

Факултет Инжењерских Наука
Универзитет у Крагујевцу



Назив студијског програма: Машинско инжењерство

Ниво студија: Мастер академске студије

Модул: Примењена механика и аутоматско управљање

Предмет: **Брза израда прототипова**

Број индекса: 350/2010

Ђорђе Д. Јовановић

Програмабилни аутоматски систем мале пиваре

МАСТЕР РАД

Комисија за преглед и одбрану:

1. Проф. др Ненад Грујовић - ментор

2. _____

3. _____

4. _____

Датум

одбране: _____

Оцена: _____

У оквиру овог мастер рада кандидат треба да презентује процес реализације и пројектовања аутоматског система управљања, проналазак оптималног физичког изгледа система користећи технологије резања и брзе израде прототипова, прављење алгоритма управљања системом и његово примењивање у процесу контроле уз коришћење Arduino микроконтролера.

Препоручена литература:

- [1] „Rapid Prototyping – брза израда прототипова“, др Ненад Грујовић, 2005
- [2] Andreas Gebhardt, “Rapid Prototyping”, Hanser, 2000
- [3] Милићев Д.: Објектно оријентисано програмирање на језику C++, Микрокњига, Београд, 1991.
- [4] Хенсен А.: Програмирање на језику C, Микрокњига, Београд, 1991.
- [5] „Getting Started With Arduino 2nd Edition“, Massimo Banzi, Maker Media, Inc October 2011

Крагујевац, 2015.

ментор:
др Ненад Грујовић, ред. проф.

Резиме рада:

Мастер рад „Програмабилни аутоматски систем мале пиваре” приказује суштину пројектовања аутоматских система на основу његовог идејног решења. Приказан је процес тродимензионалног моделирања система на основу захтева и ограничења технологије производње пива, његова оптимизација у поређењу на постојеће - конвенционалне начине, израда реалног система, састављање електро спецификације, примена софтвера и технологија за брзу израду прототипова и програмирање у програмском језику C.

Кључне речи:

Arduino, брза израда прототипова, објектно оријентисано програмирање, технологија пива

Summary of work:

Master thesis „Programmable automatic system of home brewery” is showing the essence of designing an automated systems on the basis of its conceptual design. This project explains a process of three-dimensional modeling system based on the requirements and limitations of the technology of brewing, its optimization compared to the existing - conventional ways, making the real system, assembling electrical specifications, application of software and technology for rapid prototyping and programming in the programming language C.

Key words:

Arduino, rapid prototyping, object-oriented programming, brewing technology

Садржај

Увод.....	9
1. Основне карактеристике производње пива.....	11
1.1 Вода	11
1.2 Производња сладовине	12
1.2.1 Дробљење слада	13
1.2.2 Укомљавање слада	14
1.2.3 Филтрација сладовине	15
1.2.4 Кување сладовине са хмељом.....	15
1.2.5 Хлађење и бистрење сладовине	16
1.3 Главно и накнадно врење.....	18
1.3.1 Главно алкохолно врење.....	18
1.3.2 Накнадно зрење и врење	20
1.4 Бистрење и истакање пива. Готово пиво.....	21
1.4.1 Филтрација пива.....	21
1.4.2 Истакање пива.....	22
2. ARDUINO.....	23
2.1 Arduino микроконтролерска платформа	23
2.1.1 Предности Arduino -а.....	23
2.1.2 Примарна намена Arduino -а	23
2.2 Хардвер Arduino -а	23
2.2.1 Ардуино модули (Shields)	24
2.2.2 Arduino Duemilanove/UNO	25
2.2.2.1 Унутрашње компоненте - Arduino Duemilanove.....	25
2.2.2.2 Конекције – Arduino Duemilanove	26
2.3 Програмско окружење Arduino -а.....	28
2.3.1 Изглед и функционалност Arduino IDE прозора	28
2.3.1.1 Уношење скице у микроконтролер.....	30
2.3.1.2 Biblioteke (Libraries).....	31
2.4 Примена модула и компонената	31
2.4.1 Ардуино сензорски модул (Sensor Shield V4.0)	31
2.4.2 Осмо-релејни модул (Arduino 8-relay module)	33
2.4.3 Дигитални температурни сензор DS18B20.....	35
2.4.4 Тастери	40
2.4.5 Ултразвучни сензор HC - SR04	44
2.4.6 LCD дисплеј са i2C комуникацијским протоколом	47
2.4.7 Пиезо компонента - звучник	50
3. Оптимизација процеса прављења пива	53
3.1 Физички изглед пиваре.....	53
3.1.1 Судови	54
3.1.1.1 Први суд - HLT / Кувало	56
3.1.1.2 Други суд - Мешаона / Суд за укомљавање	61

3.1.2 Систем хлађења	66
3.1.3 Пумпе	67
3.1.4 Црева за повезивање судова.....	69
3.1.5 Електро ормар	71
3.1.5.1 Распоред компоненти унутар ормара	73
3.1.5.2 Електро шема система	75
4. Алгоритам програма и повратне спреге	76
4.1 Алгоритам рада система	76
4.1.1 Функцијски објекти система	78
4.1.1.1 Void Core()	78
4.1.1.2 Void beep(параметри)	79
4.1.1.3 Void welcomescreen().....	79
4.1.1.4 Void mainmenu().....	80
4.1.1.5 Void formattime(параметри и променљиве)	81
4.1.1.6 Void formattimeseconds(променљиве).....	81
4.1.1.7 Void calcstrike()	81
4.1.1.8 Void uss().....	82
4.1.2 Основни објекти система	83
4.1.2.1 Void manualmode().....	83
4.1.2.2 Void setupmenu()	83
4.2 Повратне спреге - аутоматски мод	87
4.2.1 Void automode()	87
4.2.1.1 Case 1 - Одбројавање	88
4.2.1.2 Case 2 - Провера нивоа воде.....	89
4.2.1.3 Case 3 - Загревање воде	90
4.2.1.4 Case 4 - Утакање у мешач	92
4.2.1.5 Case 5 - Укомљавање	93
4.2.1.6 Case 6 - Покапавање	96
4.2.1.7 Case 7 - Мешање након загревања комине.....	97
4.2.1.8 Case 8 - Одлежавање комине	98
4.2.1.9 Case 9 - Претакање и рецикулација	99
4.2.1.10 Case 10 - Пуњење кувала	101
4.2.1.11 Case 11 - Подизање температуре у кувалу	102
4.2.1.12 Case 12 - Кување.....	103
4.2.1.13 Case 13 - Хлађење и филтрација.....	106
4.2.2 Крај петље void automode().....	107
5. Прављење пива.....	108
5.1 Прибор.....	109
5.2 Рецепт 1	109
5.3 Рецепт 2	110
5.4 Рецепт 3	111
6. Закључак	112
7. Литература.....	113

Увод

У свету производње пива тренутно има прилива особа које производе пиво, како из хобија, тако и из комерцијалних интереса којима је у циљу да се направи низ производа од пива. Сама пиварска индустрија добија више удела на тржишту како која година пролази, јер муштерије захтевају више укуса и више креативности у пићима које конзумирају. Чак прошле године, раст пиварске индустрије је био 12% у количини, а 16% у заради у доларима. Повећање је, такође, изазвало процват и у кућној производњи пива јер сам процес производње може да се обави са релативно малом инвестицијом. Програмабилни аутоматски систем мале пиваре улази на ово растуће тржиште и омогућава лак, аутоматизован метод за производњу пива на бази слада добијеног од житарица.

Циљ овог пројекта је приказивање и показивање путање развоја једног идејног решења и замисли што на крају представља изводљив производни процес система аутоматског управљања. Крајњи циљ је и показивање функционалности истог пројектованог система и његова крајња примена.

Прва ствар која се разматра на самом почетку пројекта је идејно решење, а то је аутоматизација процеса прављења пива уз свођење количине потребних процеса, за добијање истог, на минималну.

О пиву

Процес прављења пива је познат од давнина, али се кроз време усавршавао и допуњавао. Према древним ископинама и записима археолози су утврдили да су стари Вавилонци производили пиво у домаћинству још 7000 година пне. Имали су 16 сорти пива, а за варење су користили јечмени слад и пшеницу. Одавде се пренела производња пива у древни Египат, Персију, Грчку и друге земље. Египћани су знали да праве пиво већ 2000 година пне., а Грци су, примајући цивилизацију од њих, научили вештину производње пива, за њима Римљани, па стари Германи и онда остале европске земље. У тим временима пиво је прављено из слада, али без хмеља, тако да је добијен производ био веома кисео. Тада су у пиво додавали различите траве – пелин, лупин глог, шафран и др. Примена хмеља представља важно откриће и чини основу савремене технологије пива. Хмељ се први пут почео користити у Новгородској Русији. Усавршавање парне машине веома се повољно одразило на развитак пиварства. Најважнији технички проналазак представља проналазак машине за хлађење. Захваљујући најважнијим научно-техничким достигнућима у 19. веку створени су услови за претварање уситњених предузећа у велике механизоване фабрике пива. [1]

Мотивација

Програмабилни аутоматски систем мале пиваре (П.А.С.М.П.) омогућује и произвођачу пива у комерцијалне сврхе и произвођачу пива код куће да смање време и трошкове, а да повећају понављање рецепта и сигурност. Главни циљ самог процеса захтева додавање вруће течности измрвљеним житарицама, а онда пуштање да се свеукупна температура укомљавања стабилизује на распон од 65 до 69 степени целзијусових. Фактори који утичу на успех достизања циљане температуре укомљавања укључују термички коефицијент судова у којима се налази мешавина, температуру зрна, температуру течности, топлотну проводљивост и температуру ваздуха. П.А.С.М.П. смањује ове факторе тако што мешавина циркулише кроз размењивач топлоте у воденом купатилу. Водено купатило служи као резервоар за вруће течности, или енг. HLT (*Hot Liquor Tun*), који обезбеђује топлу воду за укомљавање, као и медијум за пренос топлоте који одржава стабилну циљану температуру укомљавања. Безбедност такође постаје проблем уколико се не користе пумпе јер вруће течности које су укључене у процес морају да се пребацују из суда у суд, а достижу запремину и до 42 литара.

Стандардна метода претварања скроба из житарица у шећер за комерцијалне сврхе или кућну

производњу захтева пет до шест сати произвођачевог времена. Овај систем смањује време за један до два сата, омогућавајући тако пивару да очисти и дезинфикује опрему која му је потребна за каснију ферментацију и кондиционирање. Пивар је такође у могућности да остави аутоматизовани систем без надзора и да буде сигуран да су све околности укључене у процес под контролом. Трошкови енергије су смањени са овим системом уз помоћ коришћења електричне енергије и добро изолованих судова. Због своје смањене величине, овај систем је аутоматизован и троши мање од 12 А што омогућава систему да буде коришћен на стандардном кућном електричном напону. Системи који су доступни пиварима који производе мале количине пива, као што су "MoreBeer 1550 brew sculpture" са S.M.A.R.T., кошта 3,035 евра за количину од 38 литара. Процес производње још увек мора да буде надгледан пошто има отворен пламен и нема аутоматски вентил нити пумпу која контролише пренос течности. П.А.С.М.П. кошта само делић од цене осталих система на тржишту и процењена цена му је око 590 евра.

Како?

Аутоматизовати процес производње пива код којег се појединачни процеси и делови производње разликују по карактеристикама окружења (запремина, температура, притисак, време, итд.) у којима се у одређеном тренутку времена нађе потенцијално пиво (вода, слад, требер, итд.), а притом их има око 9 (зависно од технологије), је био крајње занимљив.

Одређени захтеви и ограничења су постојали од старта.

Захтеви:

1. Што мања запремина и габарити система аутоматског управљања (<30 литара)
2. Могућност управљања уређајима, моторима и компонентама наизменичне и једносмерне струје
3. Што мања цена
4. Примена једноставних решења у циљу поједностављења процеса
5. Што већи степен аутоматизације
6. Доступност делова који ће чинити крајњи физички систем
7. Модуларност
8. Поновљивост процеса

Ограничења:

1. Што мања запремина и величина система аутоматског управљања (<40 литара)
2. Микроконтролерска платформа **Arduino Uno R3**
3. Што мања цена компонената система

Сва претходно наведена ограничења и захтеви представљају свакодневни део једног радног дана развојног инжењера.

1. Основне карактеристике производње пива

Под пивом се подразумева слабо алкохолно пиће, које се производи у процесу алкохолног врења из слада, хмеља, воде и пивског квасца. Вода је главни део свих напитака. Она служи као растварач. Слад даје пиву састојке екстрата од којег зависи пуноћа укуса и концентрација основног екстрата пива. Хмељ конзервира пиво и даје му угодан мирис и горак укус, док пивски квасац изазива алкохолно врење у коме шећер прелази у алкохол и угљен диоксид. Производња пива се дели на **технологију слада** и **технологију пива**.

Технологију слада обухвата:

1. Чишћење и сортирање јечма
2. Мочење јечма
3. Клијање јечма
4. Сушење зеленог слада
5. Ослобађање слада од клице и
6. Полирање.

Технологију пива обухвата:

1. Производња сладовине
2. Главно и накнадно врење
3. Бистрење и истакање пива

Пошто је тема овог рада усмерена ка самој производњи пива, објашњења и дубље залажење у материју технологије слада ће бити изостављено. Ова материја је више предмет изучавања на факултетима који се баве узгајавањем и неговањем биљака, укључујући и технолошки факултет.

1.1 Вода

У зависности од типа пива које се прави (светло, тамно, црно, итд.), мењањем квалитета воде квалитет пива се може кретати од доброг до савршеног. Циљ овог поглавља није детаљно залажење у квалитет воде који је потребан приликом прављења пива, нити хемије код овог процеса, већ један суштински приказ шта је потребно урадити са водом и на шта треба обратити пажњу.

Када се дође до термина „третман воде“, конкретно у овом случају третман воде за процес прављења пива, постоје 3 циља које пивари теже да постигну:

- **елиминисање укуса воде чији су узрочници поједина хемијска једињења**
 - ово у већини случајева представља уклањање резидуалног хлора
- **обезбедити јонски профил воде који подржава жељени укус пива**
- **обезбедити однос база и киселина у води према жељеној вредности рН**
 - овај однос зависи и од слада који ће бити коришћен

Оно што је обесхрабљујуће за многе пиваре није толико хемија воде која је потребна за добијање жељеног укуса пива, којом се релативно лако може руковати помоћу многобројних интернет калкулатора који служе за израчунавање неопходног хемијског профила воде, већ многобројни подесиви параметри који су укључени у процес.

Хлор, у води се може наћи као слободан хлор или хлорамин. Из воде за прављење пива се мора уклонити зато што формира опасно хемијско једињење - хлорфенол током процеса прављења пива. Већина пивара постижу ово филтрацијом помоћу активног угља и једноставно коришћењем хемијских средстава, у овом случају средства на бази натријума или калијум метабисулфита. Још једна нежељена супстанца у води за прављење пива је гвожђе. Ниво гвожђа у води треба да буде испод 0.3 mg/L да би се избегао укус метала у пиву.

Пивари брину о 6 примарних јона у води који су у могућности да утичу на укус пива. Наелектрисање катјона (Ca^{2+} , Mg^{2+} , Na^{+}) треба да се подударе са наелектрисањем јона (Cl^{-} , SO_4^{2-} , HCO_3^{-}) што представља и зашто се једно не може додати без другог. Испод су побројани хемијски елементи, њихова наелектрисања и како они утичу на укус воде.

Калцијум (Ca^{2+}) је благотворан за процес прављења пива пошто поспешује згрушавање, тј. флокулацију квасца. Такође је укусно неутралан и препоручени минимум је 40 mg/L. Најједноставнији начин повећања концентрације калцијума у води је додавање пиварског гипса и калцијум хлорида, који такође респективно додају сулфате и хлориде.

Магнезијум (Mg^{2+}) у води и није толико потребан, обзиром да приликом процеса прављења пива слад испушта велике количине њега. Вишак магнезијума може да проузрокује већу горчину пива. Ниво магнезијума у води треба да буде испод 50 mg/L.

Натријум (Na^{+}) може довести до сланог укуса крајњег производа. Ниво натријума у води треба да буде испод 100 mg/L. Пивари обично не додају натријум сем у случајевима када се користи сода бикарбона за подизање нивоа бикарбоната.

Хлор (Cl^{-}) даје мекши укус пиву и пожељан је у процесима прављења пива где је јечам основни састојак.

Сулафати (SO_4^{2-}) појачава хмељну горчину пива и даје суви укус пиву. Пожељна је већа концентрација код пива са више хмеља, а неки пивари иду чак и до 700 mg/L. Већина пива са вишком коришћеног хмеља ће испасти одлично са количином од 100 до 300 mg/L.

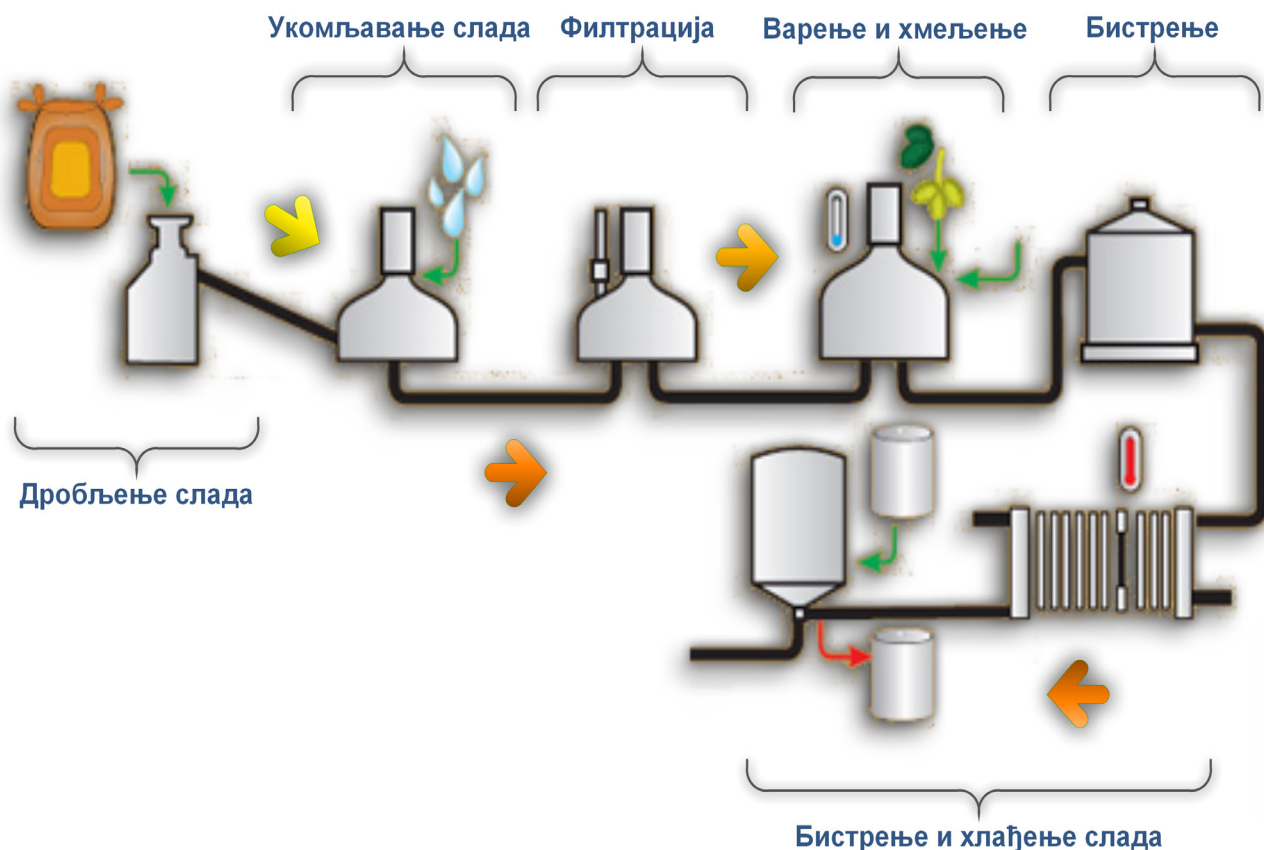
Бикарбонати (HCO_3^{-}) немају директан утицај на укус пива, али зато могу да имају индиректан утицај због њихових способности да подижу рН вредност комине ван жељеног интервала.

рН комине је резултат баланса између рН вредности активних водених јона и јечма, разних киселина и соли који су додати целокупној смеси. У већини случајева пивари желе да добију рН смеше у интервалу између 5.3 и 5.6. Овај интервал важи за охлађено скувано пиво (~ 22 степена целзијуса). Тамни сладови и киселине су примарни узрочници ниже вредности рН, док алкалност воде подиже рН. Калцијум и у мањој мери Магнезијум такође спуштају вредност рН, али не у мери да се помоћу ова два елемента врши контрола рН вредности. У већини случајева за регулацију рН вредности комине се користе киселине (лактоза или фосфорне киселине су популарни избори) приликом неутрализације алкалности воде. Светлим пивима треба више киселина да би се ниво рН чак и померио испод тренутне и довео до жељеног интервала између 5.3 и 5.6.

1.2 Производња сладовине

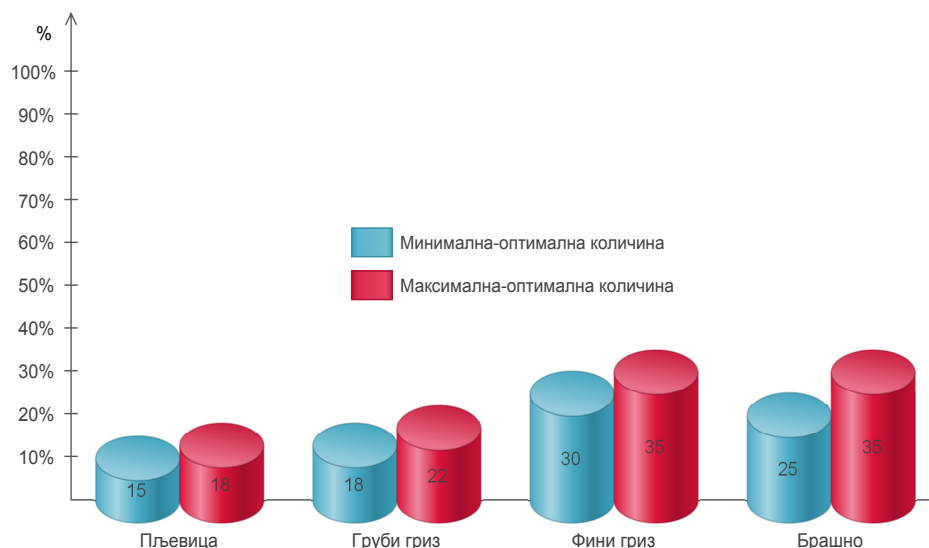
У оквиру овог поглавља биће приказан процес производње сладовине у класичној пивари и он се може поделити на следеће фазе:

- | | |
|-------------------------------------|---------------------------------|
| 1. Дробљење слада | 4. Варење и хмељење сладовине |
| 2. Укомљавање или екстракција слада | 5. Бистрење и хлађење сладовине |
| 3. Филтрација сладовине | |



1.2.1 Дробљење слада

Дробљење слада представља механичку припрему слада за екстракцију. Главни задатак дробљења слада је олакшавање и убрзавање физичких и биохемијских процеса растварања садржаја зрна у току укомљавања ради максимално могућег превозиња екстрактивних материја у водени раствор, тј. сладовине. Слад мора бити чист, сортиран и са једноличном величином зрна, што је све од посебног значаја за постизање доброг састава прекрупце. На принос екстракта утиче и степен уситњавања слада. Код дробљења слада и бољег уситњавања ендосперма поставља се задатак да се пљевике не оштете много. При нормално прекрупљеном сладу, односи појединих састојака изражени у процентима треба да буду следећи:



1.2.2 Укомљавање слада

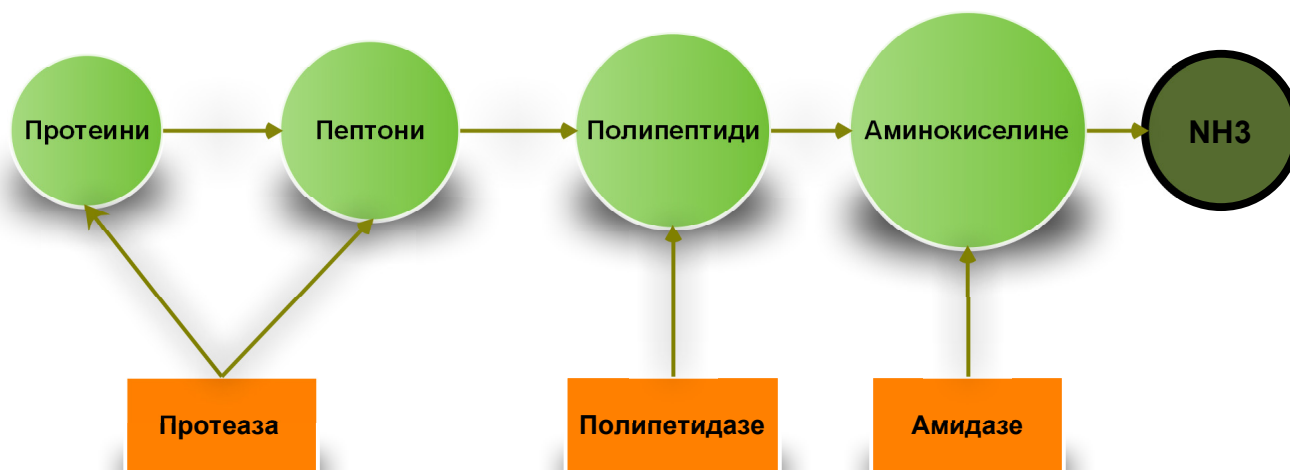
Суштина процеса укомљавања лежи у превођењу нерастворених материја слада и несладоване житарице у растворени облик путем **ензимске хидролизе**. У сладу има мало састојака који се растварају у води (10-15%) од укупне тежине суве материје. Састојци слада растворљиви у води су: шећери, део беланчевина и производа њихове разградње, те мале количине пентозана, пентоза, хексозана и хексоза, пектина, танинских и горких киселина. Све ове супстанце брзо прелазе у раствор, а најзначајнији део зрна слада, кога чини скроб, није растворљив. Поред скроба има још један део беланчевина у сладу који је у нерастворљивом облику. Да би могли прећи у раствор, главни састојци зрна: скроб и беланчевине морају се даље разградити путем ензимске хидролизе. Ова ензимска разградња одвија се под утицајем амилолитичких и протеолитичких ензима. Амилаза цепа скроб на малтозу, глукозу, малтотриозу и више декстрине. Протеаза преводи сложене беланчевинске супстанце до аминокиселина. Фитаза разграђује фитин на инозит и фосфате, а липаза разграђује масти.

