2.8.1. Režimi rada kontrolera	37
3.2.3. Komunikacija	38
3.2.4. Energetsko napajanje i zaštita	39
3.2.4.1 PTC starter i zaštita od preopterećenja	40
3.3. Algoritam i softver	42
3.3.1. Upravljački algoritam	42
3.3.2. Upravljačka logika	42
3.3.3. Softver predajnika (firmver)	43
3.3.4. Softver prijemnika (firmver)	45
4. Eksperimentalni rezultati	48

4.1. Eksperimentalno okruženje	48
4.2. Eksperimentalni parametri	51
4.3. Rezultati preliminarnog testiranja	52
5. Zaključak i budući rad	58
6. Literatura	61
7. Prilozi	63
7.1. Prilog 1	63
7.2. Prilog 2	63
7.3 Prilog 3	65
7.4. Prilog 4	70
7.5 Prilog 5	74
7.6 Prilog 6	76
7.7. Prilog 7	78

1. Uvod i problem

Kako skladištenje vina predstavlja delikatan proces koji zahtijeva strogu kontrolu određenih parametara i da vino kao proizvod ima specifične zahtjeve za očuvanje svojih svojstava i kvaliteta tokom vremena, postojeće tehnologije kućne bijele tehnike (frižideri i zamrzivači) mogu biti prilagođene kako bi se osigurali odgovarajući uslovi za vino. Tema rada se bavi istraživanjem i razvojem uređaja koji omogućuje pravilno skladištenje vina koristeći uređaje kućne bijele tehnike, uz implementaciju savremenih tehnologija za preciznu kontrolu uslova skladištenja. Cilj rada je razviti funkcionalan i svrsishodan prototip. Predmet razmatranja su načini korišćenja elektronskih komponenti i senzora u svrhu postizanja željenih parametara, ali i razvoj softverskog sistema za praćenje i kontrolu uslova u unutrašnjosti hladnjaka. Ideja je da se precizno upravljanje uslovima vrši putem integracije hardverskih i softverskih rješenja, kako bi se omogućilo dogoročno i kvalitetno skladištenje vina. Rad ima značaj jer pruža mogućnost široj javnosti da bolje iskoristi postojeće uređaje kućne bijele tehnike za specijalizovane potrebe. Osim toga, doprinosi širenju znanja o upotrebi mikroelektronike i senzora za kontrolu sistema u svakodnevnom životu.

Aktuelnost teme se ogleda u potrebi za projektovanjem jeftinog rješenja za pretvaranje postojećih uređaja za hlađenje, odnosno kućne bijele tehnike (frižidera i zamrzivača) u sisteme preciznog hlađenja za potrebe skladištenja i očuvanja poljoprivrednih proizvoda, sa akcentom na vino.

Rad daje doprinos održivom razvoju ovog sektora, jer su pametni rashladni uređaji značajan aspekt u automatizaciji i digitalizaciji agrosektora. Prikazan je put razvoja od običnog kućnog rashladnog uređaja, koji kao takav ima skoro stogodišnju tradiciju, do inovativnog, pametnog i interaktivnog hladnjaka.

Tema posebno dobija na značaju ako se ima u vidu smanjenje negativnog uticaja na životnu sredinu. Uvođenjem automatizovanih sistema za praćenje i optimizaciju rada uređaja, smanjuje se potreba za konstantnim manuelnim nadzorom i povećava energetska efikasnost. Osim toga, promovisanje ponovne upotrebe postojećih uređaja predstavlja praktično, ekološki prihvatljivo i ekonomski održivo rješenje koje daje doprinos kako pojedinačnim domaćinstvima, tako i planeti u cjelini. Takođe, tema daje doprinos promovisanju Crne Gore kao vinske destinacije, jer je vino među rijetkim proizvodima u kojima bilježimo suficit. Sve više stanovništva se bavi proizvodnjom vina, a pored "Plantaža" egzistira oko 90 manjih vinarija.

Predmet istraživanja je dizajniranje pametnog hardversko softverskog uređaja (u daljnjem tekstu kontrolera) za regulaciju temperature skladištenja poljoprivrednih proizvoda sa akcentom na vino.

Sam hladnjak, odnosno njegov hardverski dio, ostaje nepromijenjen, dok ono što ga čini "pametnim" jeste uređaj - kontroler koji reguliše protok električne energije u svrhu regulacije temperature hlađenja.

Kontroler omogućava izbor različitih referentnih vrijednosti parametara hlađenja koji odgovaraju specifičnom tipu proizvoda. Korisnički interfejs je dizajniran da bude što jednostavniji za rukovanje. Pored podrazumijevanih referentnih vrijednosti, korisnik ima mogućnost ručnog podešavanja parametara, što ovaj sistem čini fleksibilnim za različite primjene.

Povratna sprega je realizovana u odnosu na temperaturu kao procesnu promjenljivu. Cilj je da se kompenzuje greška između trenutnih i zadanih vrijednosti temperature kontrolom on/off stanja hladnjaka. Prenos podataka o temperaturi između unutrašnjosti hladnjaka i kontrolera vrši se pomoću jednog od bežičnih protokola (BLE-Bluetooth Low Energy, radiofrekvencijski-RF, ESP-NOW) vodeći računa o potrošnji i trajanju baterije.

Sistem koristi dva standardna mikrokontrolera opšte namjene, koji imaju funkcije obrađivana informacija dobijenih od senzora i upravljanja procesa uključivanja-isključivanja motora kompresora rashladnog uređaja. Prekidački element (rele) predstavlja izlazni aktuatorski dio kontrolera, odnosno vezu između upravljačkog dijela i izvora napajanja. Na taj način je obezbijedena galvanska izolacija izlaznog organa, što omogućava da se kvar u napajanju ne prenosi na upravljački dio. Dodatno galvansko razdvajanje između ulaznog organa (senzora) i kontrolera je omogućeno preko bežične veze. Potpuno galvansko razdvajanje čini uređaj veoma sigurnim za korisnika.

Trend recikliranja, odnosno ponovne upotrebe starih uređaja prisutan je posljednjih godina u svijetu. Na taj način stižu se višestruki benefiti, od zaštite okoline do smanjenja nepotrebnih troškova. Motiv ovog istraživanja jeste utvrđivanje i iskorišćavanje potencijala polovnih kućnih hladnjaka za održavanje optimalnih uslova za čuvanje poljoprivrednih proizvoda u preciznoj poljoprivredi, sa posebnim fokusom na vino. Kako je u današnjem vremenu održavanje odgovarajuće temperature ključno za očuvanje kvaliteta poljoprivrednih proizvoda, korišćenje već postojećih, starijih hladnjaka može predstavljati ekonomičnu i ekološki prihvatljivu alternativu za mala i srednja poljoprivredna gazdinstva, vinare i ljubitelje vina koji nemaju pristup profesionalnim skupim rashladnim sistemima.

Upotreba pametnih senzora omogućava precizno praćenje ulaznih parametara, kao što je temperatura, što je od presudnog značaja za očuvanje nutritivnih svojstava proizvoda i eliminaciju štetnih mikroorganizama. Na taj način se produžava njihov vijek trajanja i sprječava kvarenje usljed nepravilnog skladištenja.

Cilj ovog istraživanja je razvioj i testiranje funkcionalnog prototipa dodatnog eksternog kontrolera temperature, podešenog za postojeće rashladne uređaje u domaćinstvu, poznate kao frižideri i zamrzivači, sve u cilju skladištenja poljoprivrednih proizvoda, sa primarnim akcentom na vino.

Implementiran je sistem senzor, kontroler, izvršni organ koji jeste eksternog tipa u obliku modifikovanog produžnog kabla. Na taj način se ne prepravlja postojeći uređaj, a rješenje je prenosno i multifunkcionalno. Osim toga, izvršićena je analiza ekonomičnosti sa više aspekata: u pogledu potrošnje energije, uštede sredstava namijenjenih za kupovinu profesionalnih skupih sistema, prostora, jednostavnosti održavanja, smanjenja otpada i sl.

Istraživanje obuhvata procjenu tehničkih rješenja, ekonomskih i ekoloških prednosti korišćenja starih hladnjaka za čuvanje poljoprivrednih proizvoda u malim i srednjim proizvodnim pogonima.

Rad na projektu obuhvata sledeće projektne, istraživačke i ekperimentalne metode:

1. Proučavanje postojeće literature i stručnih radova: Neophodno je izvršiti analizu postojeće literature u oblasti sistema za čuvanje poljoprivrednih proizvoda, sa posebnim akcentom na vino. Ovaj korak omogućava identifikaciju postojećih rješenja, tehnoloških inovacija i metodologija koje se koriste u ovoj oblasti. Jasno isticanje korelacija između kvaliteta proizvoda i temperaturnog skladištenja. Analiza otvorenih problema i vizije njihovog rješavanja.
2. Dizajniranje hardverskog sistema: Razvijanje i projektovanje fizičkih komponenti kontrolera. Izbor temperaturnog senzora u pogledu preciznosti, brzine očitavanja, potrošnje, opsega, načina čitanja, načina napajanja, načina slanja podataka itd. Odabir mikrokontrolera koji kontroliše očitavanje senzora i način slanja podataka. Način slanja podataka, RF, BLE ili drugi protokol. Smanjenje potrošnje sistema autonomnog daljinskog senzora. Odabir mikrokontrolera kontrolera, ESP32 ili drugi. Način bežične komunikacije, RF ili BLE. Skladištenje podataka, logging, potrošnja, dimenzije itd. Odabir izvršnog organa, relea, mehanički poluprovodnički, vrijeme reagovanja itd. Korisnički interfejs (tasteri a postavljanje predefinisanih parametara hlađenja, kao i tasteri za ručno podešavanje). Interfejs za vizualizaciju, displej, LED signalizacija. Zaštita, prekonaponska, prekostrujna, osigurači.
3. Uvezivanje svih hardverskih komponenti u jedan sistem: Precizno povezivanje i testiranje kako bi se osigurala pouzdana funkcionalnost svih komponenti u sistemu.
4. Priprema odgovarajućeg softverskog koda i kompajliranje u odgovarajućoj softverskoj platformi i snimanje koda na mikrokontroler: Razvoj softverskog koda korišćenjem platforme Arduino-IDE ili slične, a zatim kompajliranje i snimanje na mikrokontroler ESP-32, ATtiny85 ili sličan.
5. Testiranje sistema u realnim uslovima: Nakon završetka dizajniranja i implementacije sistema neophodni su testovi u realnim uslovima rada. To uključuje praktičnu izradu prototipa i rad hladnjaka sa stvarnim proizvodima, kako bi se provjerio efikasnost sistema u kontrolisanju mikrokline unutar hladnjaka.
6. Analiza dobijenih rezultata: Na kraju, prikupljeni podaci sa testiranja se analiziraju kako bi se utvrdilo da li sistem ispunjava postavljene ciljeve, kao što su produženje roka trajanja, očuvanje kvaliteta proizvoda i povećanje efikasnosti u procesu čuvanja. Osim toga, rezultati testiranja omogućavaju identifikaciju mogućih poboljšanja sistema, kao i optimizaciju parametara za još bolje performanse.

Očekivani rezultati su sljedeći:

- Prototip uređaja u formi produžnog kabla, koji, uz pomoć jednog bežičnog senzora, održava mikroklimu unutar bilo kog hladnjaka bang-bang kontrolom, uz napomenu da je termostat na hladnjaku podešen na maksimalnu vrijednost. Uređaj karakteriše veoma širok opseg primjene, od industrijskih hladnjaka do malih kućnih. Bio bi koristan za kancelariju svake ozbiljne osobe iz sfere biznisa, ali i malom poljoprivrednom proizvođaču. Osim toga, zbog svoje pristupačne cijene biće dostupan bilo kom ljubitelju vina.
- Softversko rješenje koje je efikasno i fleksibilno za različite primjene. Softverska arhitektura mora sadržati slojeve: senzora (za očitavanje podataka o temperaturi), kontrole (za imlementaciju bang-bang kontrole i upravljanja uređajem), korisničkog interfejsa (za implementaciju parametara koje korisnik zadava putem taster, ali i za neophodan ispis na LCD ekranu) i komunikacije (BLE, RF ili slične).
- Sistematizacija znanja i vještina u smislu pravljenja sličnih rješenja za ostale poljoprivredne ili prehrambene proizvode. Istraživanje će rezultirati razvojem pametnog sistema za monitoring i kontrolu ključnih parametara koji su esencijalni za kvalitet i očuvanje poljoprivrednih proizvoda generalno, uz izmjene u softveru ili čak dodavanje drugih senzora.
- Ušteda finansijskih sredstava i energije. Zbog svoje cjenovne konkurentnosti i pouzdanosti, stari hladnjak uz upotrebu kontrolera postaje odlična alternative skupim rashladnim uređajima. Održivost se ogleda i u smanjenju elektronskog otpada i resursa potrebnih za proizvodnju novih uređaja. Poljoprivrednici ne moraju ulagati u nove, skupe uređaje, već mogu prilagoditi postojeće aparate.
- Upotreba savremenih tehnologija, kao što su CPS (Cyber Physical System), IoT (Internet of Things), AI (vještačka inteligencija). Rad doprinosi novim saznanjima u oblasti senzora i automatizacije u poljoprivredi.
- Povećanje konkurentnosti našeg vinarstva, koje ima presudnu ulogu u dugoročnom razvoju domaće privrede, ali i njenom integrisanju i prepoznatljivosti na evropskom i svjetskom tržištu. Uređaj povoljno utiče na brzinu i kvalitet razvoja produktivnih malih i srednjih preduzeća koja doprinose ekonomskom rastu Crne Gore.

Struktura master rada se sastoji od pet poglavlja:

- U drugom poglavlju su detaljno analizirani relevantni radovi, istraživanja i postojeće tehnologije koje se bave temama vezanim za: načine skladištenja vina, primjenu kućne bijele tehnike, slična elektronska i mehanička rješenja, kontrolu temperature i vlažnosti, sisteme za automatsko upravljanje u kontekstu vinskih podruma i specijalizovanih uređaja za skladištenje. Poglavlje obuhvata tehnička rješenja u vinskim frižiderima, podrumima i specijalizovanim komorama, upotrebu senzora i aktuatora u uređajima kućne bijele tehnike i pregled dosadašnjih rješenja za automatizaciju i monitoring uslova skladištenja vina putem mikrokontrolera i

pametnih uređaja

- U trećem poglavlju je obrađen metod rada. Detaljno je opisano hardversko i softversko rješenje, kao i komunikacijski protokol. Hardverske komponente komponente potrebne za implementaciju sistema su pobrojane i detaljno prezentovane, a to su: automatski osigurač (korišćen za zaštitu sistema od kratkog spoja ili preopterećenja, a dodatno je postavljen tako da ima ulogu i ON/OFF prekidača), ispravljач (korišćen za konvertovanje AC napona u DC napon potreban za napajanje elektronskih komponenti), rele (omogućava kontrolu napajanja uređaja, odnosno kompresora frižidera/zamrzivača), mikrokontroler opšte namjene (ESP32-S3 ili sličan, korišćen za prikupljanje podataka sa senzora, upravljanje uređajem i komunikaciju sa korisnikom) temperaturni senzor (korišćen za precizno mjerenje temperature unutar hladnjaka), LCD displej (omogućava korisniku uvid u parametre sistema u realnom vremenu), korisnički interfejs (tasteri za podešavanje temperature). Prikazana je šema spajanja ovih komponenti u funkcionalni sistem. Detaljno je opisan softver uređaja. Ključne komponente su komunikacioni protokol (kao što su BLE ili RF, korišćeni za bežični prenos podataka o temperaturi) i kontrolni algoritam (omogućava precizno upravljanje uređajima na osnovu podataka sa senzora). Softver je biti razvijen korišćenjem integrisanog razvojnog alata Arduino IDE. Takođe, razmotreni su princip rada i upotreba PTC startera za kontrolu rada kompresora.
- Četvrto poglavlje prikazuje rezultate testiranja prototipa uređaja za skladištenje vina uključujući eksperimentalne parametre kao što su regulacioni odziv, energetska potrošnja, domet i preciznost sistema. Rezultati će biti potkrijepljeni dijagramima i grafikonima, odnosno vizuelnim prikazom temperature tokom testiranja.
- Peto poglavlje predstavlja zaključak u kome su sažeti svi rezultati, prednosti i mane prototipa. Predstavljani su i predlozi za poboljšanje sistema, kao i mogući pravci za dalje istraživanje i razvoj. To može uključivati optimizaciju softverskog algoritma, upotrebu novih senzora za dodatno ili preciznije mjerenje parametara, kao i unapređenje dizajna korisničkog interfejsa.

2. Pregled literature i postojećih rješenja

Poglavlje sadrži detaljno analizirane relevantne radove, istraživanja i postojeće tehnologije koje se bave temama vezanim za: načine skladištenja vina, primjenu kućne bijele tehnike, slična elektronska i mehanička rješenja, kontrolu temperature i vlažnosti, sisteme za automatsko upravljanje u kontekstu vinskih podruma i specijalizovanih uređaja za skladištenje. Biće obuhvaćena tehnička rješenja u vinskim vitrinama, podrumima i specijalizovanim komorama, upotrebu senzora i aktuatora u uređajima kućne bijele tehnike i pregled dosadašnjih rješenja za automatizaciju i monitoring uslova skladištenja vina putem mikrokontrolera i pametnih uređaja.

Tokom proteklih decenija, značajan broj radova je dostupan na temu povećanja produktivnosti i efektivnosti poljoprivredne proizvodnje implementacijom strategija propisnog skladištenja proizvoda. To je naročito izraženo za slučaj vina, koje je veoma osjetljivo na temperaturu. Procesne tehnike u proizvodnji vina obuhvataju širok spektar tehnologija, od osnovnih metoda obrade do inovacija u pogledu skladištenja i kontrole mikroklimе.

Proizvodnja vina je proces koji zavisi od precizne kontrole mnogih faktora, uključujući temperature, vlagu i svjetlost, koji direktno utiču na kvalitet vina. Istraživanja u oblasti proizvodnje vina postepeno su se razvijala, sa sve većim naglaskom na optimizaciji prethodno nabrojanih faktora. U skladu s tim, upravljanje temperaturom tokom fermentacije i skladištenja, kao i kontrola mikroklimе, postalo je ključno za očuvanje kvaliteta vina. Grčki naučni članak [2] ukazuje na to da promjene tokom skladištenja vina mogu dovesti do razvoja mikroorganizama i promjene u boji i ukusu. Hladnjaci za vino sa stabilizovanom temperaturom i vlagom mogu spriječiti ove negativne efekte. Tom pojavom su se bavili i predstavnici Australijskog Instituta Vina, koji u svom naučnom članku [3], sa posebnim osvrtom na osjetljivost bijelih vina. Naime, u navedenom članku je objašnjeno kako povišena temperatura utiče na smanjenje sumpordioksida, kao i na promjenu boje – bijela vina mogu poprimiti skoro braon boju.

Prema naučnom članku o skladištenju sira [4] održanje temperature je važno kako bi se spriječilo prekomjerno sazrijevanje i očuvao kvalitet. Takođe, navodi se da se smanjenjem temperature produžuje rok trajanja, ali i da previše niska temperatura može izazvati gubitak ukusa i promjene u teksturi. Kao optimalna temperatura se navodi 5-10 stepeni Celzijusa. Kontrolisani nivoi vlage se navode kao važni kako bi se izbjeglo presušivanje i gubitak ukusa sira.

U članku [5] je detaljno opisana problematika u transportu i skladištenju lako kvarljivih prehrambenih proizvoda. Kako bi se garantovao rok trajanja proizvoda u lancu snabdijevanja hranom. Određeni su parametri koji utiču na vijek trajanja proizvoda. Zatim su odabrani odgovarajući senzori, a bežična komunikacija se vrši uz pomoć RF protokola. Dobijeno rješenje je testirano u praksi i pokazalo se da takvi sistemi omogućavaju precizno praćenje ključnih parametara, što se pokazalo kao jako važno prilikom transporta više vrsta prehrambenih proizvoda u jednoj hladnjači.

U naučnom radu [6] je predstavljen sistem koji se primjenjuje u svim fazama proizvodnje vina, od uzgoja grožđa do krajnjeg proizvoda. Temperatura se memoriše i ZigBee

komunikacijom se šalje u kontrolni centar na obradu. Sistem je testiran u laboratorijskim uslovima.

Naučni članak [7] govori o kontroli temperature kao ključnoj stvari za fermentaciju i skladištenje vina. Pri fermentaciji vina se mora voditi računa da se ono ne pregrije, dok se za skladištenje, bilo u podrumima ili u hladnjacima, sugerise stabilna temperatura od oko 13 stepeni Celzijusa.

Italijanska kompanija koja se bavi rashladnim sistemima i klimatizacijom, u prospektu za jedan od svojih proizvoda [8] navodi temperaturu i vlagu kao ključne parametre pri proizvodnji i postporizvodnji vina. Razvili su mnoge uređaje i softvere koji vrše nadzor, a neki i kontrolu temperature, kako u vinogradima, tako i u hladnjacima i vinskim podrumima. Uspješnost svojih proizvoda ističu u jednostavnosti i pouzdanosti. TempCube[9] je razvio mobilnu aplikaciju za nadzor mikroklima, dok su korišćeni senzori švajcarske proizvodnje. U članku [10] se takođe prezentuje proizvod pod nazivom RuuviTag, koji na mobilnu aplikaciju šalje podatke o temperaturi i vlažnosti vazduha u vinskom podrumu. Kao idealna temperatura se navodi 12-16 stepeni Celzijusa za crvena, 8-12 za bijela i 12 za pjenušava, dok se idealnom vlažnošću vazduha smatra 65-75%.

U naučnom radu [10] je predstavljena izrada uređaja za kontrolu procesa fermentacije vina. Podaci u realnom vremenu se stalno prenose putem bežične komunikacije na uređaje vinara. Za izradu prototipa je korišćen mikrokontroler MSP430, RN-42 Bluetooth modul, ADC konverter, dok je kod pisan u programskom jeziku C. Kao prednost se ističe cijena prototipa manju od 50 eura, dok su cijene samo senzora za kontrolu temperature vina [11] tri puta veće. Takođe, Elitech grupa [12] i Switrace Cold Chain Solutions [13] su razvili data loggere za temperaturu vina u podrumima, navodeći temperaturu i UV svjetlost kao ključne parametre prilikom skladištenja. Takođe se u [14] i [15] kao ključni parametri ističu temperatura, vlaga, osvjetljenje, vibracije i pozicioniranje vina, bilo da se radi o rashladnom uređaju ili podrumu.

