Izrada sustava za nadzor i upravljanje rashladnim procesom

Sičanica, Niko

Master's thesis / Diplomski rad

2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:759008

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-06-03**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

IZRADA SUSTAVA ZA NADZOR I UPRAVLJANJE RASHLADNIM PROCESOM

Diplomski rad

Niko Sičanica

Osijek, 2017.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac D1: Obrazac za imenovanje Povjerenstva za obranu diplomskog rada

Osijek, 20.09.2017.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za obranu diplomskog rada

Ime i prezime studenta:	Niko Sičanica
Studij, smjer:	Diplomski sveučilišni studij Računarstvo, smjer Procesno računarstvo
Mat. br. studenta, godina upisa:	D 686 R, 14.10.2014.
OIB studenta:	87580986493
Mentor:	Doc.dr.sc. Damir Blažević
Sumentor:	
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Doc.dr.sc. Tomislav Matić
Član Povjerenstva:	Dr.sc. Ivan Vidović
Naslov diplomskog rada:	Izrada sustava za nadzor i upravljanje rashladnim procesom
Znanstvena grana rada:	**Arhitektura računalnih sustava (zn. polje računarstvo)**
Zadatak diplomskog rada:	Osmisliti, dizajnirati i izraditi sustav za upravljanje rashladnim procesom u industrijskom okruženju. Za zadani slučaj osmisliti senzore, aktuatore i centralnu upravljačku jedinicu. Na zadanom primjeru izvršiti implementaciju, pokusni i pripremu za trajni rad. Sve elemente sustava temeljiti na standardnim i dostupnim rješenjima.
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	20.09.2017.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

IZJAVA O ORIGINALNOSTI RADA

Osijek, 02.10.2017.

Ime i prezime studenta:	Niko Sičanica
Studij:	Diplomski sveučilišni studij Računarstvo, smjer Procesno računarstvo
Mat. br. studenta, godina upisa:	D 686 R, 14.10.2014.
Ephorus podudaranje [%]:	2

Ovom izjavom izjavljujem da je rad pod nazivom**: Izrada sustava za nadzor i upravljanje rashladnim procesom**

izrađen pod vodstvom mentora Doc.dr.sc. Damir Blažević

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

Sadržaj

1. UVOD ... 1
   1.1. Zadatak diplomskog rada i tehnički zahtjevi ... 2
2. ALKOHOLNA FERMENTACIJA I TEMPERATURE ... 3
   2.1. Utjecaj temperature na alkoholnu fermentaciju ... 4
   2.2. Metode ... 6
      2.2.1. Spontana fermentacija ... 7
      2.2.2. Hladna fermentacija ... 7
      2.2.3. Kontrolirana fermentacija ... 8
3. KOMPONENTE RASHLADNOG SUSTAVA ... 10
   3.1. Arduino Uno Rev3 ... 10
   3.2. Senzor temperature DS18B20 ... 12
      3.2.1. Specifikacije i shema spoja ... 13
   3.3. LCD 16x2 sa I2C adapterom ... 14
   3.4. Tipkalo ... 16
4. SKLOPOVSKA REALIZACIJA RASHLADNOG SUSTAVA ... 18
   4.1. Montažna shema ... 18
   4.2. Električna shema ... 19
   4.3. Kućište ... 20
5. PRINCIP RADA RASHLADNOG SUSTAVA ... 23
   5.1. Temperatura paljenja hlađenja ... 23
   5.2. Temperatura gašenja hlađenja ... 24
   5.3. Detekcija naglog porasta temperature ... 25
6. PROGRAMSKA REALIZACIJA RASHLADNOG SUSTAVA ... 28
7. ZAKLJUČAK ... 36
8. LITERATURA ... 37
9. SAŽETAK ... 38
10. ŽIVOTOPIS ... 39
11. PRILOZI ... 40
   11.1. Programski kod ... 40
   11.2. Shema spoja ... 50

1. UVOD

Tehnika hlađenja je grana tehnike koja se bavi postupcima i pojavama hlađenja tijela. Zadatak tehnike hlađenja je postizanje temperatura nižih od okoline i njihovo održavanje. Ona obuhvaća sve procese i postupke, kao i uređaje, postrojenja i komponente koji služe postizanju, održavanju i korištenju temperatura nižih od okolišne temperature.[2] Hladiti neko tijelo znači smanjiti njegovu unutrašnju energiju, postupkom odvođenja iste, što rezultira nižom temperaturom tijela. Tehnike hlađenja su u današnje vrijeme jako rasprostranjene u raznim granama ljudske djelatnosti i proizvodnje. Jedna od njih je i prehrambena industrija, u kojoj se koriste između ostalog za:

- Konzerviranje hrane hlađenjem – čest postupak konzerviranja hrane, uz konzerviranje toplinom, sušenjem i kemijskim postupcima.
- Hlađenje u svrhu proizvodnje, očuvanja i distribucije raznih životnih namjernica, primjerice mesni proizvodi, mliječni proizvodi, proizvodnja i distribucija piva, vina i drugih alkoholnih pića.
- Hlađenje pri fermentaciji mošta prilikom proizvodnje vina – mošt se hladi zbog postizanja veće kvalitete vina.

To su samo neki od postupaka hlađenja u prehrambenoj industriji, od kojih nam je posebno zanimljivo hlađenje mošta, koje će biti detaljnije obrađeno u ovom radu. U prvom dijelu rada bit će opisan postupak alkoholnog vrenja mošta (alkoholna fermentacija) prilikom proizvodnje vina te razlozi zašto temperatura igra veliku ulogu prilikom tog postupka. U drugom dijelu rada fokus će biti na izradi sustava za nadzor i upravljanje rahladnim procesom, kojemu je primarni cilj održavanje željene temperature prilikom alkoholne fermentacije. Također, korišteni senzori, aktuatori, mikroupravljači te ostale komponente će također biti pobliže predstavljeni i opisani. Posebice Arduino, elektronička platforma otvorenog koda (eng. *open-source*) namjenjena kreiranju elektroničkih projekata, a čine ga sklopovlje (eng. *hardware*) koji je elektronički programabilni strujni krug, odnosno mikrokontroler, i programsko okruženje koji se zove Arduino IDE (eng. *Integrated Development Enviroment*) koje se koristi za programiranje i upravljanje samim mikrokontrolerom, odnosno Arduinom.

1.1. Zadatak diplomskog rada i tehnički zahtjevi

Zadatak diplomskog rada je osmisliti, dizajnirati i izraditi sustav za upravljanje rashladnim procesom u industrijskom okruženju. Za zadani slučaj osmisliti senzore, aktuatore i centralnu upravljačku jedinicu. Na zadanom primjeru izvršiti implementaciju, pokusni i pripremu za trajni rad. Sve elemente sustava temeljiti na standardnim i dostupnim rješenjima. Zadani primjer je fermentacija i skladištenje mošta, koje se odvija u inox posudama ili cisternama, gdje svaka posuda ima svoj rashladni cirkulacijski krug. Potrebno je napraviti regulaciju temperature svake pojedine posude, a regulacija se treba obavljati na način da se cirkulacijski krugovi uključivaju ili isključivaju ventilima, koji će propustiti tekućinu za hlađenje kroz cirkulacijske krugove.

Podatak koji je potreban za otvaranje ili zatvaranje cirkulacijskih krugova je trenutna temperatura mošta, stoga povratnu vezu sustava čine temperaturni senzori koji se postavljaju na svaku pojedinu posudu. Temperaturu je potrebno regulirati u rasponu od 0 $^{\circ}C$ do 30 $^{\circ}C$, no najčešća zahtjevana temperatura regulacije je u rasponu od 18 $^{\circ}C$ do 20 $^{\circ}C$. Bitno je napomenuti da je osnova hlađenja mošta pri fermentaciji sprečavanje pregrijavanja, koje se događa na temperaturama iznad 30 $^{\circ}C$, kada je zbog preburne fermentacije koja nastaje temperaturu mošta praktično nemoguće ponovno sniziti, pogotovo u velikim cisternama od primjerice 5000 litara. Zato sustav, osim trenutne temperature, treba moći prepoznati i nagli porast temperature (gradijent) i otvoriti cirkulacijski krug i prije nego mošt dostigne zadanu temperaturu.

Sustav treba biti konfigurabilan, odnosno da se za svaku posudu ili cisternu zasebice može postaviti temperatura na kojoj će se otvarati cirkulacijski krug, odnosno paliti hlađenje, kao i temperaturu pri kojoj će se zatvoriti cirkulacijski krug, odnosno isključiti hlađenje.

2. ALKOHOLNA FERMENTACIJA I TEMPERATURE

Vrenje mošta, tj. alkoholna fermentacija predstavlja jednu od osnovnih faza u procesu proizvodnje vina. Vrenjem mošta, u stvari, počinje stvaranje vina.[2] Mošt je iscijeđeni sok grožđa, koji se nakon fermentacije pretvara u vino, a faza vrenja mošta je faza koja dolazi nakon muljanja grožđa ili cijeđenja mošta. Tada kvasci, odnosno gljivice alkoholne fermentacije koje su se nalazile na koži bobice, dolaskom u tekućinu u kojoj je otopljen šećer (u mošt), počinju intezivno razmnožavanje pritom razlažući šećer na različite kemijske spojeve, no prvenstveno na alkohol i CO_2. Ova dva spoja su osnovni i najbitniji produkti rada vinskog kvasca, dok je alkohol osnovni sastojak vina, odnosno svakog alkoholnog pića dobivenog alkoholnom fermentacijom.

Uz već navedena dva osnovna produkta fermentacije, alkohol i CO_2, pojavljuju se u manjim količinama i niz raznih drugih spojeva, poput glicerina, octene kiseline, jantarne kiseline i drugih. U isto vrijeme pojavljuje se određena količina energije, koja se manifestira u vidu topline, a djelomično je iskorištena od kvasaca za životne potrebe. Tijek vinifikacije, međusobni odnos i količina nastalih spojeva, kao i količina nastale energije se razlikuju, što ovisi o mnogim činiteljim koji utječu na proces vinifikacije, što rezultira različitim odnosima pojedinih spojeva i količinom energije. Kvasci izazivaju određene promjene u moštu koje se manifestiraju na sljedeći način:

Mošt se ponajprije jako zamuti, te se počinju stvarati mjehurići i debeli sloj pjene koja počinje vreti. Pritom se pojavljuje porast temperature za $10^{\circ}C$, $20^{\circ}C$ i više. Debljina sloja pjene i intezitet vrenja ponajviše ovise o temperaturi mošta, ali i o kvaliteti sastava mošta, posudi u kojoj se odvija postupak, vrsti kvasca itd. Ovaj proces i promjene koje on izaziva naziva se alkoholno vrenje mošta ili alkoholna fermentacija.

No, prilikom postupka muljanja grožđa poznato je da u moštu sa željenim vinskim kvascima, završe i nepoželjni kvasci, kao i razni drugi štetni mikroorganizmi, koji u vinu uzrokuju neželjene procese. U ovisnosti o uvjetima u kojima su se nalazili, i kvasci se različito ponašaju, stvarajući različite količine alkohola, CO_2, i ostalih spojeva. Uz navedeno, pod utjecajem raznih neželjenih činitelja (temperatura koja nije idealna, sastav mošta, različiti mikroorganizmi i dr.) može doći do prekida postupka vrenja. Primjerice, mošt nastao od trulog grožđa teško prevrije i lako dolazi do prestanka vrenja ako se ne poduzmu odgovarajuće mjere, poput upotrebe odabranih kvasaca i slično. Također, visoke temperature dovode do prestanka vrenja.

Nepoželjan sastav mošta će također izazvati prestanak vrenja, primjerice velika količina šećera, visoke koncetracije CO_2, SO_2, kao i nedovoljan sadržaj kiselina. Hoće li fermentacija, odnosno vrenje mošta pod utjecajem željenih sorti vinskog kvasca biti uspješna, ili će vrenje prilikom djelovanja štetnih mikroorganizama krenuti u krivom pravcu, ovisi o tome jesu li u moštu kreirani uvjeti pogodni vinskom kvascu, koji su različiti za različite slučajeve.

2.1. Utjecaj temperature na alkoholnu fermentaciju

Temperatura je jedan od najvažnijih čimbenika koji utječu na početak i završetak alkoholne fermentacije, a time i na njeno trajanje. Hoće li mošt početi fermentirati prvi dan, ili nakon 8 i više dana, hoće li završiti s fermentiranjem prije ili kasnije, ili će se pak fermentacija prekinuti, hoće li kvasci stvoriti višu ili nižu razinu alkohola, sve te situacije ovise značajno o temperaturi mošta. Treba imati u vidu da su kvasci, odnosno kvaščeve gljivice živući organizmi, stoga se utjecaj temperature odražava direktno na sve njihove životne funkcije.

Ako je temperatura bila viša, alkoholna fermentacija će započeti ranije, no također će biti intenzivnija i trajati će kraće. Kao što je navedeno, kvasac je živi organizam, koji ima svoje optimalne temperature, kao što su temperature koje mu direktno pogoduju, ali i temperature koje ga uništavaju i ubijaju. Prema Schumacheru, kvasac ohlađen na -113$^{\circ}C$ nije ubijen.[3] A prema navodima nekih drugih autora, nije uginuo ni na -200 $^{\circ}C$. No s druge strane, na visoke temperature je kvasac veoma osjetljiv, te je visoka temperatura ujedno i najčešći razlog prekida alkoholne fermentacije. Kritičnim i previsokim se smatraju temperature iznad 32$^{\circ}C$ do 33$^{\circ}C$. Hoće li se alkoholno vrenje mošta prekinuti na 32$^{\circ}C$, 36$^{\circ}C$ ili čak 40$^{\circ}C$ ovisi o drugim faktorima, kao što su primjerice:

- Vrsta kvasca.
- Koncetracija šećera i kiselina
- Količina alkohola
- Zračenje mošta
- Jačina sumporenja

Način na koji temperatura utječe na početak, tijek i završetak alkoholne fermentacije je vidljiv iz tablice 2.1.

T [$^{\circ}C$]	Fermantacija započela nakon [dana]	Ostvareno alkohola u vinu [$\%$]
10	8	16.2
15	6	15.8
20	4	15.2
25	3	15.2
30	1.5	10.2
35	1	6.0

Tablica 2.1. Tijek alkoholne fermentacije ovisno o temperaturi

Dakle, prema tablici 2.1. uočava se da optimalna temperatura za početak fermentacije nije u isto vrijeme i optimalna temperatura za intezitet fermentacije i njen krajnji ishod u pogledu količine alkohola stvorenog vrenjem. Temperatura u iznosu $25^{\circ}C$ - $28^{\circ}C$ smatra se optimalnom za rad, odnosno život kvasca, međutim za stvaranje većeg postotka alkohola poželjna je, kako se da iščitati iz tablice 2.1., niža temperatura od oko $10^{\circ}C$. Ako se mošt zagrijava na temperaturu od $65^{\circ}C$ u trajanju od 10 minuta, svi kvasci će uginuti. Ova temperatura nije stalna, jer ovisi i o drugim čimbenicima kako je već navedeno, naročito o vremenu trajanja zagrijavanja. Ako se skrati vrijeme zagrijavanja, onda je temperatura na kojoj dolazi do stradanja kvasaca viša. Vrijedi i obratno, ako zagrijavanje traje duže temperatura na kojoj će kvasci uginuti je niža.