Разградња скроба

До ензимске хидролизе скроба долази услед деловања амилолитичких ензима. Скроб се састоји од амилозе и амилопектина. У почетку укомљавања молекули скроба се брзо разграђују уз хидролизу, што карактерише промену јодне реакције. Једна од најкарактеристичнијих реакција скроба је интензивно модра боја са јодом. Она указује на присуство скроба. Утицајем алфа и бета амилазе у присуству воде скроб се раствара на малтозу и декстрин. Код тог распадања скроба настаје обично 80% малтозе и 20% декстрина. Проценат није сталан већ се мења под утицајем температуре. Све у свему деловање ензима траје док се између декстрина и малтозе не постигне одређена равнотежа. Ензимска хидролиза скроба одиграва се истовремено алфа и бета амилазе. Постепеним отцепљивањем молекула малтозе од молекула скроба настаје амилодекстрин, који са раствором јода даје љубичасту боју. У даљој фази разградње ензима из амилодекстрина отцепљивањем малтозе настаје еритродекстрин, који са јодом даје црвену боју. Даљим деловањем из еритродекстрина се отцепљује малтоза па настаје ахродекстрин, који са јодом не изазива промену боје. Коначно настаје малтоза и малтодекстрин који са јодом не реагује, па не долази до промене боје. Пошто оваква разградња скроба доводи до шећера, процес се зове **ошећерење**.

Разградња беланчевина

У току процеса укомљавања сложени молекули беланчевина су под дејством ензима протеаза распадају.



Производи разградње беланчевина (полипептиди и аминокиселине) неопходни за исхрану квасца. Са друге стране, средње-молекуларне беланчевине (пептони и делимично полипептиди) условљавају пуноћу укуса и стабилност пене пива. Беланчевине великих молекула су често узрок појаве помућења, због чега се тежи њиховој разградњи.

1.2.3 Филтрација сладовине

Пошто су у току процеса укомљавања и кувања све растворене материје прешле у сладовину, нужно је да се она одвоји од нерастворених делова требера.

Поступак филтрације се врши у две фазе:

1. **Прва фаза** - отицање прве сладовине, која има већу концентрацију основног екстракта
2. **Друга фаза** - испирање екстракта из требера, која се врши путем екстракције са топлем водом.

Састав пивског требера

Све оне материје, које приликом укомљавања слада и кувања не прелазе у раствор, чине нерастворни део који се издваја у пивски требер. У њему се у првом реду налазе пљевице зрна, нерастворених беланчевина алеуронског слоја зрна, нешто заосталог скроба што се није разградио у току кувања, и мала количина нерастворног екстракта који при испирању није прешао у сладовину.

1.2.4 Кување сладовине са хмељом

Код кувања сладовине са хмељом прелазе горке и ароматске материје у сладовину, а беланчевине се издвајају у крупне пахуљице које се постепено таложе и сладовина бистри.

У току процеса кувања дешавају се следеће физичко-хемијске промене:

- Разградња и растварање хмеља и његових прерађевина, екстракта у једноставније спојеве
- Коагулација беланчевина и издвајање беланчевина у виду пахуљица
- Испаравање воде у циљу постизања стандардне концентрације основног екстракта
- Стерилизација сладовине
- Деактивација ензима
- Бојење сладовине
- Оксидација сладовине
- Образовање редуцирајућих материја

Задатак кувања је стабилизација састава сладовине уз горе наведене физичко-хемијске промене. Са додатком хмеља постиже се његова екстракција која утиче на квалитет пива (укус, биолошку стабилност, пену). Кување сладовине траје обично 1,5-2 сата. Предуго кување води до појачања боје сладовине, а прекратко делује неповољно јер не може да доведе до издвајања високомолекуларних беланчевина које изазивају слабо бистрење сладовине и помућење пива. Крај кувања се практично одређује према концентрацији екстракта у сладовини, према издвајању беланчевина у виду пахуљица и провидности вруће сладовине. Кувањем се такође уништавају и сви ензими у сладовини чиме се обезбеђује стабилност хемијског састава сладовине пре врења. У

сладовини и у пиву налазе се материје које се вежу са кисеоником и стварају нерастворни облик танинско-беланчевинских комплекса који су узрочници помутњења пива услед дугог стајања, због тога је важно да у сладовини постоје редукујуће супстанце које могу да сузбију оксидацију која настаје аерацијом. Ове супстанце су тзв. редуктони, долазе са сировинама у сладовину, и стварају се у процесу кувања.

Додавање хмеља

Количина хмеља која се додаје у сладовину варира од 200-500 грама по хектолитру пива. То зависи од више фактора:

- Захтевима и укусу потрошача
- Квалитету хмеља
- Типу и врсти пива
- Саставу воде
- Квалитету слада

На пример, светла пива у односу на тамна захтевају 20-30 % више хмеља по хектолитру пива. Разлог је да светла пива треба да имају хмељну горчину, док су тамна пива слабо ароматичног укуса.

Одвајање сладовине од хмељног тропа

Након завршеног кувања хмељне сладовине она се препумпава преко цедила за хмељ у таложњак. На дну хмељног цедила налазе се сита где остаје хмељни троп и део коагулисаних беланчевина, а бистра сладовина излази кроз сита и одлази до таложњака. Хмељни троп који остаје у цедилу садржи доста упијене сладовине која се испира са топлем водом, па се на 1 килограм додатог хмеља налази 6-7 литара сладовине.

1.2.5 Хлађење и бистрење сладовине

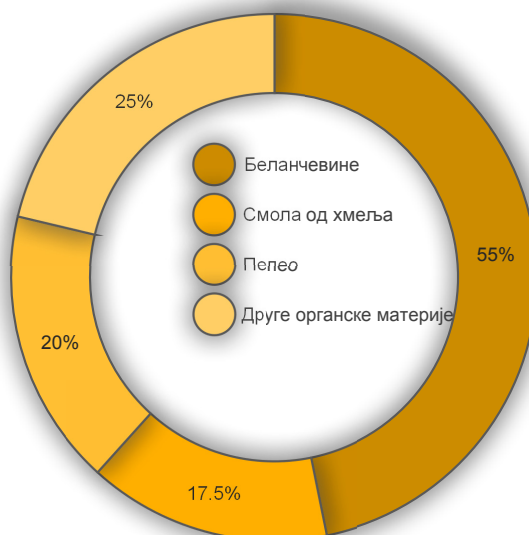
Задатак процеса хлађења је снижавање температуре која одговара потребама почетног стадијума врења и **засићавање кисеоником из ваздуха** у циљу постизања врења. Осим тога треба да се издвоји талог како би се добила бистра сладовина као једна од основних претпоставки за пиво високог квалитета. У сладовини која долази на хлађење налазе се беланчевине које чине фини талог. Снижењем температуре оне се талоче као и друге материје. За време хлађења сладовина се мора заситити кисеоником из ваздуха и ослободити грубог талог. У вези овог захтева сладовина се хлади у две фазе, па добијамо две врсте талогa - **врући и хладни талог**.

Врући талог

Врући талог коагулира у виду крупних пахуљица приликом талочења сладовине. Састав врућег талогa је следећи:

- 50 - 60 % Беланчевина
- 15 - 20 % Смоле од хмеља
- 20 - 30 % Других органских материја
- 3 - 20 % Пепела

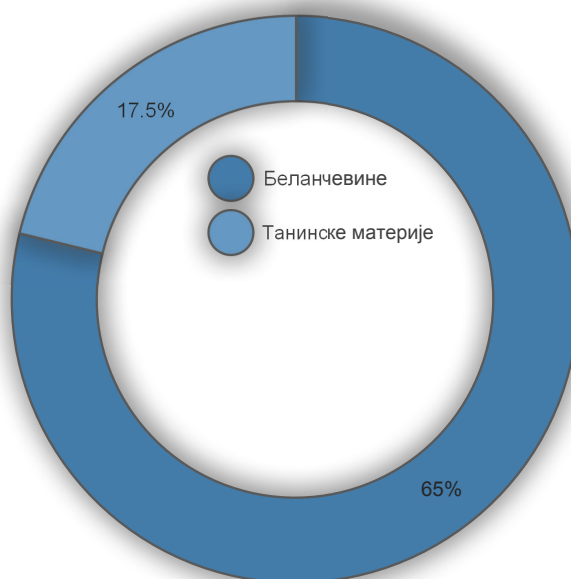
Сладовина под притиском пумпе улази у таложњак и том приликом долази до њеног кружног кретања. Услед тога формира се талог у центру посуде у облику купе која остаје на дну након испуштања сладовине.



Хладни талог

Овај талог почиње да се издваја на температури испод 60° C. Хладни талог чини:

- 60 - 70 % Беланчевина
- остатак су танинске материје



За отклањање хладног талога користи се Вентуријева цев, тј. процес који се зове **флотација**. Суштина флотације је у томе што се у расхлађену сладовину, из које је издвојен груби талог, помоћу Вентуријеве цеви убризгава ваздух. У танку се због тога ствара пена која се пење према горњим слојевима и површини, носећи за собом хладни талог. Са ваздухом се кроз цев истовремено додаје и квасац па се на тај начин врши аерација и истовремено додавање квасца, па се после тога сладовина пумпа на врење.

Оксидација сладовине

У току процеса бистрења сладовина везује кисеоник физички и хемијски, што делује на повећање боје. Код виших температура долази до хемијског везивања кисеоника, а код нижих

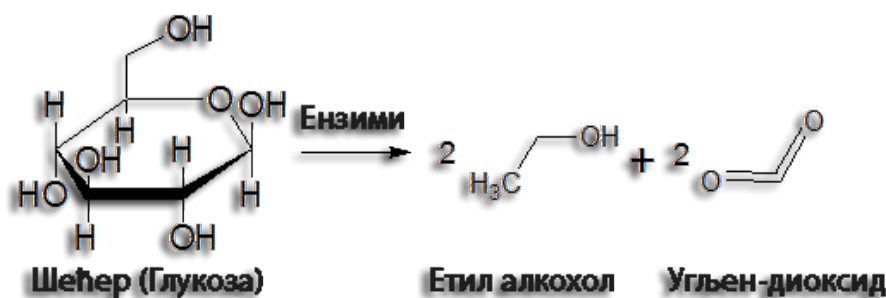


кисеоник се раствара све док раствор не постане засићен. Шећера у сладовини има далеко више него других екстрактивних материја, због чега се у току хлађења кисеоник углавном веже са шећером.

1.3 Главно и накнадно врење

1.3.1 Главно алкохолно врење

Основу технологије добијања пива сачињава алкохолно врење. Процес врења је животна појава дисања квасца, када долази до разградње шећера у алкохол и угљену киселину (Пастерова витална теорија врења).

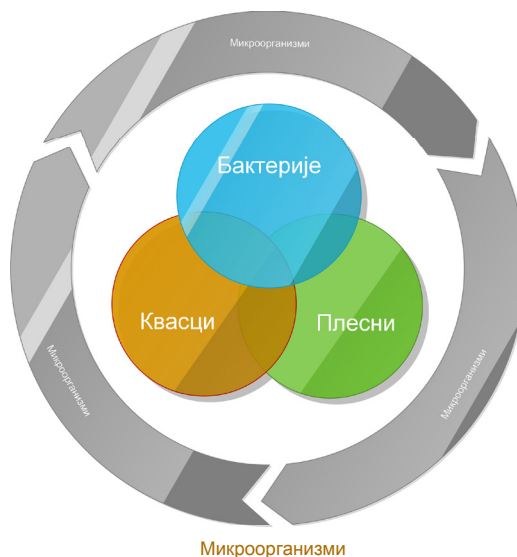


После Пастера појавила се и нова теорија према којој треба алкохолно врење посматрати последицом деловања ензима, а квасац је неопходан као носилац ензима врења. Ензим који се налази у квасцу добио је име цимаза (eng. Zymase).

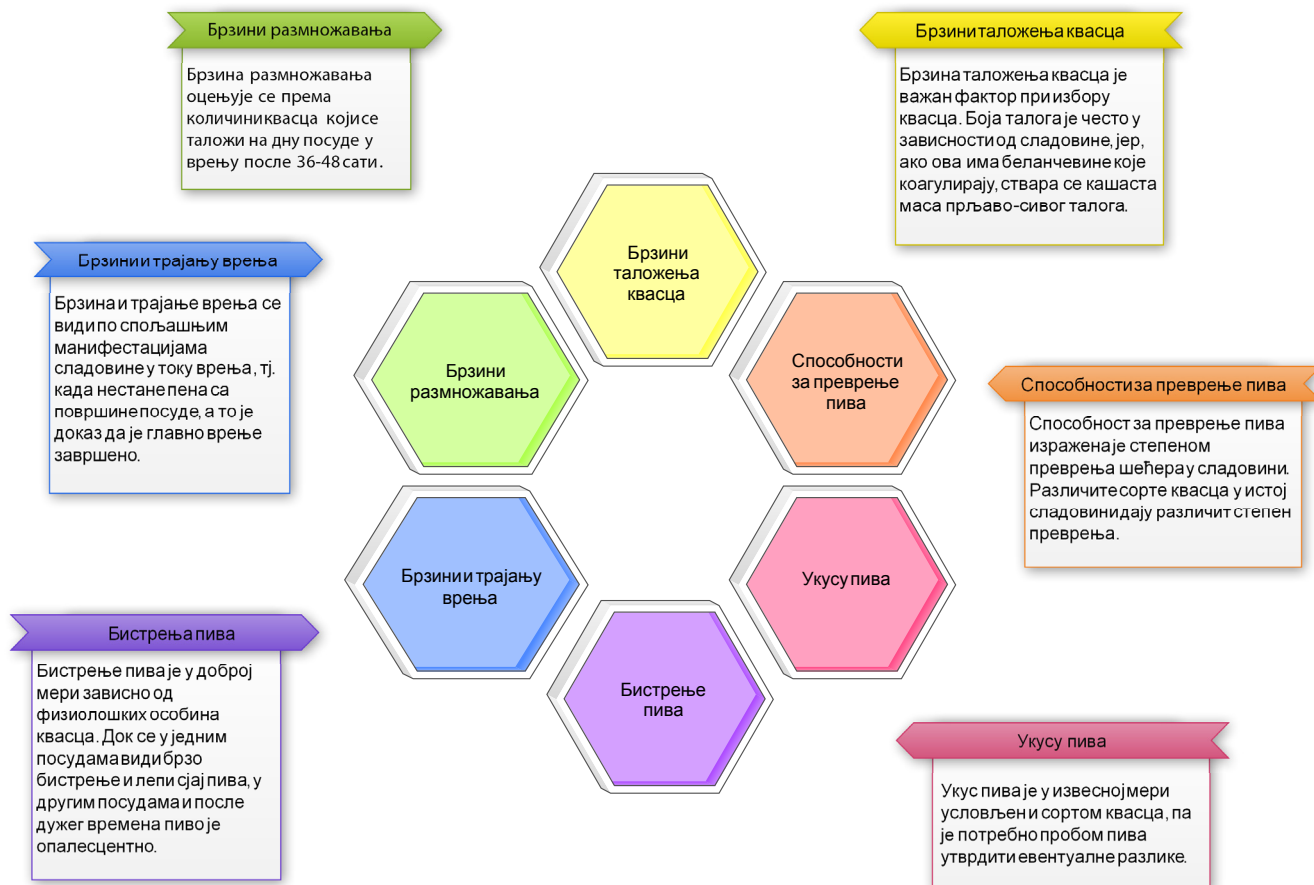
Узрочници врења

Узрочници свих врста врења су микроорганизми који производе ензиме као специфичне изазиваче хемијских промена. Све те микроорганизме можемо поделити у три главне групе, и то:

- Бактерије
- Квасци
- Плесни



За производњу пива служе **пивски квасци**. Квасци су једноћелијски организми елипсоидног, угластог, овалног, или дугуљастог облика, који се редовно размножавају пумпањем. У неповољним приликама, када квасци не располажу са довољно храњивих материја размножавају се спорама које настају унутар ћелија. У пиварству квасци се деле на пахуљасте и прашкасте. Пахуљасте квасци имају нижи степен преврења сладовине јер се згрушавају у пахуљице и имају мању контактну површину са сладовином. Прашкасти квасци имају виши степен преврења сладовине, јер је већа површина њихових ћелија у контакту са сладовином. Ови квасци имају одређене особине значајне за ток алкохолног врења. Оне се огледају у следећем:



Карактеристике врења

На сладовини приликом врења долази до видних промена. Те промене се могу поделити на следеће фазе:

1. Ниска бела пена
2. Средња пена
3. Висока пена
4. Опадање пене
5. Хлађење и таложење

Први знаци врења се показују стварањем белог покривача од пене коју ствара угљен-диоксид, а који се ослобађа приликом врења и прелази у атмосферу. При крају ове фазе врења почињу се накопљати мале количине хмељних смола и беланчевинских материја, што се назива **фаза ниске беле пене**.

У стадијуму **средње пене** мехурићи угљен-диоксида развијају се интензивније и настаје гушћа компактна пена, која се почиње подизати. Услед појачаног издвајање хмељних смола, пена добија жућкасто-смеђу боју. Боја зависи од хемијског састава и концентрације сладовине и сорте квасца и количине хмеља.

Трећа фаза представља фазу најинтензивнијег зрења. У овој фази температура достиже максимум, а настајање пене је најинтензивније.

Четврта фаза је карактеристична по томе што се у њој поред смењивања пене и њеног постепеног пропадања долази до флокулације пене и бистрења пива. Пена се постепено губи и на крају површина пива остаје покривена танким слојем пене. Производ добијен на крају ове фазе назива се **“младо пиво”**.

У овој фази се пиво хлади како би се убрзало таложење квасца и постигла температура 4-5° С. За одређивање завршетка главног зрења служи степен преврелости и спољни знаци на површини сладовине, односно **“младог пива”**. Пиво се узима у чашу од 50 ml и када се добро просветли треба да показује суспендоване честице на дну и то би био знак да се добро избистрило и да је процес главног врења завршен.

1.3.2 Накнадно зрење и врење

За накнадно врење карактеристична је лагана ферментација шећера, при којој се одигравају исти процеси као код главног врења, али они теку спорије. Брзина биохемијских реакција процеса се смањује услед нижих температура и мањег броја ћелија квасца по јединици запремина пива, јер се главни део квасца уклања по завршетку главног врења. У току накнадног врења оксидо-редукциони потенцијал пива опада, том приликом долази до оксидације нестабилних материја, услед чега се појављује фино оксидационо замућење, које се само делимично одстрањује филтрацијом, а остатак се отклања природним таложењем. Ово бистрење пива представља другу фазу врења и састоји се у томе што ћелије квасца апсорбују беланчевине замућења и друге суспендоване материје које падају на дно. У току накнадног врења коагулишу се беланчевине, танинске материје и хмељне смоле. У току зрења мења се укус младог пива, долази до смањења горчине што се објашњава процесом коагулације и разградње хмељних смола.

1.4 Бистрење и истакање пива. Готово пиво

1.4.1 Филтрација пива

Познато је да у току процеса главног врења и зрења долази до одређених физичко-хемијских и орвано-лептичких промена у саставу и особинама пива. Пиво се бистри, ћелије квасца и друге суспендоване материја беланчевинско-танинског комплекса се таложе, а ослобођени угљен-диоксид се веже за пиво, тако да поприма свој коначан састав, укус и друге особине. Филтрирано пиво треба да буде кристално бистро, са потребном трајности, укусом и осталим карактеристикама. Ради тога, филтрација има велики значај као последња фаза производног процеса, јер тек филтрирано пиво има све оне карактеристике које траже потрошачи. Данас се у свету у циљу постизања бистрења пива примењују два основна поступка, и то:

- Сепарација пива
- Филтрација пива

Сепарација пива

Бистрење пива на сепаратору одвија се на бази центрифугалне силе. Познато је да сладовина и пиво садрже извесне честице различите специфичне тежине које се таложе услед гравитационе силе. Пиво долази под притиском у бубањ у коме су смештени ротирајући тањира. Тањира су смештени на шупље вретено које служи за улаз пива. По бубњу пиво пролази између тањира и основице бубња у простор за таложење. Крупне честице се издвајају услед центрифугалне силе ка периферији, па се на тај начин врши стварање талоба. Овај талоб се у одређеним временским интервалима избацује. Пошто су сепаратори херметички затворени одржава се константан притисак и не долази до губитка пива и смањења концентрације угљен-диоксида.

Филтрација пива

Основна предност филтрације пива је могућност оштрије филтрације и постизање веће бистрине. Квалитет филтрације готовог пива зависи од правилног избора и дозирања **кисел-гура** који одговара захтевима за постизање одређеног степена бистрине пива. У већини случајева честице замућења које треба одстранити из пива су мале па је потребно створити одговарајући порозни слој који ће ове честице задржати приликом филтрације. На тај начин се из течне фазе издвајају честице тако да се добија бистра течност. За филтрирање се обично користе филтрациони улошци који су израђени у виду плоча из комбинације целулозе и памука. На филтрационим улошцима се формира прво један наплавни слој грубог кисел-гура, а затим се наноси кисел-гур финије гранулације који формира филтерски слој који задржава честице и на тај начин омогућава да кроз филтрационе слојеве пролази потпуно чист филтрат, ослобођен свих таложних материја и ћелија квасца.

Карактеристике кисел-гура

Кисел-гур као средство за филтрирање, производи се од слојева диатомеја (алге кременашице). Кад је правилно припремљен најбоље одговара захтевима који се постављају идеалном средству за филтрирање: својим обликованим честицама ствара веома пропустљиву филтер-погачу која омогућаје најбоље филтрационо деловање. У хемијском саставу кисел-гур је највећим делом састављен од силицијумових диоксида и алуминијумових оксида.

Филтрација је последња фаза у производном процесу, који претходи пуњењу пива у бурад, боце и лименке па је нужно да јој се посвети посебна пажња. Како са становништа бистрине и других технолошких захтева за квалитетно пиво, тако и са становништва економичности извођења ове

1.4.2 Истакање пива

Филтрирано пиво долази у танкове под притиском у којима треба да се задржи 8-12 часова пре пуњења. Задржавање пива у танковима има сврху да се пиво охлади на температури 1-2 °C како би се спречило пенушање у процесу отакања пива. Пиво се пуни под изобарометарским притиском. Под овим појмом се подразумева константи против-притисак угљен-диоксида који омогућава нормално пуњење пива и који спречава губитак угљен-диоксида и оксидацију пива.

Хемијски састав и особине пива

Главни састојци готових пива су вода, алкохол, угљен-диоксид и екстракт (непреврели део екстракта). Зависно од сорте пива садржај алкохола се креће од 2-6 %. Поред етил алкохола у пиву се налазе мале количине виших алкохола. У пиву нормално има 0,30-0,40 % угљен-диоксида. Присуство угљен-диоксида делује повољно на укус, даје свежину, а представља и важну компоненту за пенушавост пива. Способност држања пене зависи о количини и хемијском саставу екстракта. Од пива се тражи да при пуњењу у чашу пени, тако да се ова густа и чврста пена задржи на површини најмање 3 минута. Мехурићи угљен-диоксида који се дижу од дна ка површини помажу одржавању пене. Екстракт пива претежно чине угљени хидрати и мања количина протеина, аминокиселина, глицерина и саставних делова хмеља. Од састава екстракта зависи пуноћа укуса пива. Хемијски састав екстракта не зависи само од врсте слада, него и начину производње сладовине и вођењу ферментације, односно степену коначног преврења на крају ферментације.



2. ARDUINO

Ово поглавље упознаје читаоца са Arduino –м, микроконтролерском платформом, која је коришћена током развоја овог пројекта.

2.1 Arduino микроконтролерска платформа

Према, Масиму Банзију, једном од стваралаца Ардуина, Ардуино је микроконтролерска платформа базирана на једноставној улазно/излазној (I/O) штампаној плочи и развојном окружењу у којем је имплементиран програмерски језик.

2.1.1 Предности Arduino -а

Постоје многе доступне микроконтролерске платформе, али је Ардуино добар избор јер је:

- **Хардверски и софтверски пројекат отвореног типа**, тако да су оба, и софтвер и хардвер, приступачни, прилагодљиви и модуларни.
- **Прилагодљив** – нуди многе аналогне и дигиталне улазе, SPI, I²C протоколе комуникације серијског типа, дигиталне и PWM излазе.
- **Лак за коришћење** - повезивање са рачунаром преко USB порта и комуницира са другим уређајима преко стандардног серијског протокола, ради независно од рачунара и уједно комуницира са PC/Macintosh рачунарима.
- **Јефтин** - кошта мање од 30 евра по комаду и долази са бесплатним софтверско-развојним окружењем.
- **Подржан од стране растуће интернет заједнице**. Мноштво програмерских кодова је већ доступно и спремно за коришћење.

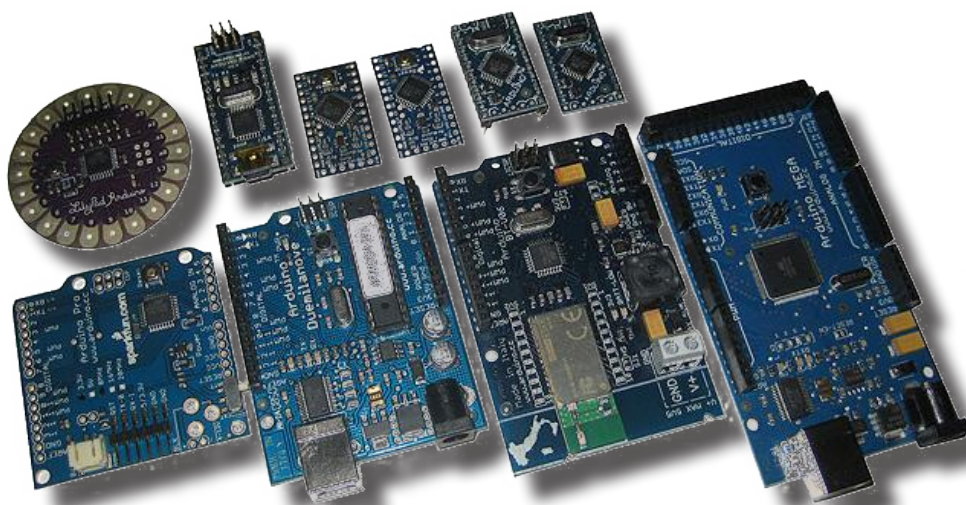
2.1.2 Примарна намена Arduino -а

Ардуино је одличан алат за развој интерактивних објеката, јер прима сигнале са различитих прекидача или сензора, и контролише мноштво светала, мотора и других излазних компонената. Ардуино пројекти могу бити самостални или могу бити повезани са рачунаром путем USB порта. Рачунар ће препознати Ардуино као стандардан серијски интерфејс.

2.2 Хардвер Arduino -а

Као што је већ речено, Ардуино представља микроконтролерску платформу која у себи садржи цео рачунар у једном малом чипу (Микроконтролер). Постоје разне верзије Ардуино платформе: разликују се у компонентама, циљу и величини, итд.. Неки примери Ардуино платформи су: **Arduino Duemilanove/UNO**, **Arduino Mega**, **Arduino Nano**, **Arduino Mini**, итд. Шеме ардуинове платформе се дистрибуирају коришћењем отворене дозволе, тако да је свако слободан да направи личну Ардуино платформу. Ардуино је регистровани бренд, тако да није могуће назвати

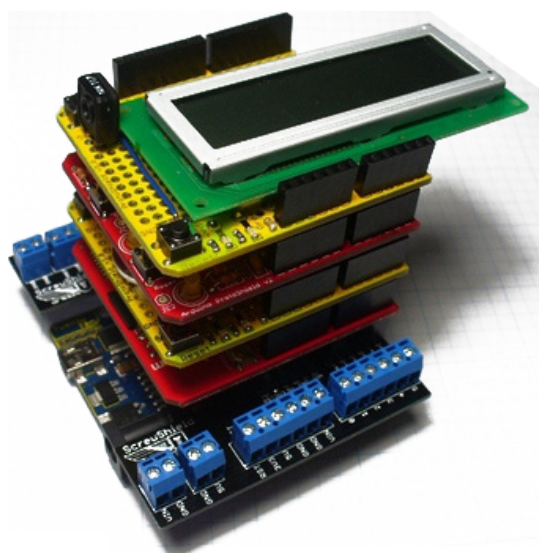
сличне платформе Ардуино: то је разлог зашто је уобичајено наићи на *дуино платформу, као што су Seeeduino, FreeDuino, Japanino, Zigduino, iDuino, итд..