Pored toga, u [16] se detaljno opisuju načini čuvanja vina, od presudne važnosti temperature, pozicije i osvjetljenosti i prezentuje se sistem za monitoring parametara. Nudi se osnovna i profesionalna verzija sistema, dok je za obje princip rada isti, bežičnom Wi-Fi konekcijom se šalju podaci sa senzora temperature, vlažnosti, vibracije i osvjetljenosti. Podaci se prikazuju na LCD displeju, dok po želji korisnika mogu slati i putem mejla ili SMS poruke. Razvijeni su i mnogi data logeri [17] koji prikupljaju podatke na svojoj unutrašnjoj memoriji, a kao glavni se ističu temperatura i vlažnost.

U [18] se ističe značaj pravilnih uslova skladištenja vina, ali i korištenja odgovarajućeg čepa kao presudnih za kvalitet starenja vina. U [19] su prikazani cjenovnici profesionalnih rashladnih uređaja za vino dostupnih u Crnoj Gori.

Postojeća rješenja za hlađenje vina su mnogobrojna, i kreću se u širokom cjenovnom opsegu. Najjednostavnija su u formi utičnice [26] i omogućuju osnovnu regulaciju, dok, sa druge strane, postoje luksuzni rashladni uređaji firmi Asko [27] i Eurocave [28], koje se smatraju renomiranim proizvođačima u tom segmentu.

Najjednostavniji kontroler proizvođača Tuya je kompatibilan sa grijalicama i uređajima

za hlađenje. Povezuje se preko 2.4 GHz Wi-Fi mreže pomoću Android/iOS aplikacije i pruža preciznost regulacije od 0.5 °C, a radi u rasponu temperature od -20°C do +90°C. Prednosti su upravo pristupačna cijena, relativno širok temperaturni opseg i preciznost. Najveća mana pri implementaciji u rashladni uređaj jeste žičani senzor. Kontroler ima oblik utičnice na koju je povezan kabl, na čijem kraju se nalazi senzor. Neophodna je prepravka i provlačenje kabla kroz vrata rashladnog uređaja, pa se za konkretni komercijalni kontroler ne može reći da ga karakteriše univerzalnost i lakoća implementacije. Kao mana se ističe i upitna pouzdanost Tuya Wi-Fi uređaja u odnosu na alternativne integracije (npr. Zigbee). Takođe, upitna je i privatnost podataka i oslanjanje na Tuya cloud. Slika 1 prikazuje izgled komercijalnog kontrolera i preuzeta je sa web-stranice putem koje se proizvođač oglašava.



Slika 1: Primjer izgleda komercijalnog kontrolera

Kao ozbiljniji kontroler ističe se Wine-Mate 24V Smart Wi-Fi Thermostat. To je robusno rješenje za upravljanje mikroklimom vinskog podruma, sa naprednim funkcijama i automatskim mjerama zaštite. Kompatibilan je samo sa kanalnim ili split hlađenim vinskim podrumima, dok ga nije moguće koristiti kod kompresorskih vinskih vitrina. Uređaj konstatno mjeri temperaturu i vlažnost vazduha i konstantno provjerava odstupanje od zadatih parametara. Na osnovu očitavanja, putem 24V releja daje signal sistemima kao što su vinska klima (za hlađenje prostorije), ovlaživač (za povećanje vlažnosti) ili čak grijač (ako temperatura padne ispod minimuma). Ovi sistemi se isključuju/uključuju automatski. Kontroler je povezan na lokalnu Wi-Fi mrežu i omogućava daljinsku kontrolu i prikaz parametara putem aplikacije. Uređaj je predviđen za unutrašnju montažu, najmanje 1.5m iznad poda i van direktnog protoka vazduha.

Po sistemima hlađenja vinske vitrine možemo podijeliti u dvije grupe – sa kompresorski i sa termoelektičnim (Peltier) sistemom hlađenja. Kompresorski sistem koristi kompresor i

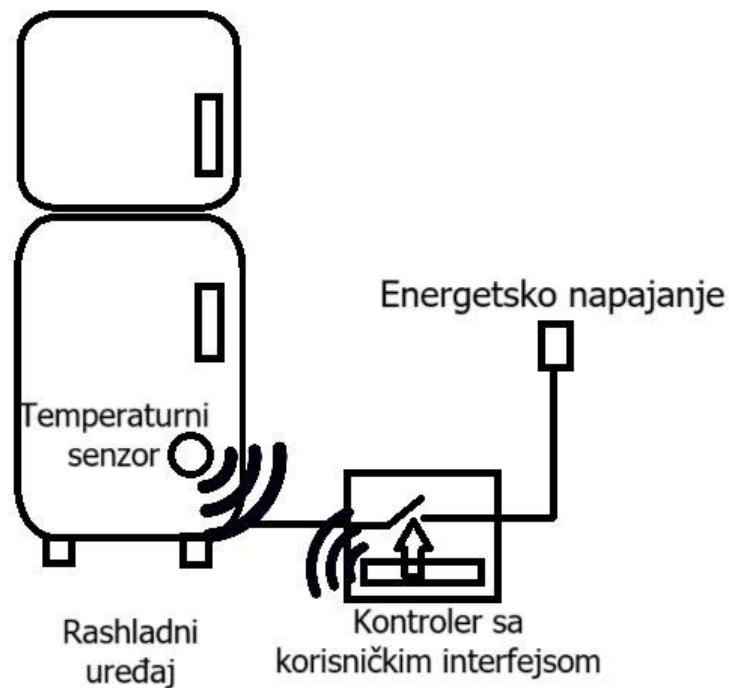
rashladni gas (freon) za kompresiju i ekspanziju gasa, dok termoelektrični sistem koristi električnu energiju i Peltierov modul da stvoru temperaturnu razliku (jedna strana se hladi, druga se grije). Prednosti kompresorskog sistema su stabilna i precizna temperatura, efikasnost i na višim temperaturama okoline i većim kapacitetima. Termoelektrični sistem ima lošiju efikasnost, teško postiže temperaturu nižu od 10°C, izuzetno je osjetljiv na spoljašnje temperature i performanse hlađenja mu opadaju ukoliko sobna temperatura prelazi 25°C. Osim toga, na efikasnost negativno utiču i često otvaranje i velika količina vina unutar uređaja, što implicira da nije za veće kapacitete. Prednosti termoelektričnog sistema hlađenja jeste odsustvo kompresora. Samim tim je daleko manji nivo buke i vibracija, koje mogu uticati na talog vina [29]. Zaključak je da su kompresorski modeli daleko pouzdaniji i pružaju stabilnu temperaturu za skladištenje svih vrsta vina, dok su Peltier modeli tihi, jeftini i jednostavni, ali zadovoljavaju samo temperaturne potrebe crvenog vina i nijesu efikasni za dugotrajno skladištenje ili upotrebu u toplim prostorijama. U Crnoj Gori se gotovo isključivo upotrebljavaju kompresorski modeli (samo takve je i moguće kupiti u maloprodaji), jer su pouzdaniji i bolje odgovaraju klimatskim uslovima.

U maloprodaji najčešći hladnjaci za vino su proizvođača Tesla, Vivax, Dometic, Bosch [30] [31]. Cijene se kreću od 300 do 3000 eura, u zavisnosti od brenda, kapaciteta i drugih karakteristika. Vinske vitrine mogu biti jednozonske, dovozone i trozonske. Jednozonske čuvaju svo vino na istoj temperaturi, dok luksuzni proizvodi te branše nude i dvije do tri temperaturne zone čuvanja, što omogućuje da se skladište do tri vrste vina istovremeno.

3. Metod

3.1. Sažetak metodologije

Predloženi sistem se sastoji iz hardvera i softvera, međusobno integriranih u cilju obezbjeđivanja optimalnih uslova za skladištenje vina. Hardverski dio obuhvata bežičnim senzorom opremljen pametni hladnjak, koji kontinualno prati temperaturu, korisnički interfejs, kao i aktuator – rele unutar kontrolera koji vrši uključivanje/isključivanje hladnjaka. Softverski dio sistema uključuje upravljački algoritam koji u realnom vremenu analizira podatke sa senzora i automatski prilagođava rad sistema za regulaciju mikroklimе unutar hladnjaka. Slika 2 prikazuje šemu uopštenog sistema.



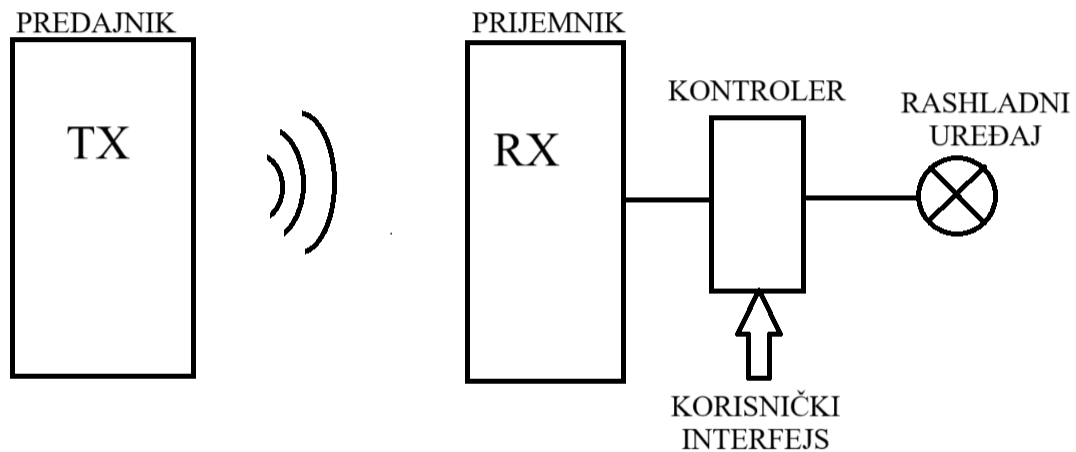
Slika 2: Šema uopštenog sistema

Metodološki pristup uključuje projektovanje i testiranje prototipa pametnog hladnjaka, kao i eksperimentalno poređenje načina za realizaciju prototipa. Praćenjem promjena u parametrima skladištenja, kao i potrošnje energije tokom vremena, vrši se analiza stabilnosti sistema, energetske efikasnosti i tehničke pouzdanosti. Cilj je validacija tehničkih performansi predloženog rješenja u realnim uslovima rada.

3.2 Hardver

Hardver predloženog rješenja se može podijeliti u dvije cjeline – predajnik i prijemnik. Idejno rješenje (Slika1) prikazuje predajnik, koji jednim od bežičnih protokola šalje informaciju (temperaturu) prijemniku. Zatim prosljeđuje tu temperaturu kontroleru koji, na osnovu željenih preferenci korisnika, vrši upravljanje rashladnim uređajem.

Slika 3 prikazuje generalnu arhitekturu hardvera sistema.



Slika 3: Generalna arhitektura hardvera sistema

Hardver predajnika se nalazi unutar rashladnog uređaja. Osim samog bežičnog komunikacionog modula mora imati i izvor napajanja (poželjno baterijski) kao i temperaturni senzor. Bežični protokol mora imati odgovarajući domet kako bi poslao vrijednost temperature prijemniku. Hardver prijemnika mora imati kontroler koji upravlja rashladnim uređajem kao i korisnički interfejs kojim korisnik unosi željene podatke.

Kontroler je u formi produžnog kabla. Ima integrisani korisnički interfejs. Unutar njega se nalaze bežični komunikacioni modul, aktuator-rele i automatski osigurač.

3.2.1 Senzor

Kako bi mjerio temperaturu unutar rashladnog uređaja, temperaturni senzor mora da zadovoljava sljedeće uslove: preciznost mjerenja od $\pm 0.5^{\circ}\text{C}$, kao i opseg mjerenja temperature od 0°C do 20°C . Pri sklanju prototipa korišteni su senzor DHT22, poznat i kao AM2302, i senzor DS18B20.

DHT22 je jedan od najčešće korišćenih senzora za mjerenje temperature i vlažnosti u IoT i hobi projektima. Karakterište ga niska potrošnja energije i laka integracija sa brojnim mikrokontrolerima, kao što su Arduino, ESP32, Rasperry Pi itd. Sposoban je da mjeri temperaturu u opsegu od -40°C do $+80^{\circ}\text{C}$, sa preciznošću od $\pm 0.5^{\circ}\text{C}$, što ga čini pogodnim za upotrebu u ovom projektu. Osim toga, sposoban je da mjeri i relativnu vlažnost vazduha, u opsegu od 0% do 100%, sa preciznošću $\pm 2-5\%$.

Senzor koristi digitalni jednosmjerni signal za komunikaciju sa mikrokontrolerom, što ga čini jednostavnim za upotrebu, budući da nije potrebna analogno-digitalna konverzija. Podržani napon napajanja je od 3.3V do 6V, dok je interval mjerenja dvije sekunde. Dimenzije senzora su 15.1mm x 25mm x 7.7mm. DHT22 u svojoj unutrašnjoj strukturi sadrži NTC termistor za mjerenje temperature, kondenzatorski senzor za mjerenje relativne vlažnosti i mikrokontroler koji obrađuje i digitalizuje podatke prije nego što ih pošalje mikrokontroleru.

DHT22 ima tri izlazna pina, odnosno signalne nožice. „+“ se povezuje na 3.3V - 6V, „-“ na masu, dok se „out“ povezuje na digitalni ulazni pin mikrokontrolera, kod ovog prototipa je korišten pin D4.

DS18B20 je digitalni temperaturni senzor koji koristi 1-Wire komunikacioni protokol. Posebno je poznat po tačnosti, stabilnosti i mogućnosti povezivanja više senzora na jednom pinu. Opseg mjerenja temperature je od -55°C do 125°C , uz preciznost od $\pm 0.5^{\circ}\text{C}$ (u opsegu od -10°C do 85°C). Podržani napon napajanja je od 3V do 5.5V. Minimalni interval mjerenja iznosi jedno mjerenje u sekundi. Dimenzije senzora su 4.5mmX4mmX3mm, što ga čini kompaktnijim u odnosu na DHT22.

DS18B20 ima dva načina napajanja – normalni i parazitski. Sam senzor ima tri izlazna pina, odnosno signalne nožice. Pri normalnom načinu napajanja „VDD” se povezuje na 3V – 5.5V, „GND” na masu, dok se „Data” izlazni pin preko otpornika povezuje na ulazni pin mikrokontrolera, konkretno kod ovog prototipa je korišten pin PB4 i normalni način napajanja. Pri parazitskom načinu napajanja se „VDD” i „GND” pinovi povezuju na masu, dok se „Data” izlazni pin preko otpornika povezuje na ulazni pin mikrokontrolera. Napajanje se vrši isključivo preko „Data” pina. Važno je napomenuti da „Data” linija ide preko $4.7\text{k}\Omega$ pull-up otpornika, o bilo kom načinu napajanja da je riječ.

3.2.1.1. Korišćenje komercijalnog senzora

U okviru ovog projekta moguće je koristiti i komercijalni senzor, kao već integrisano i ispitano rješenje. Konkretno, korišten je senzor Xiaomi Mi Temperature and Humidity Monitor 2 za mjerenje temperature unutar pametnog hladnjaka. Proizvod u svom kućištu već sadrži senzor temperature i vlažnosti, svoj mikrokontroler, BLE (Bluetooth Low Energy) predajnik i sopstveni izvor napajanja (CR2032 baterija napona 3V).

Komercijalni senzor koristi BLE za emitovanje podataka putem periodičnih paketa (tzv. advertising frames), koje može primiti i obraditi spoljni uređaj, kao što je ESP32 BLE Tracker Hub. Mikrokontroler ESP32 zatim služi kao centralna jedinica koja interpretira podatke o temperaturi, vlažnosti i stanju baterije senzora.

Senzor najčešće koristi čip Dialog DA14580 ili njegov nasljednik Dialog DA14585, koji integriše mikrokontroler i BLE modul u jednom čipu (SoC – System on Chip). Zahvaljujući tome, uređaj može da radi autonomno i sa minimalnom potrošnjom energije, omogućavajući dug radni vijek baterije.

Prednost korišćenja komercijalnog senzora je jednostavna integracija. Zahvaljujući BLE sistemu, može se lako povezati sa drugim pametnim uređajima. Sljedeća prednost jeste kompaktan dizajn i gotova hardverska integracija, pa je uređaj spreman za upotrebu bez potrebe za dodatnim razvojem hardverskih modula. Fabrički je kalibrisan i nudi stabilne rezultate pri normalnim uslovima. Energetski je efikasan i može raditi dugo bez zamjene baterije. Deklarisani opseg temperature je od 0°C do 60°C, dok je preciznost mjerenja $\pm 0.3^{\circ}\text{C}$ (u rasponu od 0°C do 40°C, dok van tog opsega preciznost može biti manja).

Mana korišćenja komercijalnog senzora je nemogućnost direktnog programiranja. Ima zatvoren firmware od strane proizvođača (u konkretnom slučaju – Xiaomi). Ne postoji dvosmjerna komunikacija, senzor samo šalje podatke, ali ne može da prima komande. Korisnik nema potpunu kontrolu nad frekvencijom slanja podataka, već zavisi od BLE broadcast intervala.

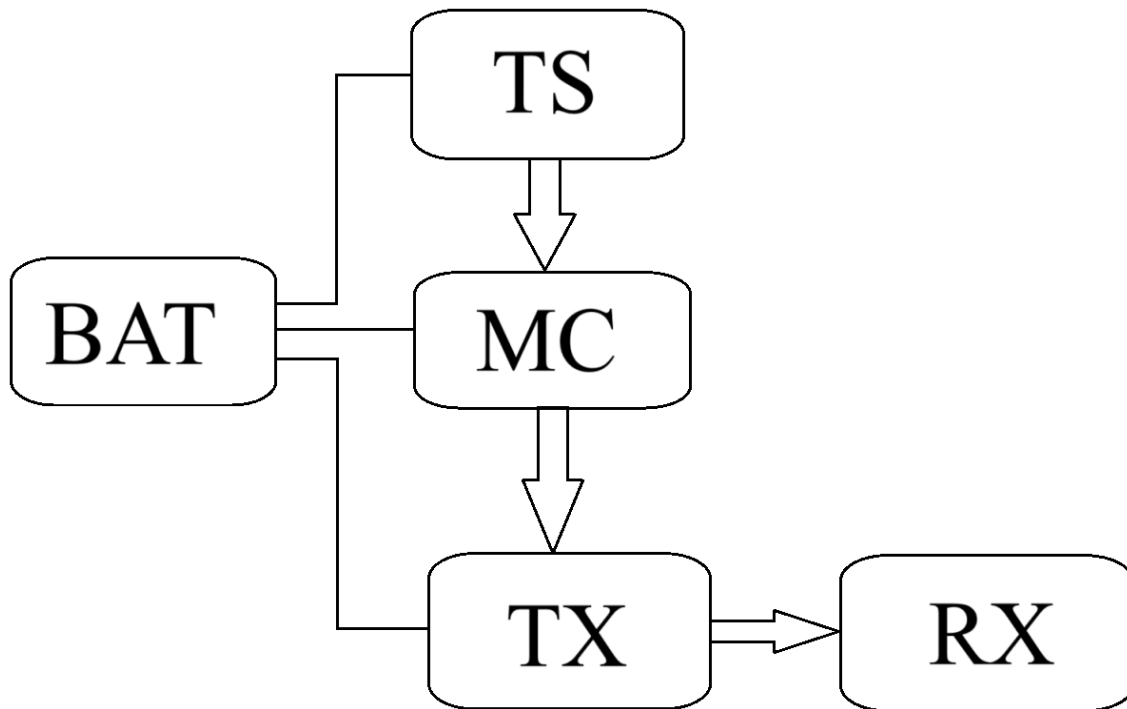
Slika 4 predstavlja fotografiju korištenog komercijalnog senzora u rashladnom uređaju (konkretno kućnom hladnjaku).



Slika 4: Fotografija korištenog komercijalnog senzora u rashladnom uređaju

3.2.2 Predajnik

Predajnik ima zadatak da, putem bežičnog protokola, šalje vrijednost izmjerene temperature prijemniku. Njegove glavne komponente su: mikrokontroler (MC), bežični komunikacioni modul (TX), temperaturni senzor (TS) i baterija (BAT). Princip rada je da senzor očitava temperaturu. Zatim, mikrokontroler prima očitane vrijednosti i prosleđuje je komunikacionom modulu, koji je, bežičnim protokolom šalje prijemniku (RX). Slika 5 prikazuje generalnu arhitekturu predajnika.



Slika 5: Generalna arhitektura predajnika

3.2.1.1. Predajnik na bazi osmobitnog RISK procesora

Mikrokontroler predstavlja mozak hardverskog sistema vezanog za predajnik. On upravlja senzorom temperature, njegovim načinom rada (da li je stalno aktivan ili u režimu štednje), kao i predajnikom i informacijama koje on šalje.

Tokom naučno-istraživačkog rada i formiranja prototipa, za potrebe predajničkog hardvera su korišteni mikrokontroleri ESP32 i ATtiny85.

Kada je riječ o mikrokontroleru ATtiny85 radi se o osmobitnom AVR RISK (Reduced Instruction Set Computer) procesoru, popularnom zbog svoje kompaktnosti, niske potrošnje i niske cijene. Ima ograničen broj resursa i optimizovan je za nisku potrošnju. Može da radi na temperaturama u opsegu od -40°C do $+85^{\circ}\text{C}$ i na naponima u opsegu od 2.7V do 5.5V.