U pogledu graničnih najnižih temperatura, vrijede sljedeća pravila: za temperature ispod $10^{\circ}C$ koriste se kvasci hladnog vrenja ili krio kvasci, koji vrše fermentaciju na temperaturama znatno nižim od običnih kvasaca. Primjerice, korištenjem hladnog kvasca se već nakon 6 dana uočava fermentacija mošta na temperaturi od $2^{\circ}C$, a pri temperaturi mošta od -$2^{\circ}C$ nakon 20 dana. Međutim, korištenjem običnog kvasca fermentacija na nižim temperaturama počinje dosta kasnije i jako dugo traje. Primjerice, ako je temperatura mošta $2^{\circ}C$, obični kvasci će početi fermentirati tek nakon 30 dana, dok na -$2^{\circ}C$ kvasci uopće ne pokazivaju znakove fermentacije.

Na porast temperature značajno utječe i materijal od kojeg su izrađene vinske posude (drvo, čelik, plastika, beton itd.), a u određenoj mjeri i veličina posuda u kojim se odvija fermentacija.

Prema nekim istraživanjima porast temperature za vrijeme vrenja u drvenim bačvama različitih veličina je bio sljedeći: u bačvi kapaciteta 6 hl pri procesu vrenja, temperatura se povećala za 11.50$^{\circ}C$, u bačvi kapaciteta od 45 hl za 17$^{\circ}C$, a u bačvi od 72 hl temperatura se povećala za 20 $^{\circ}C$.

Temperatura vrenja [$^{\circ}C$]	Trajanje vrenja [dana]
36	17
27	24
18	46
9	100

Tablica 2.2. Trajanje vrenja ovisno o temperaturi mošta

Tablica 2.2. nam pokazuje trajanje vrenja u ovisnosti od temperature mošta, no i ovdje na tijek i trajanje vrenja utječu i drugi čimbenici, stoga navedeni dani su samo relativni. Temeljeno na novijim enološkim saznanjima, poseban značaj se pripisuje nižim temperaturama fermentacije pri proizvodnji bijelih vina, posebice kvalitetnih. Za razvoj bukea (aroma vina koja ovisi o starenju), boje, ekstrakta i drugog, idealna ili optimalna temperatura vrenja kod bijelih vina je 15 $^{\circ}C$ do 18$^{\circ}C$, a kod crnih 25$^{\circ}C$ do 30$^{\circ}C$. U pogledu najkvalitetnijih bijelih vina (kod sorti sa puno aromatični materija) poželjne su i niže temperature, 10$^{\circ}C$ do 14$^{\circ}C$, uz upotrebu kvasaca za hladnu fermentaciju.

2.2. Metode

U praksi postoji, i primjenjuje se, čitav niz različitih metoda provođenja alkoholne fermentacije mošta. Postoji dosta podjela, kao što su spontane klasične i suvremene kontrolirane metode, no ovdje će biti opisano ukratko samo nekoliko najpoznatijih i najkorištenijih, koje su karakteristične za postupak pri provođenju vinifikacije. Pritom, fokus će biti na stvar koja nam je od najvećeg interesa zbog teme ovog diplomskog, a to je temperatura, optimalne temperature pri kojima se vrši postupak fermentacije u pojedinoj metodi, neželjene posljedice rasta temperature iznad optimalnih vrijednosti, prednosti fermentacije pri niskim temperaturama, kao i opasnosti koje prijete pri takvim hladnim fermentacijama.

2.2.1. Spontana fermentacija

Taložen ili ne taložen mošt, sumporen ili ne, pretače se u posude za fermentaciju (bačve, cisterne ili drugo). Ukoliko je mošt prethodno jače sumporen radi taloženja, pretače se uz prozračivanje da bi se oslobodio višak SO_2 (sumporni dioksid). Mošt od trulog grožđa obavezno se sumpori sa 15 do 25 g/hl vinobrana, ili više, ovisno koliko je grožđe trulo. Važno je sumporenje mošta ili grožđa što prije obaviti, naročito u toplijim krajevima gdje fermentacija i kvarenje grožđa započinje još u vinogradu, odnosno za vrijeme transporta gdje dolazi do gnječenja grožđa. Da bi se fermentacija mošta odvijala normalno mora se ostaviti najmanje 1/10 praznog prostora u bačvi ili spremniku. U pojedinim slučajevima, naročito u južnijim krajevima zemlje gdje je fermentacija znatno intezivnija, prazni prostor mora biti i veći, inače bi došlo do prelijevanja pjene tijekom burne alkoholne fermentacije.

Optimalna temperatura za ovakav tip fermentacije mošta je 15 $^{\circ}C$ do 18 $^{\circ}C$. Pri spontanoj fermentaciji ispod 15$^{\circ}C$ burna fermentacija se prekida, a može i u potpunosti prestati. Međutim, u južnijim krajevima burna fermentacija se često odvija i na temperaturi iznad 36$^{\circ}C$. Tako visoka temperatura tijekom fermentacije štetno utječe na kvalitetu vina, a posebice na sadržaj alkohola i aromu. Često se događa da se zbog visoke temperature vrenje mošta prekine, i u tim slučajevima odmah treba poduzeti odgovarajuće mjere, poput pretakanja na zraku i drugih, da bi osvježili skoro umrtvljeni kvasac, jer u suprotnom primat preuzimaju nepoželjni mikroorganizmi te se dobiva slatkasto ili octeno vino, mutno, gorko itd. Inače, ukoliko se ne prekine postupak fermentacije na višim temperaturama, vina gube znatno na kvaliteti, u pogledu gubitka arome, kao i alkohola koji koji zbog burnog vrenja i visoke temperature hlapi u znatnom postotku, zajedno sa znatnim količinama CO_2 i vodene pare. I kvasci u ovakvim nepovoljnim uvjetima, da bi se očuvali, troše znatno veće količine šećera nego u normalnim uvjetima fermentacije. I zaključno, uz navedeno, fermentacija u uvjetima visoke temperature uništava buke, aromu i boju vina, te prilikom proizvodnje visokokvalitetnih sortnih bijelih vina takvu fermentaciju treba izbjegavati.

2.2.2. Hladna fermentacija

Hladna fermentacija se odvija u uvjetima niske temperature, uvjetovane od strane tehnologa uz pomoć uređaja za hlađenje mošta, odnosno rashladne uređaje, i uz korištenje već spomenutog selekcioniranog kvasca za vrenje na niskim temperaturama, takozvanog frigo kvasca.

Fermentacijom mošta na nižim temperaturama (hladna fermentacija) se dobivaju vina sa većim postotkom alkohola, te izraženije arome. Također, fermentacijom na nižim temperaturama se dobijaju svježija i zdravija vina, sa više CO_2, a manje octene kiseline. Hladna fermentacija se obično odvija na temperaturi nižoj od $15^{\circ}C$, a početna temperatura mošta bi trebala iznositi $5^{\circ}C$ do $11\,^{\circ}C$. U ovoj metodi posebnu ulogu igraju kvasci hladnog vrenja (Frigofilni) koji mogu razgrađivati šećer na nižoj temperaturi, a da pritom ne izazivaju burno vrenje. Hladna fermentacija traje poprilično dugo, nekada i po nekoliko mjeseci bez prekida. I uz veoma korisni efekt hladne fermentacije, ona se kod nas ne primjenjuje puno zbog jednog nedostatka:

- Uvijek postoji opasnost da bi se uz povišenje temperature mogla pretvoriti u burnu, odnosno toplu fermentaciju.

Stoga je za nju je potrebno imati posebne uređaje za hlađenje mošta, odnosno rashladne sustave s kojima bi osigurali sigurno odvijanje hladne fermentacije. S druge strane neke od prednosti hladne fermentacije su sljedeće:

- Vino apsorbira i zadržava više CO_2, što čini hladno fermentirana vina svježijim
- Dobijaju se vina s većim postotkom alkohola
- Vina se brže bistre i ranije budu spremna za konzumiranje
- Filtracijska sposobnost vina je povećana
- Oksidacijski procesi su sporiji, time i proces starenja vina
- Štiti primarne buketne tvari i poboljšava sortni karakter vina
- Fermentacija je pravilnija i čišća
- Zahtjeva manje sumporenje i omogućuje fermentiranje u većim posudama

2.2.3. Kontrolirana fermentacija

Napredak tehnologije i znanosti je i u procesu proizvodnje vina doveo do velikih izmjena. Jedna od tih izmjena je, da prilikom same fermentacije mošta, umjesto da postupak prepuštamo spontanom vrenju, kada u pogledu inteziteta vrenja imamo visoke oscilacije i padove, a isto tako u pogledu temperature, što se nepovoljno odražava na kvalitetu. Zahvaljujući naprecima na polju tehnologije, fermentacijom se može upravljati, kako u pogledu inteziteta vrenja tako i u pogledu temperature mošta. Takve metode fermentacije se nazivaju kontrolirane ili dirigirane fermentacije, u kojima se može primjenom tehnologije obuzdati burna fermentacija i velike oscilacije temperature mošta. Ova metoda fermentacije se naziva i usporena fermentacija jer uravnotežuje proces fermentacije, odnosno spriječava velike oscilacije temperature i inteziteta.

Primjenom kontrolirane fermentacije, njen tijek teče ravnomjernije i izbjegavaju se visoke i kritične temperature, koje narušavaju buduću kvalitetu vina. Ovakav tip je naročito značajan prilikom proizvodnje kvalitetnijih vina, kod kojih se teži zadržavanju arome i svježine, za razliku od spontane fermentacije koja daje vina s nižim postotkom alkohola, te se zbog visoke temperature u procesu fermentacije i CO_2 znatno brže gubi,a time i boja i aroma, kao i ukus. Također, u ovakvim uvjetima su i oksidativni procesi znatno intezivniji. Tijekom kontrolirane ili usporene fermentacije, s obzirom da se odvija na nižim temperaturama i nema naglih skokova u intezitetu vrenja i visine temperature dobivaju se vina s više:

- Alkohola.
- CO_2 (Ugljičnog dioksida).
- Zelene boje - kod bijelih vina.
- Svježine - u pogledu okusa i mirisa.

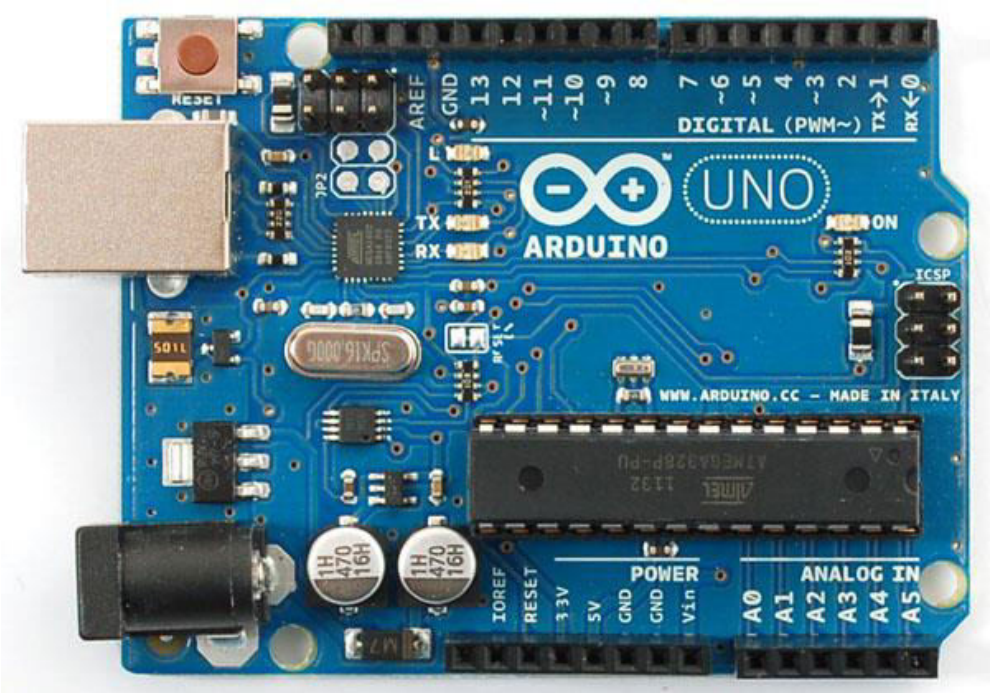
Primjenom ove metode se ne uništava aroma, kao što je čest slučaj sa spontanom fermentacijom, posebice u južnim krajevima gdje su za vrijeme berbe i prerade grožđa temperature dosta visoke. Ako izuzmemo neka jača crna vina, ili neka bijela specijalne kvalitete, moderni potrošač zahtjeva da u vinu ostane što više arome, okusa i svježine, kao i određenog ostatka CO_2. Postoji nekoliko načina izvođenja kontrolirane fermentacije, no sve se više manje zasnivaju na reguliranju temperature i CO_2, ili i jedno i drugog. Pritom se koriste različiti tehnološki uređaji, kao što su rahladni uređaji za hlađenje mošta ili specijalne metalne posude za rad pod tlakom CO_2 koji se stvara fermentacijom. Odstupanja od nekoliko stupnjeva mogu učiniti ogromnu razliku po pitanju kvalitete vina, stoga se iz svega navedenog može iščitati koliko velik značaj rashladni sustavi imaju pri ovom procesu i kakvu pouzdanost moraju imati.

3. KOMPONENTE RASHLADNOG SUSTAVA

U izradi ovog diplomskog rada bilo je potrebno koristiti dostupne komponente, omogućiti rad svake komponente te njihovo međusobno povezivanje kako bi se realizirao sklop koji će obavljati zadaću rashladnog sustava. Potrebo je osmisliti neku upravljačku jedinicu koja bi obavljala rashladnim procesom, za koju je odabran Arduino Uno Rev3, prilično jeftina mikroupravljačka pločica sposobna obavljati dosta zahtjevne zadaće. Također je bilo potrebno omogućiti praćenje procesa, u ovom slučaju temperature, te je za tu zadaću izabran senzor temperature DS18B20, koji je vodootporan i služi ta mjerenje temperature tekućine, ali i zraka. Za komunikaciju sa Arduinom koristi jednu žicu, te je moguće na jedan PIN Arduina spojit čak 127 takvih senzora. Za prikaz temperature je korišten LCD 16x2 sa IIC adapterom, koji koristi HD447780 kontroler i omogućuje prikazivanje teksta, simbola i bilo čega drugog šta dolazi s Arduina u 2 retka po 16 znakova.

3.1. Arduino Uno Rev3

Arduino Uno (Slika 3.1.) je mikroupravljačka razvojna platforma, osmišljena 2005. godine sa svrhom omogućavanja izrade različih sklopova koji bi prema napisanom kodu upravljali različitim komponentama i u konačnici sustavima. Središnja jedinica za obradu podataka Arduina je Atmelov mikrokontroler ATmega328P, u kojem se napisani kod neprestano izvršava u petlji (eng. *loop*).