Слика 2.2.1 - Lylipad, Mini, Nano (две), Pro (две), Duemilanove, Mega (с лева на десно)

2.2.1 Ардуино модули (Shields)

Функционалност ардуина може бити побољшана коришћењем модула, *ad hoc* дизајнираних РСВ плоча које имају исти шаблон распореда улазно/излазних пинова као и сам Ардуино, и притом могу бити наслагане једна на другу дајући додатне функције Ардуину. Слика испод приказује екстреман приступ коришћења ових модула.



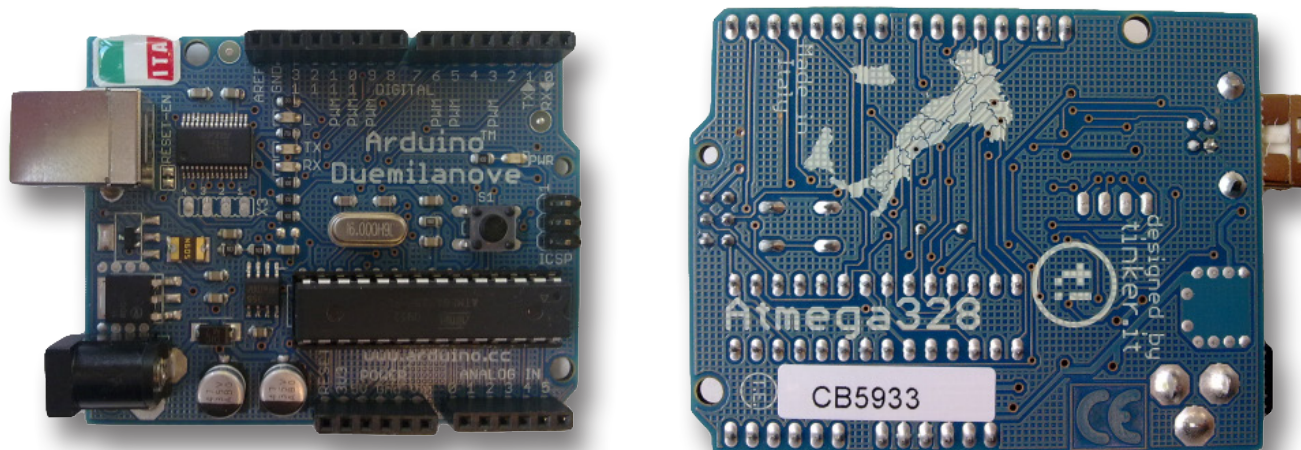
Слика 2.2.1.1 - Ардуино је на дну, а модули су напаковани један на другог (екстреман приступ)

Постоји велики број доступних модула од којих је сваки посебно дизајниран за одређену примену. Неки од њих су развијени од стране произвођача ардуина, док је већина њих развијена од стране других компанија или појединаца. Постоје модули за контролисање рада мотора једносмерне струје, Ethernet комуникацијски протокол, пуштање MP3 датотека, аналогни видео излаз, LCD

екране, итд.. Сама идеја коришћења модула је примењена због олакшавања и превазилажења проблема који се могу десити у процесу развоја неког пројекта, а притом корисника не доводи у ситуацију да сам развија електрична кола за поједине примене.

2.2.2 Arduino Duemilanove/UNO

Према речима произвођача Ардуина, Ардуино Дуемиланове представља најједноставнију платформу за коришћење и примену у пракси.

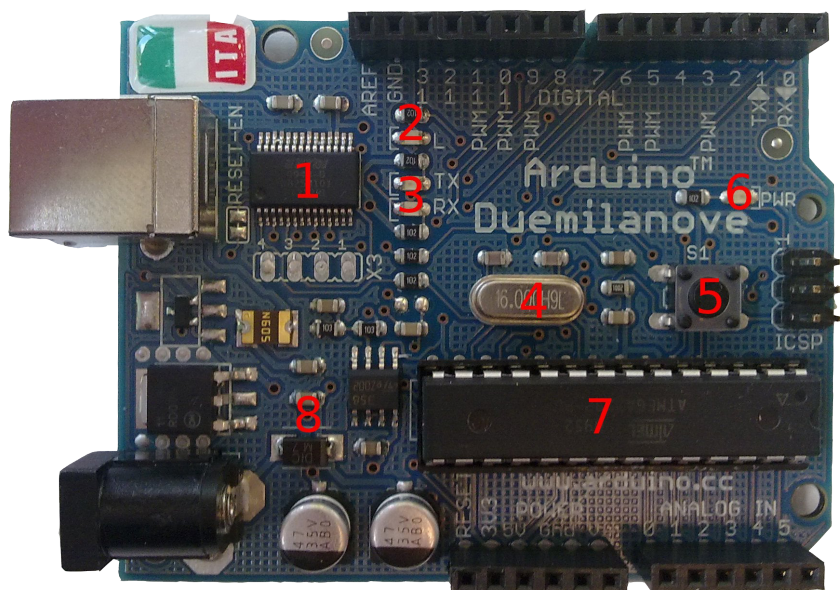


Слика 2.2.2.1 - Ардуино Дуемиланове снимана са предње и задње стране

2.2.2.1 Унутрашње компоненте - Arduino Duemilanove

На слици испод су побројане најбитније унутрашње компоненте на плочи, које ће бити у наставку описане.

1. **FTDI чип** - ово је компонента која омогућава Ардуину да комуницира са рачунаром преко USB порта. Микроконтролер Ардуина може само да има серијску комуникацију. FTDI чип конвертује серијске сигнале до USB порта и обратно. Такође има унутрашњи регулатор напона који конвертује напон од 5 V који долази са USB порта на 3.3 V.
2. **Статусна LED диода** – Повезана је са тринаестим пином преко отпорника од 1 Kohm. Сваки пут када напон са микроконтролера дође до пина 13, LED диода ће засветлети.
3. **Серијске TX и RX LED диоде** – оне служе као индикатори комуникације са рачунаром или другим серијским уређајем (кроз дигиталне пине 0 и 1).
4. **16 MHz кристал** – ово је компонента која се понаша као сат према микроконтролеру. У основи она генерише on-off сигнал који микроконтролер користи да би променио своје стање.
5. **Reset тастер** – Једном притиснут ресетује микроконтролер.
6. **LED диода** која означава да је микроконтролер под напоном – ова диода ће бити укључена када је ардуино повезан на било који извор напона, показујући да је микроконтролер укључен.



Слика 2.2.2.2.1 - Ардуино Дуемиланове (предња страна) са обележеним главним компонентама.

7. Ово представља **микроконтролер, срце Ардуина** – Дуемиланове користи Atmel ATMEGA 328p, осмобитни AVR RISC базиран микроконтролер, који комбинује 32 килобајта флеш меморије са read-while-write могућношћу, 1 килобајт EEPROM-а, 2 килобајта SRAM-а, 23 улазно/излазна порта за универзалну употребу, 32 регистра, три прилагодљива тајмера/бројача са упоредним модовима, унутрашње и спољашње прекиде (енг. interrupts), серијско програмабилни USART, бајтно оријентисани двожицни серијски интерфејс, SPI серијски порт, шестоканални 10-обитни аналогно/дигитални конвертер способан да ради и до 200 KHz, програмабилни watchdog тајмер са унутрашњим осцилатором и пет софтверско-изборних учитавајућих модова. ATMEGA328p ради између 2.7 и 5.5 V.
8. **Разне компоненте** – кондензатори, диоде и регулатори напона. Они се користе да би се стабилизовао извор напона, да би се конвертовао на тачан напон који је потребан Ардуину и да би се спречила штета од кратких спојева.

2.2.2.2 Конекције – Arduino Duemilanove

Кључна ставка због чега је изабрана Ардуино платформа је број доступних конекција. Ово су компоненте које омогућавају повезивање платформе са другим компонентама (сензорима, потрошачима, тастерима, итд.) тако да она може да има интеракцијску комуникацију са њима, а то је: читање улазних података, слање података, премештање података, итд.

Као што можете видети на слици изнад, Ардуино 2009 платформа има следеће конекције:

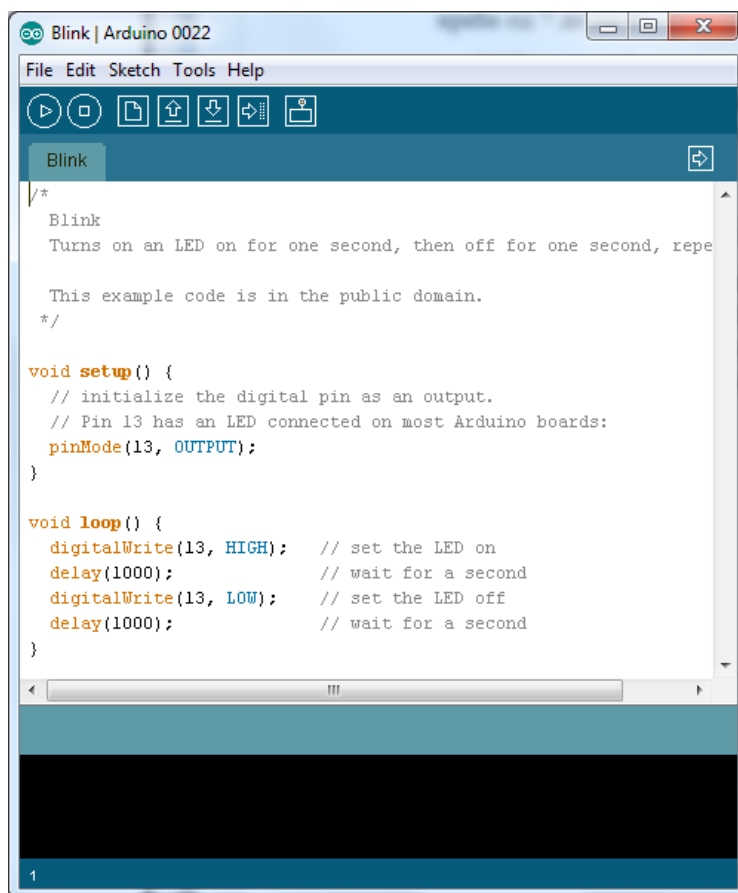
- **AREF: Analog Reference Pin** – Напон у овом пину одређује напон који ће аналогно дигитални конвертери (ADC) пријавити у облику децималних вредности у распону од 0 до 1023, што представља интервал од 0 до 5 V, или напон на коме ради Ардуино. Ово значи да ћете коришћењем овог пина бити у могућности да промените максималну вредност коју читају аналогно-улазни пинови: ово је начин да промените однос величина аналогно улазних сигнала.

AREF pin је стандардно повезан на AVCC напон од око 5 волти.

- **GND: Digital Ground** – Користи се као нула код дигиталних улаза и излаза.
- **DIGITAL 0-13: Дигитални пинови** – користе се за дигитални улаз и излаз.
 - **TX/RX Pins: 0-1** Серијски улази и излаз – Ови пинови могу бити коришћени за дигитални улаз и излаз као и дигитални пинови од 2 до 13, али не могу бити коришћени уколико се користи серијска комуникација.
 - **External Interrupts Pins: 2-3** – ови пинови могу бити конфигурисани да покрену прекид на различитим улазним стањима.
 - **PWM: 3, 5, 6, 9, 10, 11** – омогућавају осмобитни PWM излаз са `analogWrite()` функцијом.
 - **LED: 13** – постоји стандардно уграђена LED диода повезана на дигитални пин 13. Када је пину додељена вредност HIGH, онда је диода упаљена, а када је вредност пина 13 LOW, онда је диода угашена.
- **ICSP: In-circuit Serial Programmer** – Ардуино долази са бутлодером (енг. *bootloader*) који омогућује учитавање програма преко USB порта до серијског интерфејса. Напредни корисници, могу такође да учитају програме на ардуино платформу користећи екстерне учитаваче. Ово се може постићи коришћењем ICSP хедер фајла. На овај начин је могуће уштедети и до 2 килобајта меморије на меморијском простору микроконтролерског чипа.
- **ANALOG IN: 0-5 Аналогни улазни пинови** – Користе се за читање аналогног извора напона (потенциометри, фотоотпорници или температурни сензори).
- **POWER пинови** – Користе се да пруже или приме напајање Ардуино платформи.
 - **Vin** – представља пин за екстерни извор напајања. Када користите овај напонски пин, добијате исти напон који даје екстерни извор напајања. Такође је могуће користити овај конекциони пин за напајање саме Ардуино плоче.
 - **GND (2 пина)** – користе се као 0 пинови (дигиталне нуле).
 - **5 V** – Овај пин се користи за добијање 5 V са Ардуино плоче. Овај исти напон се користи за напајање микроконтролера. Овај напон се може добити и са екстерног напајања или директно са USB порта.
 - **3V3** – 3.3 V пин представља напонски извор који је генерисан са FTDI чипа. Максимална струја која може да се повуче са овог пина је 50 mA. Генерално треба избегавати коришћење овог напонског пина, зато што је осетљив на кратке спојеве и прелазак максималних 50 mA, који могу да доведу до оштећења FTDI чипа.
 - **RESET** – Довођењем логичке нуле на овај пин Ардуино се рестартује. Такође постоји дугме на плочи које служи истој сврси, али приликом коришћења ардуино модула дугме се не може више користити, јер је неприступачно.
 - **External Power Supply In** – овај улаз се користи да би се довео напон са екстерног регулатора напона. Користи се централно позитиван конектор пречника 2.1 mm. Напонски интервал ових регулатора треба да буде од 6 до 20 V, али да би се спречило прегрејавање и проблеми око стабилности рада плоче, препоручени напонски интервал се креће од 7 до 12 V.
 - **USB** – помоћу овог улаза се уноси (енг. *upload*) код, претходно формиран на рачунару. Такође, овај улаз се користи за серијску комуникацију између компјутера и микроконтролера. Ардуино се може напајати са USB порта.

2.3 Програмско окружење Arduino -а

Још једна важна компонента у концепту Ардуино платформе је његово програмско окружење (енг. Integrated Development Environment - IDE). Ардуиново програмско окружење садржи све што је потребно да би рачунар могао на правилан начин да комуницира са Ардуино платформом.



Слика 2.3.1 - Програмско окружење Ардуино-а

Програмско окружење Ардуино-а у себи садржи едитор који се користи за програмирање и формирање програма.

Коришћењем овог едитора, програм који пишемо је конвертован у C програмски језик (користи се његова синтакса), касније компајлиран помоћу *avr-gcc* компајлера, бесплатан програмски компајлер базиран на „Gnu C Compiler” (*gcc*) компајлеру, специјално дизајнираним за AVR микроконтролере. AVR микроконтролери представљају модификовану верзију харвардске осмобитне RISC архитектуре једночипних микроконтролера, дизајнираних од стране Atmel компаније 1996. године. Овај процес своди виши програмски језик на ниво бинарног кода који микроконтролер на Ардуино платформи може да процесуира и изврши.

Бинарни код се уноси у Ардуинов микроконтролер преко USB конекције. Овај процес се спроводи помоћу *avrdude* програма који у себи има имплементиран комуникацијски протокол који пребацује бинарне информације у ардуинову програмску меморију.

2.3.1 Изглед и функционалност Arduino IDE прозора

Програмско окружење Ардуина, тј. сам његов изглед је прилично једноставан. Састоји се из неколико кључних делова, а то су:

1. Уређивач текста за писање кода (енг. Text editor)

2. Простор за приказ порука
3. Текстуалну конзолу
4. Линију алатки са дугмићима за стандардне функције
5. Низ менија

Повезује се са хардвером Ардуина да би се унели потребни програми и да би комуникација са њима била могућа.

Програми који су написани уз помоћ Ардуино развојног окружења се називају скице (енг. Sketches). Ове скице су написане у уређивачу текста. Уређивач има могућности за уређивање текста као што су cut/paste и search/replace text. Простор за приказ поруке даје повратну информацију док у исто време чува и приказује текст, а има могућност да приказује и грешке. Текстуална конзола приказује текст у развојном окружењу Ардуина укључујући комплетне поруке о грешкама и друге информације. Линија алатки са дугмићима омогућава да верификујете и унесете програме у микроконтролер, да направите, отворите, сачувате скице и да отворите серијски монитор прозор:



Verify/Compile

- Проверава код и налази грешке.



Stop

- Зауставља серијски монитор или не наглашава осталу дугмад.



New

- Прави нову скицу.



Open

- Представља мени свих скица у меморији рачунара. Кликом на једну од њих, она се отвара у оквиру истог прозора у коме сте тог тренутка.

Белешка: због грешке у Јави, овај мени не може да се “скролује”. Уколико треба да отворите скицу касније, користите **File | Sketchbook** мени уместо тога.



Save

- Чува скицу.



Upload to I/O Board

- Компајлира код и убацује га у Ардуинову I/O меморију.



Serial Monitor

- Отвара серијски монитор.

Додатне команде се налазе унутар пет менија: **File, Edit, Sketch, Tools, Help**. Менији су осетљиви на садржај што значи да су доступне једино ставке које су битне за посао који се тренутно обавља.

File Edit Sketch Tools Help

- **Copy for Forum**

Копира код скице на *clipboard* у формату који одговара оном за форуме, што вам омогућава да исти поставите на форум и употпуните бојењем текста.

- **Copy as HTML**

Копира код скице на *clipboard* као HTML код, што вам омогућава да га поставите на web страницу.

- **Verify/Compile**

Проверава да ли има грешака у скици.

- **Import Library**

Додаје библиотеку вашој скици тако што убацује **#include** наредбу у код.

- **Show Sketch Folder**

Отвара фолдер са скицама на позадини рачунара.

- **Add File...**

Додаје фајл скици (биће копиран са своје тренутне локације). Нов фајл се појављује у новој картици у прозору са скицом. Фајлови могу бити уклоњени са скице користећи **tab** мени.

- **Auto Format**

Ова опција форматира код, нпр. увлачи га тако да га отварање и затварање витичасте заграде поравнавају, а коментари се увлаче више.

- **Board**

Помоћу ове опције се бира микроконтролерска платформа.

- **Serial Port**

Овај мени садржи све серијске уређаје (праве или виртуелне) на машини. Требало би да се аутоматски освежи када отворите Tools мени.

- **Burn Bootloader**

Ставке у овом менију омогућавају убацивање "bootloader"-а у микроконтролер на Ардуино платформи. Ова ставка није неопходна за стандардну употребу Ардуино платформе, али је корисна у случају стављања новог Atmega чипа (који обично долази без "bootloader"-а). Веома је важно одабрати одговарајућу платформу из менија Boards пре него што покренете убацивање "bootloader"-а. Када користите AVR ISP, мораћете да изаберете ставку из Serial Port менија.

Sketchbook

Развојно окружење Ардуина укључује и концепт фолдер са скицама (енг. Sketchbook): то је место где се складиште програми (или скице). Скице у књижици могу бити отворене из менија **File > Sketchbook** или уз помоћ **Open** дугмета у линији са алаткама. Када се програм Ардуина први пут покрене, он аутоматски направи потфолдер у основном фолдеру са скицама. Локација се може видети или променити уз помоћ опције **Preferences**.

Kartice, mnogobrojni fajlovi i kompilacije

Омогућавају да управљате већим бројем скица са више фајлова (од којих се сваки појављује на новој картици). То могу бити стандардни Ардуино фајлови (без екстензије), C фајлови (.c екстензија), C++ фајлови (.cpp) или фајлови библиотека (.h).

2.3.1.1 Уношење скице у микроконтролер

Пре него што се унесе скица, изаберу се одговарајуће ставке из менија **Tools > Board i Tools**

> **Serial Port.** Платформе су описане испод. На Мекинтош рачунарима, серијски порт је вероватно `/dev/tty.usbserial-1B1` (за USB протокол), или `/dev/tty.USA19QW1b1P1.1` (за серијски протокол повезан уз помоћ Keyspan USB-to-Serial адаптера). На рачунарима са Windows оперативним системом, вероватно је **COM1** или **COM2** (за серијски протокол) или **COM4**, **COM5**, **COM7**, или више (за USB протокол). На рачунарима са Linux оперативним системом би требало да буде `/dev/ttyUSB0`, `/dev/ttyUSB1` или слично.

Након бирања одговарајућег серијског порта и платформе, помоћу “**upload**” дугмета на линији алатки или ставка **Upload to I/O Board** из **File** менија. Тренутне Arduino платформе ће се ресетовати аутоматски и започеће унос. Услед недостатка аутоматског ресетовања код старијих платформи, мораћете да притиснете дугме за ресетовање на платформи пре почетка уноса података. На већини платформи видећете да **RX** и **TX** LED диоде трепућу док се врши унос скице. Развојно окружење Ардуина ће приказати поруку када је унос завршен или ће показати грешку уколико до ње дође.

Код уношења скице, користите Ардуино “**bootloader**”, мали програм који је лоциран у микроконтролеру на платформи. Он омогућава да се унесе код без коришћења икаквог додатног хардвера. “**Bootloader**” је активан неколико секунди када се платформа ресетује, а онда почиње скицу која је последња унета у микроконтролер. Када “**bootloader**” започне са радом (нпр. када се платформа ресетује), трепериће пин 13 LED диода.

2.3.1.2 Biblioteke (Libraries)

Библиотеке омогућавају додатну функционалност за коришћење скица, нпр. рад са хардвером или управљање подацима. Да би користили библиотеку на скици, мора се изабрати опција **Sketch > Import Library** менија. Ово ће унети једну или више **#include** наредби на врху скице и укључити библиотеку у скицу. Пошто се библиотеке уносе на платформу заједно са скицом, повећаће се количина простора који је заузет. Уколико скици више није потребна библиотека, једноставно брисањем наредбе **#include**, библиотека се неће унети у микроконтролер.

Неке библиотеке су укључене са Ардуино софтвером. Друге се могу набавити од разних извора. Да би се инсталирале библиотеке набављене на овај начин, формира се фолдер у фолдеру са именом **libraries** унутар **Sketchbook** фолдера. Затим се фајлови са екстензијом **.h** стављају у исту, претходно формирану датотеку. На пример, да би се унела **DateTime** библиотека, њени фајлови би требало да буду у `/libraries/DateTime` потфолдеру, **Sketchbook** фолдера.

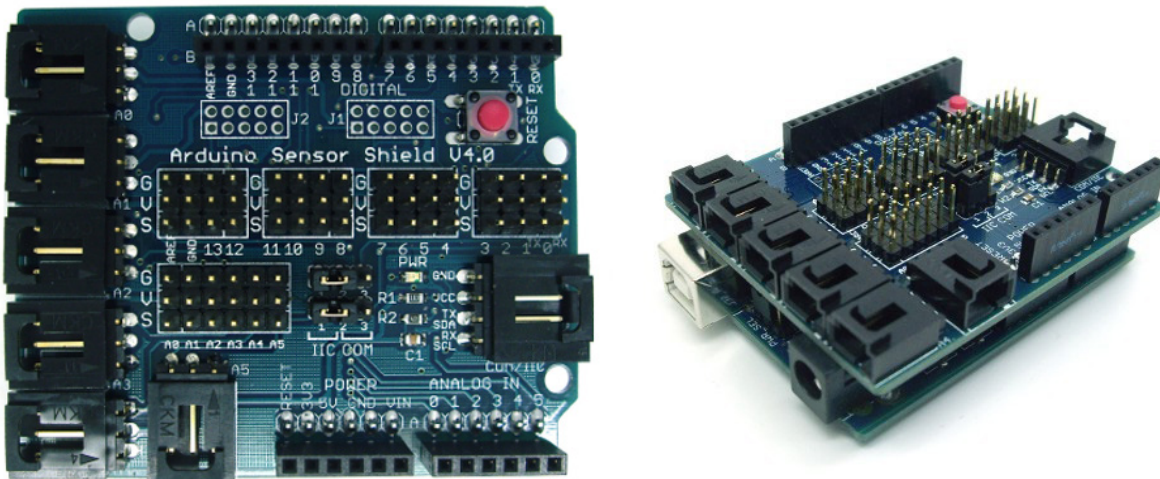
2.4 Примена модула и компонената

У оквиру овог поглавља биће описане компоненте и модули који су коришћени у оквиру овог пројекта. Биће показана спрега између хардвера и софтвера, као и комуникацијских протокола. Свако потпоглавље ће садржати суштинску намену модула или компоненте, основни пример и шему повезивања са Ардуином.

2.4.1 Ардуино сензорски модул (Sensor Shield V4.0)

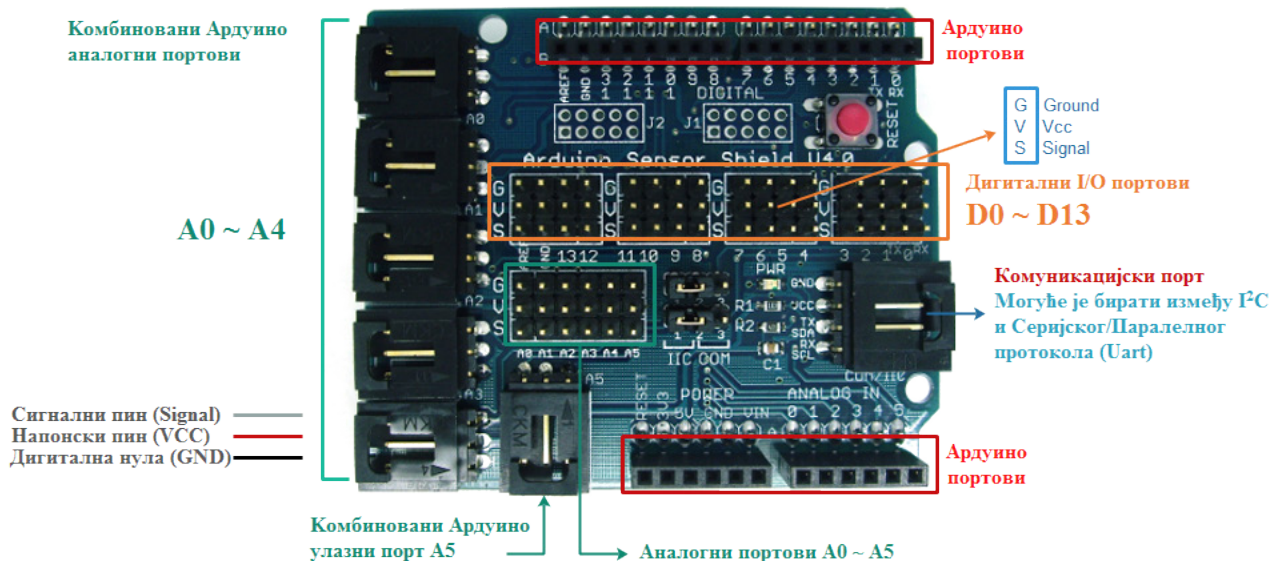
Ардуино **Sensor Shield V4.0** је ту да обезбеди кориснику да релативно лако повеже све жељене компоненте као што су сензори, серво регулатори, релеји, тастери, потенциометри, са Ардуином. Сваки функцијски модул или порт на сензорској плочи има тако конципирани конекцијски шаблон да се на сваком од њих налазе референтни напон (**VCC**), дигитална нула (**GND**), и излазни/

улазни сигнал (SIG). Као и сви Ардуино модули повезивање сензорског модула на платформу је једноставно, јер им се шема улазно/излазних пинова подударају.