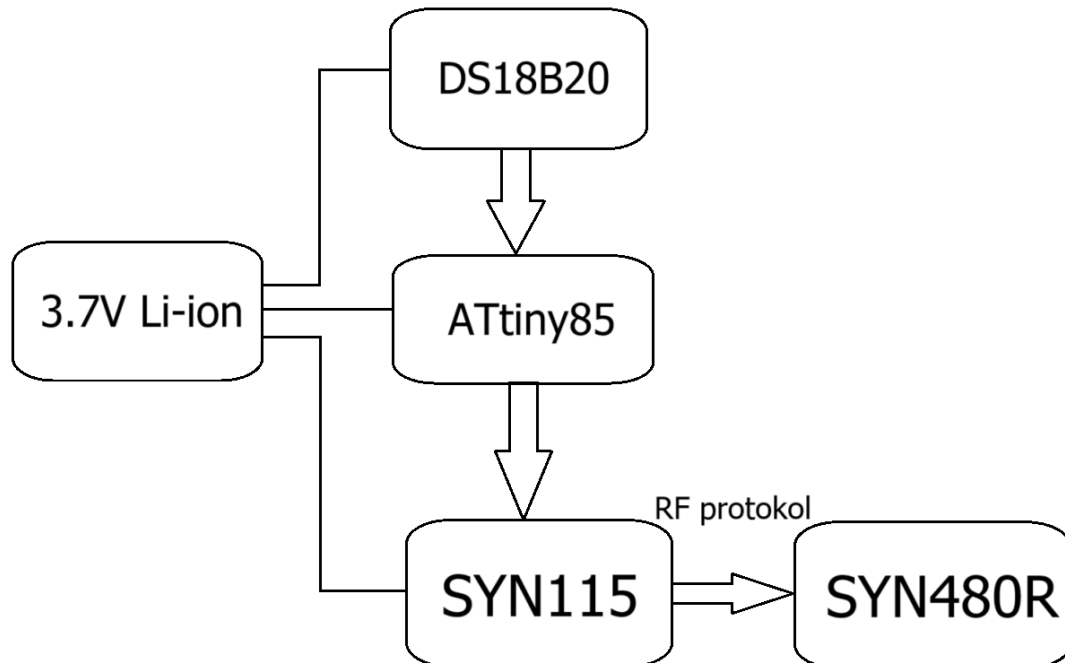
Ključne prednosti su niska cijena, niska potrošnja, koja u režimu spavanja ide i ispod $0.5\mu\text{A}$, što ga čini idealnim za dugoročne baterijske aplikacije, kao i kompaktnost – mala veličina i jednostavan dizajn, mogućnost ugradnje na male ploče. Stabilan je i predvidiv, nema multitastinga ni nepotrebnih slojeva, a prilikom programiranja ima dobru podršku Arduino proizvođača. Treba uzeti u obzir i malu memoriju od 512 B RAM, tako da nije pogodan za rad sa više senzora ili komunikacija odjednom.

Kada se uporede mikrokontroleri ESP32 i ATtiny85 po potrošnji energije, zaključuje se da je ESP32 višestruko veći potrošač. U aktivnom režimu ESP32 troši 50-70 mA, dok ATtiny85 troši 5-10 mA. Potrošnja u režimu spavanja (sleep ili snore) je takođe približno 10 puta manja u korist ATtiny85. Zaključuje se da je ATtiny idealan za aplikacije niske potrošnje energije, kao što dugotrajno očitavanje senzora upravo jeste.

Pri formiranju konkretnog prototipa sa po jednim sensorom i komunikacionim modulom, mikrokontroler ATtiny85 je u potpunosti konkurentan. Njegove karakteristike zadovoljavaju sve neophodne potrebe, dok bi se, u poređenju sa njim, za ESP32 moglo reći da je predimenzionisan i bez realne potrebe jak i složen. Takođe, kao prednost mikrokontrolera ATtiny85 se ističe i kompaktnost, uzimajući u obzir ograničen prostor u unutrašnjosti rashladnih uređaja. Važno je napomenuti da je za njegovo programiranje neophodno koristiti ISP (In-System Programming) programator.

Uz predajnik na bazi osmобitnog RISK procesora korišteni su temperaturni senzor DS18B20 i RF komunikacioni modul SYN115.

Slika 6 predstavlja šematski prikaz predajnika na bazi osmобitnog RISK procesora. Podaci se šalju prijemuniku posredstvom RF protokola.



Slika 6: Šematski prikaz predajnika na bazi osmобitnog RISK procesora

3.2.2.3. Predajnik na bazi ESP32

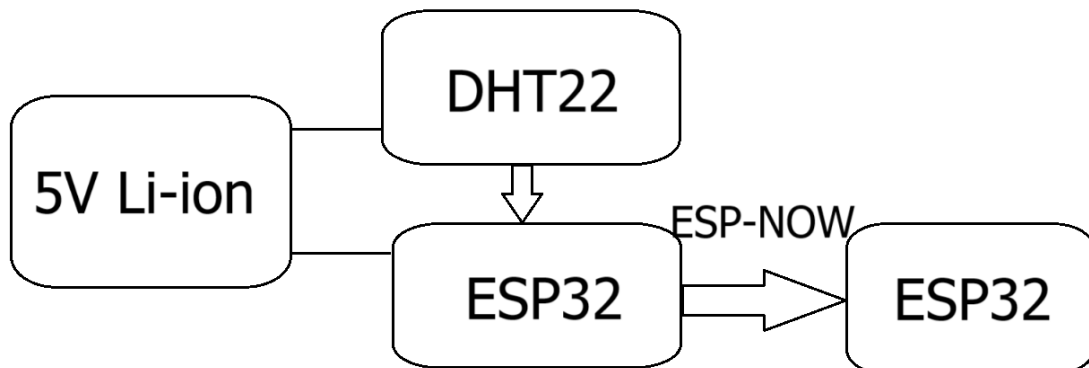
Kada je riječ o mikrokontroleru ESP32 radi se o moćnom Wi-Fi i Bluetooth modulu baziranom na ESP32 čipu, koji integriše mikroprocesor visoke efikasnosti, uz bogat skup perifernih uređaja i interfejsa. Radi na frekvenciji od 2.4 GHz i razvijen je korišćenjem TSMC 40 nm tehnologije niske potrošnje. Dizajniran je da postigne optimalne performanse kada je riječ o potrošnji energije i radio-frekvencijskoj komunikaciji, pokazujući robusnost i pouzdanost u širokom spektru aplikacija.

Jedna od ključnih prednosti ESP32-WROOM modula je njegova izuzetna fleksibilnost i jednostavnost u povezivanju sa širokim dijapazonom senzora i periferija. Za potrebe predajnika korišćen je jedan digitalni pin za komunikaciju sa sensorom, uz, naravno, pinove korišćene za napajanje i masu. ESP32 podržava biblioteke koje su dostupne u razvojnom okruženju Arduino IDE, tako da je čitanje podataka sa senzora jednostavno. Kao još jedna prednost se ističe njegov već ugrađeni protokol ESP-NOW. Time je omogućeno korištenje identičnog mikrokontrolera u hardveru vezanom za prijemnik i njihova laka međusobna komunikacija.

U poređenju sa predajnikom na bazi 8-bitnog RISK procesora, ESP32 je sposobniji mikrokontroler koji, pri eventualnoj nadogradnji predajničkog hardvera, može podržati upotrebu i komunikaciju sa većim brojem senzora, dok je ATtiny85 ograničen na pet korisnih pinova (mnogo manje nego ESP32, koji ih ima oko 30) i 512 B RAM memorije. ESP32 ima dovoljno resursa za ozbiljno unaprijeđenje sistema.

Ukoliko se upotrebljava ESP32 kao predajnik, nije neophodan poseban bežični komunikacioni modul. Sa prijemnikom (drugim ESP32 mikrokontrolerom) može komunicirati preko već ugrađenog ESP-NOW bežičnog protokola koji radi na 2.4GHz. . Kada je u pitanju brzina prenosa podataka ESP32 kao predajnik je u velikoj prednosti. Pruža zadovoljavajući domet. Za napajanje mikrokontrolera ESP32 je neophodno pažljivo upravljanje i korišćenje režima spavanja ili upotreba baterije velikog kapaciteta.

Slika 7 predstavlja šematski prikaz predajnika na bazi ESP32. Podaci se šalju prijemniku posredstvom ESP-NOW protokola.



Slika 7: Šematski prikaz predajnika na bazi ESP32

3.2.3. Modul napajanja

Budući da su mikrokontroler i senzor unutar rashladnog uređaja potrošači električne energije, za njih se mora obezbijediti odgovarajuće napajanje. Jedan od načina za obezbjeđenje napajanja je ispravljač – adapter, koji, najpogodnije provučen kroz vrata rashladnog uređaja, obezbjeđuje mikronotroleru i senzoru stalan napon od 5V. To rješenje iziskuje prepravku na samom rashladnom uređaju. Druga alternativa je baterijsko napajanje, koje je, budući da se radi o prenosnom rješenju bez dodatne konfiguracije korišteno pri izradi ovog prototipa.

Kako bi se obezbijedilo baterijsko napajanje postoji više solucija. Jedna od njih je korištenje standardne alkalne baterije 9V. Međutim, ta solucija nije pogodna, kada se uzme u obzir njihov kapacitet koji je obično oko 500mAh. Osim toga, mora da se koristi BUCK regulator, kako bi se napon od 9V sveo na 5V. Mane su im što kapacitet značajno opada pri niskim temperaturama. Iako nema trajnog oštećenja, uređaj često prestaje da radi. Bezbjedni temperaturni opseg je od 0°C do 50°C.

Mnogo bolja alternativa su litijum-jonske baterije. Karakteriše ih mogućnost brzog punjenja i par puta veći kapacitet. Kada se uporede sa klasičnim alkalnim baterijama imaju višu cijenu ali bolje performanse pri niskim temperaturama. Kod litijum-jonskih baterija postoji manji pad napona i optimalna temperatura rada im je iznad -10°C. Bolje drže kapacitet pri niskim temperaturama. Bezbjedni temperaturni opseg im je od -20°C do 60°C. Treba voditi računa o bezbjednom opsegu za punjenje, koji je od 0°C do 45°C. Punjenje ispod donje temperaturne granice je opasno, jer može dovesti do unutrašnjeg kratkog spoja.

Kao solucija se izdvaja baterija od 2200mAh napona 3.7V, koja je i korištena pri izradi prototipa. Obzirom na kapacitet baterije poželjno bi bilo koristiti mikrokontroler u operativnom

načinu dubokog sna. Kako bi baterija trajala što duže preporuka je buditi mikrokontroler na svaki minut kako bi poslao podatke, dok bi ostatak vremena proveo u režimu spavanja. Prilikom upotrebe navedene baterije i mikrokontrolera ATtiny85, senzora i komunikacionog modula, nije neophodno vršiti nikakvo svođenje napona. Za nesmetan rad je dovoljno direktno vezati bateriju i potrošače.

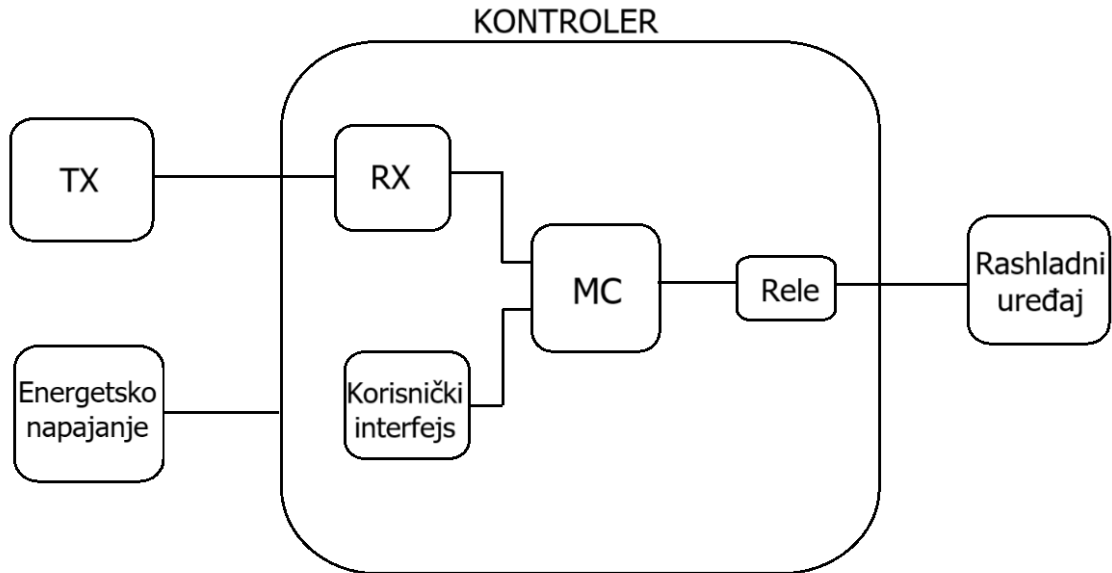
Još jedna solucija jeste upotreba eksterne baterije – prenosnog punjača (power bank). Eksterna baterija je litijum-jonska i ima sve njene karakteristike. Kapacitet eksterne baterije je obično od 5000mAh do 20.000mAh, dok je pri eksperimentalnom radu u ovom projektu korištena baterija od 10.000mAh. U odnosu na 3.7V bateriju, kapacitet je veći pet puta, dok je cijena veća preko dva puta. Ukoliko se koristi ESP32 kao predajnik, on se može napajati direktno, putem već postojećeg kabla i USB-C ulaza. Osim toga, u slučaju potrebe može istovremeno napajati i više potrošača. Najveća mana eksterne baterije jesu relativno velike dimenzije i problem smještaja u rashladni uređaj.

Kako eksterna baterija takođe radi na 3.7V internoj bateriji, a izlaz je 5V, postoje gubici u boost konverziji. Efikasnost je oko 85%. Kada se u kalkulaciju uzme maksimalna procijenjena potrošnja od maksimalnih 14mAh po satu, baterija od 2200mAh bi trajala oko pet dana, dok bi 10.000mAh eksterna baterija trajala otprilike 25 dana. Eksterna baterija omogućava rjeđu potrebu za punjenjem, nauštrb relativno velikih dimenzija i potencijalnog problema smještaja uređaja.

Baterija korištena u komercijalnom senzoru jeste CR2032, napona 3V i nominalnog kapaciteta 220-235mAh (zavisno od proizvođača). Radi se o litijumskoj dugmastoj bateriji kompaktnih dimenzija. Idealna je za uređaje koji rade povremeno, a, budući da je litijumska, bezbjedni temperaturni opseg im je od -20°C do 60°C . Važno je napomenuti da baterija nije punjiva.

3.2.2. Prijemnik

Prijemnik ima zadatak da prima informaciju o temperaturi, kao i da omogući korisniku manipulaciju tim parametrom. Korisnik preko korisničkog interfejsa mora biti u stanju da podešava temperaturu na željenu vrijednost, kao i da bira između predefinisanih vrsta vina i idealnih temperatura za njihovo hlađenje. Slika 8 predstavlja generalnu arhitekturu prijemnika.



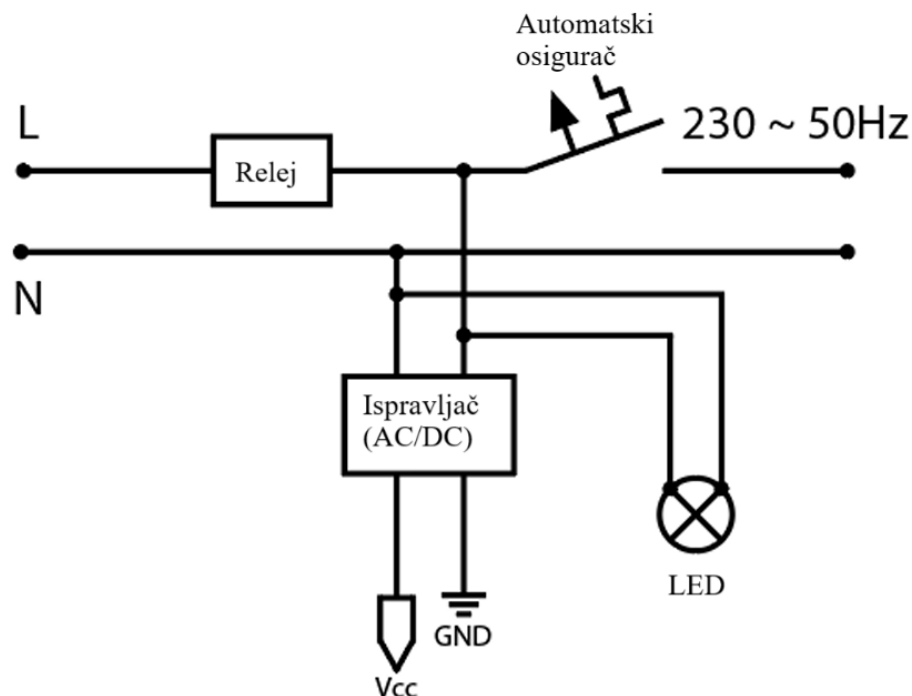
Slika 8: Generalna arhitektura prijemnika

Korišćene su sljedeće hardverske komponente:

- Mikrokontroler opšte namjene
- Ispravljač
- Automatski osigurač
- Rele
- Bežični komunikacioni modul RX
- LCD ekran
- LED signalne lampice
- Kontrolni tasteri za podešavanje parametara

Hardver prijemnika je u obliku produžnog kabla, dok se nabrojane hardverske komponente nalaze u kućištu. Pri formiranju prototipa kao kućište je poslužila razvodna kutija, dok se korisnički interfejs nalazi na njenom poklopcu.

Slika 9 predstavlja šematski prikaz strujnog kola prijemnika.



Slika 9: Strujno kolo prijemnika

Mikrokontroler predstavlja „mozak” prijemnika. On ima zadatak da, na osnovu podataka koje dobije od prijemnika i željenih preferenci korisnika upravlja relejem, koji uključuje/isključuje rashladni uređaj. Ključne uloge su prijem podataka, analiza, donošenje odluka i kontrola rada rashladnog uređaja. Po potrebi obavještava korisnika i loguje podatke.

Pri prijemu RF signala, mikrokontroler morada ga obradi. On je u početku „sirov” i mora se prevesti iz binarnog impulsa u korisne vrijednosti. Kada mikrokontroler dobije validnu temperaturu koristi uslovne naredbe da bi uključio/isključio uređaj. Nakon prijema temperature mikrokontroler obrađuje podatak – provjerava da li je temperatura viša, niža ili jednaka referentnoj vrijednosti koju je korisnik unio ili koja je predefinisana u kodu. Na osnovu analize podataka, mikrokontroler donosi sljedeće odluke:

- Ukoliko je temperatura viša od gornje granice – uključuje hlađenje
- Ukoliko je temperatura u granicama – ne radi ništa
- Ukoliko temperatura padne ispod donje granice – isključuje hlađenje

Kada mikrokontroler odluči da treba da se aktivira sistem za hlađenje, on šalje signal preko digitalnog izlaznog pina do releja, koji kontroliše napajanje kompresora. Tako mikrokontroler zapravo fizički utiče na rad rashladnog uređaja.

Mikrokontroler je povezan sa LCD ekranom i LED diodama kako bi informisao korisnika o trenutnoj temperaturi i statusu sistema. Povezan je sa kontrolnim tasterima kako bi korisnik unosio željene preference.

Uloga prijemnika (bilo da se radi o RF prijemniku ili mikrokontroleru poput ESP32 koji

prima podatke) jeste da prima informacije koje su poslate putem određenog medijuma (radio-talasa, Wi-Fi, Bluetooth itd.), te ih obrađuje i po potrebi reaktivno djeluje. Pri pisanju ovog rada korišten je prijemnik na bazi ATmega328P (konkretno Arduino UNO) i prijemnik na bazi ESP32.

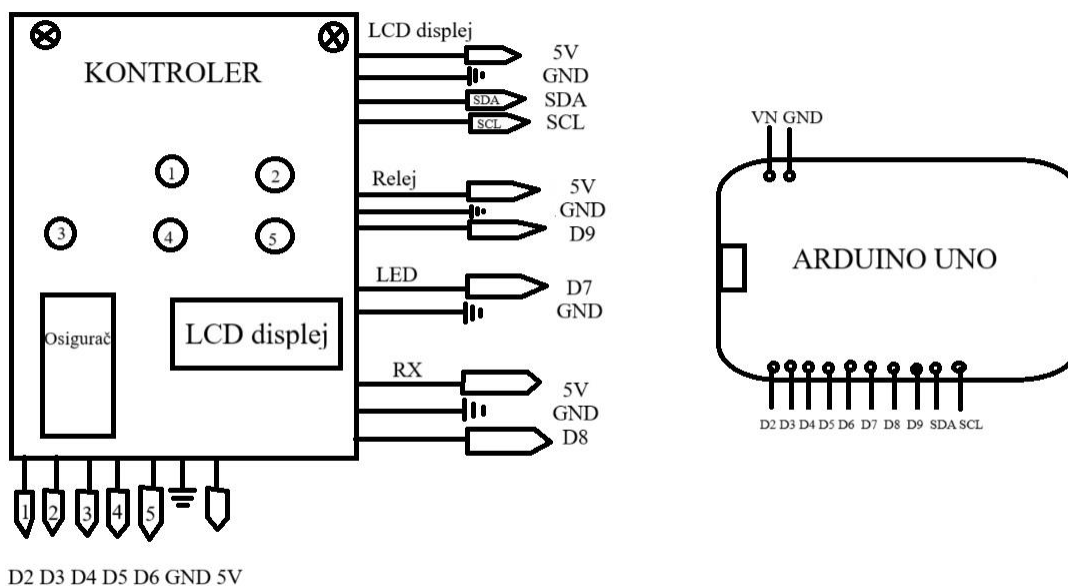
3.2.2.1. Prijemnik na bazi ATmega328P

Arduino UNO je mikrokontrolerska ploča zasnovana na ATmega328P čipu i sam po sebi nema ugrađen bežični komunikacioni modul, pa se za bežičnu komunikaciju mora dodati RF modul 433MHz.

U ovom sistemu, prijemnik na bazi ATmega328P funkcioniše kao centralni procesor koji prima digitalne podatke sa RF modula (konkretno SYN480R) povezanog na njegov digitalni ulazni pin. Radi se o pasivnom modulu koji detektuje i demoduliše radio signal poslat sa predajnika (takođe posredstvom modula RF 433 MHz). Nakon demodulacije, RF modul prenosi dekodirani signal dalje do Arduino ploče, gdje se podaci obrađuju i vrši se djelovanje na osnovu njih.

Prednosti su jednostavna implementacija sa dostupnim modulima i niska cijena sistema. Mane su nedostatak ugrađene podrške za moderne komunikacione protokole poput Wi-Fi, BLE itd. Samim tim, nema mogućnosti direktne komunikacije s komercijalnim BLE senzorima.

Slika 10 predstavlja šematski prikaz prijemnika na bazi ATmega328P, odnosno Arduino UNO mikrokontrolerske ploče.