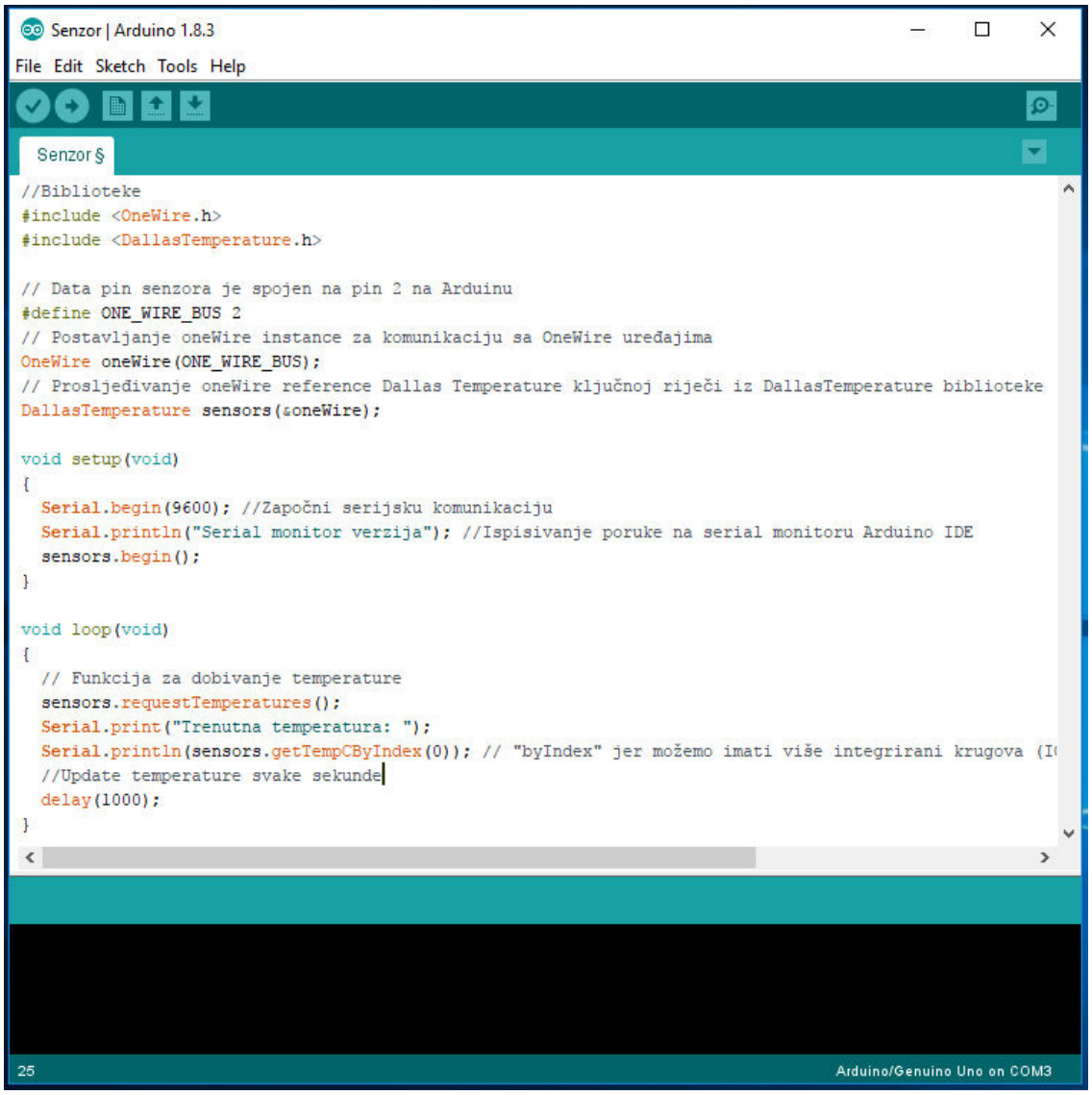
Slika 3.1. Arduino UNO mikroupravljačka platforma

Napon na kojem mikrokontroler radi iznosi 5V, stoga treba paziti da prilikom spajanja vanjskih komponenti poput raznih senzora i aktuatora ne pređemo taj napon, inače može doći do oštećenja pločice. Pisanje programskog koda koji će Arduino izvršavat se radi u Arduino IDE (eng. *Integrated Development Enviroment*) programskom okruženju i njegovo prebacivanje na razvojnu pločicu se vrši USB (eng. *Universal serial bus*) portom, kojim se također i napaja Arduino ako nemamo nekakav vanjski izvor napajanja. Tehničke specifikacije Arduino UNO Rev3 mikroupravljačke platforme vidimo u tablici 3.1. [4]

Mikrokontroler	ATmega328P
Radni napon	5V
Ulazni napon (preporučeno)	7-12V
Ulazni napon (granica)	6-20V
Digitalni I/U pinovi	14 (od kojih su 6 PWM izlazi)
PWM digitalni I/U pinovi	6
Analogni ulazni pinovi	6
DC struja po I/U pinu	20 mA
DC struja za 3.3V pin	50 mA
Flash memorija	32 KB (ATmega328P) od koje 0.5 KB koristi bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Brzina rada	16 MHz

Tablica 3.1. Tehničke specifikacije Arduino UNO Rev3

Kao što je već navedeno, programiranje Arduina se vrši u Arduino IDE programskom okruženju gdje se programiranje obavlja u prilagođenoj verziji C++ programskog jezika, a samo razvojno okruženje je napisano u Javi.[5] Programski jezik je pojednostavljen do te razine da se osnovne stvari i pojmovi mogu vrlo brzo naučiti, te svatko tko je imao dodira sa C/C++ bi se trebao snaći dosta lako. Prije početka pisanja programskog koda potrebno je uključiti biblioteke u skicu, koje dolaze uz pojedine senzore, aktuatore i ostale komponente, koje su ljudi razvili uz vrijeme i koje su besplatne za korištenje, a u sebi sadrže definirane načine rada pojedine komponente i znatno olakšavaju pisanje koda. Primjerice, biblioteka *<OneWire.h>* sadrži ključnu riječ *OneWire* koja se koristi prilikom definiranja senzora, tako da; *OneWire oneWire(ONE_WIRE_BUS),* linijom koda definiramo oneWire senzor temperature koji se nalazi na ONE_WIRE_BUS pinu Arduina, kao što vidimo iz slike 3.2., koja je ujedno i primjer izgleda Arduino IDE.



Slika 3.2. Arduino IDE programsko okruženje

3.2. Senzor temperature DS18B20

Za očitavanje temperature mošta u spremniku korišten je digitalni vodootporni senzor temperature DS18B20, koji služi za mjerenje temperature tekućine, ali i zraka. Mjerno područje mu iznosi od -55°C do 125°C, te je obložen PVC-om (poli-vinil klorid) što ga čini primjerenim za svakakve uvjete, kao i to što je digitalni pa nema degradacije signala na većim udaljenostima. Također, odlično radi sa svim vrstama mikrokontrolera, te za komunikaciju s Arduinom koristi samo jednu žicu, i posjeduje mogućnost spajanja do čak 127 senzora na jedan pin na Arduinu. Sastoji se od 3 pina koji se koriste za spajanje sa drugim uređajima, i to GND ili masa, V_{DD}ili napajanje, te DQ (eng. *Data In/ Out*) taj jedan pin ili žica koji služi za komunikaciju s Arduinom i prijenos podataka.[6] Na slici 3.3. se vidi primjer DS18B20 digitalnog vodootpornog senzora.

Slika 3.3. Senzor temperature DS18B20 i raspored pinova

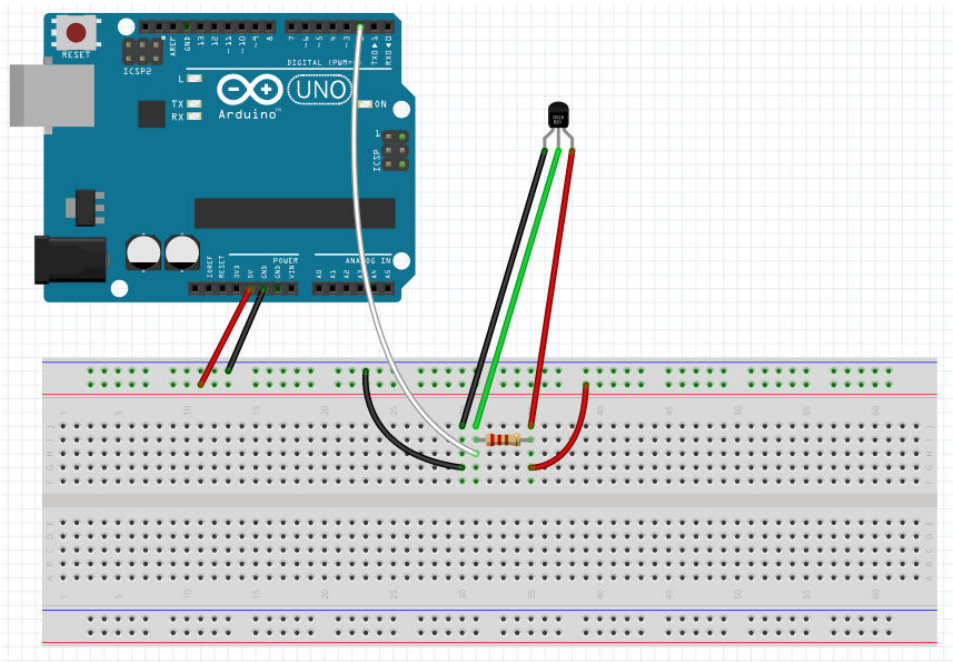
Kao što je već navedeno, na jedan digitalni pin na Arduinu se može spojiti čak 127 senzora temperature DS18B20 i mjeriti temperatura na 127 mjesta. To je omogućeno 64-bitnom oznakom koja je tvornički utisnuta i unikatna za svaki senzor, čime ih se može razlikovat prilikom korištenja i programiranja. Također, može raditi na tzv. parazitskom načinu rada gdje napajanje dobija preko podatkovnog DQ (eng. *Data*) pina, i u kojem nema potrebe za vanjskim napajanjem.

3.2.1. Specifikacije i shema spoja

Tehničke specifikacije:

- Raspon mjernog područja temperature: -55 $^{\circ}C$ do 125 $^{\circ}C$.
- Koristi tzv. 1-Wire Interface: zahtjeva samo jedan digitalni pin za komunikaciju.
- Unikatna 64-bitna oznaka: utisnuta tvornički u ROM (eng. *Read-only memory*) senzora za razlikovanje.
- Višestruki senzori mogu dijeliti jedan pin.
- $\pm$0.5 $^{\circ}C$ točnost u mjernom području od -10 $^{\circ}C$ do 85 $^{\circ}C$.
- Vrijeme očitanja promjene temperature manje od 750 ms.
- Koristi napon 3.0V do 5.5V na V_{DD} pinu za napajanje, kao i na DQ pinu.

Nedostatak je taj što se ne može koristiti u slanoj vodi, kao ni u nekim korozivnim okruženjima.



Slika 3.4. Shema spoja DS18B20 senzora temperature

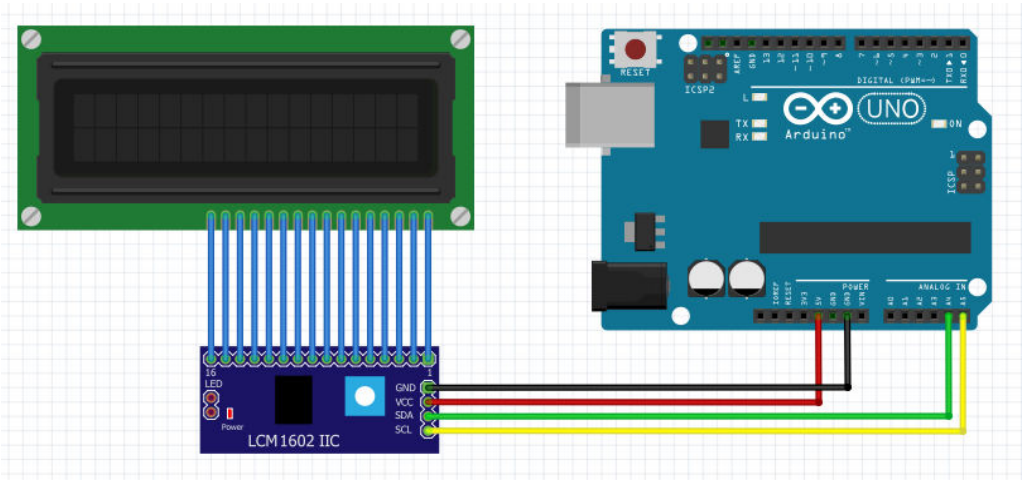
Sa Arduina dovodimo napon od 5V (crvena žica) na eksperimentalnu pločicu (eng. *breadboard*), odnosno pločicu koja služi spajanju komponenti i uređaja bez potrebe lemljenja, te prosljeđujemo napon na V_{DD} pin senzora. Također dovodimo masu sa GND pina na Arduinu (crna žica) i prosljeđujemo je na GND pin senzora. DQ pin senzora spajamo na neki od digitalnih I/U pinova na Arduinu, u ovom slučaju je to pin 2. Na shemi je vidljiv i tzv. *pull-up* otpornik u iznosu od 4.7 kΩ, spojen između V_{DD} i DQ pinova senzora, a svrha mu je da osigura zadržavanje željenog stanja logičkog kruga bez obzira na uvjete. Ako neki od pinova Arduina nije postavljen na *HIGH* ili *LOW,* on tada pluta između ta dva stanja koja se izmjenjivaju i pojavljuje se stanje visoke impedancije, što nije prihvatljivo. Takav tip otpornika nije nikakav poseban, nego samo neka stalna vrijednost spojena uglavnom preko +5V i pina Arduina.

3.3. LCD 16x2 sa I2C adapterom

Ispis temperature se vrši na LCD 16x2 (eng. *Liquid Crystal Display*) zaslonu, koji omogućuje prikaz bilo kakvih informacija na lako čitljivom LCD zaslonu. Zaslon je plave boje sa bijelim slovima, i sastoji se od dva retka po 16 polja, na kojim se mogu prikazivati bilo kakva slova, brojevi i simboli.[7] Princip rada im se temelji na tekućim kristalima, od kojih svaki tvori jedan piksel (eng. *pixel*), i oni se nalaze između dva polarizatora te zakreću svjetlost. Kada se propusti određeni napon na njih oni se ispravljaju te polarizacija ne zakreće svjetlost, tada vidimo određena polja ili piksele kao aktivne. Na slici je 3.5. je prikazan LCD 16x2 zaslon sa I2C adapterom.

Slika 3.5. LCD 16x2 ekran sa I2C adapterom

Za spajanje LCD-a potrebno je spojiti čak 11 žica na Arduino, 2 žice na 2 GND-a Arduina, 1 žica na napon od +5V, te 8 žica na digitalne I/U pinove Arduina, kojih usput budi rečeno ima ukupno 14 na Arduinu. Očito, taj način spajanja LCD-a i nije praktičan jer zauzme puno digitalnih I/U pinova na Arduinu, čime se direktno smanjuje mogućnost spajanja drugih uređaja i komponenti. Tu u priču ulazi I2C adapter koji se zalemi na stražnju stranu LCD-a te osigurava znatno jednostavnije povezivanje zaslona sa Arduinom, i još bitnije oslobađa digitalne pinove koji se onda mogu koristiti u druge svrhe.[8] Korištenjem adaptera potrebno je spojiti samo 4 žice na Arduino razvojnu pločicu, kao što je vidljivo na slici 3.6.