Слика 2.4.1.1 - Ардуино сензорски модул и његово повезивање на платформу

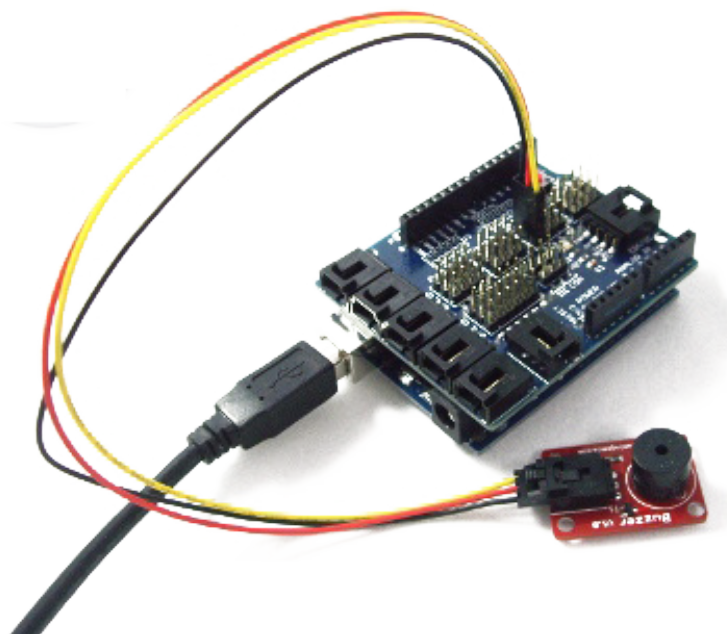
Шема на слици испод показује делове Ардуино **Sensor Shield V4.0** модула и њихове функције на њој. На њој се очигледно види зашто се сензорски модул користи у пракси.



Слика 2.4.1.2 - Функцијска шема Ардуино сензорске плоче

Примена сензорске плоче је основ за почетак сваког већег пројекта који захтева повезивање Ардуина са већим бројем сензора, релеја и других компонената који захтевају константно напајање и дигиталну нулу поред излазног/улазног сигнала. Коришћење ове плоче ни по чему не утиче на ергономичност постојеће ардуино плоче, чак је микроконтролер заштићенији као друге осетљиве компоненте на Ардуино плочи. Сви доступни пинови који се налазе на Ардуино сензорском модулу су доступни за коришћење (неки су дуплирани), тако да ова плоча пружа могућност избора врсте џампер каблова према жељи и могућностима корисника.

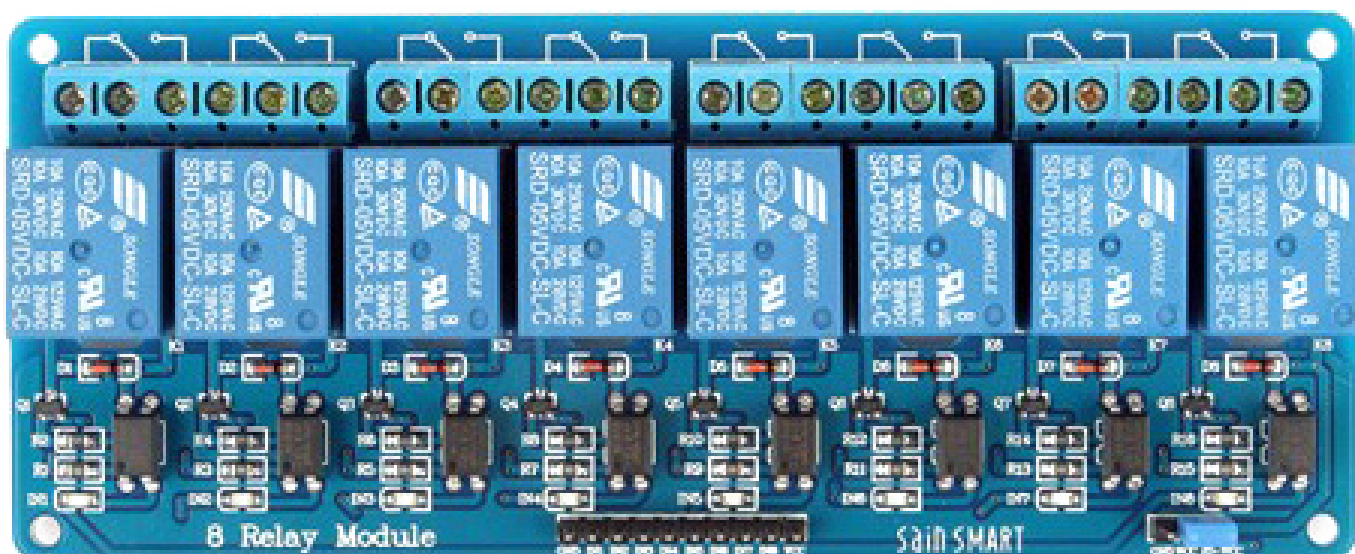
Пример повезивања пиезо звучника (енг. Piezzo Speaker) на Ардуино платформу са прикаченим сензорским модулом се налази на слици 2.4.1.3.



Слика 2.4.1.3 - Пример повезивања пиезо звучника на Ардуино платформу коришћењем Ардуино сензорског модула

2.4.2 Осмо-релејни модул (Arduino 8-relay module)

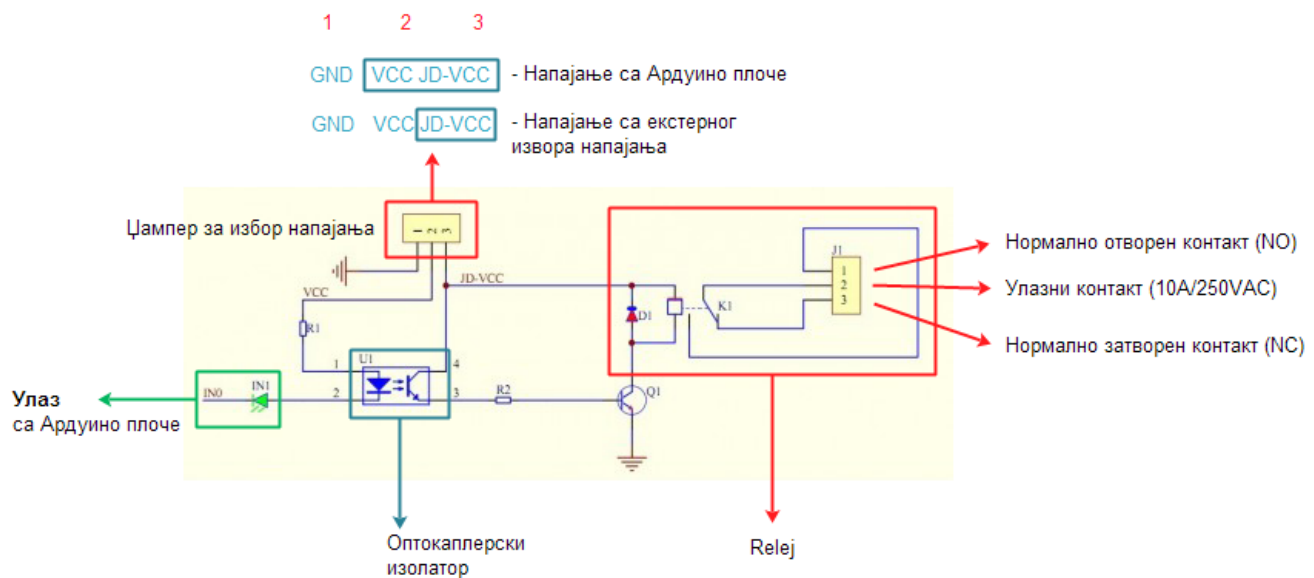
За разлику од модула који се на Ардуино повезују са шематски истим распоредом конекционих пинова, што је случај код Ардуино сензорске плоче, овај модул се не повезује на такав начин већ служи као одвојена компонента, што је у случају осморелејног модула и добро због саме природе и намене ове компоненте - контролисање и манипулисање већих напонских извора од Ардуиновог основног. Осморелејни модул који коришћен у ову сврху је производ фирме **Sein Smart** и приказан је на слици 2.4.2.1.



Слика 2.4.2.1 - Осморелејни модул произвођача Sain Smart

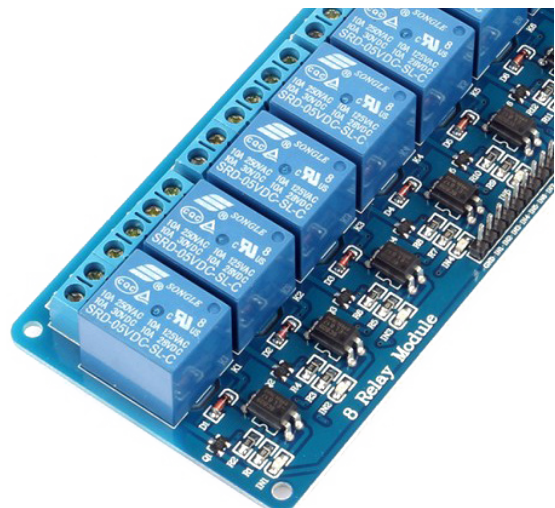
Ова плоча за разлику од стандардних релејски оријетисаних компоненти у себи има уграђене неке напредне компоненте које су карактеристичне за индустријске контролере и модуле, као што је избор између стандардне побуде калема са ардуиновог напајања у оквиру самог релеја и по жељи побуда са екстерног напајања са оптокаплерским изолаторима који у потпуности одвајају релејску плочу од Ардуиновог микроконтролера, што је у случају рада релеја са великим потрошачима (грејачи, електромотори наизменичне струје, итд.) добро због стабилности и сигурности рада микроконтролера.

На слици испод је приказана електронска шема једног релејног склопа у оквиру осморелејног модула.



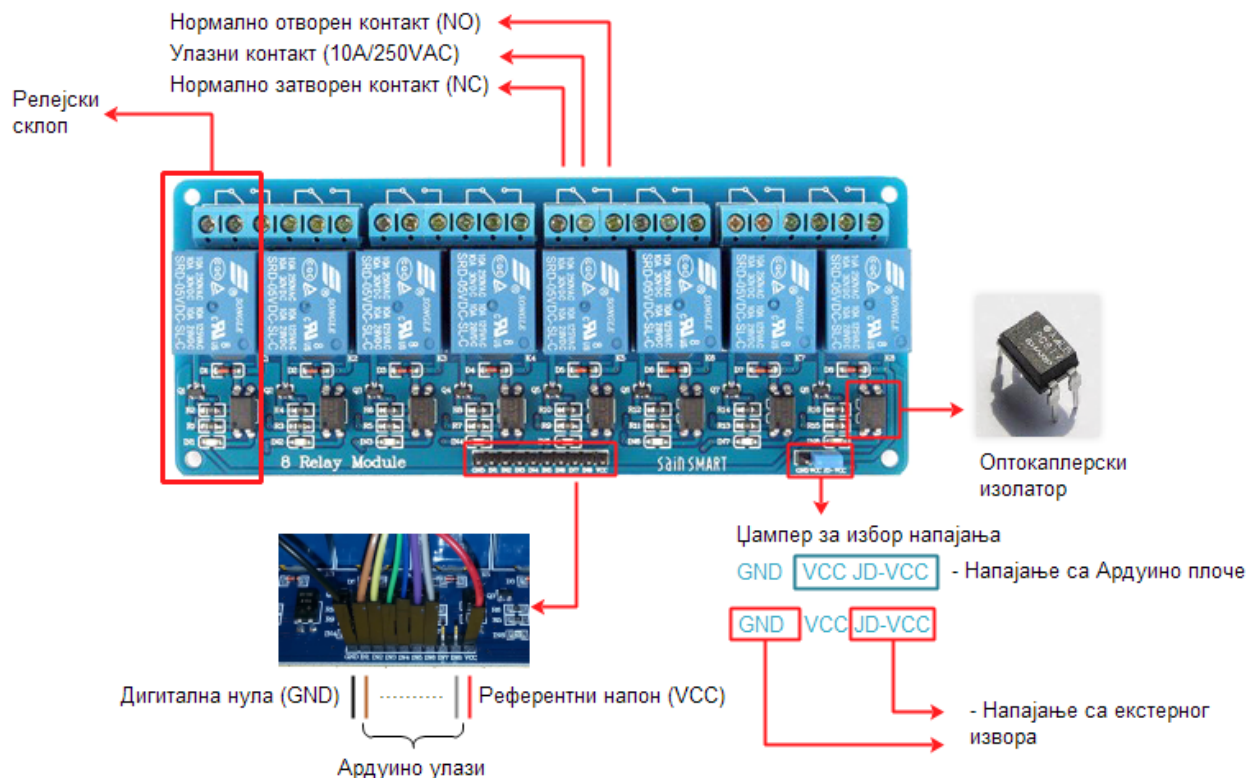
Слика 2.4.2.2 - Елетронска шема једног релејског склопа

Као што се на шеми види, у овој релејској конфигурацији да би се релеј побудио, потребно је да се на улазу са Ардуино плоче доведе логичка нула (дигитална нула). Овај принцип је уведен због самог начина функционисања Ардуино микроконтролерске платформе, јер приликом ресетовања Ардуино-а, сви излазни портови који су дефинисани као излазни у коду имају аутоматски додељену величину HIGH или логичку јединицу, а све се то дешава приликом покретања Ардуино-а пре него што се дође до основне петље унесеног програма. Из тог разлога је систем релејског склопа тако конципиран, у супротном би долазило до паљења релеја приликом покретања Ардуино-а, што није добро. Такође се види да оптокаплерском изолацијом и довођењем екстерног напајања на осморелејни модул, Ардуино платформа се у потпуности одваја од модула што се тиче извора напајања, тако да сметње које се могу десити у току рада релеја неће допрети до осетљивог микроконтролерског напајања.



Слика 2.4.2.3 - Осмо-релејски модул
сликан са стране

На слици испод се налази функционална шема осмо-релејног модула.



Слика 2.4.2.4 - Функцијска шема осморелејног модула

Код коришћења екстерног извора напајања, изузимањем дигиталне нуле са Ардуино платформе, а коришћењем исте са екстерног извора, се обезбеђује да оптокаплерска изолација буде у функцији.

2.4.3 Дигитални температурни сензор DS18B20

Једножичани дигитални температурни сензор DS18B20 произведен од стране Maxim компаније, је одлично решење за мерење температуре где је проблем обезбедити довољно конекцијских пинова на микроконтролерској платформи. Поготово када се користи више од једног температурног сензора у пројекту. Изглед коришћеног температурног сензора је приказан на слици 2.4.3.1.

Слика 2.4.3.1 - Водоотпорни температурни сензор DS18B20



Као што се на слици види, коришћена је водоотпорна верзија овог дигиталног сензора због намене у пројекту.

Када се помене реч дигитални, прво на шта се помисли су проблеми који настају у комуникацији са оваквим уређајима. Аквизиција података са овог сензора не функционише на начин на који се подаци прикупљају са аналогних температурних сензора (термопарови, термистори, итд.) већ на дигиталном нивоу, то значи да морају да се испоштују одређени комуникацијски протоколи који су имплементирани у ову компоненту. На сву срећу фирма **Maxim** се потрудила да обезбеди корисницима овог сензора библиотеку за примену истог код Ардуино пројеката. Назив ове библиотеке је **DallasTemperature**. Она не може самостално да управља овим сензором, већ је потребно имати дефинисану и **OneWire** библиотеку у оквиру Ардуино развојног окружења.

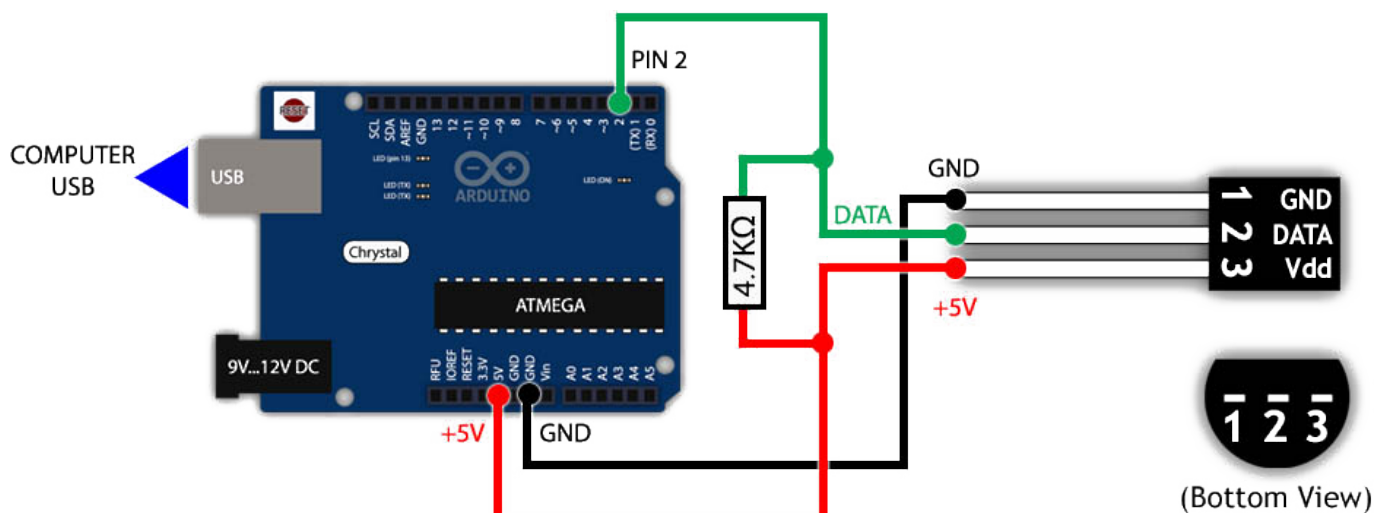
Пре него што се крене са програмирањем и коришћењем овог сензора на софтверском нивоу, потребно је правилно повезати ову компоненту. На слици 2.4.3.1 и на слици 2.4.3.2 се налазе референце пинова и њихова значења, што је неопходно знати пре повезивања на платформу.

Реч једножичани у претходном тексту се односила на комуникацијски ниво и врсту овог дигиталног сензора. Постоји два начина на које могу да се повежу компоненте које функционишу на једножичаном принципу.

- Паразитски тип напајања (потребно је само две жице GND и SIG)
- Стандардно напајање (потребно је повезати све три жице)

Пошто потреба за смањењем броја конекција са две на три жице и то на нивоу напајања не постоји због коришћења Ардуино сензорске плоче, у даљем тексту ће бити описано повезивање овог сензора применом стандардног напајања.

Приликом повезивања сензора преко стандардног напајања потребне су три жице: напајање са Ардуино-а (VCC), дигитална нула (GND) и сигнал за аквизицију података (data). Такође, је потребно повезати и **4k7 pullup** отпорник између сигналне жице и извора напајања, показано на слици 2.4.3.2.



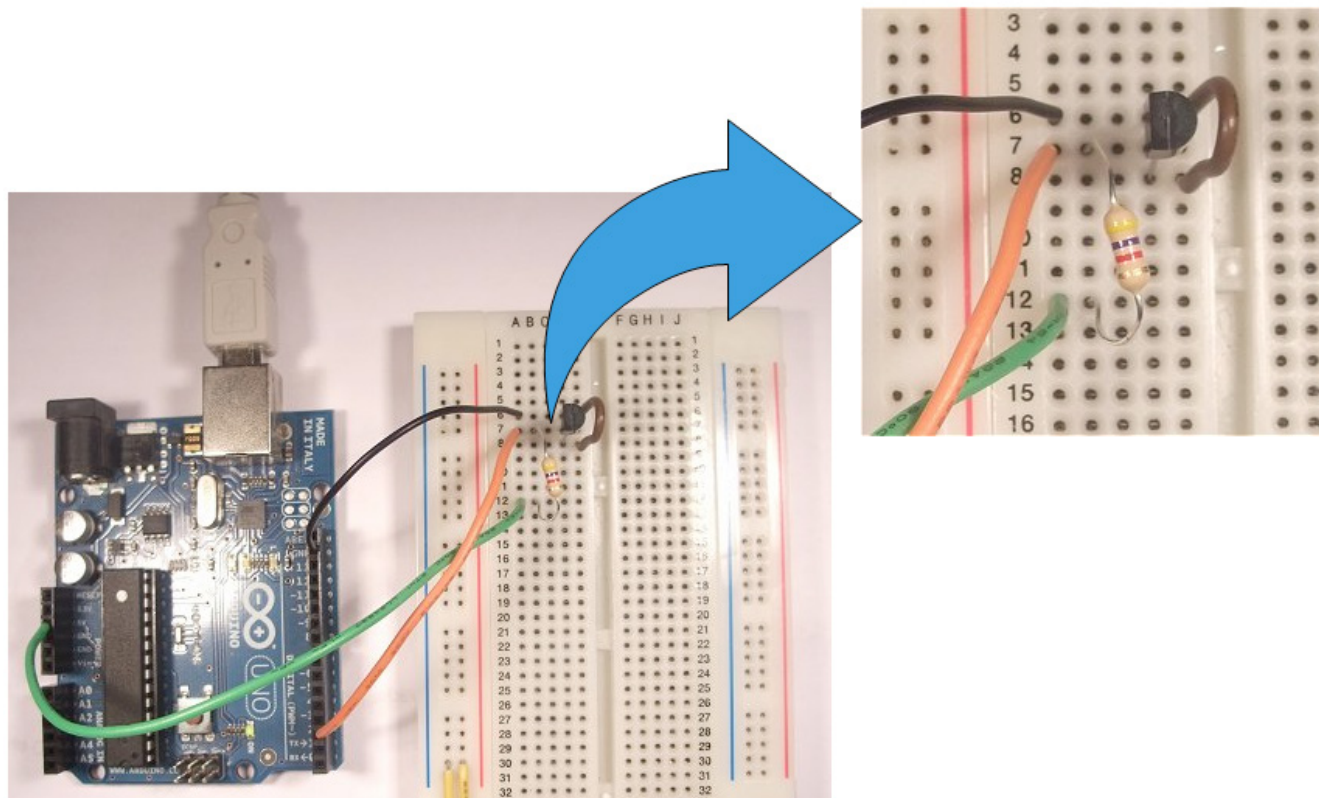
Слика 2.4.3.2 - Повезивање DS18B20 сензора са Ардуином

Такође, на овај начин микроконтролер има константан приступ подацима које даје сензор, за разлику од паразитског напајања, где постоји кашњење од минимум 750 ms (милисекунди), због природе аквизиције. Приликом коришћења овог сензора и у случају да њихов број не прелази 3 сензора на једној жици, коришћење отпорника од 4.7 kOhm -а биће довољно, али код већег броја сензора, отпор се треба смањити на 1.6 kOhm -а, због унутрашње отпорности самих дигиталних сензора.

Свака једножичана компонента (1-Wire) садржи јединствени 64-битни ROM код, који садржи

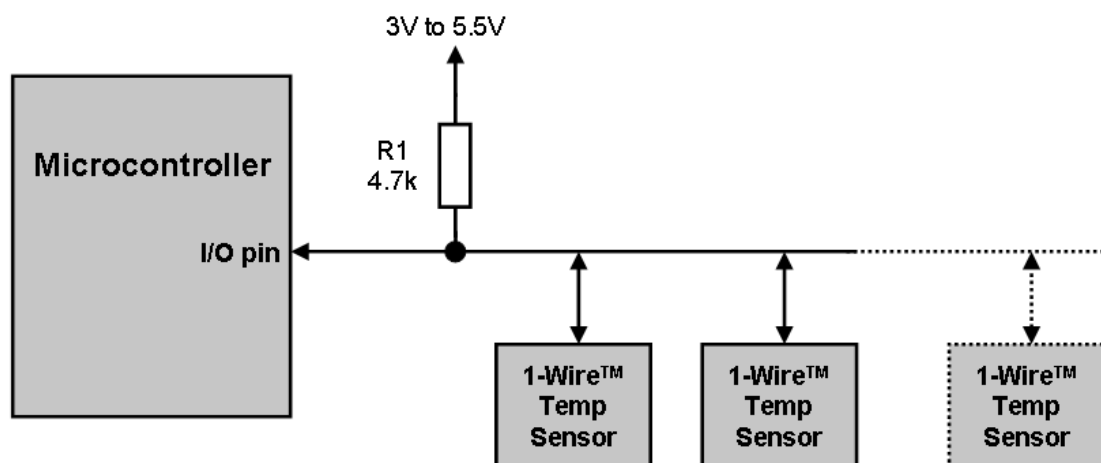
осмобитни фамилијарни код, чедрдесетосмобитни серијски број и осмобитни CRC (Cyclic Redundancy Check). CRC се користи да би се потврдио интегритет података које сензор шаље. Процес који се одвија у самом сензору приликом аквизиције неће бити описан.

За напајање сензора је потребан напон од 3 до 5 волти на VCC конектору и 0 волти на GND конектору. На слици испод је приказано повезивање DS18B20 сензора са Ардуином.



Слика 2.4.3.3 - Повезивање DS18B20 дигиталног температурног сензора са Ардуином

На слици 2.4.3.3 се види да сензор није водоотпоран, али је купљен у сврхе примене са прото-плочом (енг. protoboard). Једножичани протокол се односи на комуникацијски протокол, као што је већ поменуто, али није показано. На слици 2.4.3.4 је приказана шема и начин на који је могуће повезати више истих дигиталних температурних сензора. Примена ове технике је од изузетног значаја, када се ради на великим пројектима, а број места на микроконтролерској платформи је ограничен.



Слика 2.4.3.4 - Шематски приказ повезивања температурних сензора DS18B20

Након повезивања сензора на начин који је показан на слици 2.4.3.3 можемо почети са писањем програма у Ардуино развојном окружењу. Тест пример програма који коришћен у ове сврхе је приказан на слици 2.4.3.5.

```
#include <OneWire.h>
#include <DallasTemperature.h>

// Data конектор је прикачен на ардуинов пин 2
#define ONE_WIRE_BUS 2

// Подешавање oneWire библиотеке да би била правилно успостављена комуникација са сензором

OneWire oneWire(ONE_WIRE_BUS);

// Прослеђивање oneWire референце (библиотеке) ка библиотеци Dallas Temperature
DallasTemperature sensors(&oneWire);

void setup(void)
{
    // Покретање серијске комуникације између Ардуина и рачунара
    Serial.begin(9600);
    Serial.println("Dallas Temperature IC Control Library Demo");

    // Покретање библиотеке oneWire
    sensors.begin();
}

void loop(void)
{
    // позивање sensors.requestTemperatures() да бих добили податке о тренутној температури
    // позивање свих уређаја на мрежи
    Serial.print(" Requesting temperatures...");
    sensors.requestTemperatures(); // Слање команде ка сензорима да би добили температуре са њих
    Serial.println("DONE");

    Serial.print("Temperature for Device 1 is: ");
    Serial.print(sensors.getTempCByIndex(0)); // Зашто тражимо температуре по индексу
    // На мрежи се могу наћи и други уређаји са oneWire протоколом
    // 0 ставља референцу да је то први сензор у линији

}
```

Слика 2.4.3.5 - Пример тест програма коришћен у сврхе добијања и тестирања дигиталног сензора

Након убацивања овог тест примера у Ардуинов микроконтролер и покретањем серијског монитора у оквиру Ардуино развојног окружења почећемо да добијамо следеће податке:

Dallas Temperature IC Control Library Demo

Requesting temperatures...DONE

Temperature for Device 1 is: 24.37 Requesting temperatures...DONE

Temperature for Device 1 is: 24.37 Requesting temperatures...DONE

Temperature for Device 1 is: 24.37 Requesting temperatures...DONE

Temperature for Device 1 is: 24.37 Requesting temperatures...DONE

Temperature for Device 1 is: 24.37 Requesting temperatures...