Slika 10: Šematski prikaz prijemnika na bazi ATmega328P

3.2.2.2. Prijemnik na bazi ESP32

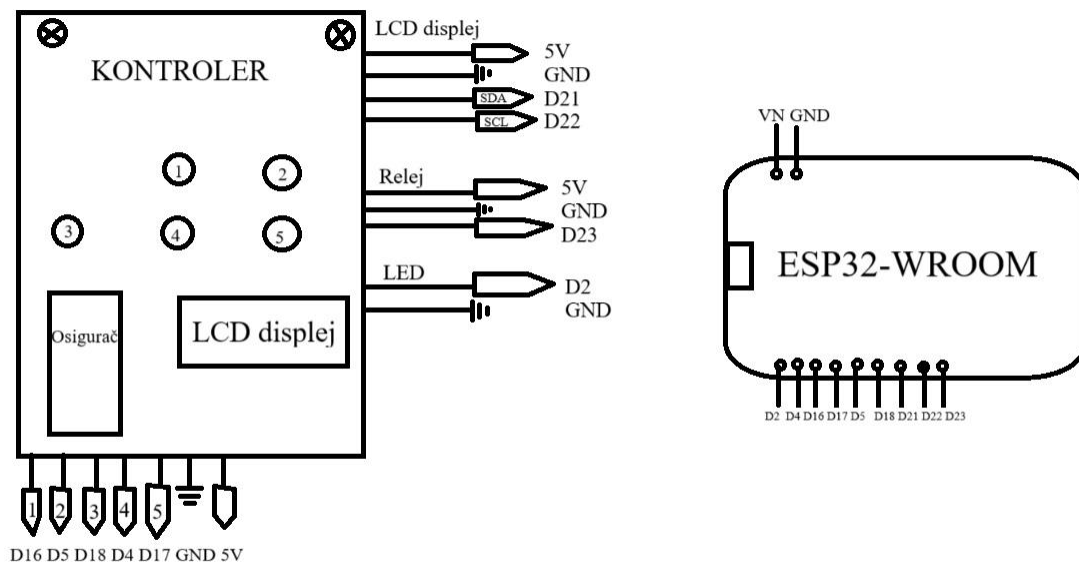
ESP32 je moćniji mikrokontrolerski sistem (SoC) koji dolazi sa ugrađenim Wi-Fi i Bluetooth (uključujući BLE) modulima, kao i dovoljno procesorske snage i memorije za napredne funkcionalnosti.

U ovom sistemu, ESP32 se koristi kao samostalni prijemnik i procesor podataka, sposoban da radi sa više različitih komunikacionih protokola. U slučaju protokola ESP-NOW, predajnik i prijemnik mogu da komuniciraju direktno. ESP-32 kao prijemnik prima podatke koje šalje drugi ESP32 (predajnik) putem Wi-Fi čipa, ali u posebnom režimu rada koji omogućava visoku pouzdanost.

Uz to, ESP32 može da prima podatke i sa komercijalnih BLE senzora, zahvaljujući ugrađenom BLE prijemniku. U ovom režimu, ESP32 prima BLE broadcast pakete koje emituje predajnik, a zatim ih parsira i koristi za dalju obradu.

Prednosti upotrebe ESP32 kao prijemnika su širok spektar upotrebe (može primati podatke i od drugih mikrokontrolera, ali i od komercijalnih BLE senzora), nema potrebu za dodatnim komunikacionim modulima (sve je integrisano). Omogućava dvosmjernu komunikaciju i lokalnu obradu podataka. Ima više memorije i procesorske snage u odnosu na prijemnik na bazi ATmega328P. Mana je veća potrošnja energije u odnosu na jednostavnije prijemnike.

Slika 11 predstavlja šematski prikaz prijemnika na bazi ESP32.



Slika 11: Šematski prikaz prijemnika na bazi ESP32

3.2.2.3. Upotreba automatskog osigurača

Automatski osigurač je električni sigurnosni uređaj namijenjen zaštiti električnog kola od oštećenja izazvanih preopterećenjem ili kratkim spojem. Njegova osnovna funkcija je da prekine protok struje kako bi zaštitio opremu i spriječio rizik od požara. Za razliku od klasičnog topljivog osigurača, automatski osigurač se može ponovo uključiti nakon što je prekinuo strujno kolo.

Ukoliko dođe do povećanja struje koja premaši dozvoljenu maksimalnu vrijednost, dolazi do povećanja temperature bimetalne trake unutar automatskog osigurača, što uzrokuje intenzivnije zagrijavanje i širenje jednog metala u odnosu na drugi. Ovo dovodi do savijanja trake i prekidanja kontakta u određenom momentu, čime se prekida i samo strujno kolo. Ova zaštita je spora – mjeri prosječno opterećenje tokom vremena. U slučaju da dođe do kratkog spoja i naglog povećanja struje, ta struja stvara jak magnetni impuls u elektromagnetnom kalemu. Tada magnet privlači metalnu polugu ili klip koji trenutno prekida strujni tok. Ova zaštita je veoma brza – vrijeme reakcije se mjeri u stotim djelovima sekunde.

Automatski osigurači se izrađuju za nazivne struje: 6A, 10A, 16A, 20A, 25A, 32A, 40A, 50A, 63A, 100A. Za većinu standardnih kućnih uređaja bijele tehnike, trenutna vrijednost struje pri pokretanju kompresora obično iznosi između 5A i 8A. Tokom normalnog rada, vrijednost struje može biti niža, obično oko 1A do 2A, u zavisnosti od efikasnosti uređaja i potreba za hlađenjem.

Pri izradi ovog prototipa je korišćen automatski osigurač od 25A – jednopolni Schneider C25A. U nazivu „25A“ označava maksimalnu struju opterećenja, dok oznaka „C“ označava da se elektromagnetna zaštita aktivira kada struja pređe 5 do 10 puta nazivnu vrijednost, odnosno 125-250A, u slučaju kratkog spoja. Proizvođač kao prednost ističe dug radni vijek i veliki broj mehaničkih (do 20.000) i električnih (do 10.000) ciklusa. Dimenzije osigurača su 78.5mm x 18mm x 85mm. Masa osigurača iznosi 215 grama. Po IEC 60529 standardu, nivo zaštite je IP20 i IP40. Radna temperatura je u opsegu od -35°C do +70°C.

3.2.2.4. Upotreba ispravljača

Za realizaciju prototipa korišćen je ispravljač (adapter) čiji se izlazni napon može mijenjati u diskretnim koracima, i to: 3V / 4,5V / 5V / 6V / 7,5V / 9V / 12V \pm 0,1V. Maksimalna izlazna snaga iznosi 12W. Ispravljač se koristi za napajanje upravljačke elektronike, koja je projektovana za napon od 5V. Zbog toga je na izlazu ispravljača podešen pomenuti nivo napona. Mikrokontroleru, releju i korisničkom interfejsu je neophodan konstantan napon od 5V za nesmetan rad. Konkretni RF prijemnik koji je korišten pri formiranju prototipa radi na naponu od 3.3V, pa je njegovo napajanje izvučeno sa mikrokontrolera.

Konkretno je korišten univerzalni AC/DC adapter proizvođača EnerGenie.

3.2.2.5. Aktuator

Za ulogu aktuatora je korišten relejni modul. On predstavlja izvršni element u tehničkom sistemu. Po odluci kontrolera sprovođi naredbu u fizički svijet, konkretno uključuje/isključuje rashladni uređaj.

Relejni modul je elektronski modul koji služi za upravljanje uređajima koji koriste visoke struje ili napone, preko niskonaponskog upravljačkog signala (koji, konkretno potiče iz mikrokontrolera).

Rele je električno napajani prekidač. Aktivira se pomoću visokog napona iz upravljačkog dijela kola, kako bi uključio ili isključio drugi dio kola, u ovom slučaju – rashladni uređaj. Prilikom rada sa relejima, obavezno je odvojiti upravljački napon od napona na kontaktima, odnosno omogućiti galvansku izolaciju između upravljačke elektronike (mikrokontrolera) i potrošača na višem naponskom nivou (rashladnog uređaja).

Prilikom izrade prototipa korišten je relejni modul 5V DC, 10A. „10A“ u oznaci označava da je maksimalna struja koju ovaj rele može da uspostavi ili prekine jednaka 10A. Rele se aktivira kada se na njegov kalem dovede DC napon od 5V, što je i standard za mnoge mikrokontrolere.

Korišteni rele pripada grupi jednopolnih preklopnih releja (SPDT-Single Pole Double Throw), što znači da ima:

- Jedan zajednički kontakt (COM)
- Jedan normalno zatvoren kontakt (NC – Normally Closed)
- Jedan normalno otvoren kontakt (NO – Normally Open)

Kod NO kontakta, kada se na kalem releja dovede napon, elektromagnet privlači polugu ka sebi, prekidač se zatvara i utiče na sekundarno električno kolo. Kada se prekine tok struje kroz kalem, elektromagnet se isključuje, opruga povlači polugu nazad i prekidač se otvara. Suprotno bi važno za NC kontakt – on je zatvoren dok rele nije aktivan i prekida se kada se rele aktivira.

Pri izradi prototipa korišten je NC kontakt. Ukoliko izmjerena temperatura premaši zadatu, linija koda „digitalWrite(relay,LOW)“ aktivira rele (aktivira se kada se pin postavi na LOW). U tom trenutku rashladni uređaj se uključuje i na serijskom monitoru se pojavljuje ispis „FRIDGE ON“. U suprotnom, linija koda „digitalWrite(relay,HIGH)“ isključuje rashladni uređaj i na serijskom monitoru se pojavljuje ispis „FRIDGE OFF“.

Rele je povezan na izlazni pin mikrokontrolera, na 5V i na masu. Na COM pin releja se veže faza iz osigurača. Fazni ulaz potrošača ide na jedan od pinova – NO ili NC (u konkretnom slučaju NC), dok drugi pin ostaje nepovezan. Rele se postavlja iza automatskog osigurača, na fazni provodnik i prekida ili spaja fazu koja ide ka potrošaču. Automatski osigurač štiti cijeli strujni krug, uključujući i rele.

3.2.2.6. LCD ekran

LCD ekran omogućava uvid u trenutno stanje kontrolnog sistema, tako što ispisuje temperaturu unutar rashladnog uređaja. Prototip koristi LCD displej modul sa I2C serijskim interfejsom 1602, odnosno 16x2 LCD ekran, koji prikazuje 16 karaktera u dva reda. Tipičan I2C LCD je zasnovan na HD44780 kontroleru i I2C adapteru za LCD. U središtu adaptera nalazi se osmobaritni I/O čip – PCF8574. Ovaj čip pretvara I2C podatke sa mikrokontrolera u paralelne podatke koji su neophodni kako bi ih LCD ekran razumio.

Svaki karakter je prikazan na LCD ekranu u matrici od 5x8 tačaka. Kontrast između ispisa i pozadinskog osvijetljenja je podešiv pomoću potencijometra, koji se nalazi sa zadnje strane I2C adaptera. Zahvaljujući adapteru, za spajanje LCD ekrana i mikrokontrolera je potrebno povezati četiri pina. Na kontroleru se izdvajaju četiri pina i to:

- GND – povezuje se na masu mikrokontrolera
- VCC – povezuje se na 5V
- SDA – Serial Data Line, služi za prenos podataka, linija za dvosmjernu komunikaciju, povezan je na GPIO 21 mikrokontrolera, koji je i podrazumijevani pin
- SCL – Serial Clock Line, služi za sinhronizaciju, mikrokontroler generiše taj signal, povezan je na GPIO 22 mikrokontrolera, koji je i podrazumijevani pin

Kako bi se omogućio valjan rad LCD ekrana korišćene su biblioteke Wire.h i LiquidCrystal_I2C.h.

LCD ekran je ekonomski efikasno i jednostavno rješenje za prikazivanje potrebnih podataka neophodnih za valjan rad prototipa. Kombinuje nisku potrošnju i jasan prikaz, što ga čini idealnim za prikaz korisniku potrebnih podataka.

3.2.2.7. Ostali hardverski elementi

Pri izradi prototipa su korištene dvije LED signalne lampice. Prva lampica pokazuje trenutno stanje (ON ili OFF) rashladnog uređaja – svijetli samo ukoliko kompresor radi. Ona je, posredstvom otpornika, povezana na izlazni pin mikrokontrolera. Druga lampica svijetli dok god je kontroler uključen i povezana je, posredstvom otpornika, na VCC sa ispravljača. Korištene su standardne LED lampice.

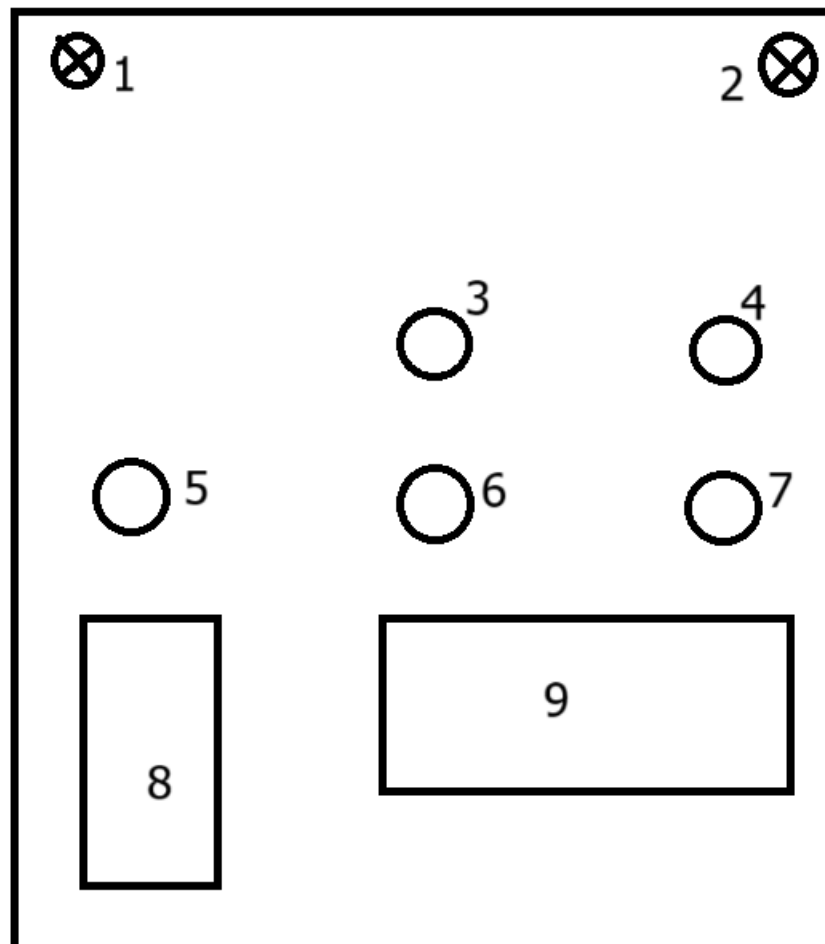
Tasteri su dio korisničkog interfejsa pomoću kojih korisnik podešava željene ili predefinisane vrijednosti temperature. Radi se o standardnim tasterima korišćenim za Arduino projekte. Povezani su na izlazne pinove mikrokontrolera.

Kako bi se svi hardverski elementi objedinili u jednu cjelinu – kontroler, kao kućište je

korištena razvodna kutija Elettrocanali, dimenzija 240mm x 190mm x 90mm.

3.2.2.8. Korisnički interfejs

Korisnički interfejs (Slika 12) predstavlja način na koji korisnik komunicira sa nekim uređajem ili softverom. Korisnički interfejs kontrolera je dizajniran da bude što lakši za upotrebu. Omogućava uvid u trenutne parametre, kao i laku manipulaciju njima.



Slika 12: Prikaz korisničkog interfejsa

Komponente korisničkog interfejsa koje se ističu su sljedeće:

1. LED signalna lampica koja svijetli ukoliko je kompresor pokrenut
2. LED signalna lampica koja svijetli ukoliko je kontroler pod naponom
3. Taster za manuelno povećavanje temperature (+)
4. Taster za provjeru trenutnog stanja (CHECK)
5. Taster za resetovanje (RESET)
6. Taster za manuelno smanjenje temperature (-)

7. Taster za odabir predefinisano načina hlađenja u zavisnosti od vrste vina (MODE)
8. Automatski osigurač
9. LCD ekran

3.2.2.8.1. Režimi rada kontrolera

Pritiskom tastera MODE biramo jedan od pet predefinisanih temperatura pogodnih za hlađenje pet vrsta vina i to:

1. Crveno (eng. Red): 15,00 °C
2. Bijelo (eng. White): 10,00 °C
3. Roze (eng. Rose): 8,00 °C
4. Pjenušavo (eng. Sparkling): 5,00 °C
5. Aromatizovano (eng. Flavored): 6,00 °C

Osim predefinisanih temperatura, korisnik može putem tastera + i – da ručno podešava određenu temperaturu, uz rezoluciju (najmanji korak u kojem očitavanje sa senzora može promijeniti vrijednost) od 0.1°C. Slika 13 prikazuje ispis LCD ekrana u zavisnosti od vrste vina.

T:0.0 Ref:10.0 Vine: White	Vine: White RefTemp: 10.0	Vine: Sparkling RefTemp: 5.0
T:0.0 Ref:15.0 Vine: Red	Vine: Rose RefTemp: 8.0	Vine: Flavored RefTemp: 6.0

Slika 13: Ispis na LCD ekranu u zavisnosti od odabira vrste vina

Kao što je i prikazano, na ekranu se uočavaju trenutna temperatura, jedna od pet vrsta vina, kao i referentna – predefinisana vrijednost temperature za hlađenje odabranog vina.

3.2.3. Komunikacija

Kako bi se omogućila komunikacija između predajnika i prijemnika neophodno je koristiti neki od bežičnih komunikacionih protokola. S obzirom da su predajnik i komponente vezane za njega baterijski napajane, neophodno je koristiti neki od protokola niske potrošnje – low power. Takvi su protokoli BLE (Bluetooth Low Energy), RF, ESP-NOW, LoRa, ZigBee. Pri pisanju ovog rada su korišteni ESP-NOW i RF bežični komunikacioni protokoli.

RF protokol podrazumijeva bežični prenos podataka putem radio-talasa. U kontekstu mikrokontrolera, najčešće se koriste moduli koji rade na frekvencijama poput 315MHz, 433MHz, 868MHz ili 915MHz, u zavisnosti od regiona i regulative. Za RF protokol neophodni su TX (predajnik) i RX (prijemnik) moduli, koji moraju međusobno da koriste istu brzinu i modulaciju. Pri formiranju prototipa korišteni su moduli koji rade na frekvencijama od 433MHz i to SYN115 (TX modul) i SYN480R (RX modul) proizvođača ACEIRMC.

RF moduli najčešće rade sa ASK (Amplitude Shift Keying) ili OOK (On-Off Keying) modulacijom. To su najjednostavniji digitalni prenosi – signal je visok (1) ili nizak (0). Komunikacija je jednosmjerna, podatak ide od jednog modula ka drugom). Nema povratne sprege, osim ako se softverski ne implementira. Protokol nije pouzdan za kompleksniju i dvosmjernu komunikaciju, a nema ni potvrde prijema. Komunikacija je bez enkripcije – svi koji prime frekvenciju čuju sadržaj. Podaci se šalju kao binarni paketi, pomoću biblioteka koje ih enkodiraju, konkretno je korištena VirtualWire.

ESP-NOW je bežični komunikacioni protokol koji je razvila kompanija Espressif System, koja je ujedno i proizvođač korištenih mikrokontrolera. Ovaj protokol omogućava komunikaciju uređaja bez korišćenja Wi-Fi mreže, čime se znatno smanjuje potrošnja energije i povećava brzina prenosa podataka.

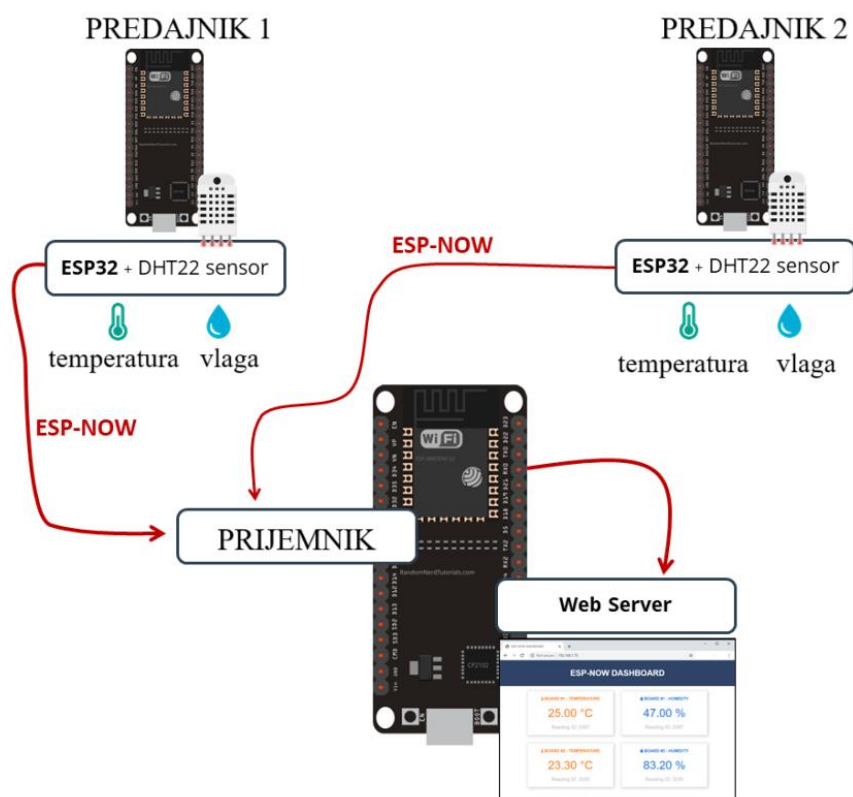
ESP-NOW koristi Wi-Fi frekvencijski opseg od 2.4GHz, ali ne zahtijeva uspostavljanje konekcije sa Wi-Fi mrežom. Ne postoje DHCP (protkol za dinamično konfigurisanje hosta), IP adresa, TCP/IP slojevi. Uređaji su potpuno nezavisni od internet konekcije. Mikrokontroleri međusobno komuniciraju direktno, koristeći svoje MAC adrese. MAC adresa predajnika korištenog za izradu prototipa je: 80:7D:3A:E2:FF:74, dok je MAC adresa prijemnika: 8C:4F:00:AD:67:1C.

ESP-NOW je pogodan za baterijski napajane sisteme, jer omogućava uređajima da brzo prenesu podatke i pređu u režim dubokog sna (deep sleep mode). Zahvaljujući činjenici da se izostavlja Wi-Fi povezivanje, kašnjenje u prenosu podataka je minimalno, što čini ESP-NOW idealnim za real-time aplikacije.

Superiornost ESP-NOW protokola se ogleda i u tome što predajnik može da šalje podatke jednom prijemniku (unicast) ili više prijemnika odjednom (broadcast). Unicast slanje podataka se vrši preko MAC adrese, dok se putem broadcasta šalje svim uređajima u dometu. Mikrokontroler koristeći ovaj protokol može komunicirati sa do 20 drugih uređaja. Naravno, može postojati i slučaj kada više predajnika šalje podatke jednom prijemniku. Slika 14 predstavlja takav slučaj, gdje prijemnik dodatno podatke šalje na Web server.