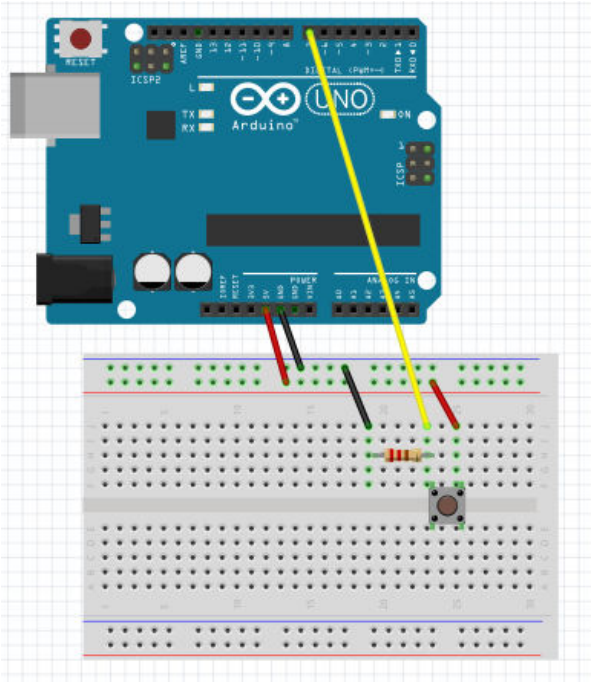
Slika 3.6. Shema spoja LCD 16x2 ekrana koristeći I2C adapter

Slika 3.6. prikazuje način spajanja LCD-a koristeći I2C adapter. Napon i masa se dovode spajanjem na +5V i GND pinove na Arduinu, to su crvena i crna žica sa slike 3.6.

Ostatak konekcije se vrši tako da se SCL i SDA konektori I2C adaptera spoje na analogne pinove Arduina, A4 i A5 (žuta i zelena žica). Pisanje programskog koda za korištenje LCD-a se vrši pomoću *LiquidCrystal* biblioteke, koja inače dolazi sa Arduino IDE i funkcionira poprilično dobro. Neke od važnijih funkcija iz biblioteke su *lcd.print()* koja služi za ispisivanje varijable, stringa ili drugog sadržaja na zaslon, *lcd.clear()* funkcija koja briše sadržaj na zaslonu, *lcd.backlight()* koja pali pozadinsko osvjetljenje zaslona, i druge.

3.4. Tipkalo

Za sustav poput rashladnog, mora postojati mogućnost brze i efikasne promjene vrijednosti temperature na kojoj će sustav uključivati hlađenje. To je postignuto pomoću tipkala, koja uz promjenu vrijednosti temperature imaju i druge uloge, poput ulaska u izbornik u kojem je moguće mjenjat temperaturu pri kojoj će se paliti i gasiti hlađenje, postavljati iznos temperature, koji ako se pređe u određenom vremenskom rasponu također pali hlađenje kako se bi se spriječio nagli porast temperature, koji bi onda teško bilo ohladit. Također se u izborniku može konfigurirat taj vremenski raspon u kojem bi došlo do naglog porasta, te vrijeme hlađenja, odnosno koliko dugo bi hlađenje radilo ako bi se upalilo prilikom naglog porasta temperature. Kako je sustav namjenjen za reguliranje hlađenja za 3 posude, tako imamo i 3 senzora temperature, te za svaki posebno možemo postavljati temperature pri kojima dolazi do paljenja i gašenja hlađenja, i to je također ostvareno pomoću tipkala.



Slika 3.7. Shema spoja tipkala

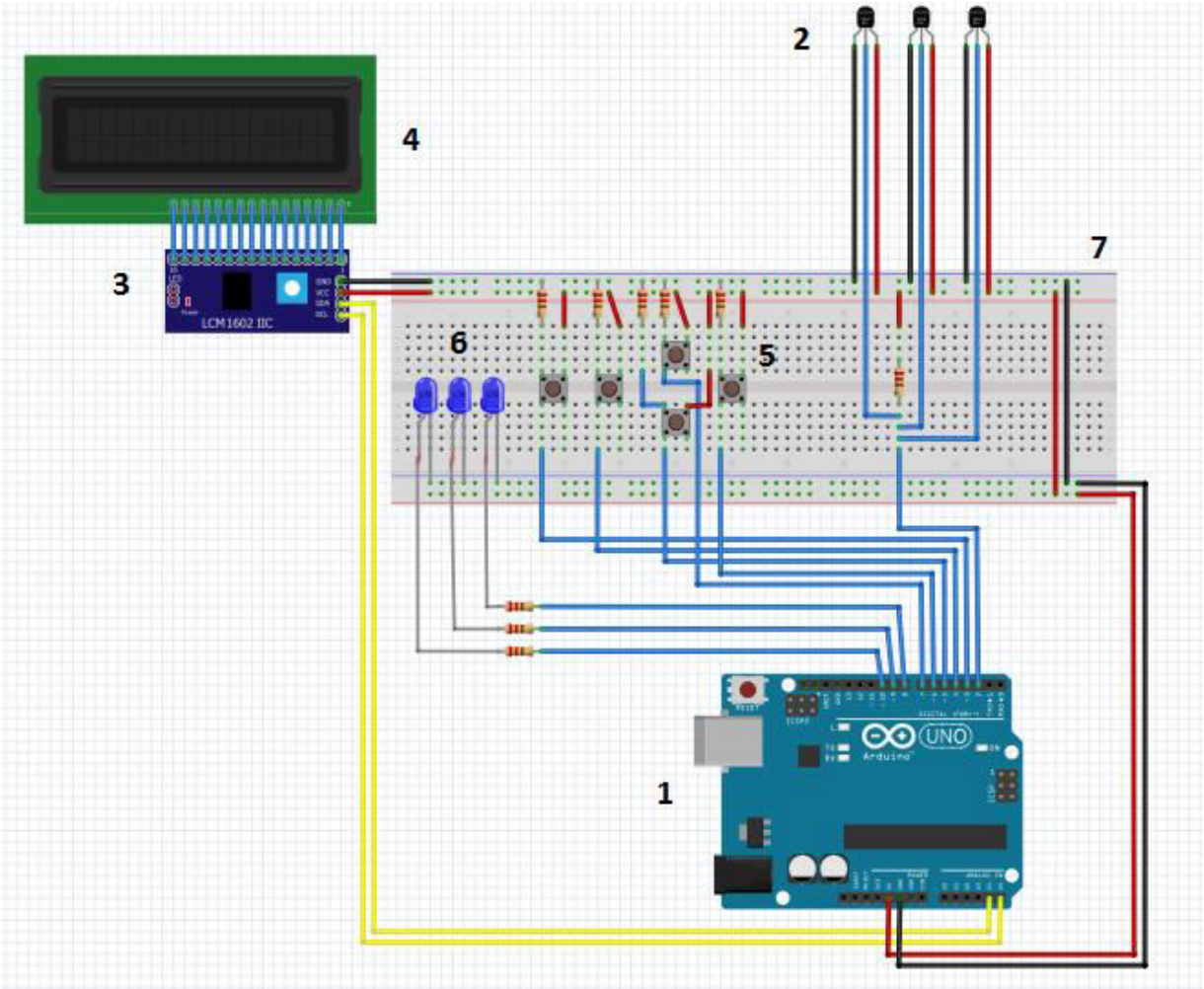
Slika 3.7. pokazuje shemu spoja tipkala. Tipkalo spajamo na +5V (crvena žica), koje će prilikom pritiska tipke proslijediti signal na primjerice digitalni pin 7 (žuta žica) kao na shemi ili neki drugi, na kojem se očitava stanje tipkala. Nožica tipkala koja je spojena na digitalni pin Arduina također je spojena i na GND, odnosno masu ili uzemljenje (crna žica), preko *pull-up* otpornika od 10 kΩ, koji je nužan kod spajanja tipkala kako bi izbjegli variranje pina (eng. *float*) između HIGH i LOW, odnosno jedinice i nule. Otpornikom osiguravamo željeno logičko vladanje.

4. SKLOPOVSKA REALIZACIJA RASHLADNOG SUSTAVA

Sklopovska realizacija rashladnog sustava je realizirana i testirana na način, da je prvo korištena eksperimentalna pločica (eng. *breadboard*) pomoću koje se mogu spajati sve komponente poput žica, otpornika, senzora, LED dioda, te samog Arduina bez potrebe lemljenja.

4.1. Montažna shema

Montažna shema je osmišljena i izrađena u CAD programu imena Fritzing.[9] Fritzing je program otvorenog koda (eng. *open-source*) namjenjen dizajniranju shema i elektroničkih sklopova koji se kasnije mogu realizirati ili pustiti u izradu. Sadrži sve elektroničke komponente, te elemente vezane za Arduino platformu poput senzora, aktuatora, zaslona i svih ostalih. Odličan alat za inovatore, dizajnere, edukaciju i najvažnije primjeren za korištenje ljudima koji nisu inženjeri.



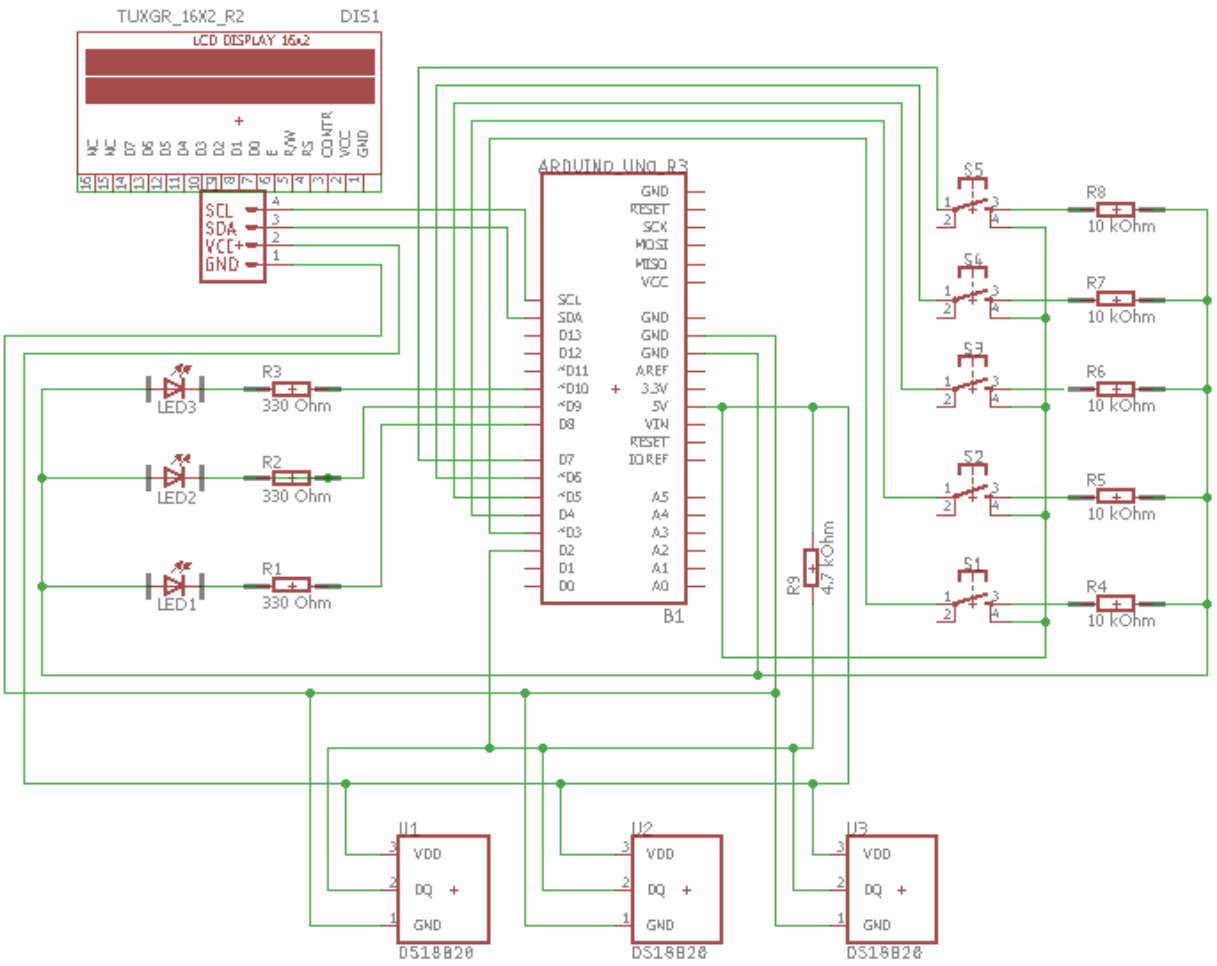
Slika 4.1. Montažna shema spoja rashladnog sustava

Slika 4.1. pokazuje montažnu shemu spoja rashladnog sustava. Na mikrokontroler Arduino UNO Rev3 razvojne pločice (1) je prije puštanja u rad učitan programski kod preko USB kabla tipa B (eng. *USB Type B*). Programski kod obavlja zadane funkcije, omogućuje željeno vladanje rashladnog sustava i komunikaciju između spojenih komponenti. Tri DS18B20 senzora (2) su spojena na digitalni pin 2 Arduina preko *pull-up* otpornika iznosa 4.7 kΩ, kao i na +5V i GND (pozitivni polaritet napona i „masa“), te očitavaju trenutnu temperaturu. Očitana temperatura sa sva tri senzora se ispisuje pojedinačno na LCD 16x2 zaslon (4) preko I2C pretvarača (3). I2C ili IIC pretvarač (eng. *I2C Converter Board*) je zalemljen za LCD te komunicira sa Arduinom putem analognih pinova A4 i A5 (SDA i SCL), te se iz tog razloga i koristi, jer koristi samo 2-žičnu komunikaciju i ostavlja druge pinove slobodne za upotrebu. Navigacija kroz izbornik i postavljanje željenih vrijednosti se vrši pomoću tipkala (5), kojih ima pet i spojeni su na digitalne pinove Arduina 3,4,5,6, i 7. Tipkala su također spojena na +5V, kao i na GND preko *pull-up* otpornika iznosa 10 kΩ. Tri LED diode (6) simuliraju paljenje i gašenje rashladnog sustava, ovisno o trenutnoj temperaturi koju očitavaju senzori temperature. Spojene su na digitalne pinove 8,9 i 10 preko otpornika iznosa 330 Ω, kao i na GND.

4.2. Električna shema

Ukoliko bi bilo potrebno realizirati rashladni sustav ponovno, ili u više primjeraka, shema povezivanja Arduino razvojne platforme sa električnim komponentama bi izgledala kao na slici 4.2. koja predstavlja električnu shemu spoja rashladnog sustava, izrađenu u programu Eagle.[10] Elementi sheme sa slike 4.2. su sljedeći:

- B1 - Arduino Uno R3 razvojna pločica sa ATmega328P mikrokontrolerom
- U1,U2 i U3 - Senzori temperature DS18B20
- DIS1 - Zaslon LCD 16x2 spojen na Arduino preko zalemljenog I2C pretvarača
- S1, S2, S3, S4 i S5 - Tipkala za navigaciju kroz izbornik i postavljanje željenih vrijednosti
- LED1, LED2, LED3 – LED diode koje simuliraju paljenje i gašenje rashladnog sustava
- R1, R2, R3 – Otpornici spojeni u seriji sa LED diodama zbog ograničavanja struje koja protječe kroz diode
- R4, R5, R6, R7, R8 – Pull up otpornici preko kojih spajamo tipkala S1-S5
- R9 – Pull up otpornik za senzore temperature U1-U3, spaja se između DATA pina senzora i +5V



Slika 4.2. Električna shema spoja rashladnog sustava

4.3. Kućište

Nakon što je sklop testiran na eksperimentalnoj pločici, te zadovoljava pri obavljanju zadanih funkcija, uslijedila je izrada kućišta, njegovo smještanje u kućište i lemljenje komponenti. Za kućište je korištena razvodna kutija izrađena od materijala ABS (akrilonitril-butadien stirel). ABS je čvrsta i žilava plastika, koja ima veliku otpornost na korozivne kemikalije i na fizičke utjecaje. Ove osobine dovode do toga da se koristi za proizvodnju širokog spektra proizvoda, primjerice, za izradu kućišta za računala, monitore, pisače, tipkovnice, zidne utičnice, automobilske branike i dr. Slika 4.3. pokazuje razvodnu kutiju koja će poslužiti kao temelj za izradu kućišta u koje će biti smješten rashladni sustav sa svim svojim elektroničkim komponentama.