На слици испод се налази код чиме је приказано адресирање више сензора у низу. Код није комплетан јер је мало опширнији због коришћења LCD екрана за исписивање вредности добијених са дигиталних температурних сензора.

```
#define Temp 10 // Сензор DS18B20 је повезан на пин 10 ардуино платформе
OneWire oneWire(Temp); // Дефинисање променљиве Temp у оквиру oneWire библиотеке
DallasTemperature sensors(&oneWire); // Постављање референце за oneWire библиотеку

void setup() /*----( SETUP: RUNS ONCE )----*/
{
    Wire.begin();
    lcd.begin(16,2); // иницијализација LCD екрана 16x2 са I2C протоколом комуникације
    lcd.backlight();
    lcd.clear();
    lcd.createChar(1,termometru);
    sensors.begin();
}
void loop() /*----( ОСНОВНА ПЕТЉА )----*/
{
    sensors.requestTemperatures();

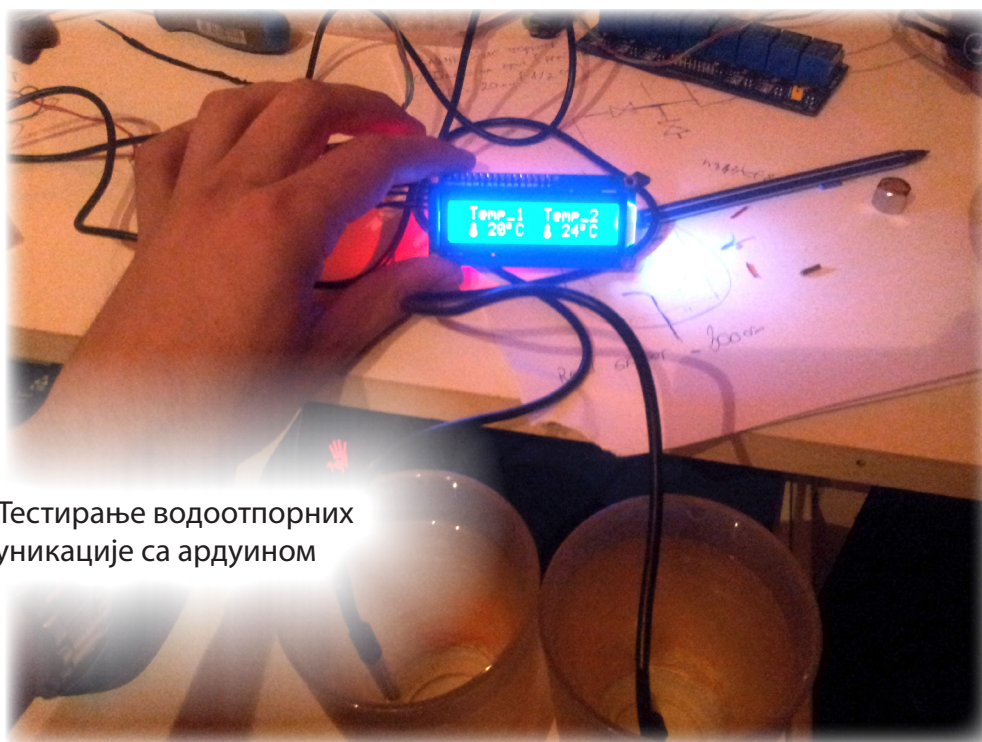
    lcd.setCursor(1, 0);
    lcd.print("Temp_1");
    lcd.setCursor(1, 1);
    lcd.write(1); // штампање знака за термометар
    lcd.setCursor(3, 1);
    lcd.print((float)sensors.getTempCByIndex(0), 1); // адресирање првог сензора у низу
    lcd.setCursor(5, 1);
    lcd.print((char)223); // знак за степен
    lcd.print("C");

    lcd.setCursor(9, 0);
    lcd.print("Temp_2");
    lcd.setCursor(9, 1);
    lcd.write(1); // штампање знака за термометар
    lcd.setCursor(11, 1);
    lcd.print((float)sensors.getTempCByIndex(1), 1); // адресирање другог сензора у низу
    lcd.setCursor(13, 1);
    lcd.print((char)223); // знак за степен
    lcd.print("C");

    delay(1000);
}
```

Слика 2.4.3.6 - Пример кода при коришћењу два сензора

Резултат је приказан на слици 2.4.3.7.



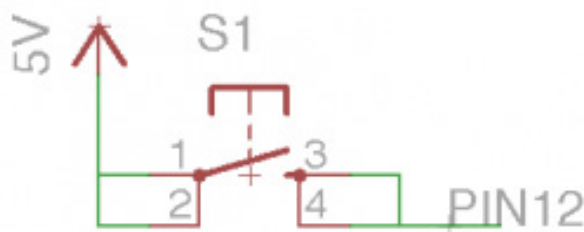
Слика 2.4.3.7 - Тестирање водоотпорних сензора и комуникације са ардуином

2.4.4 Тастери

Приликом повезивања тастера на ардуинове улазно/излазне портове, резултати читавања на тим портовима могу бити насумични, тј. несигурни. Понеки пут ће се десити да приликом стискања тастера се дешава кашњење иако је програм веома једноставан и петља кратка. Такође, може се десити да се стање улазног пина мења са HIGH ка LOW без икаквог разлога. Оно што највише може да изнервира програмера је да се стање пина мења само приближавањем прста тастеру. Сви ови примери представљају проблеме који могу да се десе приликом повезивања тастера са ардуином.

Решење за ове проблеме је једноставно - **коришћење PULLUP отпорника**.

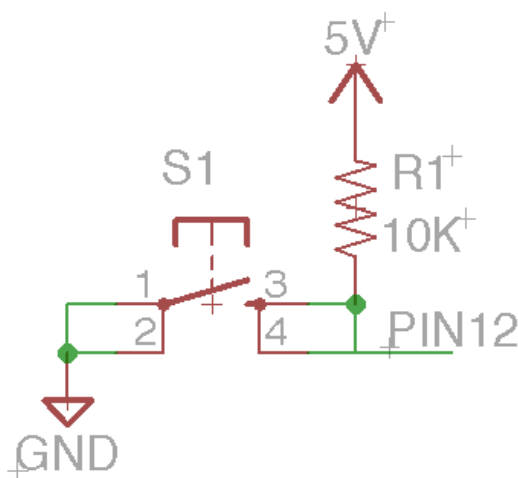
Повезивањем тастера на ардуинов улаз као на шеми испод долази до горе наведених проблема.



Шема 2.4.4.1 - Шема повезивања тастера без PULLUP отпорника

Проблем је у томе што се улазни пин када тастер није притиснут налази у лебдећем (енг. floating) стању. То значи да је сигнал који долази до улазног пина подложен променама и утицају окружења. Улазни пин мери све што долази до њега, а то могу да буду разни дигитални уређаји, близина потрошача наизменичне струје, чак и човеков прст који преноси електричне импулсе који се налазе у ваздуху.

Решење, као што је горе наведено је постављање PULLUP отпорника. То представља отпорник који је повезан за улазни пин (између прекидача и улазног пина). У том случају, улазни пин се ставља у PULLUP мод, где се напонска вредност пина подиже на 5 V, а супротна страна (после прекидача) је везана за дигиталну нулу. На шеми испод се налази изглед струјног кола.



Шема 2.4.4.2 - Шема повезивања тастера са PULLUP отпорником

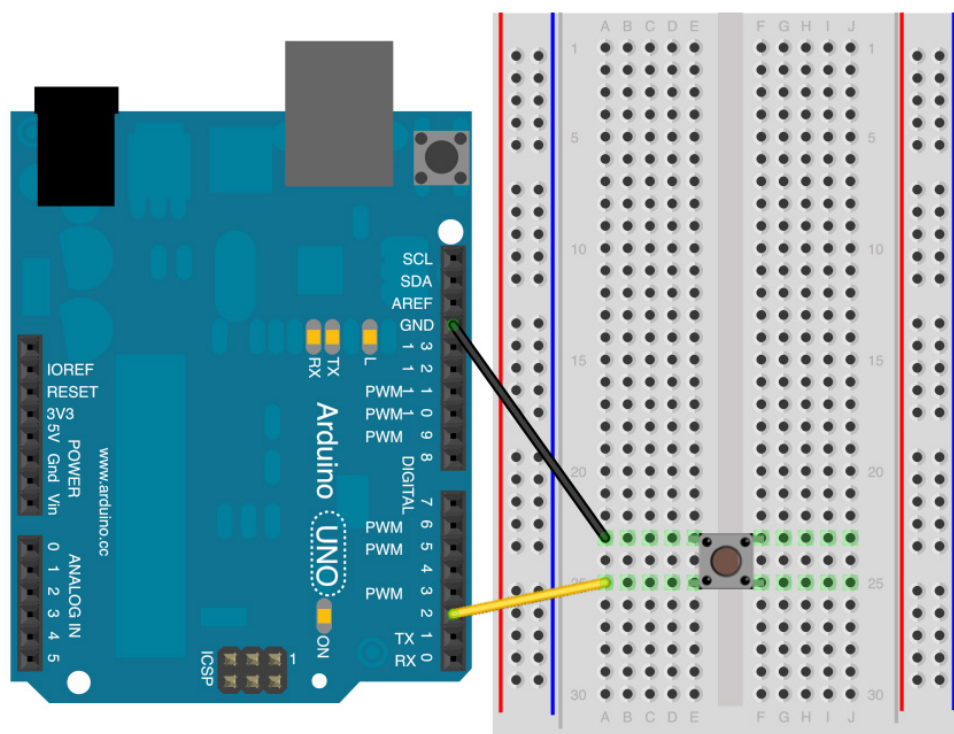
На овакав начин повезан тастер ће када није притиснут увек имати величину 5 V - HIGH, а када је притиснут имаће умањену величину за вредност сразмерну отпору и јачини струје у колу, тј. добиће се логичка нула - LOW.

Ардуино је у ове сврхе за све улазно/излазне пинове уградио у плочу по један PULLUP отпорник због решавања овог проблема и лакше примене и повезивања тастера. Кад се мислило на све, мислило се и на дигиталне и на аналогне пинове. На слици испод су приказани тастери коришћени у овом пројекту.



Слика 2.4.4.1 - Уграђени тастери и њихово повезивање

Шематски изглед повезивања тастера са Ардуином изгледа као на слици 2.4.4.2.



Слика 2.4.4.2 - Повезивање тастера са Ардуином

Код примене оваквог повезивања, ардуинова заједница се потрудила да направи библиотеку која ће да умањи количину кода која мора да се унесе да би се прочитало стање пина и свела је комплетан процес на неколико синтаксних функција и једну библиотеку - **Button Library**.

Изглед синтаксе за дефинисање пина и покретање PULLUP мода на том пину је приказана у коду испод:

```
Button(byte buttonPin,byte buttonMode)
```

Button Тастер = Button(12,PULLUP);

Button библиотека поседује неколико функција које могу да буду корисне у току програмирања понашања и реакција након притискања тастера:

void pullup()

Мења мод Button библиотеке у PULLUP конфигурацију

void pulldown()

Мења мод Button библиотеке у PULLDOWN конфигурацију

boolean isPressed()

Функција која врши проверу да ли је тастер притиснут

boolean wasPressed()

Проверава да ли је тастер био притиснут последњи пут када је позвана функција isPressed() и враћа логичку јединицу ако јесте.

boolean stateChanged()

Проверава да ли је стање дугмета промењено. Као што је промена са притиснут - не притиснут и обрнуто и враћа логичку јединицу уколико се та промена разликује од величине која је добијена последњим позивањем функције isPressed().

boolean uniquePress()

Провера да ли је стање тастера промењено и да ли је притиснут тастер. Ова функција ће вратити логичку јединицу само први пут када је дугме притиснуто, за разлику од функције isPressed() које ће враћати логичку јединицу све време док је притиснуто.

Једноставан пример коришћења Button библиотеке:

#include <Button.h>		void loop(){
/*		if(button.isPressed()){
Креирање Button улазног пина на пину 12		digitalWrite(13,HIGH);
*/		}else{
Button Тастер = Button(12,PULLUP);		digitalWrite(13,LOW);
void setup(){		}
pinMode(13,OUTPUT); //debug to led 13		}
}		

Због велике брзине рада петље код једноставнијих кодова, да би се постигла стабилност приликом притискања дугмета и мењања стања неког излазног пина у IF петљи, користи се углавном **uniquePress()** функција. На примеру испод је показано како се она користи и зашто.

```
Button taster1 = Button(14,PULLUP); // taster 1
Button taster2 = Button(15,PULLUP); // taster 2
Button taster3 = Button(16,PULLUP); // taster 3
Button taster4 = Button(17,PULLUP); // taster 4
Button mikrotaster1 = Button(3,PULLUP); // granicnik dodavaca hmelja / na 90 stepeni pomak

// Relaj za grejac

if (taster1.uniquePress())
{
    if (Readpingrejac == HIGH)
    {
        digitalWrite(pingrejac, LOW);
        lcd.setCursor(1, 1);
        lcd.print(" ON");
    }
    else
    {
        digitalWrite(pingrejac, HIGH);
        lcd.setCursor(1, 1);
        lcd.print("OFF");
    }
}

// Relaj pumpe za recirkulaciju

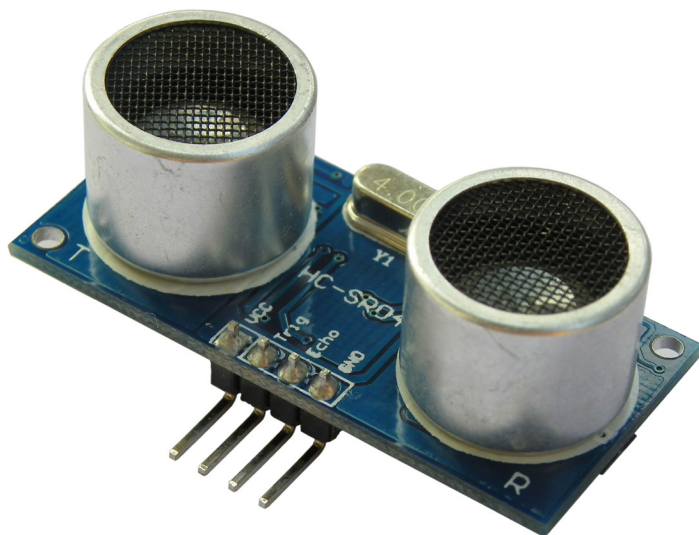
if (taster2.uniquePress())
{
    if (Readpinpumparec == HIGH)
    {
        digitalWrite(pinpumparec, LOW);
        lcd.setCursor(6, 1);
        lcd.print(" ON");
    }
    else
    {
        digitalWrite(pinpumparec, HIGH);
        lcd.setCursor(6, 1);
        lcd.print("OFF");
    }
}
```

Слика 2.4.4.3 - Део кода коришћеног за мануелно управљање релејским склопом

Због велике брзине петље, уведена је функција **uniquePress()**, да се не би десило да се релеј упали и угаси притиском тастера, што би био случај са функцијом **isPressed()**.

2.4.5 Ултразвучни сензор HC - SR04

Ултразвучни сензор користи технологију сонара да дефинише удаљеност неког предмета испред њега. Предност ове технологије је у томе што дозвољава бесконтактно мерење удаљености, у овом случају нивоа воде, са великом прецизношћу и стабилности и све то у једном малом сензору HC-SR04. На слици испод се налази изглед сензора.

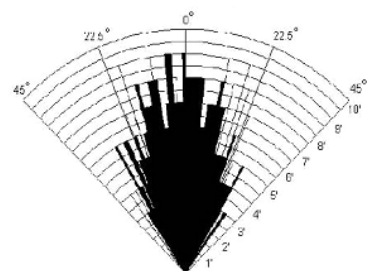
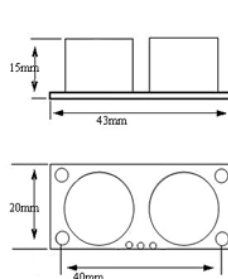


Слика 2.4.5.1 - Изглед ултразвучног сензора HC-SR04

Оперативни интервал му је од 2 до 400 центиметара. Овакав сензор не може бити ометен у раду утицајем сунчеве светлости, температуре или боје објекта од којег се одбија звук, али његова прецизност процене раздаљине између предмета који су мекани, тј. предмети израђени од не акустичних материјала (одећа, сунђери, итд.), опада. Пошто је његова намена у овом пројекту таква да се сигнал са сензора одбија од површине воде, ово неће бити проблем. Као што се и на слици види, он поседује и трансмисиони и пријемни модул у једном.

Спецификација HC-SR04:

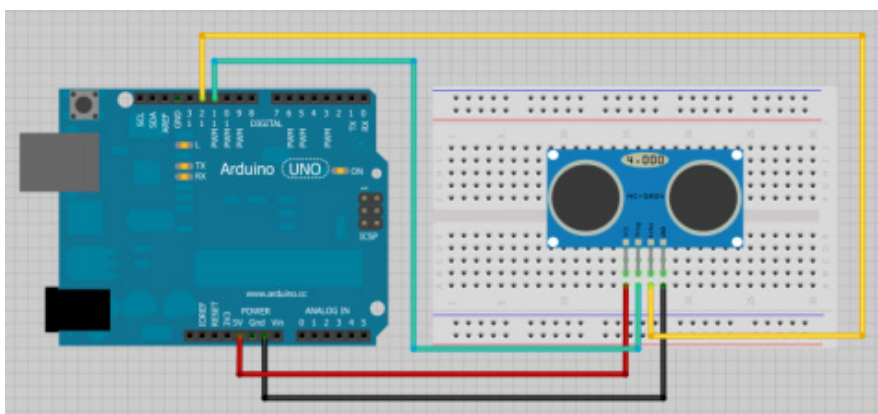
- Напајање сензора: 5V
- Струја у стању мировања: <2mA
- Радна струја: 15mA
- Ефективни угао мерења: <15 °
- Интервал мерења: 2 cm - 400 cm
- Резолуција: 0.3 cm
- Угао мерења: 30 °
- Пулсна дужина улазног сигнала: 10 μs
- Димензије: 45mm x 20 mm x 15mm



Конекциони пинови:

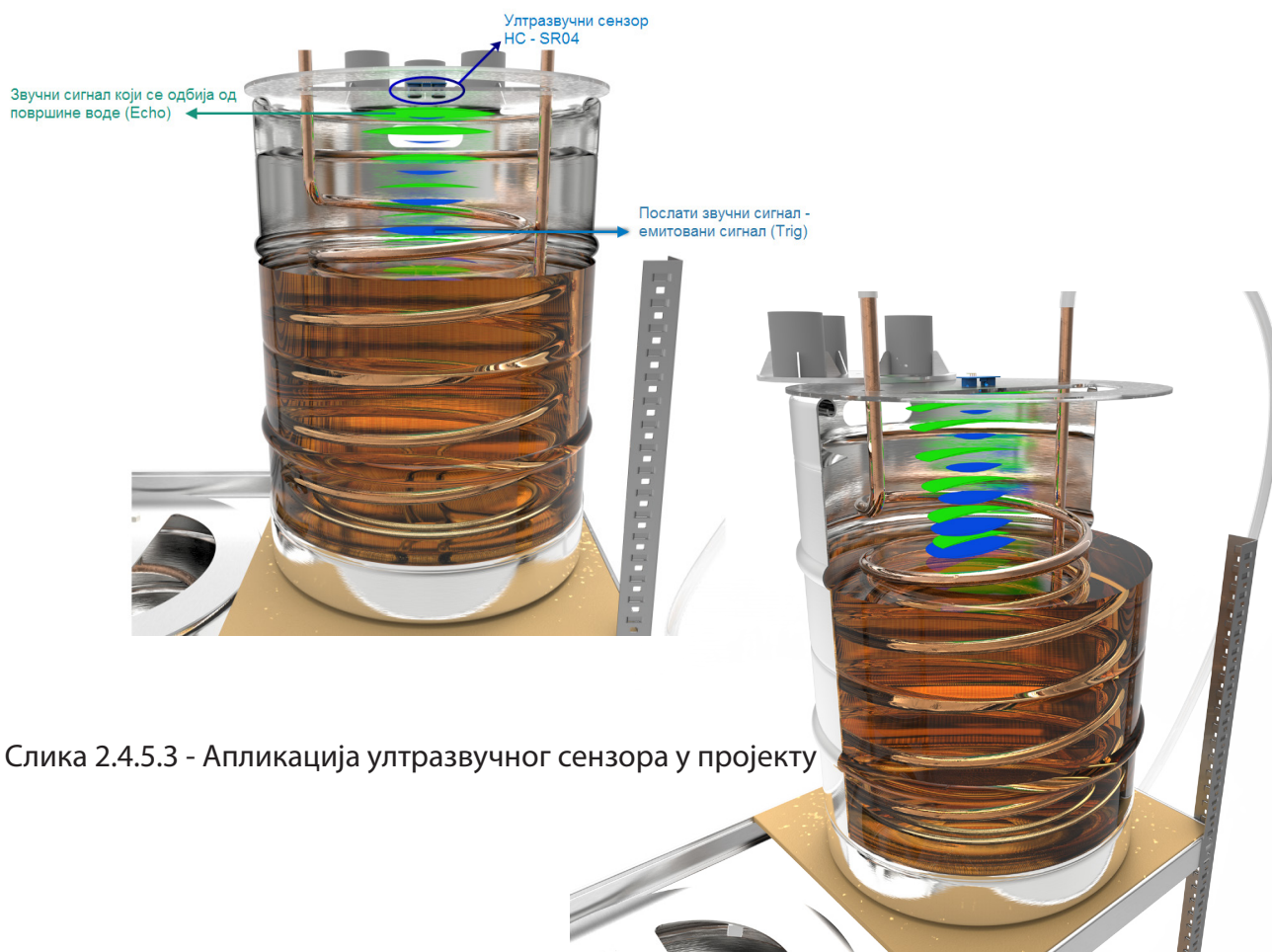
- VCC: 5VDC
- Trig: Trigger (INPUT)
- Echo: Echo (OUTPUT)
- GND: GND

Повезивање овог сензора са ардуином као и у случају других компоненти, је једноставно. Шема испод објашњава повезивање ардуина и ултразвучног сензора.



Слика 2.4.5.2 - Повезивање ултразвучног сензора са Ардуином

Апликација овог сензора је од веома важног значаја за овај пројекат, из разлога што овај сензор контролише количину воде у суду и воде која ће из њега да изађе, тиме је дошло до уштеде на додатним сензорима за мерење протока. На слици испод је приказана апликација сензора и његова позиција у склопу.

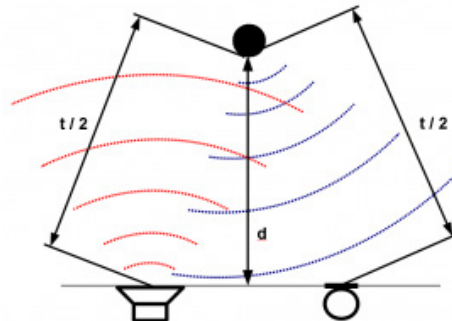


Слика 2.4.5.3 - Апликација ултразвучног сензора у пројекту

Такође, овај сензор се користи и као сигурност код прегрејавања грејача. Ако је ниво воде сувише низак, грејач се неће упалити да не би дошло до прегрејавања и уништавања компоненте (ради се о тзв. мокром грејачу). Из свега наведеног, види се да апликација овог сензора иза себе има велику одговорност.

Након повезивања овог сензора са ардуином, можемо почети програмирање и софтверску

апликацију овог сензора. Прво што треба да се узме у обзир је принцип рада ултразвучног сензора и улазно/излазне променљиве. Сигнал који се емитује је у виду високофреквентног звука који се понавља у регуларним временским интервалима. Мери се време које протекне између емитовања сигнала и враћања истог одбијеног о неки предмет. У случају ултразвучних сензора, није битна јачина звук већ време које протекне између слања и примања звука. На слици 2.4.5.4 је приказана скица ове тврдње.



Слика 2.4.5.4 - Скица у функцији времена између слања и примања ултразвучног сигнала

Коришћењем емпиријских података и закона физике (брзина звука) дошло се до облика кода који је потребан да би ова компонента дала прецизан резултат. Функција која ће се користити у оквиру овог сензора је **pulseIn()** функција. Ова функција мења стање променљиве којој је додељена на следећи начин: Функција чека одређену величину или логичку нулу или логичку јединицу, зависно шта јој је додељено, када је величина покрене, она мери време од почетка до краја деловања или важења те величина на улазу, тј. одговарајућем конекцијском пину. Синтакса за коришћење ове функције је:

pulseIn(pin, value)

pulseIn(pin, value, timeout)

pin - Конекцијски пин на који је прикључен Echo

value - Вредност коју очекује функција (HIGH или LOW)

timeout - Време које ће чекати функција да прими сигнал, ако не прими, враћа логичку нулу [микросекунде]

Пример кода који је коришћен у пројекту заједно са рачунањем средње вредности променљиве дужине, због прецизније и инертније промене променљиве је приказан на слици 2.4.5.5. У коду се види да је време емитовања сигнала 10 микросекунди или 0,00001 секунди, али и то је довољно да би се сигнал емитовао. Након тога код чека ехо сигнал наредних 5 милисекунди, уколико га не добије враћа логичку нулу.

```
void uss()
{
    digitalWrite(trigpin, LOW); //
    delayMicroseconds(5); // AKTIVIRANJE "ULTRASONIC" SENZORA
    digitalWrite(trigpin, HIGH); //
    delayMicroseconds(10); //
    digitalWrite(trigpin, LOW); //

    trajanje = pulseIn(echopin, HIGH, 5000); // OBRADA SIGNALA SA "ULTRASONIC" SENZORA
    razdaljinauss = (trajanje/2) / 29.1; //

    if (razdaljinauss <= 34 || razdaljinauss <= 0){

        // Oduzeti od poslednjeg citanja signala
        total -= citanje[index];
        // Ocitanje sa senzora
        citanje[index] = razdaljinauss;
        // Dodato ocitanje totalu
        total += citanje[index];
        // Napredovanje index promenljive do sledece pozicije u nizu
        index = index + 1;

        srvrednostuss = total / brccitanja; // Racunanje srednje vrednosti
    }
    else {

        razdaljinauss = 00; // prijavljivanje greska
        srvrednostuss = 00; // prijavljivanje greska
    }

    if (srvrednostuss != srvrednostussp) // smanjenje frekvencije updateovanja na displej
    {
        srvrednostussp = srvrednostuss;
    }

    if (index >= brccitanja) // vraćanje na pocetak racunanja srednje vrednosti razdaljine
    {
        index = 0;
    }
}
```

Слика 2.4.5.5. - Пример кода коришћен за одређивање раздаљине између површине воде и сензора

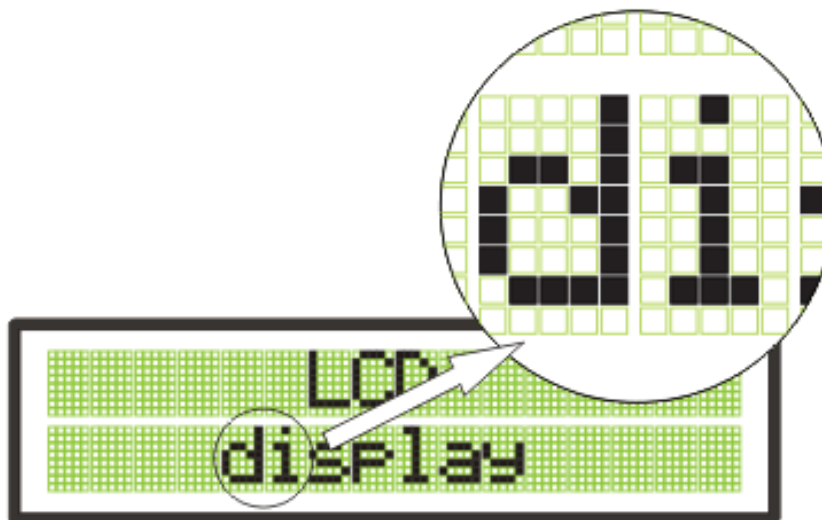
2.4.6 LCD дисплеј са i2C комуникацијским протоколом

Сваки пројекат који захтева надгледање променљивих које нису дискретне, као што је упаљено-угашено, има-нема, итд., захтева визуелну везу са корисником на нивоу екрана. У овом случају, пошто се ради о систему који неће имати везе са рачунаром, серијски монитор се аутоматски искључује као комуникација са корисником. У ове сврхе је узет матрични LCD екран величине 20x4. На слици испод се налази LCD екран који је коришћен у пројекту.