Arhitektura ESP-NOW komunikacije je jednostavna. Predajnik šalje paket podataka direktno prijemniku. Paket sadrži: MAC adresu prijemnika, podatke koje je definisao korisnik (u konkretnom primjeru to je temperatura) i CRC (kod za kontrolu tačnosti prenijetih podataka) i druge kontrolne podatke. Maksimalno ograničenje ESP-NOW paketa iznosi 250 bajta korisničkih podataka.

Pomenuti komunikacioni protokol karakteriše i veoma veliki domet (čak do 200m). Navedeno, ali i prednosti kao što su brza lokalna bežična komunikacija, niska potrošnja energije i jednostavnost bez mrežne infrastrukture, čine ga idealnim u postupku izrade prototipa.



Slika 14: Primjer ESP protkola sa dva predajnika i slanjem podataka na web server

3.2.4. Energetsko napajanje i zaštita

Zbog čestog uključivanja i isključivanja rashladnog uređaja može doći do oštećenja kompresora usljed visoke početne vrijednosti struje. Iz tog razloga je neophodno ograničiti početnu struju. Konvencionalni rashladni uređaji kućne bijele tehnike uglavnom sadrže integrisanu zaštitnu opremu. Kod komercijalnih modela je najčešća termička zaštita. To je zaštita zasnovana na bimetalnim sklopkama ili termalnim prekidačima unutar kompresora ili u blizini motora. Reaguje na prekomjerno zagrijavanje – kada temperatura pređe prag, prekida strujni tok. Sve više modernih vinskih vitrina, sa više zona hlađenja i digitalnim upravljanjem, posjeduju pametnu zaštitu. To je vrsta zaštite koja koristi mikrokontroler unutar rashladnog uređaja koji

mjeri temperaturne profile i struju kompresora, a kod naprednijih sistema i pritisak freona i status vrata (senzorima). Kada otkrije neregularnost, rashladni uređaj može da isključi kompresor ili prikaže grešku (kod naprednijih sistema šalje i poruku korisniku).

3.2.4.1 PTC starter i zaštita od preopterećenja

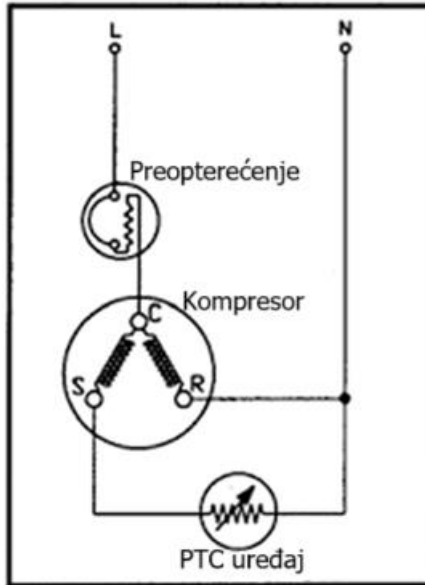
PTC rele je skraćenica za Positive Temperature Coefficient rele, što znači da otpornik povećava svoju otpornost sa porastom temperature. U kontekstu rashladnih uređaja, PTC starter se koristi da privremeno omogući rad startnog kalema motora kompresora, a zatim se automatski isključi kako se zagrijava.

PTC rele je redno povezan sa sekundarnim namotajem (startnim namotajem) motora i paralelno povezan sa glavnim (radnim) namotajem motora (Slika18 i Slika20). Kada se na oba namotaja dovede napon od 230V naizmjenične struje, dolazi do fazne razlike usljed čega se stvara eliptično rotirajuće magnetno polje i stvara se startni obrtni moment potreban za normalan rad motora.

Nakon toga, pošto je PTC rele u stanju visoke otpornosti, startni krug je skoro potpuno prekinut i samo radni namotaj napaja motor za daljnji rad.

PTC rele, poznat i kao beskontaktni starter, je u stvari termistorski rele. Kada se kompresor pokreće, temperatura PTC komponente je niska i otpornost mala, pa se može smatrati skoro kratkospojenim (provodnim) kolom. Ako početna struja poraste 4-6 puta u odnosu na normalnu, temperatura elementa brzo raste. Uz to, otpor raste višestruko, a struja koja prolazi kroz PTC komponentu postaje mala, što se može smatrati skoro prekidom kola.

Zaštita od preopterećenja (Slika 15) obično sadrži bimetalnu traku ili rele koji reaguju na promjene temperature izazvane povećanom strujom. Kada se aktivira, zaštita od preopterećenja prekida električno kolo ka kompresoru i isključuje napajanje motora.



Slika 15: Šematski prikaz PTC releja

3.3. Algoritam i softver

Kada je riječ o softveru, pisan je u razvojnom okruženju Arduino IDE, kako za predajnik tako i za prijemnik. Dva mikrokontrolera međusobno komuniciraju putem bežičnog komunikacijskog protokola. Tokom snimanja eksperimentalnog dijela korišteni su programi Python i Matlab.

3.3.1. Upravljački algoritam

Upravljačke komande se zadaju putem operativnog sistema u realnom vremenu FreeRTOS. To je namjenski operativni sistem za mikrokontrolere kao što je ESP32, Arduino UNO i sl. Njegova upotreba omogućava mikrokontroleru da upravlja sa više zadataka (taskova) istovremeno, tj. da se više funkcija izvršava istovremeno (paralelno) na organizovan način.

FreeRTOS omogućava multitasking – više stvari može da se odvija u isto vrijeme (npr. očitavanje senzora, slanje podataka, uspavljivanje uređaja). Pri pisanju ovog rada korišten je FreeRTOS, ali indirektno, bez kreiranja direktnih RTOS zadataka. Npr. funkcija loop() u kodu funkcionise kao FreeRTOS task. Osim toga, RTOS se koristi posredno i pri ESP-NOW komunikaciji, jer Wi-Fi protokol radi u okviru FreeRTOS zadataka. Koristi se i za senzore za mjerenje temperature, jer upravljanje vremenom i prekidima obavlja FreeRTOS.

Opis za dijagram toka FreeRTOS zavisno od konkretne funkcije jeste:

setup():

- Inicijalizuje serijsku komunikaciju, LCD, ESP-NOW
- Postavlja ulaze i izlaze, kreira FreeRTOS task za tastere

loop():

- Periodično poziva funkciju relayControl()

buttonReading():

- Očitava vrijednost svih tastera i ažurira vrstu vina i referentnu temperaturu

onDataRecv():

- Prima vrijednost temperature, ažurira je i osvježava LCD

LCD_display():

- Ažurira prikaz u zavisnosti od moda (0: standardni, 1:promjena, 2:provjera)

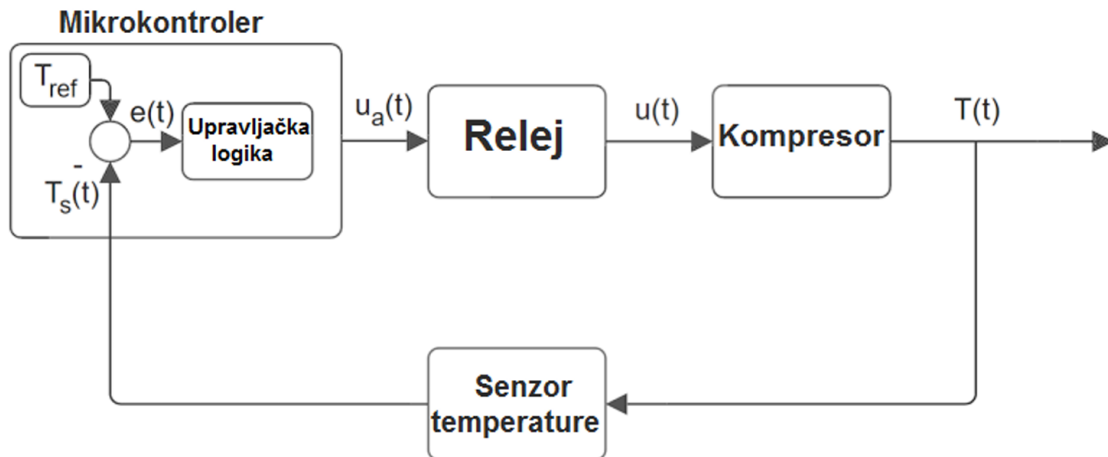
3.3.2. Upravljačka logika

Upravljačka logika se može definisati u sljedećim koracima:

1. Predajnik mjeri temperaturu pomoću temperaturnog senzora
2. Temperatura se šalje putem bežičnog protokola ka prijemniku
3. Prijemnik upoređuje primljenu temperaturu sa referentnom
4. Ako je temperatura veća od referentne za vrijednost višu od 0.5°C – uključuje se

- rele (uključuje se i kompresor)
- 5. Kada temperatura padne ispod donje granice – isključuje se rele
- 6. Ciklus se ponavlja

Upravljačka logika radi na principu zatvorene povratne sprege (Slika 16).



Slika 16: Upravljačka logika – povratna sprega

3.3.3. Softver predajnika (firmver)

Softver predajnika ima sljedeće zadatke:

- Inicijalizuje temperaturni senzor
- Mjeri temperaturu svake sekunde
- Ako je mjerenje uspješno, šalje vrijednost bežičnim protokolom prijemniku
- Štampa podatke na serijskom monitoru

Kod je pisan u razvojnom okruženju Arduino IDE, sačuvan kao .ino fajl i snimljen na predajnički mikrokontroler. Dat je u Prilogu 1 i Prilogu 2. Prilog 1 prikazuje kod predajnika, pisan za upotrebu ESP32 mikrokontrolera u kombinaciji sa DHT22 senzorom i ESP-NOW komunikacionim protokolom. Prilog 2 prikazuje kod predajnika, pisan za upotrebu ATtiny85 mikrokontrolera u kombinaciji sa DS18B20 senzorom i RF komunikacionim protokolom.

Slijedi analiza koda predajnika iz Priloga 1:

Uključuju se tri biblioteke: WiFi.h, esp_now.h i DHT.h. Prva biblioteka omogućava WiFi funkcionalnosti mikrokontrolera i zajedno sa drugom bibliotekom omogućava ESP-NOW bežičnu komunikaciju između predajnika i prijemnika. Treća biblioteka olakšava rad sa DHT senzorima (korišteni senzor temperature)

Inicijalizacija DHT senzora: Dodjeljuje mu se pin, definiše tip senzora (DHT22) i kreira

se instanca DHT objekta.

Definisanje MAC adrese prijemnika. MAC adresa jeste 8C:4F:00:AD:67:1C, što se u kodu upisuje na sljedeći način:

```
uint8_t receiverMAC[] = {0x8C, 0x4F, 0x00, 0xAD, 0x67, 0x1C};
```

Definisanje strukture za slanje podataka: Definiše se struktura *struct_message*, koja sadrži samo jedno polje – *float temperature*. Ta struktura se koristi za slanje temperature preko ESP-NOW, a instanca te strukture koja će se popunjavati i slati nosi naziv *dataToSend*.

Funkcija *setup()*: U njoj se vrši pokretanje serijske komunikacije, brzina 115200bps. Inicijalizuje se DHT senzor funkcijom iz njemu potrebne biblioteke. Zatim slijedi pauza od dvije sekunde, kako bi se sistem stabilizovao prije početka rada. Zatim se ESP postavlja u Station mod i poziva se funkcija *WiFi.disconnect()*, koja je dodatna mjera da se isključi WiFi ukoliko je ranije bio povezan. Zatim se vrši pokretanje ESP-NOW, koje, ukoliko ne uspije, ispisuje grešku i prekida dalje izvršavanje koda. Dalji korak je kreiranje objekta za informacije o drugom mikrokontroleru (*peer-u*). MAC adresa prijemnika se kopira u strukturu i *peer* se dodaje u ESP-NOW listu.

Funkcija *loop()*: U njoj se prvo vrši očitavanje temperature sa senzora. Ukoliko je očitavanje uspješno, očitana vrijednost se postavlja u strukturu i ti podaci se šalju ka prijemniku, kao niz bajtova u veličini strukture. Zatim se na serijskom monitoru štampa poslana temperatura, dok se, ukoliko očitavanje nije uspjelo, štampa poruka o grešci. Kako se *loop*-petlja stalno ponavlja (*vrta u krug*), na kraju je potrebno dodati pauzu od 5 sekundi prije sljedećeg mjerenja.

Slijedi analiza koda predajnika iz Priloga 2:

U početku se pozivaju biblioteke – *avr/io* za pristup direktno AVR registrima, *DallasTemperature* i *OneWire* za očitavanje temperature, *VirtualWire* za RF komunikaciju i *tinysnore* za nisku potrošnju energije u režimu spavanja.

Zatim se definišu pinovi, isključuje napajanje senzora na početku, pokreće se 1-Wire komunikacija i inicijalizuje se *VirtualWire* na vrlo niskom baudrate-u, što je odlično za pouzdanost i domet.

U *loop* petlji se formira buffer veličine jednog bajta – šalje samo cjelobrojnu temperaturu. Uključuje se senzor i LED signalizira da aktivnost. Čita se temperatura kao *float* i pretvara se u 8-bitni integer, cijeli broj u vrijednosti od 0 do 255. Ako je greška u komunikaciji, šalje vrijednost 127 kao oznaku disfunkcije. Zatim se šalje jedan bajt (temperatura) bežično. Na kraju se isključuje senzor i LED i time se postiže ušteda energije. ATtiny85 ide u *low-power* mod na 60 sekundi. Ova funkcija koristi *Watchdog Timer* za buđenje.

Važno je napomenuti da se programiranje ATtiny85 mikrokontrolera ne može izvršiti direktno, već posredstvom ISP programera. ISP programeri kao što su USBasp ili Arduino kao ISP su standardni način da bi se programirao čip. Pri izradi rada je korišten Arduino kao ISP. Da

bi Arduino ušao u ISP režim, na njega se mora snimiti kod ArduinoISP (već gotov kod, koji se nalazi u primjerima unutar Arduino IDE, File>Examples>11.ArduinoISP). ATtiny nema bootloader kao Arduino, pa iz tog razloga je neophodno ISP programiranje, kako bi se kod zapisao direktno u flash memoriju čipa. Kada se Arduino koristi kao ISP povezuje se sa ATtiny mikrokontrolerom na sljedeći način: pin 10 na RESET, 11 na PB0, 12 na PB1, 13 na PB2, 5V na VCC i masa na GND. Dok god se čip programira, Arduino mora biti u ISP modu, a kod se snima naredbom „Upload Using Programmer”.

3.3.4. Softver prijemnika (firmver)

Softver prijemnika ima sljedeće zadatke:

- Inicijalizuje funkcije za prijem podataka
- Na prijem temperature ažurira promjenjivu `receivedTemp`, kao i LCD prikaz
- U loop-petlji svake dvije sekunde poziva funkciju `releyControl()` koja uključuje/isključuje rele u zavisnosti od temperature
- Koristi FreeRTOS task za tastere (ciklus od 200ms)

Kod je pisan u razvojnom okruženju Arduino IDE, sačuvan kao .ino fajl i snimljen na prijemnik. Ima oblik dat u Prilogu 3 i Prilogu 4. Prilog 3 prikazuje kod prijemnika u kome se kao mikrokontroler koristi ESP32 i bežični protokol ESP-NOW, dok Prilog 4 prikazuje kod prijemnika u kome se kao mikrokontroler koristi Arduino UNO, bežični RF protokol i RF komunikacioni modul.

Slijedi analiza koda prijemnika iz Priloga 3:

Uključivanje biblioteka: Uključuju se biblioteke neophodne za ESP-NOW komunikaciju, kao kod predajnika. Dodatno se uključuju biblioteke *Wire.h* i *LiquidCrystal_I2C.h*, neophodne za rad LCD ekrana i laku manipulaciju njime. Zahvaljujući njima LCD ekran se inicijalizuje kao 0x27 16x2 ekran.

Definisanje pinova za tastere (4, 5, 16, 17, 18), LED lampicu (2), rele (23). Pinovi za LCD (SDA i SLC) se ne moraju definisati, jer je povezan na već preodređene pinove za njegov rad. Dodaje se muteks – mehanizam za sinhronizaciju pristupa LCD ekranu kada se koristi FreeRTOS, kako bi više taskova moglo koristiti LCD bez međusobnog konflikta u pristupu.

Unošenje parametara aplikacije: Parametri aplikacije su lista vinskih režima, kao i njihove referentne temperature. Definišu se promjenjive *receivedTemp* koja predstavlja posljednju primljenu vrijednost temperature (inicijalno se postavlja na nulu) i *delta* koja predstavlja toleranciju pri odlučivanju da li treba uključiti rashladni uređaj i iznosi 0.5.

Tajmeri i intervali: Tajmeri za vraćanje LCD prikaza u početni režim, nakon što korisnik promijeni postavke. Nakon pritiska MODE tastera interval za vraćanje na početak iznosi pet sekundi, dok za CHECK dugme iznosi dvije sekunde.

Struktura za ESP-NOW podatke: Jednostavna struktura za primanje temperature putem

ESP-NOW.

ESP-NOW Callback: Kada se primi podatak, provjerava se dužina i kopira se u promjenjivu `incomingData`, gdje se zapisuju dobijeni podaci. Ažurira se i promjenjiva `receivedTemp`, koja predstavlja dobijenu vrijednost temperature i prikazuje se na LCD (zaštićeno muteksom).

Funkcija `setup()`: U `setup`-u se prvo pokreće LCD, tasteri se postavljaju kao `INPUT_PULLUP`, rele se postavlja na `HIGH`, inicijalizuje se ESP-NOW i registruje callback funkcija. Zatim se kreira muteks za LCD i poziva funkcija `setParameters()` za početni režim. Na kraju se pokreće RTOS task za čitanje tastera.

Funkcija `loop()`: Kako se interakcija vrši preko RTOS task-a, u funkciji `loop` se samo odlučuje da li se uključuje rele.

RTOS task – `buttonReading()`: To je task za obradu ulaza tastera. Svakih 200ms provjerava stanje svih pet tastera, a svako od njih ima specifičnu funkciju.

Funkcija `releyControl()`: Uključuje rele (postavlja na `LOW`) ako je temperatura previsoka (iznad referentne plus tolerancija).

Funkcija `setParameters()`: Postavlja referentnu temperaturu prema izabranom vinu. Sadrži pet slučajeva (`case`).

Funkcija `makeStringToDisplay()`: Vezana je za LCD ekran. Formira string sa trenutnom i referentnom temperaturom. Računa broj praznina, kako bi se osiguralo estetski lijepo poravnanje prikaza na LCD ekranu. Dodatno se koristi i pomoćna funkcija `padToLength()`, koja dopunjuje string prazninama do dužine 16 karaktera (kako je riječ od 16x2 displeju) za ispravno poravnanje.

Slijedi analiza koda prijemnika iz Priloga 4:

Na početku se pozivaju biblioteke vezane za LCD ekran i bežičnu RF komunikaciju. Zatim se definišu pinovi i parametri, kao i vrste vina i predefinisane temperature.

U funkciji `setup()` se vrši inicijalizacija serijske komunikacije i LCD-a, kao i RF prijemnika. Postavljaju se početni parametri.

Funkcije vezane za čitanje stanja tastera, kontrolu releja, LCD, tajmere su iste kao u prethodnom prilogu. Ono što se razlikuje jeste neophodna RF komunikacija u `loop`-petlji. Prima se jedan bajt temperature sa senzora, 127 se koristi kao kod za grešku i, ako je vrijednost validna, ista se prikazuje na LCD-u.

Kodovi iz oba priloga omogućavaju funkcionalnost prototipa. Imaju 5 režima za različite vrste vina i korisniku je omogućen odabir. Imaju identičan displej i ispis, kao i rele za kontrolu hlađenja. Prilikom RF komunikacije (Prilog 4) podaci se primaju u vidu jednog bajta, što je

dovoljno precizno. ESP-NOW komunikacijom (Prilog 3) se primaju podaci u obliku *struct* sa *float* vrijednostima, što omogućuje veću preciznost. Korištenje zasebnog zadatka (FreeRTOS) za čitanje stanja tastera, koje mikrokontroler ESP32 podržava, pruža bolji odziv i stabilnost sistema.

Zaključak je da je verzija sa Arduino UNO mikrokontrolerom dovoljna i jednostavna, dok je ESP32 verzija superiornija i uz nevelike modifikacije pruža mogućnosti slanja podataka na server, mobilnu aplikaciju ili u cloud. Može se proširiti tako da podrži više senzora ili vrsta vina.

4. Eksperimentalni rezultati

4.1. Eksperimentalno okruženje

Kako bi se simulirao rad i efikasnost kontrolera, korišten je standardni jednodijelni kućni hladnjak proizvođača Candy. Kako bi se snimili dobijeni podaci, korišteni su programski jezici Python i Matlab.

Python je služio kao pomoćni program, kako bi se podaci o vrijednosti temperature i stanja rashladnog uređaja (uključen ili isključen) u datom momentu sačuvali u tekstualni .txt fajl.

Na početku, u Python-ov .py fajl se dodaju neophodne biblioteke i to – *serial* (omogućava komunikaciju sa serijskim portom, odnosno povezivanje sa USB-om) i *time* (koristi se za upravljanje vremenom, konkretno timestamp).

Zatim se vrši konfiguracija parametara. To su parametri koji određuju serijski port (na koji je povezan mikrokontroler tokom korišćenja programa), brzina serijske komunikacije (koja se mora poklapati sa brzinom serijske komunikacije mikrokontrolera, konkretno 115200), naziv fajla u koji se čuvaju podaci (tekstualni .txt fajl) i trajanje logovanja u sekundama (konkretno 10800s za tri sata).

Potom slijedi pokušaj otvaranja serijskom porta (ispisuje se greška i program se prekida ukoliko otvaranje nije uspjelo) i kratka pauza za reset kontrolera. Mikrokontroler ESP32 se često resetuje kada se uspostavi serijska veza, pa pauza slijedi prije nego što Python počne da čita.

Dalje se vrši početak logovanja gdje se bilježi trenutno vrijeme – za mjerenje trajanja logovanja i posljednja primljena temperatura – koja se koristi kada stigne ON/OFF poruka. Glavna petlja za čitanje i upis podataka otvara fajl za pisanje, a poziva se i petlja koja traje dok je proteklo vrijeme manje od željenog trajanja logovanja (3 sata).