Slika 4.3. Razvodna kutija za izradu kućišta

Kućište se prilagođava kako bi LCD, tipkala i LED diode bile na vidljivom položaju čime se olakšava očitavanje temperature, kao i navigacija kroz izbornik i postavljanje željenih vrijednosti. Također LED diode se postavljaju na vidljivo mjesto kako bi lakše primjetili simulaciju paljenja i gašenja rashladnog sustava. Te tri komponente su postavljene na poklopac kutije. Sa strane kutije su postavljeni izlazi za napajanje Arduina kao i izlazi za senzore temperature. Pri postavljanju LCD-a, LED dioda, kabla za napajanje, te senzora bilo je potrebno samo izrezati utore za iste, na razvodnoj kutiji. Međutim, tipkala se prilikom postavljanja moralo zalemiti na bušenu univerzalnu tiskanu pločicu (eng. *stripboard*), koja se zatim zalijepila s unutrašnje strane poklopca kutije. Nakon izvršenih preinaka na razvodnoj kutiji, pristupilo se smještanju elektroničkih komponenti u prilagođeno kućište, testiranju sustava te spajanju i lemljenu komponenti. Naposljetku se kućište zatvara pomoću vijaka, dovodi se napajanje te se rashladni sustav može uključiti. Slika 4.4. pokazuje završni proizvod, rashladni sustav smješten u kućište koje sadrži LCD zaslon na prednjoj strani, ispod njega tipkala S (SELECT), L (LEFT), R (RIGHT), PLUS i MINUS za navigaciju kroz izbornik i postavljanje željenih vrijednosti, iznad zaslona se nalaze LED diode koje simuliraju paljenje i gašenje rashladnog sustava. Također na slici 4.4. se vide i tri senzora temperature DS18B20, presvučena PVC-om (poli-vinil klorid) kako bi se osigurala vodootpornost.

Slika 4.4. Rashladni sustav smješten u kućište

5. PRINCIP RADA RASHLADNOG SUSTAVA

Prilikom prvog pokretanja sustava započinje inicijalizacija, postavljanje određenih varijabli na zadane početne vrijednosti te se na zaslonu ispisuju trenutne temperature koje mjere sva tri senzora, a koje se mjenjaju ovisno o promjeni temperature koju senzori detektiraju (Sl.5.1.).

Slika 5.1. Početni zaslon s trenutim vrijednostima temperature

5.1. Temperatura paljenja hlađenja

Pritiskom na tipkalo „S“ dok je sustav na početnom zaslonu se ulazi u prvi izbornik (Sl. 5.2.). U prvom izborniku postavljamo temperaturu na kojoj će sustav otvorit ventil kroz koji će proteći hladna voda, odnosno temperaturu na kojoj će kada se dostigne započeti hlađenje. Temperatura se postavlja za 1. senzor, a mjenja se pritiskanjem „+“ ili „-“ minus tipkala, što rezultira povećanjem ili smanjenjem zadane temperature za 0.1 $^{\circ}C$.

Slika 5.2. Postavljanje temperature na kojoj će se paliti hlađenje za 1. senzor

Pritiskom na tipkala „L“ i „R“ dok je sustav u prvom izborniku se prebaciva na 2. i 3. senzor, tako da pritiskom na tipkalo „R“ se dolazi na postavljanje temperature hlađenja za 2. senzor, kao što se vidi na slici 5.3. Postavljanje željene temperature se vrši na isti način kao i za 1. senzor, pritiskom na „+“ i „-“ tipkala, što povećava ili smanjuje zadanu temperaturu za 0.1 $^{\circ}C$.

Slika 5.3. Postavljanje temperature na kojoj će se paliti hlađenje za 2. senzor

Pritiskom na tipkalo „R“ se dolazi do 3. senzora, naravno vrijedi i obrnuto, ako bi se pritisnulo tipkalo „L“ vratili bi se na 1. Za 3. senzor vrijedi sve isto kao i za prva dva, pritiskanjem „+“ i „-“ tipkala se povećava, odnosno smanjuje željena temperatura za 0.1 $^{\circ}C$. Slika 5.4. pokazuje temperaturu paljenja hlađenja za 3. senzor.

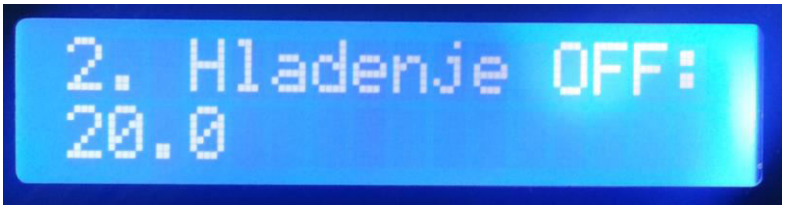
Slika 5.4. Postavljanje temperature na kojoj će se paliti hlađenje za 3. senzor

5.2. Temperatura gašenja hlađenja

Nakon što je postavljena temperatura na kojoj će se paliti hlađenje kad bude dostignuta, sljedeće na redu je postavljanje temperature na kojoj će se isključiti hlađenje, odnosno donji prag. To se postiže pritiskom na „S“ tipkalo dok je sustav u prvom izborniku.

Slika 5.5. Postavljanje temperature na kojoj će se gasiti hlađenja za 1. senzor

Slika 5.5. pokazuje izbornik za postavljanje temperature na kojoj će se gasiti hlađenje za 1. senzor. Temperatura se postavlja identično kao i u prošlim slučajevima, pomoću tipkala „+“ i „-“, što rezultira povećanjem ili smanjenjem temperature za 0.1 $^{\circ}C$. Navigacija između senzora je identična kao i u prošlom slučaju, što znači da pritiskom na tipkalo „R“ se dolazi u postavke 2. senzora, kao što slika 5.6. pokazuje.



Slika 5.6. Postavljanje temperature na kojoj će se gasiti hlađenja za 2. senzor

Nakon što je namještena željena temperatura gašenja hlađenja za 2. senzor, ponovnim pritiskom na tipkalo „R“ naposljetku se dolazi do 3. senzora. Slika 5.7. pokazuje postavke temperature pri kojoj će se gasiti hlađenje za 3. senzor. Željena temperatura se kao i uvijek namješta pomoću „+“ i „-“ tipkala. Navigacija prema ostalim senzorima je moguća u svakom trenutku pomoću „L“ i „R“ tipkala.

Slika 5.7. Postavljanje temperature na kojoj će se gasiti hlađenja za 3. Senzor

5.3. Detekcija naglog porasta temperature

Kako je već ranije navedeno, prilikom alkoholnog vrenja mošta može doći do burne fermentacije u kojoj temperatura mošta nekontrolirano raste i u takvim situacijama ju je jako teško obuzdati i ohladiti mošt, posebice u spremnicima velike zapremine. U takvom scenariju mošt bi proveo previše vremena na visokoj temperaturi dok ga sustav hlađenja ohladi, što direktno utječe na kvalitetu vina. Kako bi takav scenarij bio spriječen, u rashladni sustav je implementirana detekcija naglog porasta temperature, odnosno temperaturnog gradijenta, koja bi trebala primjetiti nagli porast temperature ovisno o zadanim parametrima, te upaliti hlađenje i prije nego što temperatura dođe do vrijednosti koja je zadana za paljenje hlađenja. Funkcionira na način da pritiskom na tipkalo „S“ dok je sustav u drugom izborniku, odnosno u postavkama za postavljanje temperature na kojoj će se gasiti hlađenje, se ulazi u treći izbornik, izbornik za postavljanje parametara za detekciju naglog porasta temperature. U tom izborniku postoje tri parametra koja je potrebno postaviti:

- Vremenski raspon u kojem će se detektirati nagli porast temperature - minuta, sat ili bilo koja druga proizvoljna vrijednost
- Temperaturni skok, odnosno vrijednost za koju će temperatura porasti u prethodno zadanom vremenskom rasponu – 1 $^{\circ}C$, 2 $^{\circ}C$ ili bilo koji drugi iznos.
- Vrijeme hlađenja, odnosno vrijeme koje označava koliko dugo će raditi rashladni sustav ako se upali prilikom naglog porasta temperature – minuta, sat ili bilo koja druga proizvoljna vrijednost.

Slika 5.8. Vremenski raspon za detekciju naglog porasta temperature

Prilikom ulaska u izbornik za postavljanje parametara za detekciju naglog porasta temperatura, prva vrijednost koju treba postaviti je vremenski raspon (Sl. 5.8.). Željena vrijednost se postavlja tipkalima „+“ i „-“, što povećava vrijednost za 1 sekundu, odnosno smanjuje. Za tu vrijednost se odabire vremenski raspon u kojem će se pratiti porast temperature i taj parametar se postavlja u skladu sa drugim parametrom, temperaturnim skokom. Na postavke temperaturnog skoka se prebaciva pritiskom na tipkalo „R“, dok je sustav u postavkama vremenskog raspona (Sl. 5.9.).

Slika 5.9. Temperaturni skok naglog porasta temperature

Detekcija naglog porasta temperature funkcionira na način, da se odabere vremenski raspon (Sl.5.8.) u kojem će temperatura porasti za broj stupnjeva celzijusa, koji je postavljen parametrom temperaturni skok (Sl. 5.9.). Primjerice, ako se u vremenskom rasponu postavi vrijeme iznosa 1 minute, a u temperaturnom skoku se postavi temperatura iznosa 3 $^{\circ}C$. U tom slučaju, ako temperatura mošta poraste unutar jedne minute za 3 $^{\circ}C$ ili više, upalit će se sustav hlađenja, neovisno o temperaturi koja je zadana kao temperatura na kojoj će se upaliti hlađenje. Ako temperatura raste polako, i ne aktivira algoritam za detekciju naglog porasta temperature, tada će se hlađenje upaliti kada temperatura dostigne vrijednost postavljenu u prvom izborniku.

Ukoliko je hlađenje upaljeno detekcijom naglog porasta temperature, tada će raditi onoliko dugo koliko je postavljeno trećim parametrom, vremenom hlađenja. Slika 5.10. pokazuje postavke vremena hlađenja, u koje se ulazi pritiskom na tipkalo „R“ dok je sustav u postavkama temperaturnog skoka.

Slika 5.10. Vrijeme hlađenja u izborniku naglog porasta temperature

Vrijeme hlađenja se postavlja pritiskom na tipkala „+“ i „-“, što rezultira povećanjem, odnosno smanjenjem vrijednosti za 1 minutu. Vrijeme hlađenja je parametar koji označava koliko dugo će sustav hlađenja raditi ukoliko je aktiviran naglim porastom temperature. Primjerice, ako postavimo parametar vrijeme hlađenja na 5 minuta i sustav hlađenja je aktiviran naglim porastom temperature, hlađenje će raditi 5 minuta i isključiti se. No valja napomenuti, ako se unutar tih 5 minuta temperatura spusti ispod vrijednosti koja je postavljena u drugom izborniku (temperatura na kojoj se gasi hlađenje), tada će se hlađenje isključiti i prije nego istekne tih 5 minuta. To je napravljeno iz razloga da nema preklapanja, u prvom izborniku se postavlja temperatura na kojoj se bezuvjetno pali hlađenje, u drugom izborniku se postavlja temperatura na kojoj se bezuvjetno gasi hlađenje, tako da detekcija naglog porasta temperature funkcionira samo unutar te dvije vrijednosti, odnosno između njih, što je i logično.

6. PROGRAMSKA REALIZACIJA RASHLADNOG SUSTAVA

Kao što je već navedeno, prilikom prvog pokretanja sustava na zaslonu se ispisuju trenutne temperature koje mjere sva tri senzora, a koje se mjenjaju ovisno o promjeni temperature koju senzori detektiraju. Slika 6.1. pokazuje programski kod za očitavanje i dohvaćanje trenutne temperature sa sva tri senzora, kao i njihovo ispisivanje na LCD.

```
1   float temp1 = 0;
2   float temp2 = 0;
3   float temp3 = 0;
4
5   void setup(void)
6   {
7     Serial.begin(9600);
8     sensors.begin();
9
10    lcd.begin (16,2);
11    lcd.setBacklightPin(3,POSITIVE);
12    lcd.setBacklight(HIGH);
13
14    sensors.requestTemperatures();
15    temp1 = sensors.getTempCByIndex(0);
16    temp2 = sensors.getTempCByIndex(1);
17    temp3 = sensors.getTempCByIndex(2);
18    t1 = temp1;
19    t2 = temp2;
20    t3 = temp3;
21  }
22  void loop(void)
23  {
24    sensors.requestTemperatures();
25    temp1 = sensors.getTempCByIndex(0);
26    temp2 = sensors.getTempCByIndex(1);
27    temp3 = sensors.getTempCByIndex(2);
28
29    lcd.setCursor (2,0);
30    lcd.print("Temperatura:");
31    lcd.setCursor (0,1);
32    lcd.print(temp1,1);
33    lcd.setCursor (6,1);
34    lcd.print(temp2,1);
35    lcd.setCursor (12,1);
36    lcd.print(temp3,1);
37    delay(500);
38  }
```

Slika 6.1. Dohvaćanje i ispis trenutne temperature

Kao što slika 6.1. pokazuje, kreirane su varijable *temp1*, *temp2* i *temp3* tipa *float* koje će sadržavati očitanu vrijednost temperature. Pomoću *sensors.begin();* se učitava biblioteka temperaturnog senzora, kao i biblioteka LCD zaslona pomoću *lcd.begin(16,2);* gdje su u zagradi navedene dimenzije zaslona. Nakon što su učitane biblioteke i pokrenuti senzor i LCD, pozivanjem *sensors.requestTemperatures();* se šalje globalni zahtjev za temperaturama svim senzorima spojenim na isti pin Arduina, u ovom slučaju na sva tri senzora.

Nakon zahtjeva za temperaturama, pozivanjem *sensors.getTempCByIndex(0);* se dohvaća temperatura i sprema u kreirane varijable, a u zagradi se nalazi indeks pomoću kojeg razlikujemo spojene senzore, u ovom slučaju su indeksi 0,1 i 2 za sva tri senzora. Kada su sve temperature dohvaćene, sa *lcd.setCursor();* i *lcd.print();* se postavlja pokazivač na željeno mjesto na LCD zaslonu i ispisuju varijable koje sadrže trenutne temperature. Nakon izvršavanja iznad opisanog koda, na LCD zaslonu na sklopu se prikazuje trenutna temperatura koju mjere sva tri senzora.