Слика 2.4.6.1 - LCD матрични екран величине 20x4

Величина матричних екрана се изражава у броју под-матрица. На слици 2.4.6.1 се види да под-матрицу представља простор једног знака, у овом случају слова. Стандардна под-матрица је величине 7x5 пиксела. На слици 2.4.6.2 се види облик и изглед под матрице.



Слика 2.4.6.2 - Распоред пиксела у под-матрицама LCD матричног екрана

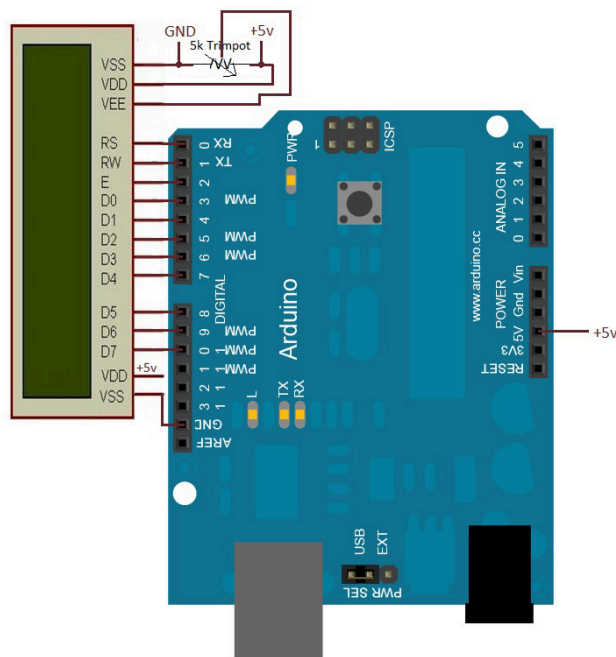
Да не би даље залазили у технологију течних кристала и начина функционисања самог LCD екрана, што и није неопходно знати за реализацију пројекта, прећи ћемо одмах на повезивање екрана са Ардуином. Да би повезали такав екран за ардуинову платформу потребно је да искористимо приличан број комуникацијских пинова. Тачно 7 улазно/излазних пинова. Да бисмо избегли заузимање великог броја комуникацијских пинова, користимо i2C комуникацијски протокол. На слици 2.4.6.3 се налази шема повезивања ардуинове платформе преко стандардног

комуникацијског протокола.

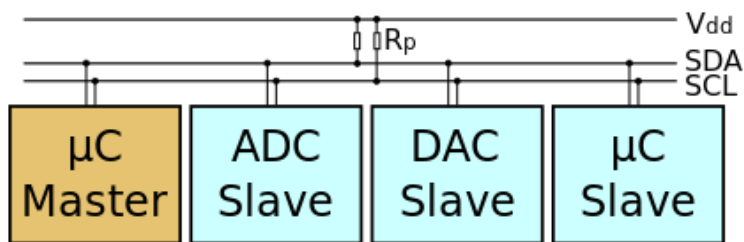
Комуникацијски протокол **i2C** или **iiC** представља један вид бас (енг. bus) комуникације. У буквалном преводу i2C или IIC значи унутрашње-интегрисано коло (енг. inter-integrated circuit). У свет електронике га је увела фирма **Phillips Semiconductor**.

Овај комуникацијски протокол користи две двосмерне комуникацијске линије или везе, а то су SDA (Serial Data Line) и SCL (Serial Clock Line), pullup-оване са pullup отпорницима. Типичне напонске величине са којима ради овај протокол су 5V и 3.3V.

Адресирање уређаја повезаних на овај комуникацијски протокол је у виду седмобитне или десетобитне адресе (зависно о ком уређају или компоненти је реч). Оваква бас комуникација додељује две улоге уређајима повезаних на мрежу, а то су: мастер и слејв мод. На слици 2.4.6.4 су приказани модови и начин повезивања уређаја или компоненти.



Слика 2.4.6.3 - Повезивање Ардуино -а са LCD екраном

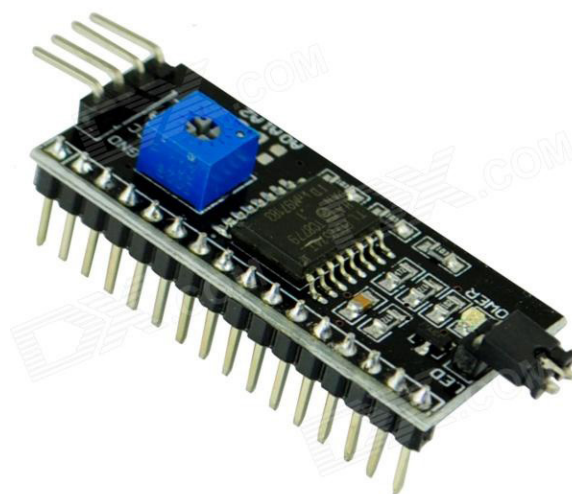


Слика 2.4.6.4 - Модови и повезивање компоненти на мрежу IIC

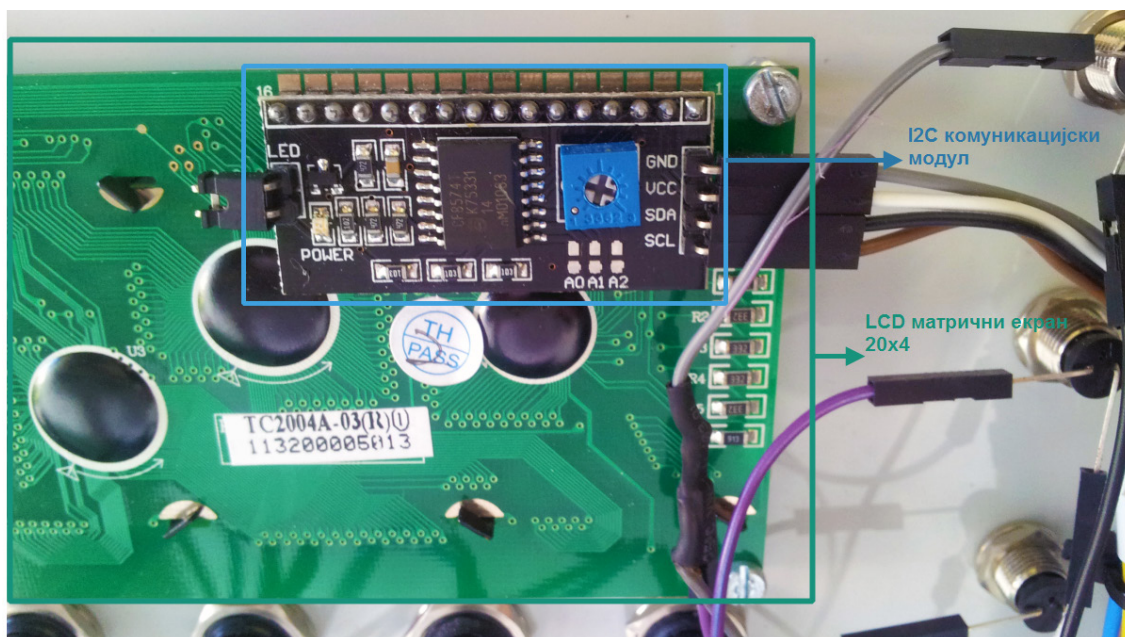
Да би омогућили ардуину да комуницира са LCD екраном на овај начин, морамо да уградимо модул који ће да обезбеди конверзију сигнала послатих преко ове две бас конекције на свих 7 конекцијских пинова на LCD матричном екрану. Овај модул је I2C модул и приказан је на слици 2.4.6.5.

Повезивање модула на екран се врши лемљењем конекционих пинова I2C модула на штампану плочу LCD екрана. Пошто овај модул долази одвојен од екрана, овај процес је одрађен накнадно, ручном техником лемљења, не машински, тако да је захтевана велика велика прецизност извршиоца овог захвата. На слици 2.4.6.1 се види да сваки конекциони пин има своју површину пресвучену бакарном легуром коју треба прекрити везивним средством, у овом случају калајем и тако везати конекционе пинове IIC модула за плочу LCD екрана.

На слици 2.4.6.6 се види резултат претходно описаног захвата заједно са уградњом у кућиште електронских компонената.



Слика 2.4.6.5 - I2C комуникацијски модул



Слика 2.4.6.6 - Спојени I2C комуникацијски модул и LCD екран

У случају Ардуино верзија Duemilanove/UNO конекцијски пинови за SDA и SCL су респективно аналогни улаз/излаз A4 и A5. На сензорском модулу за Ардуино који је коришћен у овом пројекту се налази издвојена група конекцијских пинова намењена само у ове сврхе, тј. остваривања комуникације са I2C уређајима. Поменута група конекцијских пинова се налази на слици 2.4.1.2.

Након спајања ова два модула, сада помоћу само два комуникацијска пина можемо остварити комуникацију између ардуиновог микроконтролера и LCD екрана.

Да би се пронашла адреса неког уређаја који се налази на бас мрежи, приступа се поступку идентификације. Оно што је прво, а то је, да на почетку сваког кода где желимо да покренемо комуникацију са I2C уређајем дефинишемо библиотеку **Wire.h**. Ова библиотека служи за комуникацију ардуиновог микроконтролера са I2C уређајем. Коришћењем кода за идентификацију компоненти на бас мрежи, дошло се до резултата приказаног на слици 2.4.6.7.

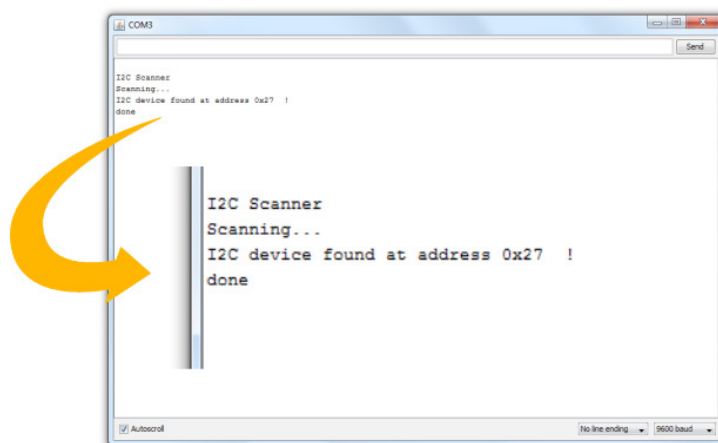
```
byte error, address;
int nDevices;

Serial.println("Scanning...");

nDevices = 0;
for(address = 1; address < 127; address++)
{
  // The I2C_scanner uses the return value of
  // the Write.endTransmission() to see if
  // a device did acknowledge to the address.
  Wire.beginTransmission(address);
  error = Wire.endTransmission();

  if (error == 0)
  {
    Serial.println("I2C device found at address 0x");
    if (address<16)
      Serial.print("0");
    Serial.print(address,HEX);
    Serial.println("  !");
    nDevices++;
  }
  else if (error!=4)
  {
    Serial.println("Unknown error at address 0x");
    if (address<16)
      Serial.print("0");
    Serial.println(address,HEX);
  }
}
if (nDevices == 0)
  Serial.println("No I2C devices foundin");
else
  Serial.println("done");

delay(5000); // Wait 5 seconds for next scan
```



Слика 2.4.6.7 - Пример кода за идентификацију адресе LCD екрана на I2C бас мрежи

Након идентификације екрана, прелазимо на програмирање и приказивање величина на LCD екрану. За комуникацију са екраном као што је штампање и слање знакова на њега, користи се библиотека **LiquidCrystal_I2C.h**. Ова библиотека поседује функције као и свака друга за олакшано исписивање и слање података на LCD екран. На слици 2.4.6.8 се налази пример кода који је коришћен у сврхе исписивања вредности са сензора и у сврхе исписивања карактера. Пошто је код део коначног кода који је коришћен за управљање комплетним процесом прављења пива, узети су делови из њега и приказани у функционалном облику.

```

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

#define I2C_ADDR 0x27 // <<----- Upisati adresu vasesg LCD ekrana.  Nadjite ga pomocu I2C skenera
#define BACKLIGHT_PIN 3
#define En_pin 2
#define Rw_pin 1
#define Rs_pin 0
#define D4_pin 4
#define D5_pin 5
#define D6_pin 6
#define D7_pin 7

LiquidCrystal_I2C lcd(I2C_ADDR,En_pin,Rw_pin,Rs_pin,D4_pin,D5_pin,D6_pin,D7_pin, BACKLIGHT_PIN, POSITIVE); // Adresiranje pinova u biblioteci "LiquidCrystal_I2C"

void setup() // run once, when the sketch starts
{
  Wire.begin(); // Inicijalizacija biblioteke OneWire
  lcd.begin(20,4); // inijalizacija ekrana LCD 20 x 4
  lcd.backlight(); // Paljenje pozadinskog osvetljenja ekrana PIN 13
  lcd.clear(); // Brisanje ekrana

void loop() // pocetak stvarne skice, i vrti se non stop u krog
{
  lcd.setCursor(0, 3);
  lcd.print("TK ");
  lcd.setCursor(3, 3);
  lcd.write(1);
  lcd.setCursor(7, 3);
  lcd.print((char)223); // karakter oznake stepena
  lcd.print("C");

```

Укључивање библиотеке у код

Дефинисање пинова и излазних променљивих из I2C модула

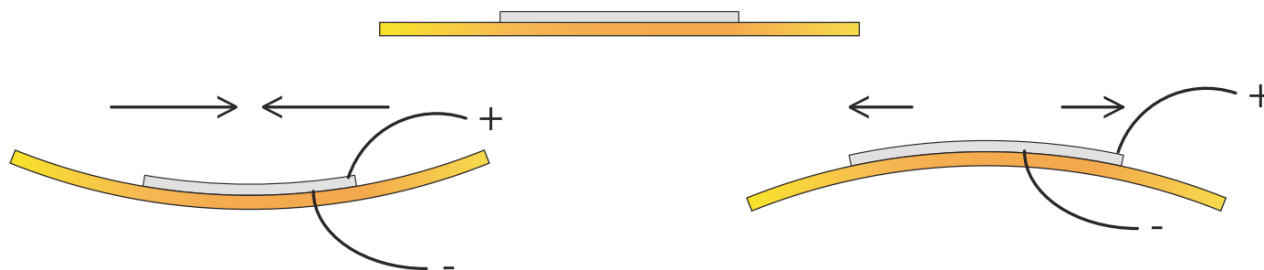
Позивање LiquidCrystal_I2C библиотеке

Слика 2.4.6.8 - Пример издвојен из коначног кода у функционалном облику

2.4.7 Пиезо компонента - звучник

Пошто сваки производни процес, што је и тема овог рада, захтева одређену количину времена за добијање коначног производа и одређене операције морају да буду извршене у тачним временским интервалима, пиезо компонента - звучник је уведен да би оператер у сваком тренутку знао шта се дешава са одређеним кораком у процесу и као сигурносна компонента.

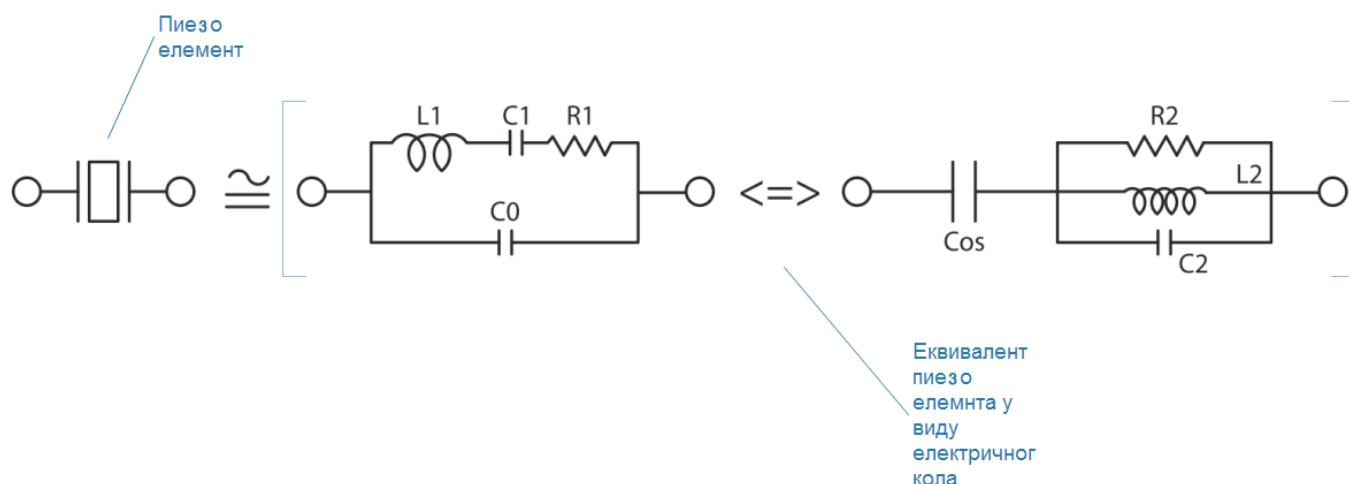
Срце сваког пиезо елемента, било то сензор или у овом случају актуатор је керамички диск који производи електрицитет применом силе на њега или се деформише под утицајем једносмерне струје. Повећањем напона на овом диску, што је и приказано на слици 2.4.7.1, пиезо керамички диск се деформише више или мање у зависности од напона.



Слика 2.4.7.1 - Деформација пиезо керамичког диска на површини металне мембране

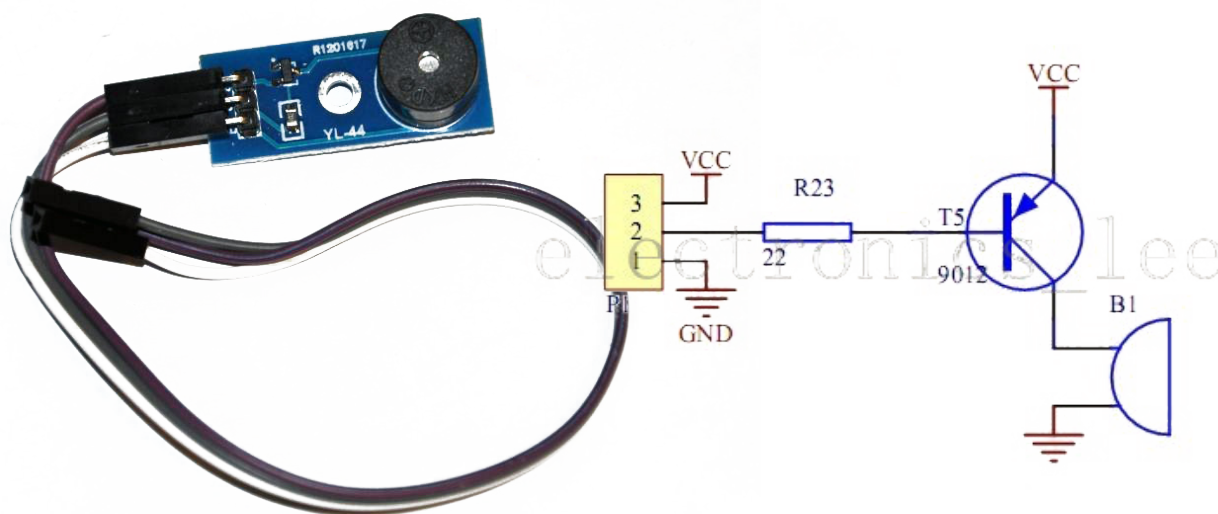
Што је већа разлика између напонских амплитуда то је звук јачи, што је већа фреквенција промена амплитуда, то је тон виши. Пиезо елемент има комплексан еквивалент електричног кола, тј. математички модел, али углавном се могу гледати као кондензаторски потенцијал у интервалу од 10 нано фарада до 1 микро фарада. На слици 2.4.7.2 је приказано електрично коло које је

еквивалент пиезо елемента или пиезо керамичког диска.



Слика 2.4.7.2 - Пиезо елемент и његови еквиваленти

Кондензаторски капацитет пиезо звучника је најважнија његова карактеристика. Као што је већ речено, да би се звук емитовао потребно је да постоје разлике у улазном напону саме компоненте. Пошто је сигнал емитован са дигиталног излазног пина, напонска разлика је била 5 V, а фреквенција је усвојена из табеле дате на веб страници <http://www.phy.mtu.edu/~suits/notefreqs.html>. Компонента која је коришћена у овом пројекту је у виду модула и приказана је на слици 2.4.7.3 као и њена



Слика 2.4.7.3 - Пиезо модул

електронска шема. Повезивање ове компоненте са ардуино је крајње једноставно, потребно је довести напон од 5 V на одговарајући улаз модула, дигиталну нулу и повезати сигнални конекциони пин са жељеним конекционим пином на ардуину. Након повезивања може се почети са програмирањем функције за емитовање звучног сигнала (слика 2.4.7.3). Код који се налази на слици 2.4.7.4 је узет из коначног кода и приказан је у облику функције коју је могуће позвати у било ком делу кода и звук ће бити емитован. Функција која дефинише висину тона је уведена да би се звук у сваком кораку процеса једноставно позивао и да би му се једноставно мењала фреквенција, што у неким случајевима може да буде од користи. Усвојени тон на основу коришћене фреквенције је тон C7, што је прилично висок тон, али се тонови могу мењати и тако би се могли раздвојити звукови пред-упозорења, упозорења, грешке или кvara.

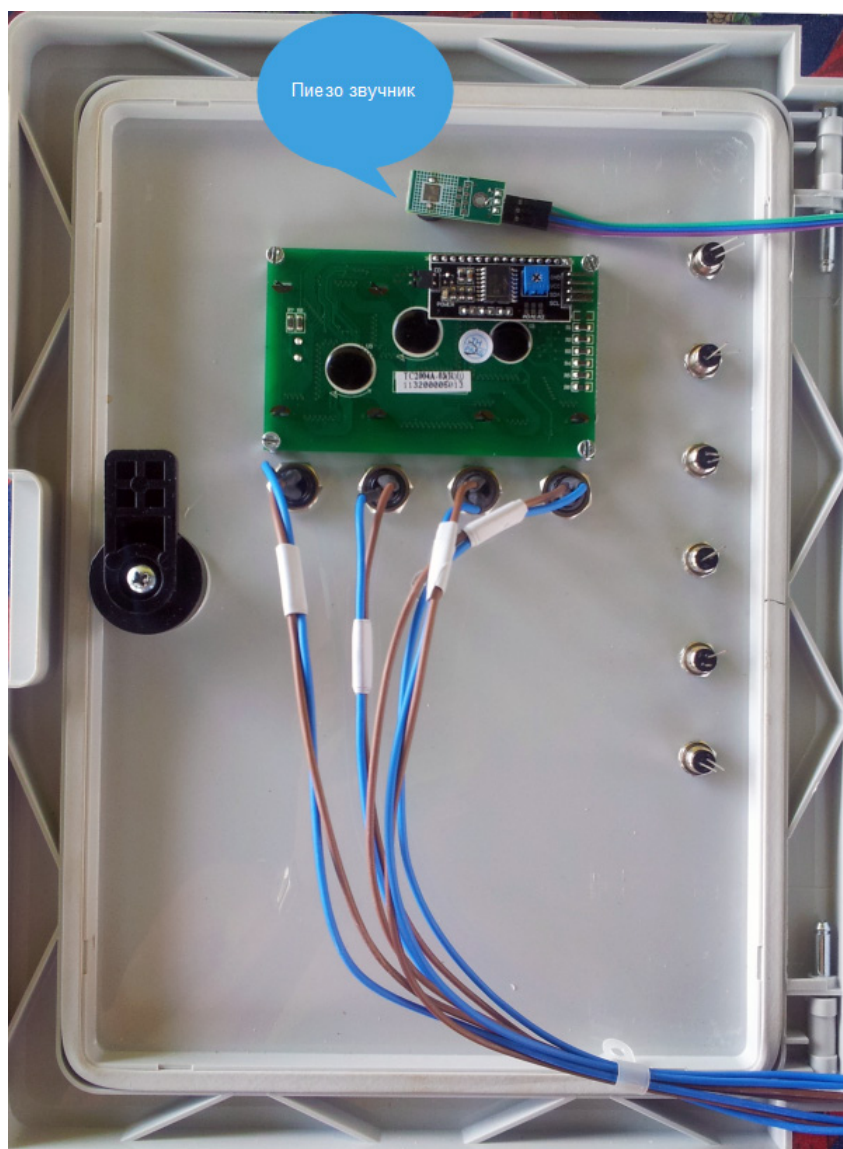
На слици 2.4.7.4 се види уграђени пиезо звучник и његова позиција у склопу.


```

void beep (unsigned char PinZvucnik, int frequencyInHertz, long timeInMilliseconds) // Funkcija za proizvodnju zvuka :)
{
    int x;
    long delayAmount = (long)(1000000/frequencyInHertz);
    long loopTime = (long)((timeInMilliseconds*1000)/(delayAmount*2));
    for (x=0;x<loopTime;x++)
    {
        digitalWrite(PinZvucnik,LOW);
        delayMicroseconds(delayAmount);
        digitalWrite(PinZvucnik,HIGH);
        delayMicroseconds(delayAmount);
    }
}

```

Слика 2.4.7.3 - Функција за производњу звука помоћу пиезо модула



Слика 2.4.7.4 - Пиезо звучник у подклопу електро ормара

3. Оптимизација процеса прављења пива

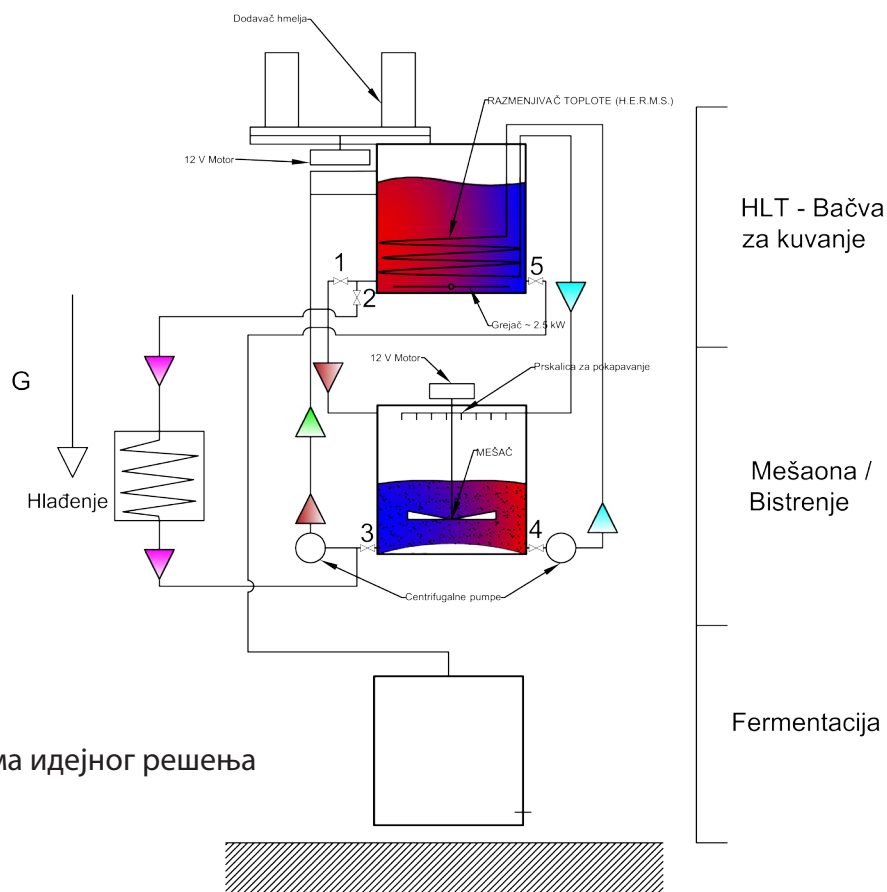
3.1 Физички изглед пиваре

Главни циљ овог пројекта је садржан у наслову, као споредни циљеви ту су поједностављивање процеса прављења пива, тј. свођење на суштину уз добијање истог финалног производа као и код конвенционалних начина. Такође, ту је и пројектовање самог изгледа оваквог система и прављење спецификације потребних елемената за израду пиваре.