Vrši se čitanje i obrada linije sa serijskog porta. Čita se linija po linija, konvertuju bajtovi u string i uklanjaju razmaci i novi redovi. Ako je linija prazna, preskače se. Nakon toga se vrši ispis na ekran, prikazuje se šta je primljeno na serijskom portu u realnom vremenu. Dodaje se i vremenska oznaka (timestamp), kako bi se u tekstualnom fajlu znalo kad se šta desilo.

Na kraju se provjerava da li linija sadrži temperaturu i stanje rashladnog uređaja. Ukoliko dođe do greške (npr. loša linija), samo se ispiše poruka i nastavlja se dalje. Obavezno se na kraju vrši zatvaranje porta da ne bi ostao zaključan i ispisuje poruka o završetku logovanja.

Krajnji rezultat u tekstualnom fajlu ima sljedeći oblik:

```
2025-06-06 16:31:41,9.7,OFF
2025-06-06 16:31:43,9.7,OFF
2025-06-06 16:31:45,9.7,OFF
2025-06-06 16:31:47,9.7,OFF
2025-06-06 16:31:49,9.7,OFF
.
.
.
2025-06-06 19:25:31,15.6,ON
```

Fajlovi pisani u Python-u se nalaze u Prilogu 5 (verzija sa mikrokontrolerom ESP32) i u Prilogu 6 (verzija sa mikrokontrolerom Arduino UNO).

Rezultat je tekstualni fajl eksperiment.txt.

Matlab na osnovu pomenutog tekstualnog fajla pravi grafikon promjene temperature u realnom vremenu.

Na početku se vrši učitavanje fajla. Definiše se naziv tekstualnog .txt fajla iz kojeg se čita temperatura. CSV fajl se čita kao tabela, definiše se struktura reda (datum+vrijeme, temperatura i stanje rashladnog uređaja) i na osnovu toga se vrši imenovanje kolona.

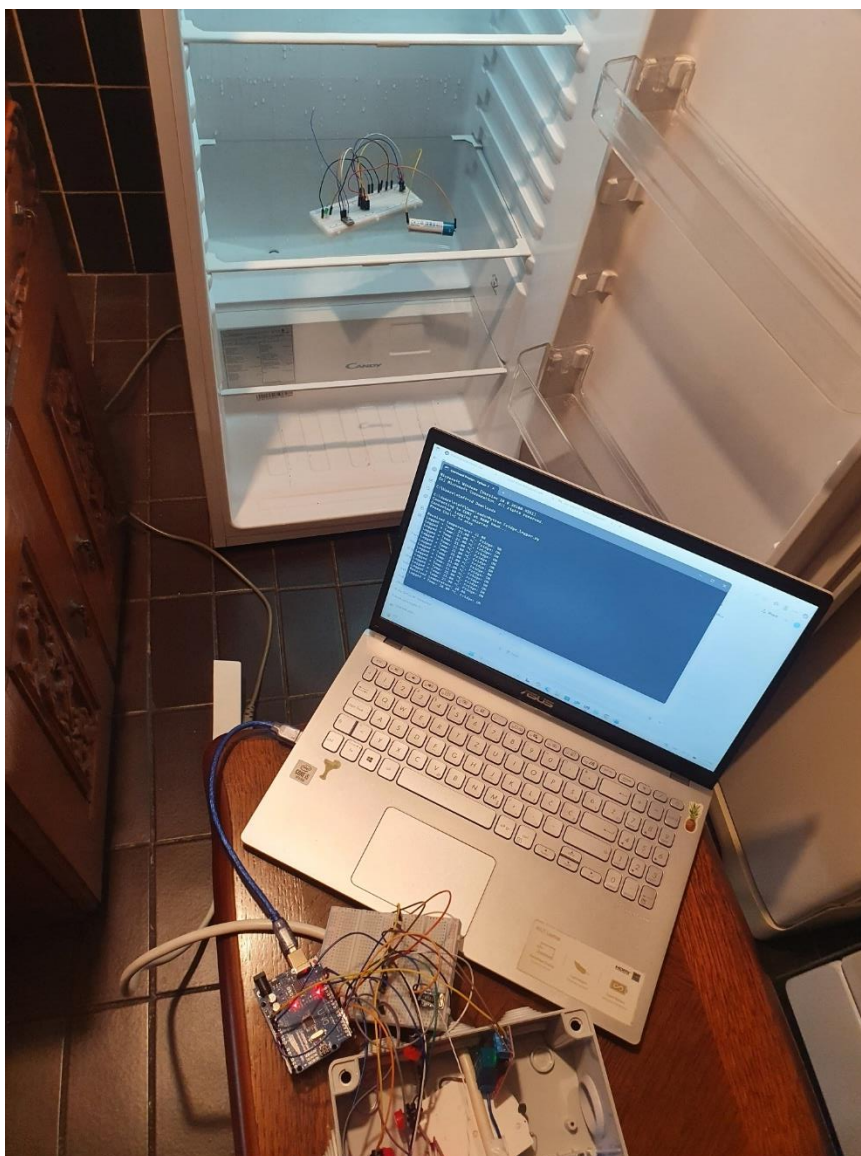
Zatim se vrši obrada stanja rashladnog uređaja. Prvo se uklanjaju razmaci iz stringova i vraća logički niz gdje piše „ON”. Rezultat je niz logičkih vrijednosti, 1 za uključen rashladni uređaj i 0 za isključen.

Izračunava se prosjek temperature kako bi kriva imala ljepši – glatki vizuelni izgled. Svaka vrijednost temperature je prosječna i zavisi od sopstvene vrijednosti i od vrijednosti dvije susjedne.

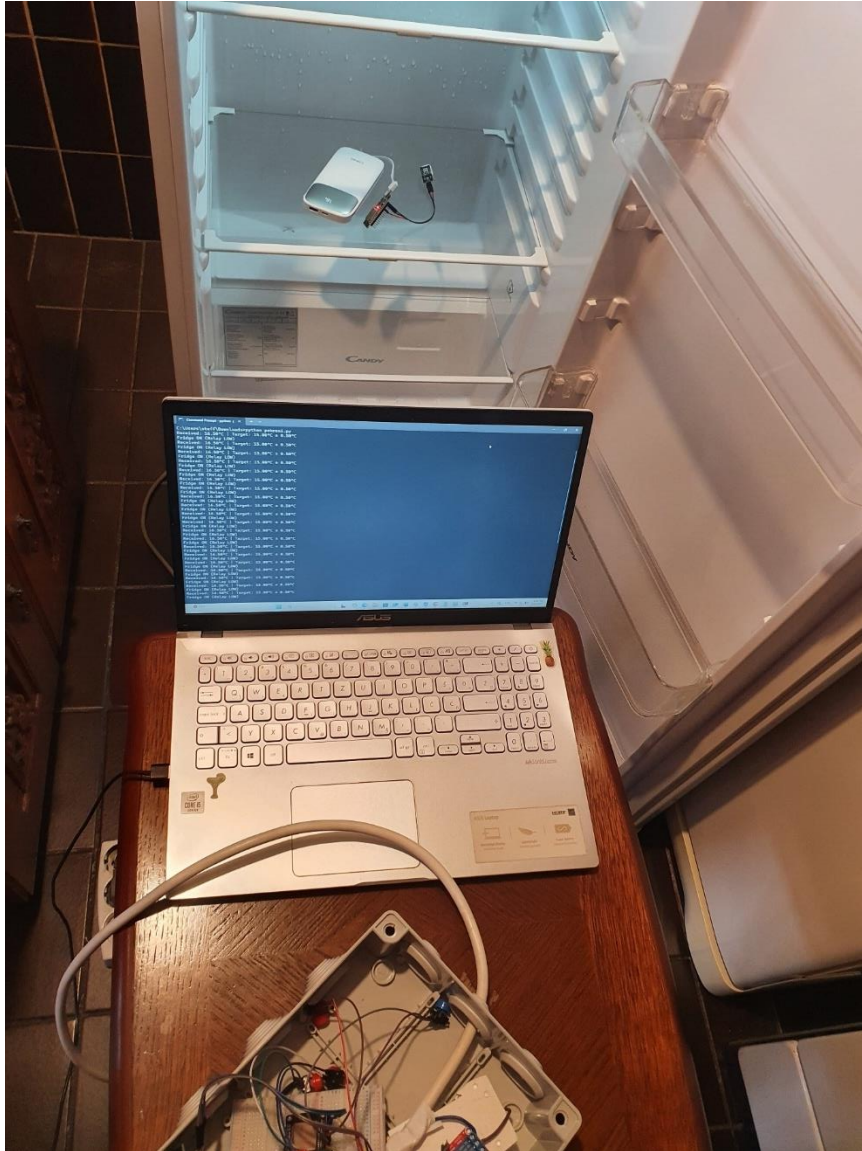
Grafik uvijek sadrži referentnu temperaturu u svom opsegu, a vrši se sijenčenje grafika u periodu kada je rashladni uređaj uključen blagom crvenom bojom, uz postotak providnosti od 50%. Kriva temperature je plave boje, crtana debljom neisprekidanom linijom, dok je horizontalna crna isprekidana linija upravo referentna linija.

Fajl pisan u Matlab-u je dat u Prilogu 7.

Slika 17 i Slika 18 predstavljaju fotografije eksperimentalnog okruženja. Slika 17 je nastala tokom eksperimenta sa osmobicnim RISK procesorom kao predajnikom, dok je Slika 18 nastala tokom eksperimenta sa ESP32 kao predajnikom.



Slika 17: Fotografija eksperimentalnog okruženja, 8-bitni RISK procesor kao predajnik



Slika18: Fotografija eksperimentalnog okruženja, ESP32 kao predajnik

Na fotografijama se uočava predajnik na bazi osmobitnog RISK procesora i prijemnik na bazi Atmega328P, kao i predajnik i prijemnik na bazi ESP32. Fotografije su nastale tokom snimanja eksperimenata. Budući da je za logovanje potrebno povezati mikrokontrolere sa kompjuterom, korisnički interfejs je nepotpun, poklopac nije postavljen i tasteri i ekran nijesu fiksirani u svojim ležištima. Uočava se da je kompresor rashladnog uređaja uključen u oba slučaja (temperatura je viša od referentne) i podaci se loguju putem Pythona.

4.2. Eksperimentalni parametri

Eksperimentalni parametri predstavljaju mjerljive veličine koje se prate, analiziraju i koriste za procjenu performansi razvijenog sistema. Ključni eksperimentalni parametri u kontekstu ovog rada su:

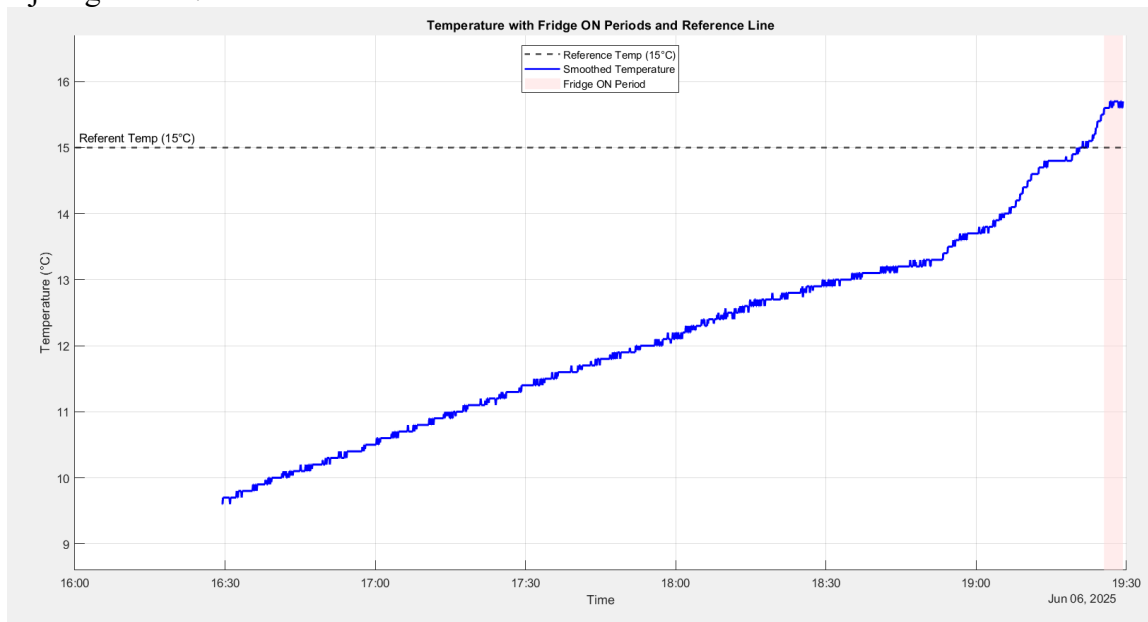
- Regulatorni odziv – način na koji sistem reaguje na odstupanje izmjerene temperature od zadate vrijednosti. Mjerna jedinica je vrijeme, amplituda je u stepenima Celzijusa. Pokazuje koliko brzo i efikasno sistem reaguje na promjene i vraća temperaturu u željeni opseg.
- Domet sistema – maksimalna udaljenost između bežičnog senzora i prijemnog modula unutar koje se prenose podaci bez gubitka signala. Mjerna jedinica je metar.
- Potrošnja sistema – odnosi se na količinu električne energije koju sistem troši tokom rada. Mjerna jedinica je miliamper.
- Preciznost očitavanja – zavisi prvenstveno od samog senzora, mjerna jedinica je stepen Celzijusa.

4.3. Rezultati preliminarnog testiranja

Regulacija temperature je vršena metodom uključi/isključi (on/off). U analiziranom sistemu, procesna varijabla jeste temperatura unutar rashladnog uređaja.

Protok toplote u rashladnom uređaju je tipičan toplotni proces. Temperatura, nakon uključivanja kompresora bez regulatora, bi opadala eksponencijalno do minimalne temperature. Nakon isključenja kompresora, temperatura bi rasla eksponencijalno do temperature okoline.

Radi provjere kvaliteta rada opisanog temperaturnog regulatora, vremenska promjena temperature rashladnog uređaja određena je eksperimentalno. Snimanje je trajalo tri sata. Podešena je referentna temperatura koja odgovara crvenom vinu. Slika 19 prikazuje odgovarajući grafikon.

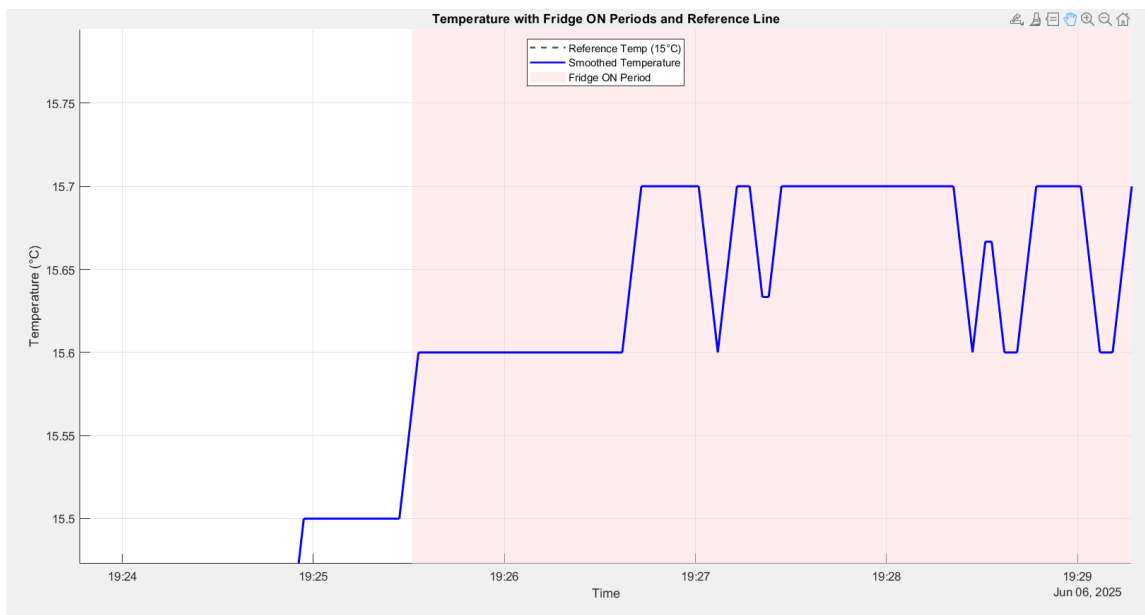


Slika 19: Grafikon promjene temperature tokom vremena

Kako je rashladni uređaj prethodno radio na temperaturi bliskoj 10°C, pri odabiru moda hlađenja crvenog vina (pri čemu je referentna temperatura hlađenja 15°C), uređaj je većinu vremena proveo sa isključenim kompresorom. Kako bi postigao referentnu temperaturu, trebalo mu je skoro tri sata.

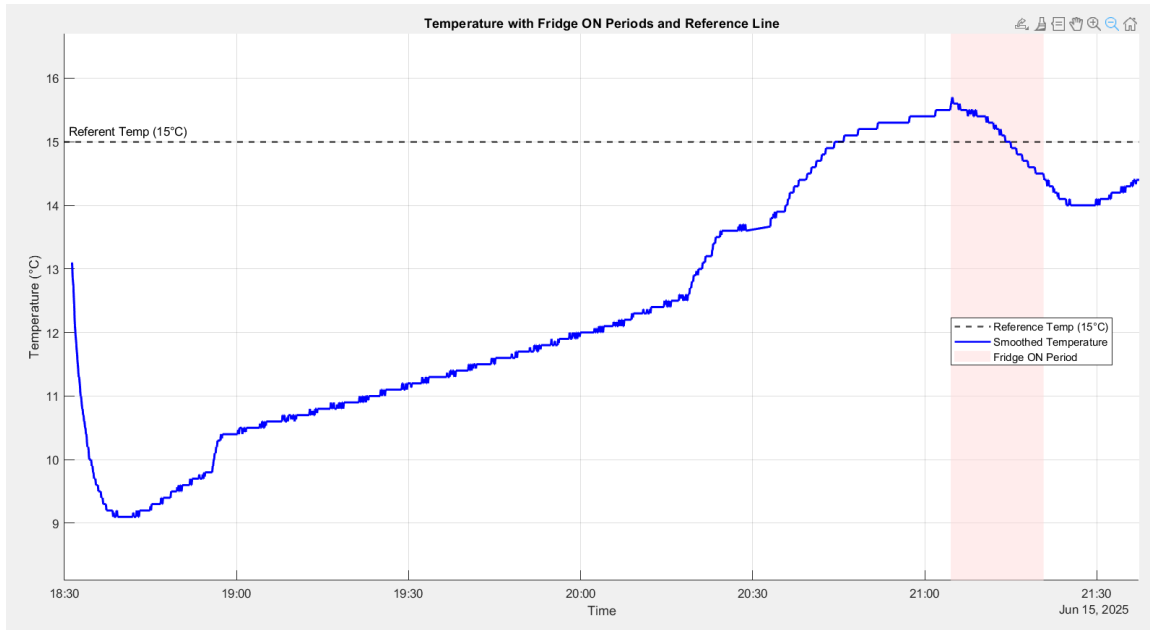
Tolerancija pri uključivanju rashladnog uređaja (delta) iznosi 0.5°C, što znači da se uključivanje vrši kada temperatura premaši vrijednost za 0.5°C veću od referentne. Kako je rezolucija temperaturnog senzora 0.1°C, u konkretnom eksperimentu uključivanje rashladnog uređaja se izvršilo na 15.6°C. Inertnost promjene temperature unutar hladnjaka i termostat koji je podešen na maksimalnu vrijednost, usloveli su da temperatura ni u jednom momentu ne pređe 15.7°C.

Preciznost regulacije se najbolje uočava na uvećanom grafikonu rada rashladnog uređaja (Slika 20).



Slika 20: Uvećani grafikon rada rashladnog uređaja

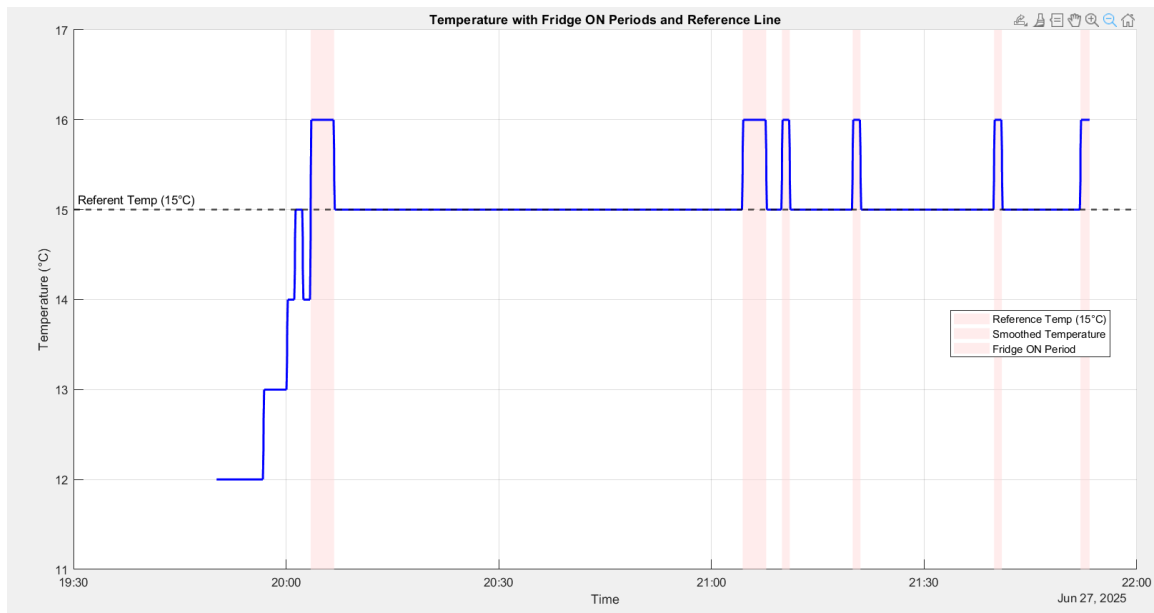
Slika 21 prikazuje eksperiment dužeg trajanja, s obzirom da je rashladnom uređaju koji je bio u normalnoj kućnoj upotrebi potreban period kako bi dostigao temperaturu pogodnu za čuvanje crvenog vina. Kontroler uspješno održava temperaturu tako što uključuje kompresor kada vrijednost iste pređe 15.5°C, a isključuje kada vrijednost padne ispod 14.5°C. Ni u kom momentu vrijednost temperature ne pada ispod 14°C. Temperatura je uspješno regulisana naizmjeničnim uključivanjem i isključivanjem rashladnog uređaja. Interval rada kompresora je iznosio približno 15 minuta.