Kako bi bilo moguće postavljati temperature uključivanja hlađenja, gašenja i druge postavke, potrebno je realizirati navigaciju kroz izbornike (Sl. 6.2.). To je realizirano na sljedeći način, *digitalRead(buttonSelectPin);* provjerava stanje „S“ tipkala i sprema ga u varijablu *buttonSelectState,*ako je pritisnuto (HIGH) sustav ulazi u prvi izbornik, izbornik za postavljanje temperature pri kojoj se pali hlađenje.

```
1   void setup(void)
2   {
3     Serial.begin(9600);
4
5     pinMode(buttonSelectPin, INPUT);
6     pinMode(buttonLeftPin, INPUT);
7     pinMode(buttonRightPin, INPUT);
8     pinMode(buttonUpPin, INPUT);
9     pinMode(buttonDownPin, INPUT);
10
11    pinMode(ledPin1, OUTPUT);
12    pinMode(ledPin2, OUTPUT);
13    pinMode(ledPin3, OUTPUT)
14  }
15  void loop(void)
16  {
17   buttonSelectState = digitalRead(buttonSelectPin);
18    if (buttonSelectState == HIGH) {
19      lcd.clear();
20      trigger = 0;
21      delay(1000);
22      while(trigger != 1){
23
24        buttonSelectState = digitalRead(buttonSelectPin);
25        if (buttonSelectState == HIGH) {
26          delay(500);
27          trigger = 1;
28        }
29        buttonLeftState = digitalRead(buttonLeftPin);
30        if (buttonLeftState == HIGH) {
31          delay(500);
32          trig = trig - 1;
33        }
34        buttonRightState = digitalRead(buttonRightPin);
35        if (buttonRightState == HIGH) {
36          delay(500);
37          trig = trig + 1;
38        }
39        if (trig > 2){
40          trig = 0;
41        }
42        if (trig < 0){
43          trig = 2;
44        }
45  }
```

Slika 6.2. Ulazak u izbornik za postavljanje temperature uključivanja hlađenja

While petlja zadržava taj izbornik, sve dok se ponovo ne pritisne tipkalo „S“. U tom izborniku su postavke za sva tri senzora, a pomicanje prikaza s jednog na drugi se vrši tipkalima „L“ ili „R“ kako je već objašenjeno ranije, a to se vrši na način da se očitava stanje „L“ i „R“ tipkala pomoću *digitalRead(buttonLeftPin); i digitalRead(buttonRightPin);* te se ta stanja spremaju u varijable *buttonLeftState* i *buttonRightState.* Pomoćna varijabla *trig* služi za razlikovanje senzora i pomicanje prikaza s jednog na drugi. Kada je prikaz na posljednjem senzoru ponovnim pritiskom na tipkalo „R“ se dolazi na prvi, to je omogućeno uvjetima sa slike 6.2., linije koda od 39 do 44. Ako je prikaz na željenom senzoru, može se postaviti željenu temperaturu sa „+“ i „-“.

```
1   if (trig == 0){
2     buttonUpState = digitalRead(buttonUpPin);
3     if (buttonUpState == HIGH){
4       borderTemp1 = borderTemp1 + 0.1;
5       lcd.setCursor (0,1);
6       lcd.print(borderTemp1,1);
7       delay(500);
8     }
9     buttonDownState = digitalRead(buttonDownPin);
10    if (buttonDownState == HIGH){
11      if(lowTemp1 < borderTemp1){
12        borderTemp1 = borderTemp1 - 0.1;
13      }
14      lcd.setCursor (0,1);
15      lcd.print(borderTemp1,1);
16      delay(500);
17    }
18    lcd.home ();
19    lcd.print("1. Hladenje ON:");
20    lcd.setCursor (0,1);
21    lcd.print(borderTemp1,1);
22    lcd.setCursor (15,1);
23    lcd.print("C");
24  }
25
26  if (trig == 1){
27    buttonUpState = digitalRead(buttonUpPin);
28    if (buttonUpState == HIGH){
29      borderTemp2 = borderTemp2 + 0.1;
30      lcd.setCursor (0,1);
31      lcd.print(borderTemp2,1);
32      delay(500);
33    }
34    buttonDownState = digitalRead(buttonDownPin);
35    if (buttonDownState == HIGH){
36      if(lowTemp2 < borderTemp2){
37        borderTemp2 = borderTemp2 - 0.1;
38      }
39      lcd.setCursor (0,1);
40      lcd.print(borderTemp2,1);
41      delay(500);
42    }
43    lcd.home ();
44    lcd.print("2. Hladenje ON:");
45    lcd.setCursor (0,1);
46    lcd.print(borderTemp2,1);
47    lcd.setCursor (15,1);
48    lcd.print("C");
49  }
50
51  if (trig == 2){
52    buttonUpState = digitalRead(buttonUpPin);
53    if (buttonUpState == HIGH){
54      borderTemp3 = borderTemp3 + 0.1;
55      lcd.setCursor (0,1);
56      lcd.print(borderTemp3,1);
57      delay(500);
58    }
59    buttonDownState = digitalRead(buttonDownPin);
60    if (buttonDownState == HIGH){
61      if(lowTemp3 < borderTemp3){
62        borderTemp3 = borderTemp3 - 0.1;
63      }
64      lcd.setCursor (0,1);
65      lcd.print(borderTemp3,1);
66      delay(500);
67    }
68    lcd.home ();
69    lcd.print("3. Hladenje ON:");
70    lcd.setCursor (0,1);
71    lcd.print(borderTemp3,1);
72    lcd.setCursor (15,1);
73    lcd.print("C");
74  }
```

Slika 6.3. Postavljanje temperature uključivanja hlađenja

Slika 6.3. pokazuje kod kojim je realizirano postavljanje željene temperature pri kojoj će se uključiti hlađenje, a funkcionira na način da se provjerava varijabla *trig* pomoću koje se razlikuju senzori (0,1 i 2). Ako je prikaz na željenom senzoru, očitava se stanje „+“ i „-“ tipkala pomoću *digitalRead(buttonUpPin);* i *digitalRead(buttonDownPin);* te se njihovo stanje sprema u varijable *buttonUpState* i *buttonDownState.* Ako je pritisnuto tipkalo „+“, stanje varijable *buttonUpState* je HIGH i varijabla *borderTemp* se povećava za 0.1 $^{\circ}C$, i tako svakim pritiskom na tipkalo. Ako je pritisnuto tipkalo „-“, stanje varijable *buttonDownState* je HIGH i varijabla *borderTemp* se snižava za 0.1 $^{\circ}C$. Varijabla *borderTemp* je varijabla koja označava temperaturu na kojoj se uključuje hlađenje i početna joj je vrijednost 30 $^{\circ}C$, no može se staviti bilo koja. Postupak postavljanja temperature je isti za sva tri senzora, te svaka promjena temperature se registrira na LCD zaslonu, što je vidljivo u kodu na slici 6.3.

Nakon postavljanja temperature uključivanja hlađenja, pritiskom na tipkalo „S“ se ulazi u sljedeći izbornik, izbornik u kojem se postavlja temperatura isključivanja hlađenja. Princip je potpuno isti kao u prošlom slučaju, očitavaju se stanja „L“ i „R“ i spremaju u varijable *buttonLeftState* i *buttonRightState* čime se prikaz prebacuje s jednog senzora na sljedeći (Sl. 6.4.). Pomoćna varijabla *trig* služi za razlikovanje senzora i prebacivanje prikaza s jednog na drugi, a *trigger* da je to drugi izbornik.

```
237     delay(1000);
238     while(trigger != 2){
239
240       buttonSelectState = digitalRead(buttonSelectPin);
241       if (buttonSelectState == HIGH) {
242         delay(500);
243         trigger = 2;
244       }
245
246       buttonLeftState = digitalRead(buttonLeftPin);
247       if (buttonLeftState == HIGH) {
248         delay(500);
249         trig = trig - 1;
250       }
251
252       buttonRightState = digitalRead(buttonRightPin);
253       if (buttonRightState == HIGH) {
254         delay(500);
255         trig = trig + 1;
256       }
257
258       if (trig > 2){
259         trig = 0;
260       }
261       if (trig < 0){
262         trig = 2;
263     }
```

Slika 6.4. Ulazak u izbornik za postavljanje temperature isključivanja hlađenja

Postavljanje željene temperature pri kojoj se hlađenje gasi je identično postavljanju temperature pri kojoj se hlađenje pali. Na slici 6.5. je prikazan kod za prvi senzor, no sve je identično i za ostala dva. Očitava se stanje tipkala „+“ i „-“ i sprema u varijable *buttonUpState* i *buttonDownState*. Ako je jedno od ta dva tipkala pritisnuto varijabli *lowTemp1* se dodaje ili oduzima vrijednost 0.1 što rezultira povećanjem ili smanjenjem temperature na kojoj se isključuje hlađenje za 0.1 $^{\circ}C$.

```
267     if (trig == 0){
268       buttonUpState = digitalRead(buttonUpPin);
269       if (buttonUpState == HIGH){
270         if(lowTemp1 < borderTemp1){
271           lowTemp1 = lowTemp1 + 0.1;
272         }
273         lcd.setCursor (0,1);
274         lcd.print(lowTemp1,1);
275         delay(500);
276       }
277       buttonDownState = digitalRead(buttonDownPin);
278       if (buttonDownState == HIGH){
279         lowTemp1 = lowTemp1 - 0.1;
280         lcd.setCursor (0,1);
281         lcd.print(lowTemp1,1);
282         delay(500);
283       }
284       lcd.home ();
285       lcd.print("1. Hladenje OFF:");
286       lcd.setCursor (0,1);
287       lcd.print(lowTemp1,1);
288       lcd.setCursor (15,1);
289       lcd.print("C");
290     }
291
```

Slika 6.5. Postavljanje temperature isključivanja hlađenja

Nakon postavljanja temperature isključivanja rashladnog sustava, ponovnim pritiskom na tipkalo „S“ se ulazi u treći i posljednji izbornik, koji sadrži postavke za nagli porast temperature. Te postavke su temperaturni skok ($^{\circ}C$), vremenski raspon (sekunde) i vrijeme hlađenja (minute).

```
383     if (trig == 0){
384       buttonUpState = digitalRead(buttonUpPin);
385       if (buttonUpState == HIGH){
386         timeDiff = timeDiff + 1;
387         lcd.setCursor (0,1);
388         lcd.print(timeDiff);
389         delay(500);
390       }
391       buttonDownState = digitalRead(buttonDownPin);
392       if (buttonDownState == HIGH){
393         if (buttonDownState == HIGH and timeDiff == 10){
394           lcd.clear();
395         }
396         timeDiff = timeDiff - 1;
397         lcd.setCursor (0,1);
398         lcd.print(timeDiff);
399         delay(500);
400       }
401       lcd.home ();
402       lcd.print("Vremenski razmak");
403       lcd.setCursor (0,1);
404       lcd.print(timeDiff);
405       lcd.setCursor (13,1);
406       lcd.print("sec");
407     }
```

Slika 6.6. Postavljanje vremenskog raspona za nagli porast temperature

Postavljanje vremenskog raspona u kojem će se pratiti nagli porast temperature funkcionira na isti način kao i postavljanje temperature u prošlim slučajevima. Pritiskom na tipkala „+“ ili „-“ se mjenja stanje *buttonUpState* i *buttonDownState* varijabli, što povećava ili smanjuje varijablu *timeDiff* za 1. To rezultira povećanjem ili smanjenjem vremenskog raspona za 1 sekundu (Sl.6.6). Postavljanje temperaturnog skoka i vremena hlađenja funkcionira na isti način. Kod temperaturnog skoka se povećava ili smanjuje varijabla *tempDiff* za $0.1\,^{\circ}C$, što rezultira povećanjem ili smanjenjem iznosa temperature za koji trenutna temperatura mora porast u vremenskom rasponu da bi aktivirala hlađenje. Postavljanje vremena hlađenja je također isto implementirano,pritiskom na tipkala varijabla *coolingTime* se povećava ili smanjuje za 1 minutu.

Nakon objašnjenja koda za postavljanje željenih vrijednosti, na slici 6.7. se nalazi kod koji provjerava trenutnu temperaturu i uključuje hlađenje (u ovom slučaju LED-ice za potrebe simulacije) ako je trenutna temperatura veća od postavljene za sve tri posude. Ako su varijable *temp1*, *temp2* ili *temp3* veće ili jednake od postavljeni vrijednosti za uključivanje hlađenja *borderTemp1*, *borderTemp2* i *borderTemp3* upalit će se LED-ice koje simuliraju otvaranje ventila i cirkulacijskog kruga. Ako se hlađenje uključilo, radit će sve dok vrijednost trenutne temperature ne padne ispod postavljene temperature gašenja hlađenja, koju predstavljaju varijable *lowTemp1*, *lowTemp2* i *lowTemp3*, kada će se i LED-ice ugasit, što bi značilo zatvaranja ventila i cirkulacijskog kruga.

```
473  if (temp1 >= borderTemp1){
474    digitalWrite(ledPin1, HIGH);
475    cool1 = 0;
476  }
477
478  if (temp2 >= borderTemp2){
479    digitalWrite(ledPin2, HIGH);
480    cool2 = 0;
481  }
482
483  if (temp3 >= borderTemp3){
484    digitalWrite(ledPin3, HIGH);
485    cool3 = 0;
486  }
487
488
489
490  if (temp1 <= lowTemp1){
491    digitalWrite(ledPin1, LOW);
492  }
493
494  if (temp2 <= lowTemp2){
495    digitalWrite(ledPin2, LOW);
496  }
497
498  if (temp3 <= lowTemp3){
499    digitalWrite(ledPin3, LOW);
500  }
```

Slika 6.7. Uključivanje i isključivanje hlađenja

Detekcija naglog porasta temperature se vrši prema kodu na slici 6.8., a funkcionira na način da je prilikom definiranja varijabli definirana varijabla *k*, koja ima vrijednost *k= timeDiff / 0.5* i ona predstavlja odbrojavanje zadanog vremenskog raspona. Prilikom prvog prolaska kroz kod trenutna temperatura se sprema u varijable *t1*, *t2*, i *t3* i kada je odbrojano uzima se trenutna vrijednost temperature od koje se oduzme vrijednost prve spremljene temperature, te usporedi s temperaturnim skokom. Ako je ta razlika veća od *tempDiff*, odnosno zadanog temperaturnog skoka, pali se hlađenje, tj. LED dioda. Varijabla *k* predstavlja koliko puta treba proći kroz petlju, i svaki prolazak je smanjuje. Kada dođe do 0, tada se gleda razlika trenutne vrijednosti temperature od prvotne i uspoređuje s vremenskim skokom, a *k* se resetira. Jedan *k* predstavlja pola sekunde. Varijable *cool1=1*, *cool2=1*, *cool3=1* govore da je upaljeno hlađenje.

```
517   k = k - 1;
518   if (k <= 0){
519     if ((temp1 - t1) >= tempDiff){
520       digitalWrite(ledPin1, HIGH);
521       cool1 = 1;
522     }
523
524     if ((temp2 - t2) >= tempDiff){
525       digitalWrite(ledPin2, HIGH);
526       cool2 = 1;
527     }
528
529     if ((temp3 - t3) >= tempDiff){
530       digitalWrite(ledPin3, HIGH);
531       cool3 = 1;
532     }
533
534     t1 = temp1;
535     t2 = temp2;
536     t3 = temp3;
537     k = timeDiff / 0.5;
538   }
```