Гледање са аспекта технологије пива, минимална количина потребних судова за производњу пива је 4 суда. Ти судови су:

1. Суд са врућом течности (енг. HLT - Hot Liquor Tun)
2. Мешаона - суд за укомљавање
3. Кувало
4. Ферментор

Сагледана су многа решења колега који су тежили ономе чему и овај пројекат тежи - оптимизацији и прилагођавању технологије прављења пива кућним условима и дошло се до закључка да један од судова може да служи у више сврха, а то је суд за кување. Суд за кување према личној идеји може да се користи и за процес регулације температуре и загревање воде за укомљавање. Тако да се дошло до конфигурације система са три суда. Да би схватили у потпуности како се може процес прављења пива, који има приличан број етапа и захтева, сместити у само три суда приказана је шема идејног решења на слици испод.



Слика 3.1.1 - Шема идејног решења

Процес прављења пива данашњице се састоји из великог низа етапа и све то у сврси производње бистрог и што квалитетнијег пива. Процесу који је овим пројектом оптимизован се приступило на следећи начин:

Шта је потребно радити са водом током процеса прављења пива?

- Загревати је и одржавати на одређеној температури
- Претакати
- Мерити њену запремину у два суда
- Филтрирати
- Задржати физичку-хемијску и бактериолошку чистоћу
- Хладити је већом брзином од природне брзине хлађења

Тиме се дошло до закључака да је потребно имати судове направљене од нерђајућег челика - за складиштење воде која ће мењати температурна стања, потребно је имати цеви за претакање које су отпорне на високе температуре, вентиле - за контролу протока, пумпе - за претакање, сензоре за мерење нивоа воде у судовима, сензоре за мерење температуре, грејаче - за подизање температуре воде, расхладне компоненте - за спуштање температуре воде.

Након свега сагледаног по питању воде, се приступило моделирању конфигурације система на основу оптимизованог процеса прављења пива и резултат је приказан на слици 3.1.1.

3.1.1 Судови

Налажење судова - бачви за процес прављења пива није био лак. Пошто бачве морају да буду израђене од нерђајућег челика или неког другог кисело-отпорног метала, одговарајућа запремина таквих судова није могла лако да се нађе. Предлагане запремине, усвајане на основу искуства других (Слика 3.1.1.1) су у сваком случају биле веће од 30 литара.



Слика 3.1.1.1 - Типови судова за производњу сладовине

Та запремина је потребна због одржавања температуре приликом укомљавања слада (процес траје дуго), као и због дугог периода кувања због чега долази до губитака велике количине течности у виду испарења. Такође, већа запремина судова омогућава, већу ефикасност система. Тако су изабране пивске бачве за амбалажирање пива запремина 50 литара и 60 литара (Купљена је једна бачва од 50 литара, а друга је направљена од две бачве од по 30 литара), Слика 3.1.1.2.



Слика 3.1.1.2 - Одабрани судови

Пошто су судови за амбалажирање пива херметички затворени, било је потребно ослободити бачве од резидуалног притиска и након тога направити отворе са горње стране оба суда. Наравно, на суду који је састављен од два, било је и додатног сечења, као и заваривања. **Коришћена технологија:** брусница, електро-лучно заваривање са прохромским електродама.

Због лакше аутоматизације процеса и смањења цене производње, судови ће бити коришћени у више сврха:

- **Први суд** - Кување слада, контрола температуре укомљавања, филтрација
- **Други суд** - Укомљавање слада и филтрација
- **Трећи суд** - Ферментација

Суд који је усвојен за процес ферментације је специјално произведен у те сврхе - ферментор. Направљен је од квалитетне пластике која се користи за јестиве намирнице људске исхране. Горња ивица је специјално ојачана како би подизање и преношење ферментора било могуће. Захваљујући врењачи која је постављена на врху, ферментација се одвија а да је при томе пиво заштићено од спољне контаминације, оксидације или мува. Ферментор има уграђену славину помоћу које се врши точење у флаше након ферментације. Ферментор се налази на слици 3.1.1.3.

Тако је усвојен систем у коме ће се налазити три суда који ће служити за различите процесе и етапе прављења пива,



Слика 3.1.1.3 - Ферментор

тј. производње сладовине.

На шеми (слика 3.1.1) се види и ознака смера гравитационе силе, тај знак је постављен да би се истакла важност распореда судова по висини. Претакање из суда за кување у суд за мешање преко вентила 1, се врши помоћу гравитационе силе, тј. постојећег хидростатичког притиска у суду за кување. Исто тако, претакање из суда за кување у суд за хлађење и истакање финалног производа у ферментор се врши према истом принципу. Све ово се увело због смањења потребних цеви, вентила и пумпи.

3.1.1.1 Први суд - НЛТ / Кувало

Обзиром на намену првог суда у њега је било потребно уградити следеће компоненте:

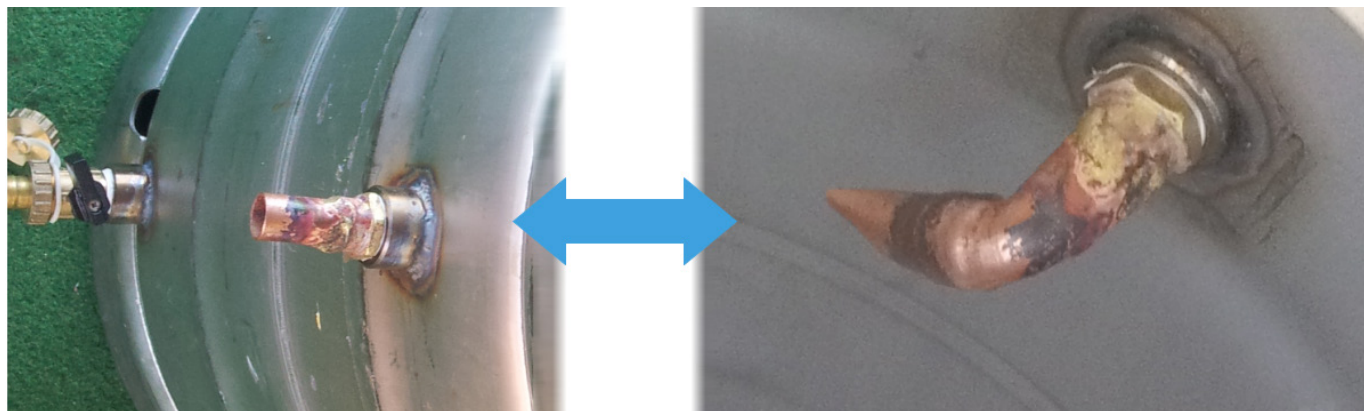
1. Прикључци за утакање и истакање

Два прикључка у доњем делу суда су куглични вентили заварени за суд. Они служе за процес истакања из суда у више етапа производног процеса. Има их укупно три. Са једне стране суда се налазе два вентила повезана преко Т рачве због скретања смера и тока течности у појединим етапама. На слици 3.1.1.1 се налазе прикључци везани за суд.



Слика 3.1.1.1.1 - Прикључци у доњем делу првог суда

У горњем делу суда се налази један прикључак са млазницом специјално направљеном за процес центрифугалног филтрирања. Овај прикључак, за разлику од прикључака у доњем делу суда нема куглични вентил. Направљен је од бакарне цеви пречника 15 милиметара и са унутрашње стране суда је савијен под углом од 90 степени и нагњечен на врху. На слици испод се види његов изглед са спољашње и унутрашње стране суда (слика 3.1.1.1.2).



Слика 3.1.1.1.2 - Изглед горњег прикључка

Разлог нагњечења цеви на врху и угла под којим се налази је повећање брзине протока и усмеравање течности ка зиду суда чиме се обезбеђује вртложно кретање течности у суду.

Тај ефекат је потребан због центрифугалне филтрације.

2. Механички филтер

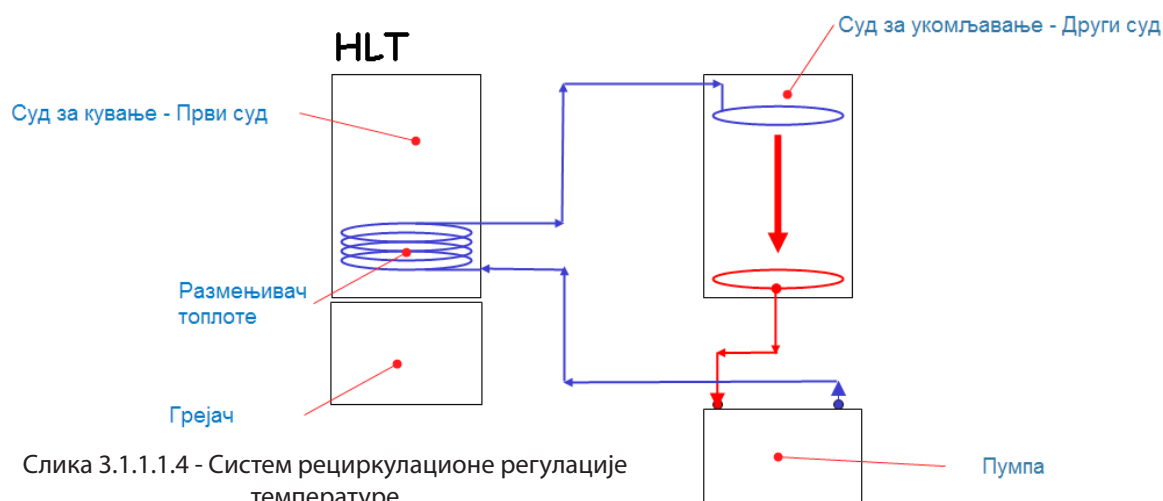
Механички филтер је направљен од бакарних цеви пречника 15 mm које формирају квадратну структуру на дну суда. Обзиром да је дно суда облика калоте, филтер се о дно ослања у четири тачке чиме је функционални део филтера одвојен од дна. Филтрација се врши преко отвора, пречника 4 mm, дуж функционалних делова структуре са страна цеви окренутих ка дну. Филтер је прикључен преко Т рачви са две стране на доње одводне прикључке. На слици 3.1.1.1.3 се налази уграђен механички филтер.



Слика 3.1.1.1.3 - Уграђен механички филтер квадратне структуре

3. Размењивач топлоте - H.E.R.M.S. систем

Да би се у суду за укомљавање - други суд одржавала константна температура комине, било је неопходно постојање система регулације температуре. Због немогућности уградње грејача у суд за укомљавање (технолошка ограничења процеса прављења пива), уграђен је рециркулациони систем одржавања температуре у суд за укомљавање, а размењивач топлоте је морао бити уграђен у суд за кување - први суд. HERMS (енг. Heat Exchange Recirculation Mash System) представља затворени систем рециркулације течности у суду за укомљавање. Суштина рециркулације је у томе што испумпана течност из суда за укомљавање пролази кроз бакарни размењивач топлоте који налази у суду за кување и затим поново враћа у исти. Притом, у суду за кување се налази одређена количина температурно регулисане течности која загрева течност у размењивачу. На слици 3.1.1.1.4 је приказана шема функционисања овог система.



Слика 3.1.1.1.4 - Систем рециркулационе регулације температуре

Размењивач топлоте је направљен од спирално савијене цеви пречника 15 mm израђене од меканог бакра (Слика 3.1.1.1.5).

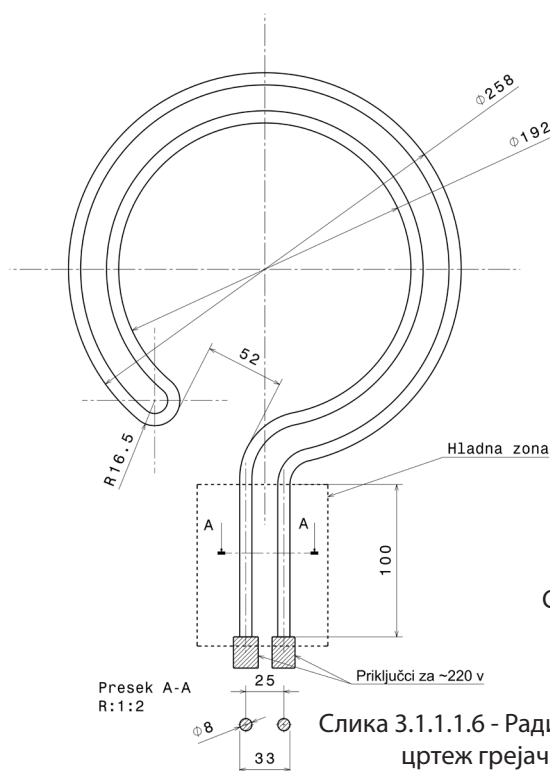


Слика 3.1.1.1.5 - Размењивач топлоте (у изради)

Дужина цеви за овај размењивач топлоте је износила **8 метара**.

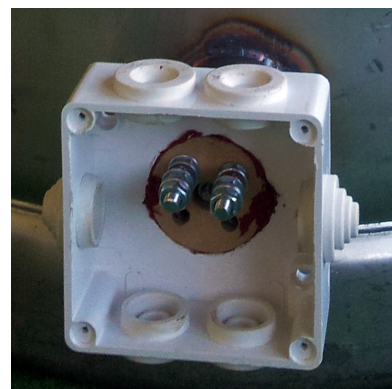
4. Грејач

За загревање течности у првом суду коришћен је специјално направљен “мокри грејач”. Мокри грејач представља грејно тело које у сваком тренутку свог рада мора бити у контакту са течношћу. Разлог томе је превелика снага која може у тренутку да истопи медијум носача грејног тела. Радионички цртеж према коме је израђен грејач је приказан на слици 3.1.1.1.6.



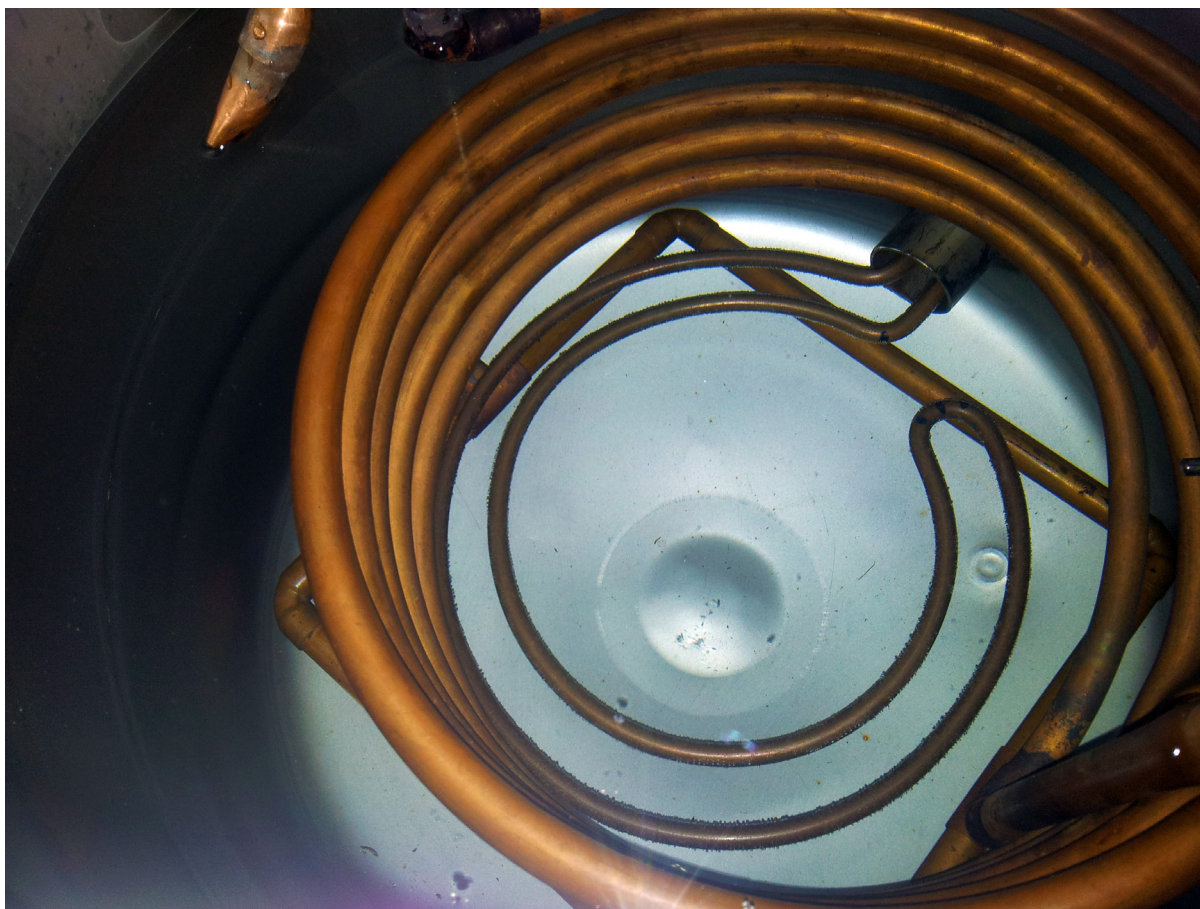
Процењена снага овог грејача је око 2.5 kW. На цртежу се види да постоји такозвана “хладна зона”, она је направљена да би се температурно изоловао грејач од суда и тиме спречило расипање топлоте. Облик грејача је условљен захтеваном снагом и позицијом у склопу. Мана ове компоненте је у томе, што уз прикључке за одвод и довод течности, није модуларна, тј. немогуће је компоненту одвојити од суда. Прикључак се везује за суд завртњевима и херметички заптива ватроотпорним силиконом (Слика 3.1.1.1.7).

Слика 3.1.1.1.7 - Прикључак грејача



Слика 3.1.1.1.6 - Радионички цртеж грејача

На слици 3.1.1.1.8 је приказан коначан изглед унутрашњости суда са свим уграђеним компонентама, укључујући и грејач.



Слика 3.1.1.1.8- Коначан изглед унутрашњости првог суда

5. Модуларни подсклоп - Први суд

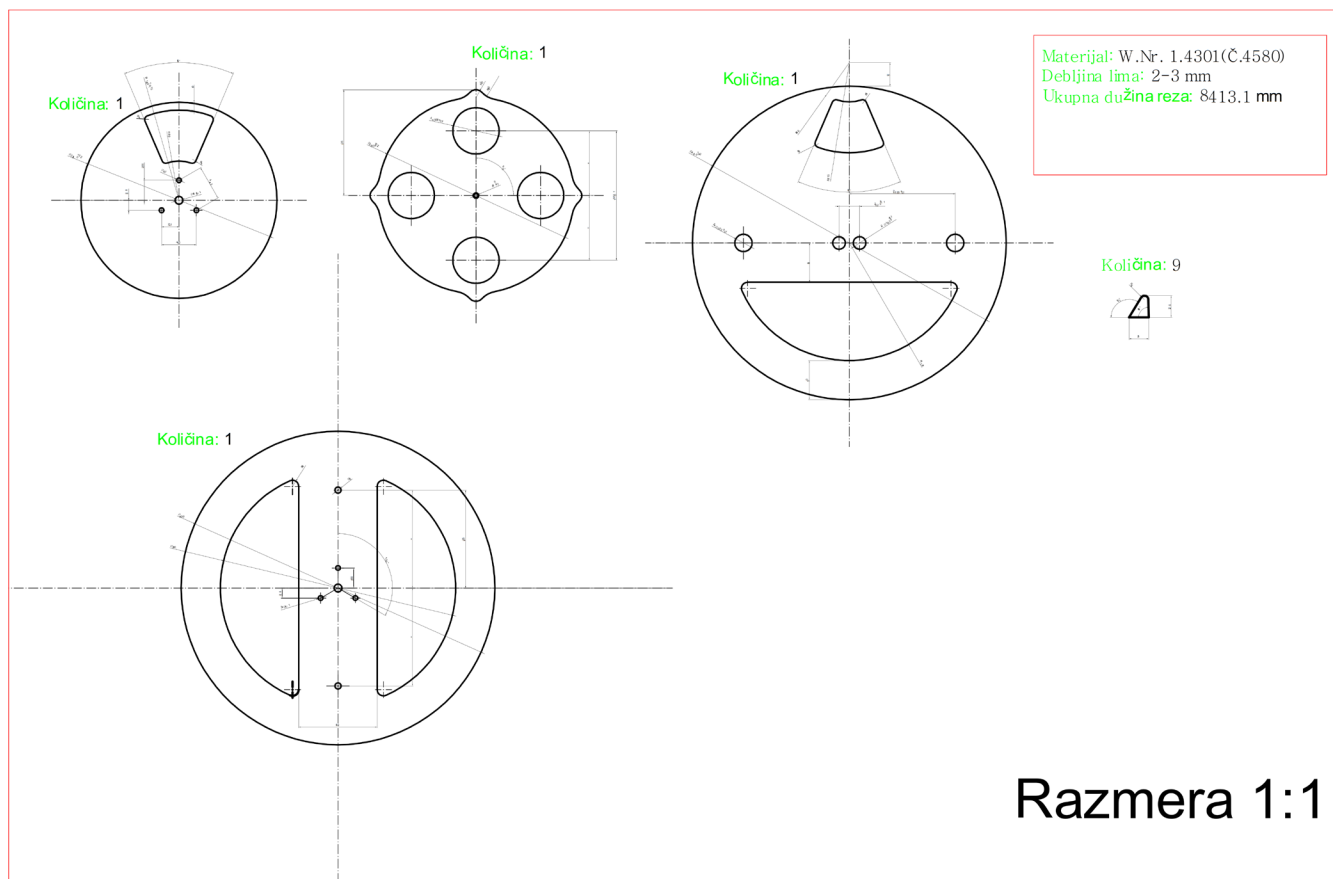
На врху првог суда се налази систем који представља одвојен подсклоп у виду поклопца. Овај подсклоп има низ функција које обавља, а то су:

- Носач размењивача топлоте
- Носач механизма додавача хмеља
- Топлотни изолатор
- Носач ултразвучног сензора
- Носач микро-прекидача додавача хмеља

Обзиром на његову намену, делови поклопца и носача механизма додавања хмеља су морали бити израђени од нерђајућег челика дебљине 3 mm и прецизно, према радионичком цртежу, израђени.

Радионички цртежи су израђени на основу 3D модела урађених према габаритним мерама првог суда. Да би се прецизно направили отвори у челику дебљине 3 милиметра плочасте форме, морало је да се изабере производна технологија доступна на тржишту Србије.

Изабрана технологија је технологија сечења тврдог медијума под млазом воде и абразива, такозвани *WaterJet*. Ова технологија је веома прецизна и испунила је захтеве за израду носачких плоча. На слици 3.1.1.1.9 је приказана радионичка документација на основу којих су сечене носеће



Слика 3.1.1.1.9 - Радионичка документација за производњу носећих плоча

плоче и компоненте за механизам додавача хмеља.

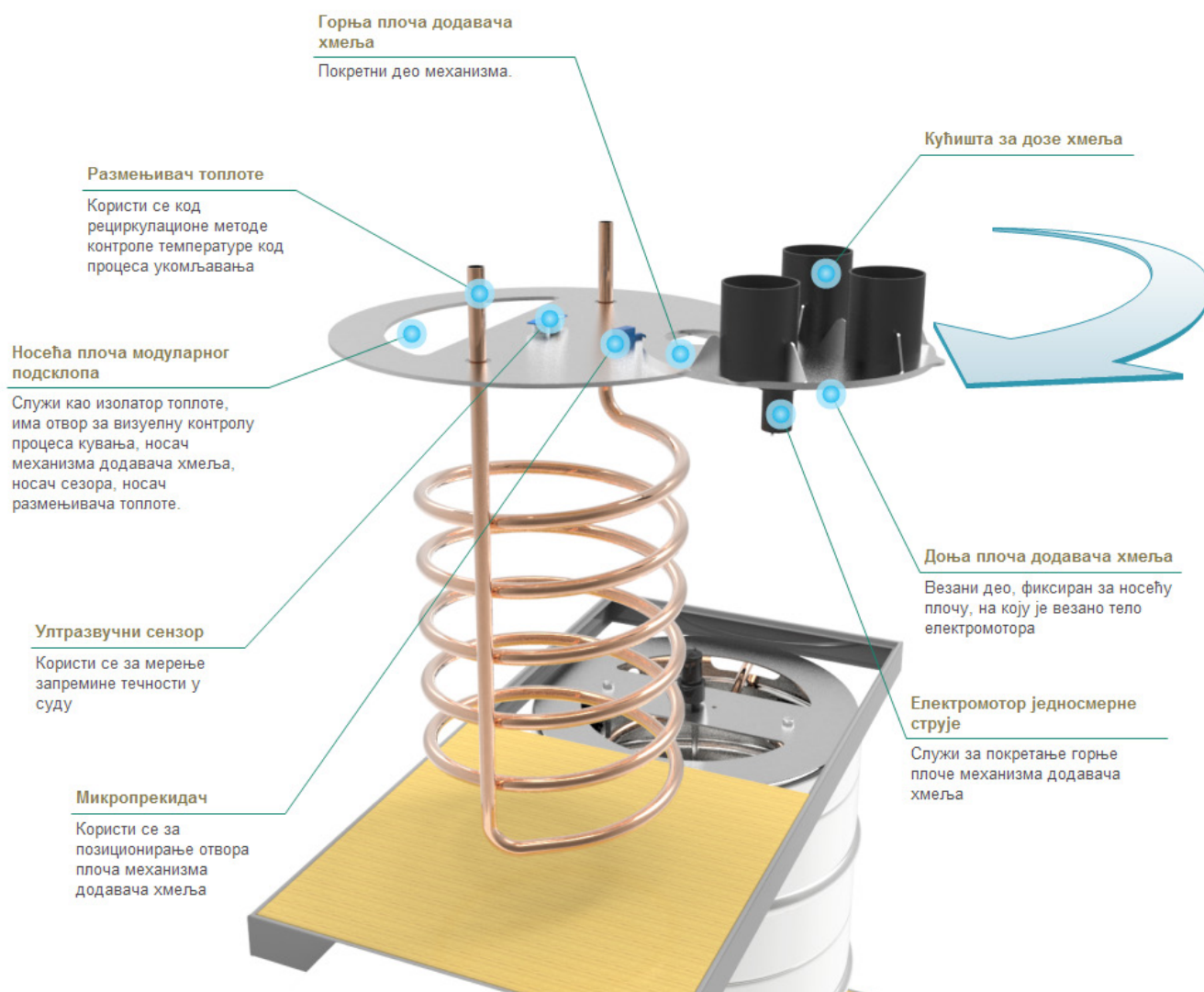
На слици 3.1.1.1.10 се налази коначан изглед плоча израђених по претходно описаној технологији.