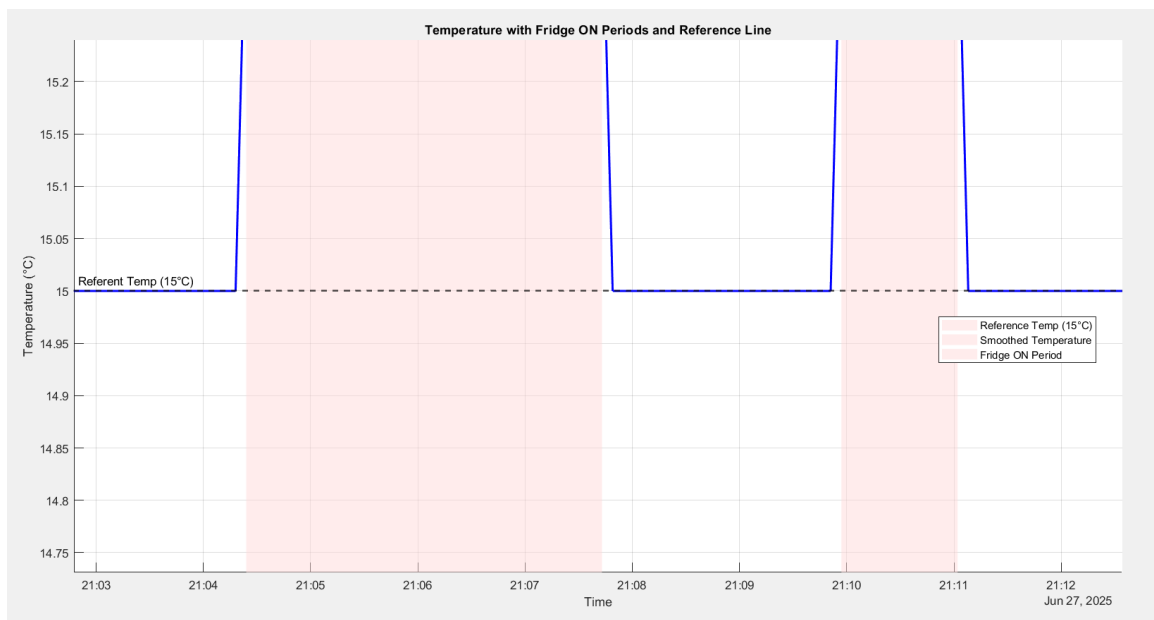
Slika 21: Eksperiment dužeg trajanja

Frekvencija odabiranja iznosi približno 0.33 Hz, odnosno slanje podataka se vrši svake tri sekunde. Navedeno, uz rezoluciju senzora, za posljedicu ima veoma brz odziv kontrolera i brzo uključivanje/isključivanje rashladnog uređaja. Preciznost regulacije je $\pm 1^\circ\text{C}$. Sama regulacija se tokom eksperimenta pokazala kao uspješna, sa temperaturom koja je konstantno u opsegu između 14°C i 15.7°C , što u potpunosti odgovara opsegu idealne temperature za čuvanje crvenog vina.

Treći eksperiment je izvršen sa značajnom izmjenom – rashladni uređaj se uključuje kada temperatura pređe 15.5°C , ali se i isključuje se kada temperatura padne ispod navedene vrijednosti. Razlog tome je eksponencijalni pad temperature tokom dugog rada kompresora. Naime, prilikom izvođenja prethodnog eksperimenta je uočeno da prilikom uključivanja kompresora, u rashladnom uređaju temperatura pređe gornju temperaturnu granicu za 0.1°C , dok se pri isključivanju temperatura smanji za 0.5°C ispod temperature granice. Na ovaj način je uži opseg promjene temperature i njena vrijednost je u najvećem dijelu vremena između 14.5°C i 15.5°C , odnosno unutar granica regulacije. Slika 22 prikazuje eksperiment bez granične vrijednosti, a Slika 23 daje njegov uvećani prikaz.



Slika 22: Eksperiment bez granične vrijednosti



Slika 23: Eksperiment bez granične vrijednosti – uvećani graf

Eksperiment je trajao dva sata. Uočava se da je rashladni uređaj bio uključen oko 10 minuta. Intervali uključenosti su trajanja od minut do tri minuta. Program je napisan tako da zaokružuje vrijednosti temperature na cijele brojeve. Fluktuacije u temperaturi su smanjenje u odnosu na prethodne eksperimente. Kompresor se uključuje češće, ali ukupno radi kraće. Kako interval slanja podataka sa senzora iznosi jedan minut, to je ujedno i minimalno vrijeme tokom kojeg kompresor može biti uključen ili isključen. Mane ovakve regulacije su češće uključivanje/isključivanje kompresora, rashladni uređaj radi u intervalima od minut do tri minuta. Dešava se da senzor uzastopno očitava vrijednosti bliske graničnoj – jednu malo (za

vrijednost rezolucije senzora) manju, drugu malo veću, što uslavljava uzastopne periode uključenosti/isključenosti koji traju po jedan minut. Mana regulacije u prvom i drugom eksperimentu je eksponencijalni pad temperature tokom rada kompresora. Iako se očitavanja sa senzora vrše mnogo češće – 20 puta u minutu, ne može se izbjeći da temperatura, kada dođe do granice isključivanja rashladnog uređaja, nastavi da eksponencijalno pada sve do vrijednosti koja je za 1 manja od optimalne.

Kada su u pitanju eksperimentalni parametri, regulacioni odziv direktno zavisi od načina na koji je sistem podešen. Kod bang-bang kontrole odziv je neposredan, ali zavisi od toga koliko često senzor vrši očitavanje. Vrijeme je moguće softverski podesiti bilo da se radi o ESP-NOW protokolu ili o RF protokolu (tokom eksperimenta podešavano je od minimalne dvije sekunde, pa do jednog minuta). Međutim, kada je riječ o komercijalnim BLE senzorima, imaju zatvoren firmware i nemaju mogućnost direktnog programiranja. Konkretno, za pomenuti Xiaomi Mi2 senzor regulacioni odziv iznosi od 5 do 8 minuta, zavisno od ciklusa oglašavanja. Tabela 1 prikazuje regulacioni odziv sistema.

Predajnik:	Komercijalni senzor	8-bitni RISK	ESP32
Regulacioni odziv:	5-8min	1 min	$\geq 2s$

Tabela 1: Regulacioni odziv sistema

Problemi sa dometom se nijesu javljali tokom eksperimentalnog rada. Sama udaljenost kontrolera fizički ne može biti toliko velika da bi se remetio domet predajnika. Za konkretni komercijalni senzor kao domet se navodi 5-15 metara, dok se kao domet ESP-NOW protokola navodi 20-50 metara. Domet RF veze predajnika i prijemnika (oba imaju antene) iznosi od 30 pa sve do 100 metara, u zavisnosti od uslova i prepreka. Tabela 2 prikazuje domet korištenih bežičnih protokola.

Bežični protokol:	BLE	RF	ESP-NOW
Domet:	5-15m	30-100m	20-50m

Tabela 2: Domet korištenih bežičnih protokola

Kako je predajnik baterijski napajan uređaj veoma važna stavka je njegova potrošnja i proračun power budget-a. Kada je u pitanju komercijalni senzor, proizvođač Xiaomi deklarise njegovu potrošnju od 10–20 μA . Kada je riječ o osmobitnom RISK mikrokontroleru, programiran je da naklon slanja podataka provede minut u režimu spavanja (snore). Pri radu u takvom režimu potrošnja predajnika iznosi oko 10 μA . ESP32 je, sa druge strane, mnogo veći potrošač. Njegova potrošnja u aktivnom režimu iznosi 50-70mA, dok potrošnja uz režim spavanja od jednog minuta (i uz korišćenje protokola ESP-NOW) iznosi oko 2.6 – 2.7mA. Potrošnja se može dodatno sniziti isključivanjem Wi-Fi modula prije svakog ulaska u režim spavanja (dodavanje esp_wifi_stop() funkcije). Međutim, navedeno može potrošnju sniziti samo za 100-200 μA , pa u konačnici razlika nije velika i potrošnja se kreće oko 2.5mA. Zaključak je da je ESP32 višestruko veći potrošač u odnosu na alternativne predajnike. Pri njegovom korištenju obavezno je baterijsko napajanje većeg kapaciteta. Tabela 3 prikazuje potrošnju navedenih predajnika.

Predajnik:	Komercijalni senzor	8-bitni RISK	ESP32
Potrošnja:	10–20 μ A	\approx 10 μ A	2.5-2.7mA

Tabela 3: Potrošnja predajnika

Preciznost komercijalnog senzora iznosi $\pm 0.3^{\circ}\text{C}$, dok je preciznost oba u prototipu korišćena senzora $\pm 0.5^{\circ}\text{C}$. Tabela 4 prikazuje preciznost korištenih senzora.

Senzor:	Komercijalni senzor	DS18B20	DHT22
Preciznost:	$\pm 0.3^{\circ}\text{C}$	$\pm 0.5^{\circ}\text{C}$	$\pm 0.5^{\circ}\text{C}$

Tabela 4: Preciznost korištenih senzora

Zaključuje se da je formirani prototip u potpunosti konkurentan komercijalnim rješenjima.

5. Zaključak i budući rad

Kao rezultat istraživanja je nastao prototip uređaja u formi produžnog kabla, koji, uz pomoć jednog bežičnog senzora, održava mikroklimu unutar bilo kog hladnjaka bang-bang kontrolom, uz napomenu da je termostat na hladnjaku podešen na maksimalnu vrijednost. Uređaj karakteriše veoma širok opseg primjene, od industrijskih hladnjaka do malih kućnih. Bio bi koristan za kancelariju svake ozbiljne osobe iz sfere biznisa, ali i malom poljoprivrednom proizvođaču. Osim toga, zbog svoje pristupačne cijene biće dostupan bilo kom ljubitelju vina.

Kada je riječ o načinima realizacije prototipa, po pitanju ekonomičnosti u prednosti je kontroler sa predajnikom na bazi osmobitnog RISK procesora, kao i sa predajnikom na bazi komercijalnog senzora. Idealni su za baterijske sisteme, održivi i ekonomični i predstavljaju odličan izbor za manje kolekcije vina (jedan kontroler – jedan rashladni uređaj). Sa druge strane, kontroler sa predajnikom na bazi ESP32 jeste veći potrošač, ali i predstavlja odličan temelj za nadogradnju i korištenje u ozbiljnijim sistemima. Podržava mnogo složeniju logiku, više senzora i periferija i odličan je za mrežnu komunikaciju (IoT). Takav sistem se može unaprijediti u smislu centralne jedinice za monitoring cjelokupnog vinskog podruma, tako što će automatizirati upravljanje sistemom za klimatizaciju, ventilaciju i grijanje. Prikupljene podatke sa više senzora može slati u cloud ili aplikaciju i omogućiti sisteme za upozorenje (alarm, e-mail, notifikacija) ako parametri izađu iz dozvoljenih granica.

Softversko rješenje je efikasno i fleksibilno za različite primjene. Softverska arhitektura sadrži slojeve: senzora (za očitavanje podataka o temperaturi), kontrole (za implemtaciju bang-bang kontrole i upravljanja uređajem), korisničkog interfejsa (za implementaciju parametara koje korisnik zadava putem taster, ali i za neophodan ispis na LCD ekranu) i komunikacije.

Izvršena je sistematizacija znanja i vještina u smislu pravljenja sličnih rješenja za ostale poljoprivredne ili prehrambene proizvode. Istraživanje je rezultiralo razvojem pametnog sistema za monitoring i kontrolu ključnih parametara koji su esencijalni za kvalitet i očuvanje poljoprivrednih proizvoda generalno, uz izmjene u softveru ili čak dodavanje drugih senzora.

Kao pogodnost se ističe ušteda finansijskih sredstava i energije. Zbog svoje cjenovne konkurentnosti i pouzdanosti, stari hladnjak uz upotrebu kontrolera postaje odlična alternative skupim rashladnim uređajima. Održivost se ogleda i u smanjenju elektronskog otpada i resursa potrebnih za proizvodnju novih uređaja. Poljoprivrednici ne moraju ulagati u nove, skupe uređaje, već mogu prilagoditi postojeće aparate.

Značajna je i upotreba savremenih tehnologija, kao što su CPS (Cyber Physical System), IoT (Internet of Things), AI (vještačka inteligencija). Rad doprinosi novim saznanjima u oblasti senzora i automatizacije u poljoprivredi.

Rad daje doprinos povećanju konkurentnosti našeg vinarstva, koje ima presudnu ulogu u dugoročnom razvoju domaće privrede, ali i njenom integrisanju i prepoznatljivosti na evropskom i svjetskom tržištu. Uređaj povoljno utiče na brzinu i kvalitet razvoja produktivnih malih i srednjih preduzeća koja doprinose ekonomskom rastu Crne Gore.

Zahvaljujući relativno jednostavnoj konstrukciji, pouzdanom radu i dovoljno preciznoj regulaciji temperature, moguće je preći sa datog prototipa regulatora na njegovu komercijalnu proizvodnju. Implementacijom regulatora temperature ovog tipa, stari frižideri i zamrzivači se mogu veoma lako transformirati u moderne i pametne hladnjake za vino. Sistem predstavlja isplativ način da se savremena tehnologija integriše u svakodnevni život, bez potrebe za potpunom zamjenom postojećih uređaja. Dakle, umjesto da se već korišteni rashladni uređaji odbace, njihova ponovna upotreba i pretvaranje u pametne uređaje predstavlja praktično, ekološki prihvatljivo i ekonomski isplativo rješenje, koje pozitivno doprinosi kako domaćinstvima, tako i planeti u cjelini.

Iako korišćenje starih hladnjaka u kombinaciji sa senzorima u poljoprivredi može donijeti značajnu uštedu resursa i smanjenje troškova, postoje određena ograničenja u tom smislu. Jedno od njih je smanjena energetska efikasnost starijih modela hladnjaka, što može dovesti do povećanja potrošnje energije i smanjenja ekonomičnosti cjelokupnog sistema. Takođe, upitna je pouzdanost i rok trajanja tog starog hladnjaka. Međutim, i pored toga je pametni sistem prikazan ovim radom je u velikoj prednosti, ukupno ekonomski i cjenovno gledano.

Metodološki, istraživanje u ovoj oblasti suočava se s izazovima vezanim za testiranje i implementaciju takvih sistema na širokoj skali. Potrebno je razviti standarde i protokole za testiranje senzora i hladnjaka, kako bi se osigurao validan uzorak za procjenu efikasnosti, efektivnosti i održivosti sistema.

Dalji pravci istraživanja trebali bi se fokusirati na unapređenje i integraciju pametnih sistema sa obnovljivim izvorima energije. Takva integracija bi smanjila negativan uticaj na životnu sredinu i povećala održivost.

Uz ove pravce važno je razmotriti i socio-ekonomske aspekte, prvenstveno konkurentnost u cijeni pametnog uređaja. To direktno utiče na troškove implementacije u malim poljoprivrednim gazdinstvima, ali i ostalim potencijalnim korisnicima. Jednostavnost, pouzdanost i konkurentna cijena bi obezbijedili širok pristup ovoj tehnologiji.

Uz minimalne izmjene mogli bi se pratiti i podešavati ostali parametri kao što su vlažnost, ali i uz dodavanje drugog senzora i osvijetljenost, kao još jedan parametar. Sistem bi se, uz nevelike izmjene, mogao koristiti i za mnoštvo drugih, u Crnoj Gori popularnih, poljoprivrednih proizvoda kao što su sir i maslinovo ulje.

U narednim fazama razvoja sistema, fokus će biti na unapređenju funkcionalnosti, proširenju oblasti primjene i povećanju stepena automatizacije. Planirano je uvođenje više bežičnih senzora koji bi omogućili precizno mjerenje i upravljanje mikroklimom unutar hladnjaka po zonama, kao i u više odvojenih uređaja. Time bi se obezbijedilo bolje mapiranje temperaturnih zona, što je od posebnog značaja kod industrijskog skladištenja vina i sličnih proizvoda.

Pored osnovnog parametra – temperature, sistem će biti proširen za praćenje i regulaciju drugih relevantnih faktora, kao što su vlažnost vazduha i osvijetljenost. Time se omogućava kontrola specifičnih uslova potrebnih za čuvanje osjetljivih proizvoda. Dodavanjem novih senzora i proširenjem softverske podrške, sistem postaje primjenjiv u širem spektru prehrambene

i poljoprivredne proizvodnje.

Dalji razvoj uključuje integraciju sa mobilnim aplikacijama koje omogućavaju korisnicima daljinski nadzor, promjenu postavki i primanje obavještenja o odstupanju od zadatih parametara. Takođe, biće implementirana mogućnost priključivanja na Internet putem Wi-Fi mreže, čime sistem postaje dio šire IoT infrastrukture i omogućava upravljanje putem cloud-a.

Jedan od ciljeva budućeg razvoja je i unapređenje energetske efikasnosti sistema kroz optimizaciju algoritma kontrole, kao i istraživanje mogućnosti integracije sa obnovljivim izvorima energije. Time bi se dodatno smanjio ekološki otisak i povećala održivost sistema.

Planiran je i razvoj naprednijih kontrolnih strategija, uključujući primjenu vještačke inteligencije i mašinskog učenja za predikciju temperaturnih oscilacija, optimizaciju potrošnje i detekciju kvarova u sistemu. Ovakav pristup omogućava dugoročno autonomno funkcionisanje sistema uz minimalnu ljudsku intervenciju.

Na kraju, cilj je da se razvije komercijalno isplativ, pouzdan i jednostavan uređaj koji može transformisati konvencionalne hladnjake u pametne uređaje, doprinoseći razvoju poljoprivrede i povećanju konkurentnosti domaćih proizvođača na tržištu.

6. Literatura

- [1] Otašević, M., Golubović, M., Raičević, S., & Stojanović, R. Smart Wine Refrigerator.
- [2] Agriopoulou, S., Stamatelopoulou, E., & Varzakas, T. (2020). Advances in occurrence, importance, and mycotoxin control strategies: Prevention and detoxification in foods. *Foods*, 9(2), 137.
- [3] Scrimgeour, N., Nordestgaard, S., Lloyd, N. D. R., & Wilkes, E. N. (2015). Exploring the effect of elevated storage temperature on wine composition. *Australian journal of grape and wine research*, 21, 713-722.
- [4] Leclercq-Perlat, M. N., Sicard, M., Trelea, I. C., Picque, D., & Corrieu, G. (2012). Temperature and relative humidity influence the microbial and physicochemical characteristics of Camembert-type cheese ripening. *Journal of dairy science*, 95(8), 4666-4682.
- [5] La Scalia, G., Settanni, L., Micale, R., & Enea, M. (2016). Predictive shelf life model based on RF technology for improving the management of food supply chain: A case study. *International Journal of RF Technologies*, 7(1), 31-42.
- [6] Wang, M., Gao, F., Wu, Q., Zhang, J., Xue, Y., Wan, H., & Wang, P. (2018). Real-time assessment of food freshness in refrigerators based on a miniaturized electronic nose. *Analytical methods*, 10(39), 4741-4749.
- [7] Boquete, L., Cambralla, R., Rodríguez-Ascariz, J. M., Miguel-Jiménez, J. M., Cantos-Frontela, J. J., & Dongil, J. (2010). Portable system for temperature monitoring in all phases of wine production. *ISA transactions*, 49(3), 270-276.
- [8] <https://morewinemaking.com/category/temperature-control-1.html>
- [9] <https://tempcube.io/pages/wine-cellar-temperature-monitoring7>
- [10] https://www.arsystem.pl/carel/dokumentacje2003/02%20GB/01%20Broszury%20informacyjnej%20+3%2002235211_GB.pdf
- [11] Wang, Y. (2021). Multi-sensor temperature and humidity control system of wine cellar based on cooperative control of intelligent vehicle and UAV. *International Journal of Computer Applications in Technology*, 67(2-3), 99-110.
- [12] <https://ruuvi.com/guide-to-wine-storage-humidity-and-temperature-monitoring/>
- [13] Sainz, B., Antolín, J., López-Coronado, M., & de Castro, C. (2013). A novel low-cost sensor prototype for monitoring temperature during wine fermentation in tanks. *Sensors*, 13(3), 2848-2861.
- [14] [.https://sensorist.com/wine](https://sensorist.com/wine)
- [15] <https://thewinecaverns.co.uk/blogs/news/expert-guide-to-perfecting-wine-storage-at-your-home>
- [16] <https://www.elitecheu.com/blogs/elitecheu/wireless-temperature-monitoring-system-for-wine-storage>
- [17] <https://dataloggerinc.com/resource-article/wine-temperature-verification/>
- [18] Echave, J., Barral, M., Fraga-Corral, M., Prieto, M. A., & Simal-Gandara, J. (2021). Bottle aging and storage of wines: A review. *Molecules*, 26(3), 713.
- [19] <https://www.tehnomax.me/bijela-tehnika/rashladne-i-vinske-vitrine>
- [20] <https://okmarts.com/news/refrigerator-ptc-relay-wiring-method.html>
- [21] <https://lastminuteengineers.com/esp32-ble-tutorial/>
- [22] <https://www.nutsvolts.com/magazine/article/a-real-time-operating-system-for-the-arduino>

- [23] Alberto Leva, Luigi Piroddi, Massimiliano Di Felice, Alessandro Boer, Raffaele Paganini, “Adaptive relay-based control of household freezers with on–off actuators”, Politecnico di Milano, Dipartimento di Elettronica e Informazione, Via Ponzio 34/5, 20133 Milano, Italy, DOI: 10.1016/j.conengprac.2009.09.008
- [24] Antoni Ryniecki, Jolanta Wawrzyniak, Agnieszka Anna Pilarska, “Basics of Process Control: the On-Off Control System”, November 2015, doi: 10.15199/65.2015.11.6
- [25] <https://www.grandviewresearch.com/industry-analysis/wine-cooler-market>
- [26] https://a.aliexpress.com/_EjQw92G
- [27] <https://us.asko.com/products/kitchen/refrigerators-freezers/wine-cooler/VITRINE-VCS5197TPG-WCN311942G-ASK/p/000000000000740495>
- [28] <https://www.eurocave.com/en/>
- [29] <https://www.decanter.com/learn/advice/why-are-vibrations-wine-harmful-394264/>
- [30] <https://www.multicom.me/bijela-i-kuhinjska-tehnika/vinska-vitrina/svi.html>
- [31] ananas.me/kategorije/bijela-tehnika/rashladne-i-vinske-vitrine/vinske-vitrine