Slika 6.8. Detekcija naglog porasta temperature

Ako su varijable *cool1*, *cool2* i *cool3* jednake jedan, to znači da je došlo do paljenja hlađenja usred naglog porasta temperature, i tada se gledaju dva slučaja. Prvi je da se upalilo hlađenje zbog naglog porasta temperature ali nije došlo do granične vrijednosti paljenja hlađenja koje smo zadali, u tom slučaju hlađenje će raditi samo onoliko vremena koliko smo postavili u izborniku. Drugi slučaj je da je hlađenje aktivirano naglim porastom temperature i ona je narasla iznad granične vrijednosti postavljene, tada se preskače kod s vremenom trajanja hlađenja i hlađenje radi sve dok ne spusti temperaturu do vrijednosti koja je postavljena za isključenje hlađenja, *lowTemp*.Varijable *c1*, *c2* i *c3* predstavljaju odbrojavanje koliko treba biti upaljeno hlađenje. Slika 6.9. prikazuje provjeru ovih uvjeta.

```
540   if (cool1 == 1){
541     c1 = c1 - 1;
542     if (c1 <= 0 and temp1 < borderTemp1){
543       digitalWrite(ledPin1, LOW);
544       cool1 = 0;
545       c1 = coolingTime*60 / 0.5;
546     }
547     else if (c1 <= 0){
548       c1 = coolingTime*60 / 0.5;
549     }
550   }
551
552   if (cool2 == 1){
553     c2 = c2 - 1;
554     if (c2 <= 0 and temp2 < borderTemp2){
555       digitalWrite(ledPin2, LOW);
556       cool2 = 0;
557       c2 = coolingTime*60 / 0.5;
558     }
559     else if (c2 <= 0){
560       c2 = coolingTime*60 / 0.5;
561     }
562   }
563
564   if (cool3 == 1){
565     c3 = c3 - 1;
566     if (c3 <= 0 and temp2 < borderTemp2){
567       digitalWrite(ledPin3, LOW);
568       cool3 = 0;
569       c3 = coolingTime*60 / 0.5;
570     }
571     else if (c3 <= 0){
572       c3 = coolingTime*60 / 0.5;
573     }
574   }
575
576   if (cool1 == 0){
577     c1 = coolingTime*60 / 0.5;
578   }
579
580   if (cool2 == 0){
581     c2 = coolingTime*60 / 0.5;
582   }
583
584   if (cool3 == 0){
585     c3 = coolingTime*60 / 0.5;
586   }
```

Slika 6.9. Vrijeme trajanja hlađenja ako je aktivirano naglim porastom temperature

7. ZAKLJUČAK

Rashladni sustavi su jedna od najbitnijih stavki u mnogim industrijskim procesima, tako i u prehrambenoj industriji gdje se koriste u razne svrhe. Jedna od tih svrha je i hlađenje prilikom proizvodnje vina, odnosno rashladni sustav koji održava željenu temperaturu mošta pri alkoholnoj fermentaciji. U ovom radu je opisana alkoholna fermentacija i njezine značajke, kao i temperature pri kojoj se odvija, te je opisana izrada rashladnog sustava koji bi se mogao koristiti u tu svrhu. Sustav je izrađen pomoću razvojne pločice Arduino Uno Rev3 koja koristi mikrokontroler ATmega328P, cijenom prihvatljiv način regulacije koji pruža popriličan broj mogućnosti. Jedna od najvećih prednosti Arduina je otvoreni kod, i upotreba biblioteka koje korisnici izrađuju za pojedine komponente i objavljuju na internetu besplatno, te znatno olakšavaju programiranje. Bitna stvar je i da su sve komponente nabavljene u Hrvatskoj, što govori kako je Arduino platforma dosta popularna, i da sklopovi i sustavi bazirani na njoj imaju sve veću primjenu i u Hrvatskoj.

Nakon nabave svih elektroničkih komponenata pristupilo se izradi sheme i spajanju komponenti. Kada je sve bilo spojeno i provjereno krenulo se sa realizacijom programskog koda, zaduženim za upravljanje sustavom. U kod su se prvo uključile sve biblioteke zadužene za rad s pojedinom komponentom, te su se definirale korištene varijable. Korištenjem varijabli i ključnih riječi (eng. *keywords*) iz biblioteka realizirale su se funkcije za pojedine elemente, koje su obavljale željene zadaće. Nakon što je kod napisan i optimiziran za rad sa izrađenim sklopom, pristupilo se izradi kućišta u koje je sklop smješten, koji je prikazan u diplomskom radu i opisan princip rada. Rashladni sustav, kojeg čine sklop i programski kod, je pružio zadovoljavajuće rezultate pri simuliranju rada, no ako bi bilo potrebe za njegovom realizacijom kao gotovog proizvoda, i upotrijebilo u okruženju koje je opisano u uvodu, trebalo bi ga unaprijediti. Unaprijeđenje bi bilo na način da se na sustav spoje relej i ventili, kako bi relejom upravljali otvaranjem i zatvaranjem ventila, koji služe za otvaranje i zatvaranje cirkulacijskog kruga.

8. LITERATURA

[1] B. Pavković, Tehnika hlađenja, Tehnički fakultet, Rijeka

http://www.riteh.uniri.hr/zav_katd_sluz/zvd_teh_term_energ/nas/tehnika_hladjenja_07/00_Sadrzaj_popis_literature.pdf [18.9.2017.]

[2] S. Muštović, Alkoholno vrenje mošta (Alkoholna fermentacija)

http://www.vinogradarstvo.com/preporuke-i-aktualni-savjeti/aktualni-savjeti-vinarstvo/za-one-koji-zele-znati-nesto-vise/418-pretakanje-vina [13.9.2017.]

[3] A. Jorgensen, Micro-organisms and fermentation, 1911, Fascimile Publisher, Amazon, 2015.

https://archive.org/details/microorganismsan017179mbp [13.9.2017.]

[4] Arduino Uno R3 Specifications Datasheet

https://www.farnell.com/datasheets/1682209.pdf [18.9.2017.]

[5] Arduino Software (IDE)

https://www.arduino.cc/en/Guide/Environment [18.9.2017.]

[6] Programmable Resolution 1-Wire Digital Thermometer

https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf [18.9.2017.]

[7] Hitachi HD44780U (LCD-II) Datasheet

https://www.sparkfun.com/datasheets/LCD/HD44780.pdf [18.9.2017.]

[8] PCF8574 - Remote 8-bit I/O expander for I2C-bus Datasheet

https://www.nxp.com/docs/en/data-sheet/PCF8574_PCF8574A.pdf [18.9.2017.]

[9] Fritzing 0.9.3b

http://fritzing.org/download/ [18.9.2017]

[10] Autodesk Eagle 8.3.2

https://www.autodesk.com/products/eagle/free-download [18.9.2017]

9. SAŽETAK

IZRADA SUSTAVA ZA NADZOR I UPRAVLJANJE RASHLADNIM PROCESOM

Ovaj rad opisuje važnost rashladnih sustava, te hlađenje pri fermantaciji prilikom proizvodnje vina. Opisan je i postupak izrade rashladnog sustava, kojemu je zadaća nadzor temperature i upravljanje hlađenjem tijekom fermentacije. Spomenute su i ukratko opisane tehnologije i električne komponente koje su se koristile prilikom izrade sustava, kao i princip rada. Uz princip rada su priložene slike koje pokazuju navigaciju kroz izbornike sustava, kao i mogućnosti sustava. Ukratko su objašnjeni i najvažniji dijelovi programskog koda, kojim je realizirano upravljanje elektroničkim komponentama i rashladnim sustavom.

ABSTRACT

COOLING PROCESS MONITORING AND MANAGING SYSTEM

This paper describes importance of cooling systems, and cooling process during fermentation, while making wine. It also describes manufacture and development procedure of cooling system, which task is to monitor and manage cooling during fermentation. Tehnologies and electrical components used during cooling system manufacturing are also mentioned and described, as well as operating principle. With operating principle there are also images attached, which are showing navigation through system menu, as well as system features. Paper also describes most important parts of source code, which is used to enable electronic components and cooling system to work properly.

10. ŽIVOTOPIS

Niko Sičanica rođen je 26. rujna 1989. godine u Doboju, BIH. Osnovnu školu je završio u Žepču, 2004. godine. Srednju elektrotehničku školu završava 2008. godine u Žepču kada i upisuje Stručni studij računarstva na Fakultetu strojarstva i računarstva u Mostaru. Godine 2013. upisuje razlikovnu godinu na Elektrotehničkom fakultetu u Osijeku, te 2014. godine diplomski studij, smjer Procesno računarstvo, na istom fakultetu.