7. Prilozi

7.1. Prilog 1

```
void setup() {
  Serial.begin(115200);
  dht.begin();
  delay(2000);

  WiFi.mode(WIFI_STA);
  WiFi.disconnect();

  if (esp_now_init() != ESP_OK) {
    Serial.println("ESP-NOW init failed!");
    return;
  }

  esp_now_peer_info_t peerInfo = {};
  memcpy(peerInfo.peer_addr, receiverMAC, 6);
  peerInfo.channel = 0;
  peerInfo.encrypt = false;
  esp_now_add_peer(&peerInfo);
}

void loop() {
  float temp = dht.readTemperature();
  if (!isnan(temp)) {
    dataToSend.temperature = temp;
    esp_now_send(receiverMAC, (uint8_t *)&dataToSend, sizeof(dataToSend));
    Serial.printf("Sent temperature: %.1f °C\n", temp);
  } else {
    Serial.println("Failed to read from DHT22");
  }
  delay(5000);
}
```

7.2. Prilog 2

```
#include <avr/io.h>
#include <DallasTemperature.h>
#include <OneWire.h>
#include <VirtualWire.h>
#include <tinysnore.h>
```

```

#define LED_PIN PB0
#define TRANSMIT_PIN PB1
#define POWER_PIN PB3
#define ONE_WIRE_BUS PB4

// Setup a oneWire instance to communicate with any OneWire devices
OneWire oneWire(ONE_WIRE_BUS);
// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire);

void setup()
{
  pinMode(ONE_WIRE_BUS, INPUT);
  pinMode(LED_PIN, OUTPUT);
  pinMode(POWER_PIN, OUTPUT);
  digitalWrite(POWER_PIN, LOW);

  sensors.begin(); // Start up the library
  // Initialise the IO and ISR
  vw_set_tx_pin(TRANSMIT_PIN);
  vw_setup(300); // Bits per sec 1200
}

void loop()
{
  char a[1] = {'a'};
  digitalWrite(POWER_PIN, HIGH);
  digitalWrite(LED_PIN, HIGH);
  sensors.requestTemperatures(); // Send the command to get temperature readings
  float tempC = sensors.getTempCByIndex(0);

  if (tempC != DEVICE_DISCONNECTED_C) // Check if reading was successful
    a[0] = (uint8_t)tempC;
  else
    a[0] = 127;

  vw_send((uint8_t *)a, 1);
  vw_wait_tx(); // Wait until the whole message is gone

  digitalWrite(POWER_PIN, LOW);
  //delay(10000); // oko sekunde

  digitalWrite(LED_PIN, LOW);
  snore(60000); // 1 minut
}

```

7.3 Prilog 3

```
#include <WiFi.h>
#include <esp_now.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// LCD setup
LiquidCrystal_I2C lcd(0x27, 16, 2);

// Button & I/O Pins
const int temperaturePlus = 16;
const int temperatureMinus = 4;
const int vineMode = 17;
const int checkButton = 5;
const int resetButton = 18;
const int signalLed = 2;
const int relay = 23;
SemaphoreHandle_t lcdMutex;

// Wine modes
String vine[] = {"Red", "White", "Rose", "Sparkling", "Flavored"};
String currentVine;
int indexOfVine = 0;
int mode = 0;
float refTemperature = 15.0;
float receivedTemp = 0.0;
float delta = 0.5;

unsigned long modeTimer = 0;
unsigned long checkTimer = 0;
const unsigned long modeInterval = 5000;
const unsigned long checkInterval = 2000;

TaskHandle_t Task;

// ESP-NOW message struct
typedef struct struct_message {
    float temperature;
} struct_message;

struct_message incomingData;
```

```

// ESP-NOW receive callback
void onDataRecv(const esp_now_recv_info_t *info, const uint8_t *data, int len) {
  if (len == sizeof(struct_message)) {
    memcpy(&incomingData, data, sizeof(struct_message));
    receivedTemp = incomingData.temperature;
    Serial.printf("Received temperature: %.1f°C\n", receivedTemp);

    // Display updated temperature on LCD (safely with mutex)
    if (xSemaphoreTake(lcdMutex, pdMS_TO_TICKS(100))) {
      LCD_display(mode);
      xSemaphoreGive(lcdMutex);
    }
  } else {
    Serial.println("Unexpected data size!");
  }
}

void setup() {
  Serial.begin(115200);
  Wire.begin();
  lcd.begin();
  lcd.backlight();
  lcd.clear();
  lcdMutex = xSemaphoreCreateMutex();

  // Buttons as INPUT_PULLUP (important!)
  pinMode(temperaturePlus, INPUT_PULLUP);
  pinMode(temperatureMinus, INPUT_PULLUP);
  pinMode(vineMode, INPUT_PULLUP);
  pinMode(checkButton, INPUT_PULLUP);
  pinMode(resetButton, INPUT_PULLUP);

  pinMode(signalLed, OUTPUT);
  pinMode(relay, OUTPUT);
  digitalWrite(relay, HIGH); // Relay off by default

  WiFi.mode(WIFI_STA);
  if (esp_now_init() != ESP_OK) {
    Serial.println("ESP-NOW init failed!");
    while (true) delay(1000);
  }
  esp_now_register_recv_cb(onDataRecv);

  setParameters(indexOfVine);
  LCD_display(mode);
}

```

```

// Button read task
xTaskCreatePinnedToCore(buttonReading, "ButtonTask", 4096, NULL, 1, &Task, 0);
}

void loop() {

    relayControl(receivedTemp, refTemperature, delta);
    //LCD_display(mode);
    delay(2000);

}

void buttonReading(void *parameter) {
    while (true) {
        vTaskDelay(pdMS_TO_TICKS(200)); // Debounce timing

        if (!digitalRead(temperaturePlus)) {
            Serial.println("Button + Pressed");
            refTemperature += 0.1;
            mode = 1;
            modeTimer = millis();
            LCD_display(mode);
            while (!digitalRead(temperaturePlus)) vTaskDelay(pdMS_TO_TICKS(50));
        }

        if (!digitalRead(temperatureMinus)) {
            Serial.println("Button - Pressed");
            refTemperature -= 0.1;
            mode = 1;
            modeTimer = millis();
            LCD_display(mode);
            while (!digitalRead(temperatureMinus)) vTaskDelay(pdMS_TO_TICKS(50));
        }

        if (!digitalRead(vineMode)) {
            Serial.println("Button MODE Pressed");
            indexOfVine = (indexOfVine + 1) % 5;
            setParameters(indexOfVine);
            mode = 1;
            modeTimer = millis();
            LCD_display(mode);
            while (!digitalRead(vineMode)) vTaskDelay(pdMS_TO_TICKS(50));
        }

        if (!digitalRead(checkButton)) {
            Serial.println("Button CHECK Pressed");
        }
    }
}

```

```

    mode = 2;
    checkTimer = millis();
    LCD_display(mode);
    while (!digitalRead(checkButton)) vTaskDelay(pdMS_TO_TICKS(50));
}

if (!digitalRead(resetButton)) {
    Serial.println("Button RESET Pressed");
    indexOfVine = 0;
    setParameters(indexOfVine);
    mode = 0;
    LCD_display(mode);
    while (!digitalRead(resetButton)) vTaskDelay(pdMS_TO_TICKS(50));
}

if (millis() - modeTimer > modeInterval && mode == 1) {
    mode = 0;
    LCD_display(mode);
}

if (millis() - checkTimer > checkInterval && mode == 2) {
    mode = 0;
    LCD_display(mode);
}
}
}

void relayControl(float temperature, float refTemp, float delta) {
    Serial.printf("Received: %.2f°C | Target: %.2f°C ± %.2f°C\n", temperature, refTemp,
delta);

    if (temperature > refTemp + delta) {
        digitalWrite(relay, LOW); // Turn ON fridge (active LOW)
        digitalWrite(signalLed, HIGH);
        Serial.println("Fridge ON (Relay LOW)");
    }
    else if (temperature < refTemp - delta) {
        digitalWrite(relay, HIGH); // Turn OFF fridge
        digitalWrite(signalLed, LOW);
        Serial.println("Fridge OFF (Relay HIGH)");
    }
    else {
        Serial.println("Temperature within acceptable range, no change.");
    }
}
}

```

```

void setParameters(int index) {
    currentVine = vine[index];
    switch (index) {
        case 0: refTemperature = 15; break;
        case 1: refTemperature = 10; break;
        case 2: refTemperature = 8; break;
        case 3: refTemperature = 5; break;
        case 4: refTemperature = 6; break;
    }
    LCD_display(mode);
}

String makeStringToDisplay(String indicator) {
    String result = "";
    if (indicator == "Temperature") {
        result = "T:" + String(receivedTemp, 1) + " Ref:" + String(refTemperature, 1);
    }
    int pad = 16 - result.length();
    String emptySpaces = "";
    for (int i = 0; i < pad; i++) emptySpaces += " ";
    int mid = result.indexOf("R");
    if (mid == -1) mid = 0;
    return result.substring(0, mid) + emptySpaces + result.substring(mid);
}

String padToLength(String input, int length) {
    while (input.length() < length) {
        input += " ";
    }
    return input;
}

void LCD_display(int mode) {
    lcd.clear();
    String line1 = "", line2 = "";

    switch (mode) {
        case 0:
            line1 = makeStringToDisplay("Temperature");
            line2 = "Vine: " + currentVine;
            break;

        case 1:

```

```

    case 2:
        line1 = "Vine: " + currentVine;
        line2 = "RefTemp: " + String(refTemperature, 1);
        break;
    }

    // Pad both lines to 16 characters
    line1 = padToLength(line1, 16);
    line2 = padToLength(line2, 16);

    lcd.setCursor(0, 0);
    lcd.print(padToLength(line1, 16));

    lcd.setCursor(0, 1);
    lcd.print(padToLength(line2, 16));
}

```

7.4. Prilog 4

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <VirtualWire.h>

// LCD setup (I2C address may differ)
LiquidCrystal_I2C lcd(0x27, 16, 2);

// Pin definitions
const int rfRxBPin = 8;
const int temperaturePlus = 2;
const int temperatureMinus = 3;
const int vineMode = 4;
const int checkButton = 5;
const int resetButton = 6;
const int signalLed = 7;
const int relay = 9;

// Wine modes
String vine[] = {"Red", "White", "Rose", "Sparkling", "Flavored"};
String currentVine;
int indexOfVine = 0;
int mode = 0;
float refTemperature = 15.0;
float receivedTemp = 0.0;

```

```

float delta = 0.5;

unsigned long modeTimer = 0;
unsigned long checkTimer = 0;
const unsigned long modeInterval = 5000;
const unsigned long checkInterval = 2000;

void setup() {
  Serial.begin(9600);
  lcd.begin();
  lcd.backlight();
  lcd.clear();

  pinMode(temperaturePlus, INPUT_PULLUP);
  pinMode(temperatureMinus, INPUT_PULLUP);
  pinMode(vineMode, INPUT_PULLUP);
  pinMode(checkButton, INPUT_PULLUP);
  pinMode(resetButton, INPUT_PULLUP);

  pinMode(signalLed, OUTPUT);
  pinMode(relay, OUTPUT);
  digitalWrite(relay, HIGH); // Relay off

  // RF Receiver init
  vw_set_rx_pin(rfRxPin);
  vw_setup(300); // Match TX baud rate
  vw_rx_start();

  setParameters(indexOfVine);
  LCD_display(mode);
}

void loop() {
  checkButtons();

  uint8_t buf[1];
  uint8_t buflen = sizeof(buf);
  if (vw_get_message(buf, &buflen)) {
    if (buf[0] != 127) {
      receivedTemp = (float)buf[0];
      Serial.print("Received temperature: ");
      Serial.println(receivedTemp);
      LCD_display(mode);
    } else {
      Serial.println("Invalid temperature received.");
    }
  }
}

```

```

}

releyControl(receivedTemp, refTemperature, delta);

if (millis() - modeTimer > modeInterval && mode == 1) {
    mode = 0;
    LCD_display(mode);
}

if (millis() - checkTimer > checkInterval && mode == 2) {
    mode = 0;
    LCD_display(mode);
}

delay(200); // Adjust as needed
}

void checkButtons() {
    if (!digitalRead(temperaturePlus)) {
        refTemperature += 0.1;
        mode = 1;
        modeTimer = millis();
        LCD_display(mode);
        waitForRelease(temperaturePlus);
    }

    if (!digitalRead(temperatureMinus)) {
        refTemperature -= 0.1;
        mode = 1;
        modeTimer = millis();
        LCD_display(mode);
        waitForRelease(temperatureMinus);
    }

    if (!digitalRead(vineMode)) {
        indexOfVine = (indexOfVine + 1) % 5;
        setParameters(indexOfVine);
        mode = 1;
        modeTimer = millis();
        LCD_display(mode);
        waitForRelease(vineMode);
    }

    if (!digitalRead(checkButton)) {
        mode = 2;
        checkTimer = millis();

```

```

    LCD_display(mode);
    waitForRelease(checkButton);
}

if (!digitalRead(resetButton)) {
    indexOfVine = 0;
    setParameters(indexOfVine);
    mode = 0;
    LCD_display(mode);
    waitForRelease(resetButton);
}
}

void waitForRelease(int pin) {
    while (!digitalRead(pin)) delay(50);
}

void relayControl(float temperature, float refTemp, float delta) {
    if (temperature > refTemp + delta) {
        digitalWrite(relay, LOW); // Turn ON fridge (active LOW)
        digitalWrite(signalLed, HIGH);
    }
    else if (temperature < refTemp - delta) {
        digitalWrite(relay, HIGH); // Turn OFF fridge
        digitalWrite(signalLed, LOW);
    }
}

void setParameters(int index) {
    currentVine = vine[index];
    switch (index) {
        case 0: refTemperature = 15; break;
        case 1: refTemperature = 10; break;
        case 2: refTemperature = 8; break;
        case 3: refTemperature = 5; break;
        case 4: refTemperature = 6; break;
    }
    LCD_display(mode);
}

String makeStringToDisplay(String indicator) {
    String result = "";
    if (indicator == "Temperature") {
        result = "T:" + String(receivedTemp, 1) + " Ref:" + String(refTemperature, 1);
    }
    int pad = 16 - result.length();
}

```

```

String emptySpaces = "";
for (int i = 0; i < pad; i++) emptySpaces += " ";
int mid = result.indexOf("R");
if (mid == -1) mid = 0;
return result.substring(0, mid) + emptySpaces + result.substring(mid);
}

String padToLength(String input, int length) {
    while (input.length() < length) {
        input += " ";
    }
    return input;
}

void LCD_display(int mode) {
    lcd.clear();
    String line1 = "", line2 = "";

    switch (mode) {
        case 0:
            line1 = makeStringToDisplay("Temperature");
            line2 = "Vine: " + currentVine;
            break;

        case 1:
        case 2:
            line1 = "Vine: " + currentVine;
            line2 = "RefTemp: " + String(refTemperature, 1);
            break;
    }

    line1 = padToLength(line1, 16);
    line2 = padToLength(line2, 16);

    lcd.setCursor(0, 0);
    lcd.print(padToLength(line1, 16));

    lcd.setCursor(0, 1);
    lcd.print(padToLength(line2, 16));
}

```

7.5 Prilog 5

```

import serial
import time

```

```

# === CONFIGURATION ===
serial_port = 'COM3'      # Replace with your port
baud_rate = 115200
output_file = 'eksperiment.txt'
log_duration = 10800      # 30 minutes

# === SETUP ===
try:
    ser = serial.Serial(serial_port, baud_rate, timeout=1)
except Exception as e:
    print(f'Could not open serial port {serial_port}: {e}')
    exit()

time.sleep(2) # Wait for ESP32 to reset

print(f'Logging started. Saving to {output_file} for {log_duration} seconds...')
start_time = time.time()
last_temp = None

with open(output_file, 'w') as f:
    while time.time() - start_time < log_duration:
        try:
            line = ser.readline().decode('utf-8').strip()
            if not line:
                continue

            print(line) # Show serial output

            timestamp = time.strftime('%Y-%m-%d %H:%M:%S')

            if line.startswith("Received temperature:"):
                temp_str = line.split(':')[1].replace("°C", "").strip()
                last_temp = float(temp_str)

            elif "Fridge" in line and last_temp is not None:
                state = "ON" if "ON" in line else "OFF"
                f.write(f"{timestamp},{last_temp},{state}\n")
                f.flush()

        except Exception as e:
            print(f'Error: {e}')
            continue # Don't exit on error

ser.close()
print("Logging complete.")

```

7.6 Prilog 6

```
import serial
import time
from datetime import datetime

# --- Configuration ---
SERIAL_PORT = 'COM3'    # Change to your port (e.g., '/dev/ttyUSB0' on
Linux/Mac)
BAUD_RATE = 9600
LOG_FILE = 'exp.txt'

# --- Thresholds ---
TEMP_ON_THRESHOLD = 15.5 # Above this = fridge turns ON
TEMP_OFF_THRESHOLD = 14.5 # Below this = fridge turns OFF
MAX_LOG_INTERVAL = 5     # seconds to force logging if no new data

# --- Parse temperature from Arduino line ---
def parse_temperature(line):
    if "Received temperature:" in line:
        try:
            temp_str = line.strip().split(":")[-1]
            return float(temp_str)
        except ValueError:
            return None
    return None

def log_entry(log_file, temp, state):
    timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
    entry = f"{timestamp},{temp:.2f},{state}\n"
    log_file.write(entry)
    log_file.flush()
    print(f"Logged → Temp: {temp:.2f} °C, Fridge: {state}")

def main():
    print(f"Connecting to {SERIAL_PORT} at {BAUD_RATE} baud...")
    try:
        ser = serial.Serial(SERIAL_PORT, BAUD_RATE, timeout=1)
        time.sleep(2)
        print("Connected. Logging started.\nPress Ctrl+C to stop.\n")
    except Exception as e:
        print(f"Failed to open serial port: {e}")
    return
```

```

fridge_state = "OFF"
last_temp = None
last_log_time = time.time()

with open(LOG_FILE, 'a') as log_file:
    while True:
        try:
            line = ser.readline().decode('utf-8', errors='ignore').strip()

            now = time.time()
            updated = False

            if line:
                print(line)
                temp = parse_temperature(line)

                if temp is not None:
                    last_temp = temp
                    updated = True

                # Update fridge state
                if temp > TEMP_ON_THRESHOLD:
                    fridge_state = "ON"
                elif temp < TEMP_OFF_THRESHOLD:
                    fridge_state = "OFF"

            if updated or (last_temp is not None and now - last_log_time >
MAX_LOG_INTERVAL):
                log_entry(log_file, last_temp, fridge_state)
                last_log_time = now

        except KeyboardInterrupt:
            print("\nLogging stopped by user.")
            break
        except Exception as e:
            print(f"Error: {e}")
            time.sleep(1)

ser.close()

if __name__ == '__main__':
    main()

```

7.7. Prilog 7

```
clear; clc; close all;

% Load file
filename = 'eksperiment.txt';
data = readtable(filename, 'Delimiter', ',', ...
    'Format', '% {yyyy-MM-dd HH:mm:ss}D %f %s', ...
    'ReadVariableNames', false);

% Rename columns
data.Properties.VariableNames = {'Timestamp', 'Temperature', 'FridgeState'};

% Convert fridge state to logical
fridgeState = strcmpi(strtrim(data.FridgeState), 'ON');

% Smooth temperature
tempSmooth = movmean(data.Temperature, 3);

% Plot
figure;
hold on;

% Define y-axis range
yMin = min(tempSmooth) - 1;
yMax = max([tempSmooth; 15]) + 1;

% Shade fridge ON regions
fridgeStatePadded = [false; fridgeState(:); false];
edges = find(diff(fridgeStatePadded) ~= 0);

for i = 1:2:length(edges) - 1
    t1 = data.Timestamp(edges(i));
    t2 = data.Timestamp(edges(i+1)-1); % ensure valid index

    area([t1 t2], [yMax yMax], yMin, ...
        'FaceColor', [1 0.85 0.85], 'EdgeColor', 'none', 'FaceAlpha', 0.5);
end

% Plot smoothed temperature
plot(data.Timestamp, tempSmooth, '-b', 'LineWidth', 1.5);

% Add reference temperature line at 15 °C
yline(15, '--k', 'Referent Temp (15°C)', 'LineWidth', 1.5, ...
    'LabelHorizontalAlignment', 'left');
```

```
% Final plot formatting
ylabel('Temperature (°C)');
xlabel('Time');
ylim([yMin yMax]);
title('Temperature with Fridge ON Periods and Reference Line');
legend('Fridge ON Period', 'Smoothed Temperature', 'Reference Temp (15°C)', 'Location',
'best');
grid on;
```

Izjava o istovjetnosti štampane i elektronske verzije master rada

Ime i prezime autora: Stefan Raičević

Broj indeksa/upisa: 7/23

Studijski program: Automatika i industrijska elektrotehnika

Naslov rada: Prilagođenje kućnih uređaja za hlađenje potrebama skladištenja vina

Mentor: Prof. dr Radovan Stojanović

Potpisani/a: Stefan Raičević

Izjavljujem da je štampana verzija mog master rada istovjetna elektronskoj verziji koju sam predao/la za objavljivanje u Digitalni arhiv Univerziteta Crne Gore.

Istovremeno izjavljujem da dozvoljavam objavljivanje mojih ličnih podataka u vezi sa dobijanjem akademskog naziva master nauka, odnosno zvanja master umjetnosti, kao što su ime i prezime, godina i mjesto rođenja, naslov master rada i datum odbrane rada.

U Podgorici
11.07.2025.

Potpis magistranda

Stefan Raičević

IZJAVA O KORIŠĆENJU

Ovlašćujem Univerzitetsku biblioteku da u Digitalnom arhivu Univerziteta Crne Gore pohrani moj master rad pod nazivom: **Prilagođenje kućnih uređaja za hlađenje potrebama skladištenja vina**

koji je moje autorsko djelo.

Master rad sa svim prilogima predao/la sam u elektronskom formatu pogodnom za trajno arhiviranje.

Moj master rad pohranjen u Digitalnom arhivu Univerziteta Crne Gore mogu da koriste svi koji poštuju odredbe sadržane u odabranom tipu licence Kreativne zajednice (Creative Commons) za koju sam se odlučio/la.

1. Autorstvo
2. Autorstvo – nekomercijalno
3. Autorstvo – nekomercijalno – bez prerade
4. Autorstvo – nekomercijalno – dijeliti pod istim uslovima
5. Autorstvo – bez prerade
6. Autorstvo – dijeliti pod istim uslovima

Potpis magistranda

Stefan Ravićević

U Podgorici

11.07.2025.