11. PRILOZI

11.1. Programski kod

```
#include <OneWire.h>
#include <DallasTemperature.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,2,1,0,4,5,6,7);

//Definiranje na koje digitalne pinove na Arduinu ce biti spojeni senzori,
tipkala i LED

#define ONE_WIRE_BUS 2
const int buttonSelectPin = 3;
const int buttonLeftPin = 4;
const int buttonDownPin = 5;
const int buttonRightPin = 6;
const int buttonUpPin = 7;
const int ledPin1 =  8;
const int ledPin2 =  9;
const int ledPin3 =  10;

//Definiranje pocetnog statusa tipki 0(LOW) znaci da su iskljucene na
pocetku, 1(HIGH) znaci da su pritisnute

int buttonSelectState = 0;
int buttonLeftState = 0;
int buttonRightState = 0;
int buttonUpState = 0;
int buttonDownState = 0;

//Granicna vrijednost temperature kada se pali hladenje

float borderTemp1 = 30;
float borderTemp2 = 30;
float borderTemp3 = 30;

//Najniza vrijednost temperature kada se gasi sustav za hladenje postavljeno
je iznad 18 stupnjeva ako se nastavi hladiti nakon gasenja

float lowTemp1 = 20;
float lowTemp2 = 20;
float lowTemp3 = 20;

//Pomocne varijable za while petlju

int trigger = 0;
int trig = 0;

//Varijable koje prikazuju temperaturu trenutnu

float temp1 = 0;
float temp2 = 0;
float temp3 = 0;

int timeDiff = 10;              //Razlika izmedu ocitanja prve i druge
vrijednosti temperature da se vidi koliko je temperatura porasla (sekunde)
float tempDiff = 5;             //Koliki treba biti skok temperature u
odredenom vremenskom razmaku da se upali sustav hladenja (stupnjevi C)
```

```
int coolingTime = 1;            //Na koliko da se upali sustav hladenja ako
se upali zbog naglog porasta ali ostane ispod granicne vrijednosti (minute)

int k = timeDiff / 0.5;         //Pomocna varijabla koja predstavlja
odbrojavanje kada treba uzeti drugu temperaturu i gledati razliku

//Pomocne varijable za racunanje razlike temperature
float t1;
float t2;
float t3;

//Pomocne varijable za aktiviranje hladenja pri naglom skoku temperature
int cool1 = 0;
int cool2 = 0;
int cool3 = 0;

//Odbrojavanje koliko treba biti upaljeno hladenje
int c1 = coolingTime*60 / 0.5;
int c2 = coolingTime*60 / 0.5;
int c3 = coolingTime*60 / 0.5;

//OneWire sluzi za spajanje vise senzora na jedan pin Arduina, ovo definira
koji ce to pin biti
OneWire oneWire(ONE_WIRE_BUS);

DallasTemperature sensors(&oneWire);

void setup(void)
{
  Serial.begin(9600);
  sensors.begin();

  lcd.begin (16,2); // 16 x 2 LCD module
  lcd.setBacklightPin(3,POSITIVE);
  lcd.setBacklight(HIGH);

  //Postavljanje pinova za tipke na input(primanje podataka)
  pinMode(buttonSelectPin, INPUT);
  pinMode(buttonLeftPin, INPUT);
  pinMode(buttonRightPin, INPUT);
  pinMode(buttonUpPin, INPUT);
  pinMode(buttonDownPin, INPUT);

  //Postavljanje pinova za LED na output(slanje podataka ili pustanje struje)
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  pinMode(ledPin3, OUTPUT);

  //Dohvacanje temperature sa senzora
  sensors.requestTemperatures();

  //Svaki temperaturni senzor se ocitava pomocu indeksa koji se dodjeljuju
tako sto najblizi spojen dobije prvi indeks (0,1,2,3,...)
  temp1 = sensors.getTempCByIndex(0);
  temp2 = sensors.getTempCByIndex(1);
  temp3 = sensors.getTempCByIndex(2);
  t1 = temp1;
  t2 = temp2;
  t3 = temp3;
}
```

```
void loop(void)
{
  // Pozivanje sensors.requestTemperatures() da se dobiju temperature sa svih
senzora na istom pinu
  sensors.requestTemperatures();
  temp1 = sensors.getTempCByIndex(0);
  temp2 = sensors.getTempCByIndex(1);
  temp3 = sensors.getTempCByIndex(2);

//-----------------------------------------------------
//Odabir vrijednosti pri kojoj ce se ukljuciti hlađenje

  buttonSelectState = digitalRead(buttonSelectPin);   //Ocitavanje statusa
tipke select je li HIGH ili LOW
  if (buttonSelectState == HIGH) {    //Ako je HIGH onda pokreni prvi menu,
odabir granicne vrijednosti
    lcd.clear();
    trigger = 0;
    delay(1000);
    while(trigger != 1){    //While petlja koja zadrzava menu odabira
granicne vrijednosti dok se ponovno ne pritisne select tipka

      buttonSelectState = digitalRead(buttonSelectPin);
      if (buttonSelectState == HIGH) {
        delay(500);
        trigger = 1;
      }
      //Left i right sluze za namjestanje ostalih senzora, pomicu prikaz sa
jednog na drugi senzor
      buttonLeftState = digitalRead(buttonLeftPin);
      if (buttonLeftState == HIGH) {
        delay(500);
        trig = trig - 1;
      }
      buttonRightState = digitalRead(buttonRightPin);
      if (buttonRightState == HIGH) {
        delay(500);
        trig = trig + 1;
      }
      //Provjeravnje kada se izvrti do treceg senzora da se ode na prvi
      if (trig > 2){
        trig = 0;
      }
      if (trig < 0){
        trig = 2;
      }
      if (trig == 0){   //Trig sluzi za oznacavanje razlicitih izbornika prvi
koji prikazuje prvi senzor je na indeksu 0 i tako 1 i 2
        buttonUpState = digitalRead(buttonUpPin);
        if (buttonUpState == HIGH){
          borderTemp1 = borderTemp1 + 0.1;
          lcd.setCursor (0,1);
          lcd.print(borderTemp1,1);
          delay(500);
        }
        buttonDownState = digitalRead(buttonDownPin);
        if (buttonDownState == HIGH){
          if(lowTemp1 < borderTemp1){
            borderTemp1 = borderTemp1 - 0.1;
          }
          lcd.setCursor (0,1);
```

```
        lcd.print(borderTemp1,1);
        delay(500);
      }
      lcd.home ();
      lcd.print("1. Hladenje ON:");
      lcd.setCursor (0,1);
      lcd.print(borderTemp1,1);
      lcd.setCursor (15,1);
      lcd.print("C");
    }
    if (trig == 1){
      buttonUpState = digitalRead(buttonUpPin);
      if (buttonUpState == HIGH){
        borderTemp2 = borderTemp2 + 0.1;
        lcd.setCursor (0,1);
        lcd.print(borderTemp2,1);
        delay(500);
      }
      buttonDownState = digitalRead(buttonDownPin);
      if (buttonDownState == HIGH){
        if(lowTemp2 < borderTemp2){
          borderTemp2 = borderTemp2 - 0.1;
        }
        lcd.setCursor (0,1);
        lcd.print(borderTemp2,1);
        delay(500);
      }
      lcd.home ();
      lcd.print("2. Hladenje ON:");
      lcd.setCursor (0,1);
      lcd.print(borderTemp2,1);
      lcd.setCursor (15,1);
      lcd.print("C");
    }
    if (trig == 2){
      buttonUpState = digitalRead(buttonUpPin);
      if (buttonUpState == HIGH){
        borderTemp3 = borderTemp3 + 0.1;
        lcd.setCursor (0,1);
        lcd.print(borderTemp3,1);
        delay(500);
      }
      buttonDownState = digitalRead(buttonDownPin);
      if (buttonDownState == HIGH){
        if(lowTemp3 < borderTemp3){
          borderTemp3 = borderTemp3 - 0.1;
        }
        lcd.setCursor (0,1);
        lcd.print(borderTemp3,1);
        delay(500);
      }
      lcd.home ();
      lcd.print("3. Hladenje ON:");
      lcd.setCursor (0,1);
      lcd.print(borderTemp3,1);
      lcd.setCursor (15,1);
      lcd.print("C");
    }
  }
  trig = 0;
  lcd.clear();
```

```
//-----------------------------------------------
//Odabir vrijednosti kada se gasi sustav halđenja

    delay(1000);
    while(trigger != 2){

      buttonSelectState = digitalRead(buttonSelectPin);
      if (buttonSelectState == HIGH) {
        delay(500);
        trigger = 2;
      }
      buttonLeftState = digitalRead(buttonLeftPin);
      if (buttonLeftState == HIGH) {
        delay(500);
        trig = trig - 1;
      }
      buttonRightState = digitalRead(buttonRightPin);
      if (buttonRightState == HIGH) {
        delay(500);
        trig = trig + 1;
      }
      if (trig > 2){
        trig = 0;
      }
      if (trig < 0){
        trig = 2;
      }

      if (trig == 0){
        buttonUpState = digitalRead(buttonUpPin);
        if (buttonUpState == HIGH){
          if(lowTemp1 < borderTemp1){
            lowTemp1 = lowTemp1 + 0.1;
          }
          lcd.setCursor (0,1);
          lcd.print(lowTemp1,1);
          delay(500);
        }
        buttonDownState = digitalRead(buttonDownPin);
        if (buttonDownState == HIGH){
          lowTemp1 = lowTemp1 - 0.1;
          lcd.setCursor (0,1);
          lcd.print(lowTemp1,1);
          delay(500);
        }
        lcd.home ();
        lcd.print("1. Hladenje OFF:");
        lcd.setCursor (0,1);
        lcd.print(lowTemp1,1);
        lcd.setCursor (15,1);
        lcd.print("C");
      }
      if (trig == 1){
        buttonUpState = digitalRead(buttonUpPin);
        if (buttonUpState == HIGH){
          if(lowTemp2 < borderTemp2){
            lowTemp2 = lowTemp2 + 0.1;
          }
          lcd.setCursor (0,1);
          lcd.print(lowTemp2,1);
          delay(500);
```

```
        }
        buttonDownState = digitalRead(buttonDownPin);
        if (buttonDownState == HIGH){
          lowTemp2 = lowTemp2 - 0.1;
          lcd.setCursor (0,1);
          lcd.print(lowTemp2,1);
          delay(500);
        }
        lcd.home ();
        lcd.print("2. Hladenje OFF:");
        lcd.setCursor (0,1);
        lcd.print(lowTemp2,1);
        lcd.setCursor (15,1);
        lcd.print("C");
      }

      if (trig == 2){
        buttonUpState = digitalRead(buttonUpPin);
        if (buttonUpState == HIGH){
          if(lowTemp3 < borderTemp3){
            lowTemp3 = lowTemp3 + 0.1;
          }
          lcd.setCursor (0,1);
          lcd.print(lowTemp3,1);
          delay(500);
        }
        buttonDownState = digitalRead(buttonDownPin);
        if (buttonDownState == HIGH){
          lowTemp3 = lowTemp3 - 0.1;
          lcd.setCursor (0,1);
          lcd.print(lowTemp3,1);
          delay(500);
        }
        lcd.home ();
        lcd.print("3. Hladenje OFF:");
        lcd.setCursor (0,1);
        lcd.print(lowTemp3,1);
        lcd.setCursor (15,1);
        lcd.print("C");
      }
    }
    trig = 0;
    lcd.clear();
//-----------------------------------------------------------------
//Odabir vremenskog raspona, temperaturnog skoka, vremena hladenja

    delay(1000);
    while(trigger != 3){

      buttonSelectState = digitalRead(buttonSelectPin);
      if (buttonSelectState == HIGH) {
        delay(500);
        trigger = 3;
      }
      buttonLeftState = digitalRead(buttonLeftPin);
      if (buttonLeftState == HIGH) {
        delay(500);
        trig = trig - 1;
        lcd.clear();
      }
```

```
buttonRightState = digitalRead(buttonRightPin);
if (buttonRightState == HIGH) {
  delay(500);
  trig = trig + 1;
  lcd.clear();
}
if (trig > 2){
  trig = 0;
}
if (trig < 0){
  trig = 2;
}

if (trig == 0){
  buttonUpState = digitalRead(buttonUpPin);
  if (buttonUpState == HIGH){
    timeDiff = timeDiff + 1;
    lcd.setCursor (0,1);
    lcd.print(timeDiff);
    delay(500);
  }
  buttonDownState = digitalRead(buttonDownPin);
  if (buttonDownState == HIGH){
    if (buttonDownState == HIGH and timeDiff == 10){
      lcd.clear();
    }
    timeDiff = timeDiff - 1;
    lcd.setCursor (0,1);
    lcd.print(timeDiff);
    delay(500);
  }
  lcd.home ();
  lcd.print("Vremenski razmak");
  lcd.setCursor (0,1);
  lcd.print(timeDiff);
  lcd.setCursor (13,1);
  lcd.print("sec");
}

if (trig == 1){
  buttonUpState = digitalRead(buttonUpPin);
  if (buttonUpState == HIGH){
    tempDiff = tempDiff + 0.1;
    lcd.setCursor (0,1);
    lcd.print(tempDiff,1);
    delay(500);
  }
  buttonDownState = digitalRead(buttonDownPin);
  if (buttonDownState == HIGH){
    if (buttonDownState == HIGH){
      lcd.clear();
    }
    tempDiff = tempDiff - 0.1;
    lcd.setCursor (0,1);
    lcd.print(tempDiff,1);
    delay(500);
  }
  lcd.home ();
  lcd.print("Temp. skok");
  lcd.setCursor (0,1);
  lcd.print(tempDiff,1);
```

```
        lcd.setCursor (15,1);
        lcd.print("C");
      }

      if (trig == 2){
        buttonUpState = digitalRead(buttonUpPin);
        if (buttonUpState == HIGH){
          coolingTime = coolingTime + 1;
          lcd.setCursor (0,1);
          lcd.print(coolingTime);
          delay(500);
        }
        buttonDownState = digitalRead(buttonDownPin);
        if (buttonDownState == HIGH){
          if (buttonDownState == HIGH){
            lcd.clear();
          }
          coolingTime = coolingTime - 1;
          lcd.setCursor (0,1);
          lcd.print(coolingTime);
          delay(500);
        }
        lcd.home ();
        lcd.print("Vrijeme hladenja");
        lcd.setCursor (0,1);
        lcd.print(coolingTime);
        lcd.setCursor (13,1);
        lcd.print("min");
      }
    }
    trig = 0;
    lcd.clear();
  }
//-----------------------------------------------------------------------
//Ovaj dio provjerava ako je temeperatura nekog od senzora veca od granicne
vrijednosti i ukljucuje rashladne uredaje u ovom slucaju LED

  if (temp1 >= borderTemp1){ //Ako je temperatura prvog spremnika veca ili
jednaka granicnoj vrijednosti za prvi spremnik tada pali rashladni uređaj
    digitalWrite(ledPin1, HIGH);
    cool1 = 0;
  }

  if (temp2 >= borderTemp2){
    digitalWrite(ledPin2, HIGH);
    cool2 = 0;
  }

  if (temp3 >= borderTemp3){
    digitalWrite(ledPin3, HIGH);
    cool3 = 0;
  }
//-----------------------------------------------------------------------
//Provjeravanje jesu li se temperature spustile do optimalne vrijednosti i
pritom se gase rashladni uredaji

  if (temp1 <= lowTemp1){ //Ako se temperatura prvog spremnika smanjila ispod
granicne vrijednosti za prvi spremnik tada gasi rashladni uređaj
    digitalWrite(ledPin1, LOW);
  }
```

```
  if (temp2 <= lowTemp2){
    digitalWrite(ledPin2, LOW);
  }

  if (temp3 <= lowTemp3){
    digitalWrite(ledPin3, LOW);
  }

//-------------------------------------------------------------------
//Provjeravanje je li nakon istecenog vremena doslo do razlike u temperaturi
koja je postavljena
//Svaki put kad prođe kroz glavnu petlju neka spusti k, a k je broj koji
predstavlja koliko treba puta proci kroz petlju da ocita drugu vrijednost kad
k dođe na nulu ocitava drugu vrijednost temperature i k se restira, jedna k
predstavlja pola sekunde
  k = k - 1;
  if (k <= 0){ //ako je k=0 ili manji od 0 ocitaj drugu vrijednost
    if ((temp1 - t1) >= tempDiff){ //Ako je razlika prošle i ove nove
temperature veca od postavljene razlike koja smije biti onda pali rashladne
uređaje
      digitalWrite(ledPin1, HIGH);
      cool1 = 1; //Varijabla koja govori da su rashladni uređaji upaljeni
    }
    if ((temp2 - t2) >= tempDiff){
      digitalWrite(ledPin2, HIGH);
      cool2 = 1;
    }
    if ((temp3 - t3) >= tempDiff){
      digitalWrite(ledPin3, HIGH);
      cool3 = 1;
    }

    t1 = temp1;
    t2 = temp2;
    t3 = temp3;
    k = timeDiff / 0.5;
  }

  if (cool1 == 1){ //Ako je rashladni uređaj upaljen samo zato što je došlo
do naglog porasta temperature onda pregledava 2 slucaja:
//1.ako temperatura nakon zadanog vremena, koje se može mijenjati, ostane
ispod granicne vrijednosti to znaci da nije došlo do dovoljno naglog porasta
temperature i nakon isteka vremena rashladni uređaji se gase
//2.ako je došlo do naglog skoka temperature i ona nastavi rast iznad
granicne vrijednosti tada se preskace ovaj dio koda i pali se gore dio koji
gasi sistem samo kada se temperatura spusti ispod određene vrijednosti

    c1 = c1 - 1;
    if (c1 <= 0 and temp1 < borderTemp1){
      digitalWrite(ledPin1, LOW);
      cool1 = 0;
      c1 = coolingTime*60 / 0.5;
    }
    else if (c1 <= 0){
      c1 = coolingTime*60 / 0.5;
    }
  }

  if (cool2 == 1){
    c2 = c2 - 1;
    if (c2 <= 0 and temp2 < borderTemp2){
```

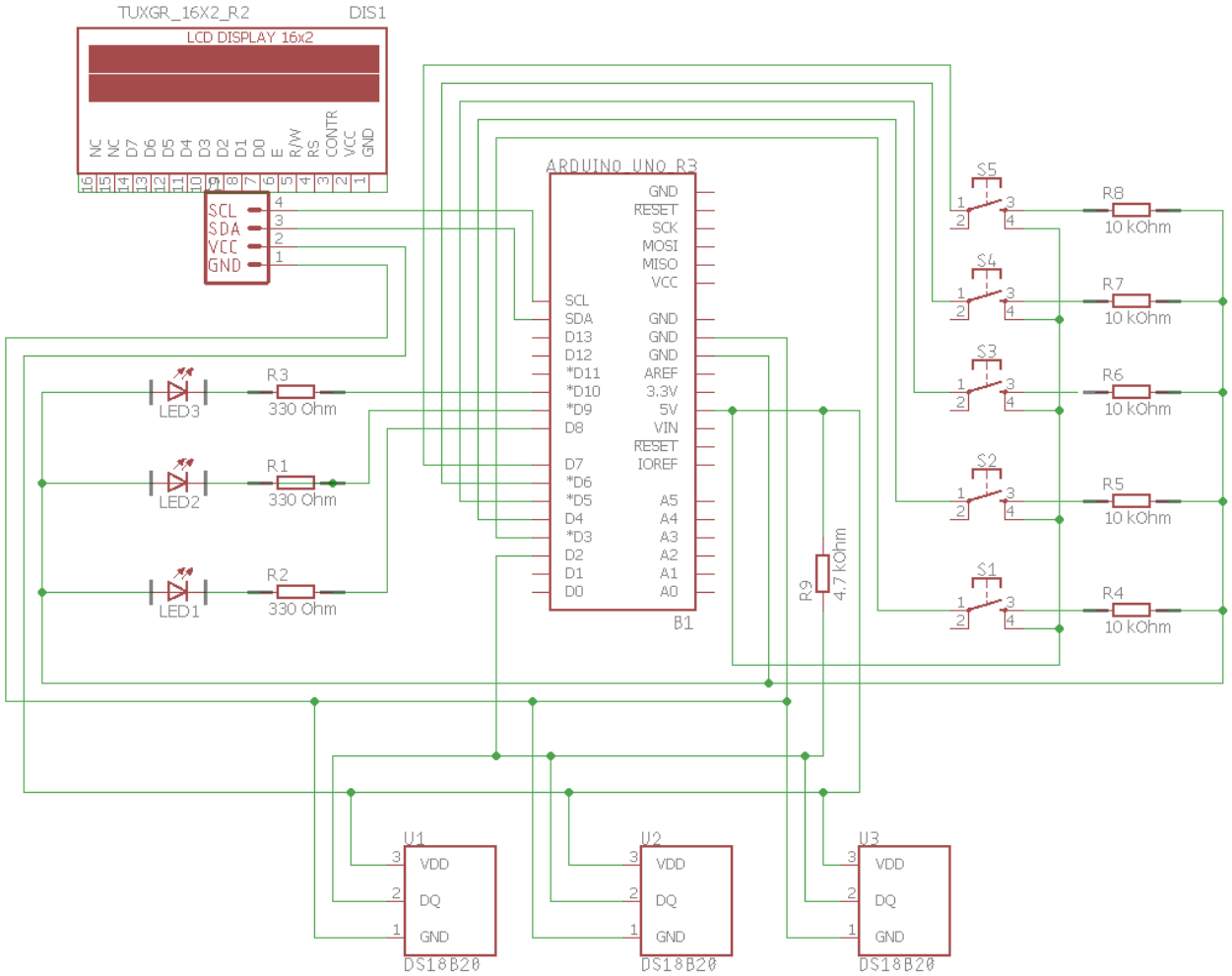
```
      digitalWrite(ledPin2, LOW);
      cool2 = 0;
      c2 = coolingTime*60 / 0.5;
    }
    else if (c2 <= 0){
      c2 = coolingTime*60 / 0.5;
    }
  }
  if (cool3 == 1){
    c3 = c3 - 1;
    if (c3 <= 0 and temp2 < borderTemp2){
      digitalWrite(ledPin3, LOW);
      cool3 = 0;
      c3 = coolingTime*60 / 0.5;
    }
    else if (c3 <= 0){
      c3 = coolingTime*60 / 0.5;
    }
  }

  if (cool1 == 0){
    c1 = coolingTime*60 / 0.5;
  }

  if (cool2 == 0){
    c2 = coolingTime*60 / 0.5;
  }

  if (cool3 == 0){
    c3 = coolingTime*60 / 0.5;
  }

//-------------------------------------------------------------------

  lcd.setCursor (2,0);  //Postavljanje pokazivaca na mjesto 2,0 na LCD-u
  lcd.print("Temperatura:");
  lcd.setCursor (0,1);
  lcd.print(temp1,1);
  lcd.setCursor (6,1);
  lcd.print(temp2,1);
  lcd.setCursor (12,1);
  lcd.print(temp3,1);
  delay(500);
}
```

11.2. Shema spoja