У оквиру овог подсклопа се налази механизам додавача хмеља. Састоји се од две плоче са отворима на себи, од којих је једна фиксирана за носећи део подсклопа методом заваривања, а друга саосно постављена изнад прве и везана за електромотор једносмерне струје. Цртежи и облик те две плоче се налазе у левом углу слике 3.1.1.1.9. Тело електромотора је везано завртњевима за доњу плочу механизма додавача хмеља. Тиме се омогућила ротација горње плоче и наизменично преклапање отвора обе плоче. Овај механизам је уведен због аутоматизовања процеса додавања хмеља. Да би се прецизно позиционирали отвори на плочама, на горњој плочи моделирани су брегови за покретање

Слика 3.1.1.1.10 - Коначан изглед носеће плоче и плоче механизма додавача хмеља



микропрекидача. На слици испод се налази готов модуларни подсклоп првог суда заједно са објашњењем функционисања додавача хмеља (Слика 3.1.1.1.11).



Слика 3.1.1.1.11 - Модуларни подсклоп првог суда са објашњењима

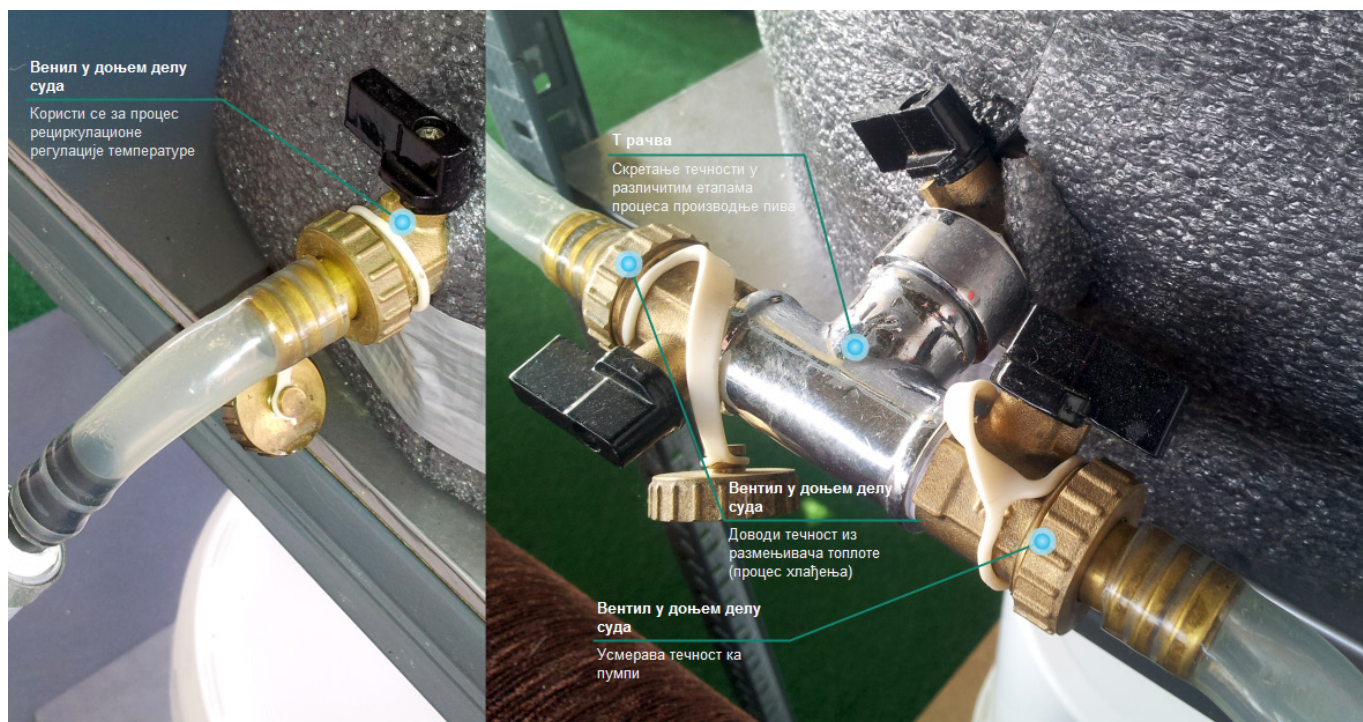
3.1.1.2 Други суд - Мешаона / Суд за укомљавање

Други суд се користи, као што је претходно наведено, за процес укомљавања и мешања сладовине. Да би то било изводљиво било је потребно уградити следеће компоненте:

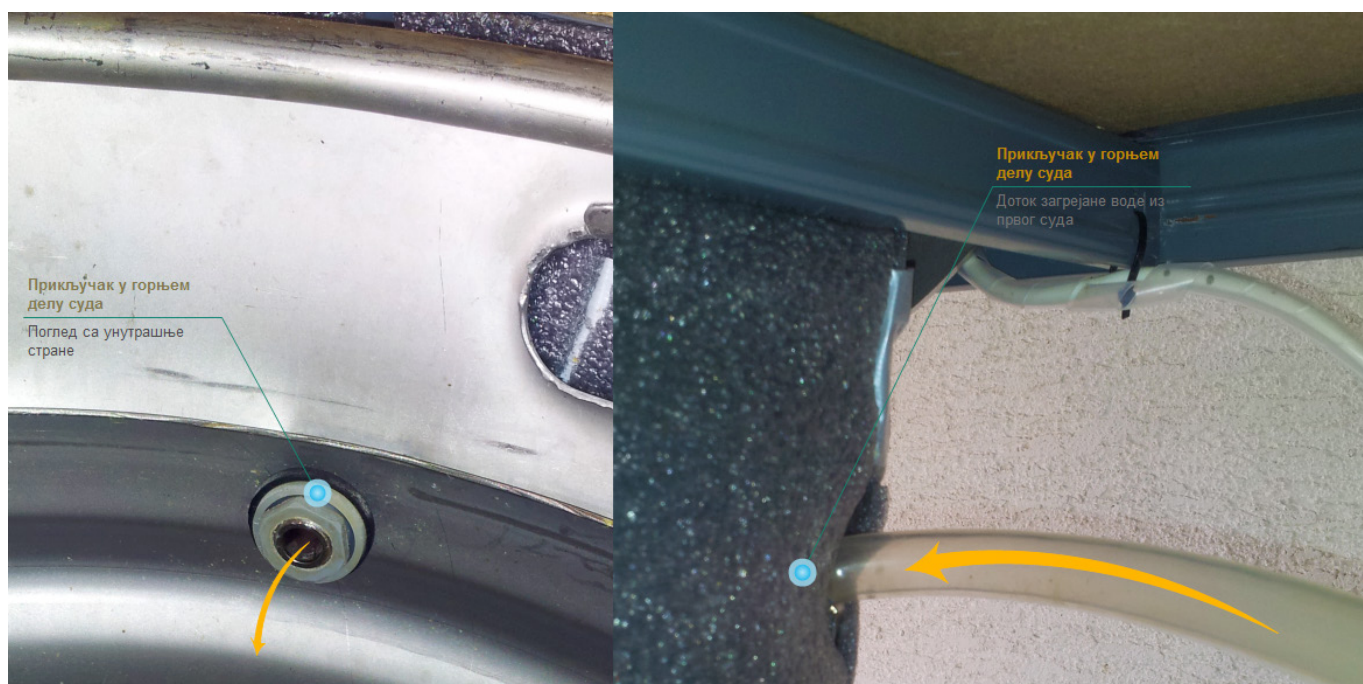
1. Прикључци за утакање и истакање

Као и у првом суду морали су да постоје прикључци са вентилима у доњем делу суда, на слици 3.1.1.2.1 су приказани прикључци. Прикључни делови су направљени од прохромске цеви пречника 30 mm, са урезаним навојем од 3/4 цола. Са судом су повезани нераскидивом везом методом заваривања. На прикључке су повезани куглични вентили. Са једне стране суда, као и у првом суду, се налази Т рачва са два куглична вентила за скретање смера протока течности.

У горњем делу суда постоје још два прикључка. Они не поседују кугличне вентиле. Један од њих се користи за утакање загрејане течности из првог суда, а намена другог ће бити описана у даљем тексту. На слици 3.1.1.2.2 се налази прикључак за довод течности из првог суда.



Слика 3.1.1.2.1 - Вентили у доњем делу суда са функционалним објашњењима



Слика 3.1.1.2.2 - Прикључак у горњем делу суда са функционалним објашњењем

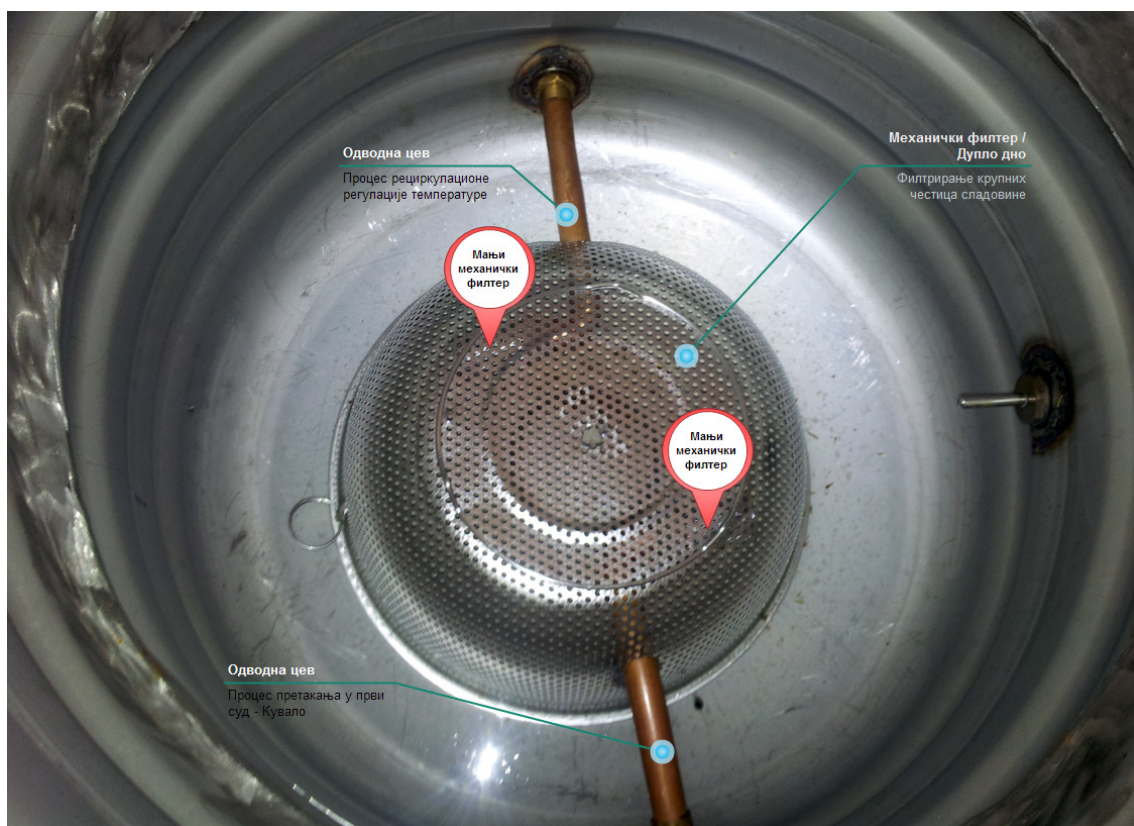
2. Механички филтер - Дупло дно

Механички филтер у другом суду има знатно важнију улогу филтрирања него исти у првом суду. Он мора да одвоји велику масу требера (укомљена сладовина) од два мања механичка филтера унутар њега. Његова примарна намена је да задржи крупне честице сладовине (пљевике, крупна зрна јечма, итд.) да не прођу до мањих филтера који су повезани на систем одвода течности из другог суда.

Дупло дно - механички филтер је направљен од нерђајућег перфорираног лима са пречником отвора 1 mm. Конструкција дуплог дна је у облику калоте која заједно са судом формира сочивасту запремину на самом дну суда. У овако формираној запремини се налази филтрирана сладовина која обезбеђује несметан рад пумпи за истакање течности. У супротном, дошло би до

нагомилавања честица сладовине у цевима и пумпама, чиме би се процес истакања блокирао. У сврси додатне оптимизације и појефтинјења конструкције система, овај филтер служи и као држач мешача сладовине.

На слици 3.1.1.2.3 се налази изглед дуплог дна са одводним монтираним одводним цевима и мањим механичким филтерима унутар њега.



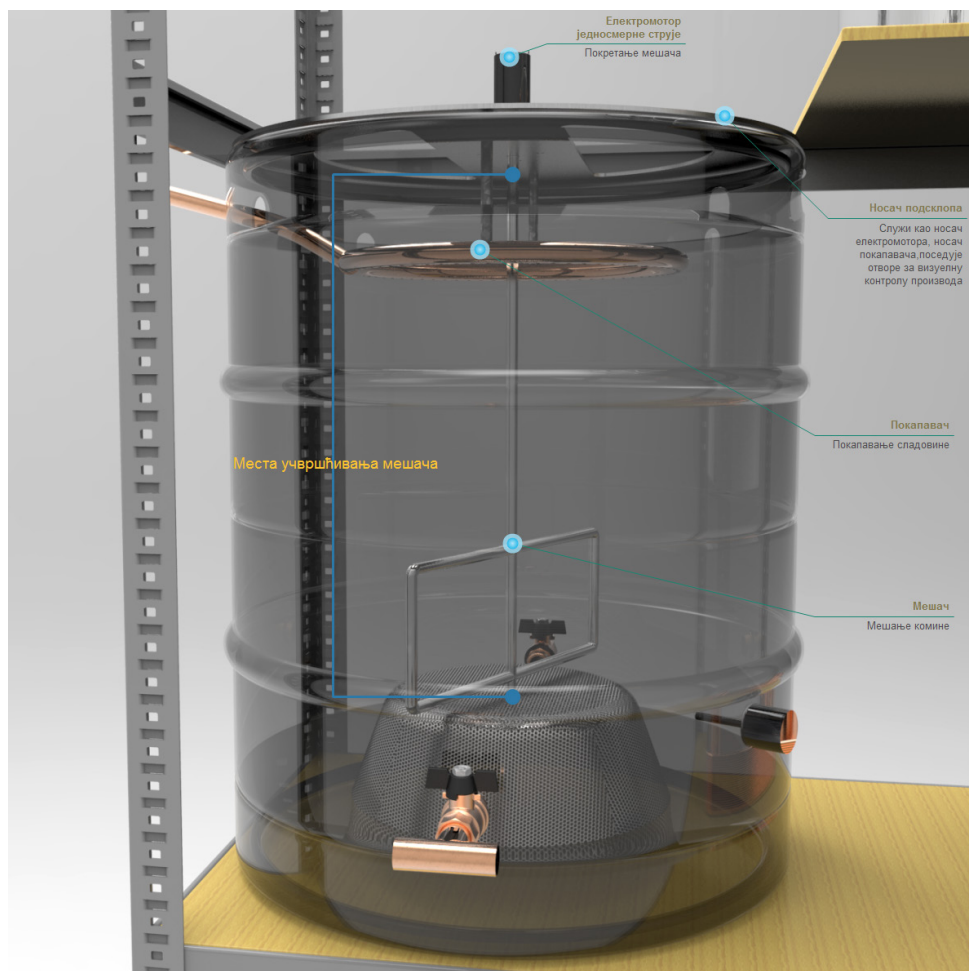
Слика 3.1.1.2.3 - Изглед дуплог дна са одводним цевима и мањим механичким филтерима

3. Мешач / Подсклоп носача електромотора

Мешач служи за мешање сладовине у процесу укомљавања, чиме поспешује разградњу ензима на просте шећере. Направљен је од материјала шипкастог облика - прохромски челик. Методом заваривања се направила једноставна конструкција, правоугаоног облика која је омогућила мешање сладовине. Мешач је прикључен завртањском везом за електромотор једносмерне струје и тиме се омогућило његово покретање. Тело електромотора је причвршћено за прохромски носач, плочастог облика који је израђен заједно са модуларним подсклопом првог суда. Радионички цртеж носача се налази на слици 3.1.1.1.9 у доњем левом углу.

Због велике масе подсклопа носача електромотора, није било потребе за додатним учвршћивањем истог за суд. Обзиром на велику густину комине која прави велики отпор кретању мешача, а он је само завртањском везом везан за електромотор, било је потребно додатно учврстити доњи део мешача за дупло дно. У супротном ротационо кретање мешача би било поремећено и његова оса ротације би била померена.

Да би се боље разумела проблематика мешача, на слици 3.1.1.2.4 је приказан моделиран склоп мешаоне, тј. суда за укомљавање заједно са сликом реалног изгледа (слика 3.1.1.2.5).



Слика 3.1.1.2.4 - Моделиран склоп мешаоне



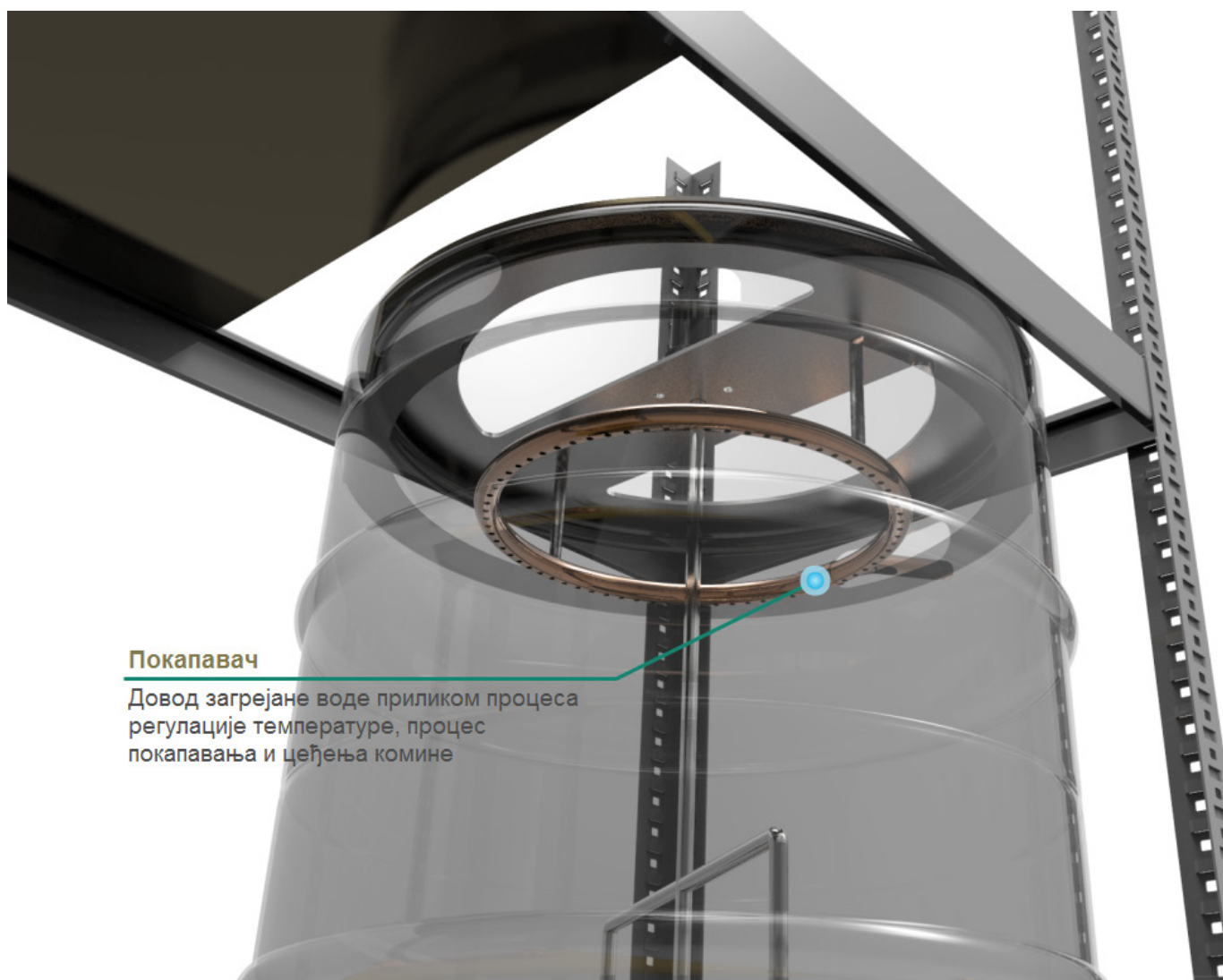
Слика 3.1.1.2.5 - Слика реалног изгледа мешаоне

4. Покапавач / Други горњи прикључак

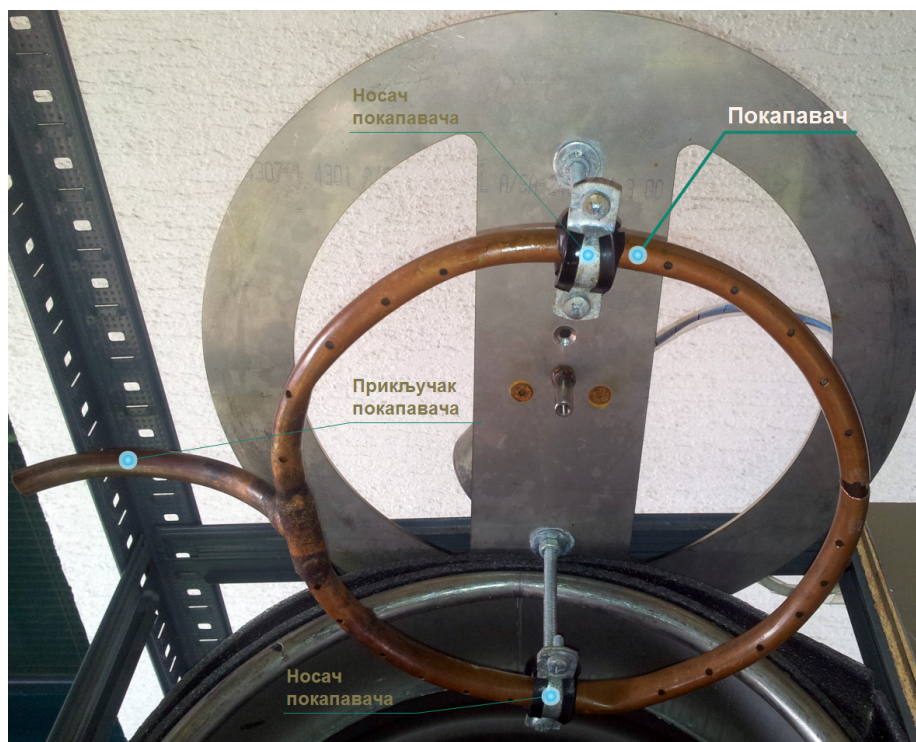
Овај део је издвојен од дела где су објашњени прикључци овог суда, због веће везе прикључка са функционисањем система рецикулационе регулације температуре (H.E.R.M.S.), јер није део суда већ подсклопа носача електромотора.

Прикључак је облику торуса. Направљен је од мекане бакарне цеви пречника 15 mm и перфориране са стране цеви окренуте ка дну. Пречник отвора перфорације на цеви је 4 mm. Ови отвори служе за покапавање комине током посебног дела процеса укомљавања укључујући и процес регулације температуре. Тај посебан део се односи на део претакања течности из суда за укомљавање у суд за кување, где је потребно са што тањим и што већим бројем млазова прекрити већу површину комине. Овај део производње пива је специфичан, јер температура течности у том процесу је до 15 степени целзијуса већа од температуре укомљавања чиме се постиже максимално убирање беланчевина из сладовине.

Покапавач служи и у сврхе рецикулационе регулације температуре, где се помоћу њега загрејана вода из размењивача топлоте равномерно распоређује по површини комине. На слици 3.1.1.2.6 се налази покапавач који је моделиран и убачен у склоп мешаоне, такође се налази његов реалан изглед на носачу електромотора на слици 3.1.1.2.7.



Слика 3.1.1.2.6 - Моделиран склоп мешаоне са истакнутим покапавачем



Слика 3.1.1.2.7 - Реалан изглед покапавача у подсклопу носача електромотора

3.1.2 Систем хлађења

Након процеса укомљавања и кувања неопходан је систем хлађења да би се температура пива брзо спустила на температуру захтевану технолошким процесом.

Систем хлађења се налази у средишњем делу конфигурације система приказаном на слици 3.1.1. Позиција размењивача топлоте је таква да би се хидростатичким притиском из првог суда омогућио несметан проток течности кроз њега до пумпе. Принцип хлађења је на почетку пројекта био статички, тј. врућа течност је циркулисала кроз размењивач топлоте - хладњак који се налази у суду са расхладним медијумом (мешавина воде и леда). Овакав начин хлађења није задовољио технолошке захтеве процеса из разлога што је разлика у запремини расхладног медијума и вруће течности била велика, тако да се расхладни медијум брзо загревао. На слици 3.1.2.1 приказан је изглед система хлађења.

Суд у коме се налази размењивач топлоте је термоизолациона посуда. Размењивач топлоте је цевастог типа израђен од бакра. Његова примарна намена је била загревање воде у гасним котловима. Сви размењивачи имају веома добре температурно-проводне карактеристике, самим тим што је за израду оваквог примерка било потребно око 8 метара цеви. На дну размењивача топлоте се налазе расхладна ребра везана за цеви која додатно одводе топлоту од медијума који протиче кроз њих.

Као што је горе наведено, статички принцип расхлађивања није испунио захтеве технолошког процеса, па је уведена рециркулациона метода расхладног медијума. Принцип функционише на повременом утакању и истакању хладне течности у суд са размењивачем. Између осталог, ово је једна од мана овог система - цеваста размењивач.



Слика 3.1.2.1. - Размењивач топлоте

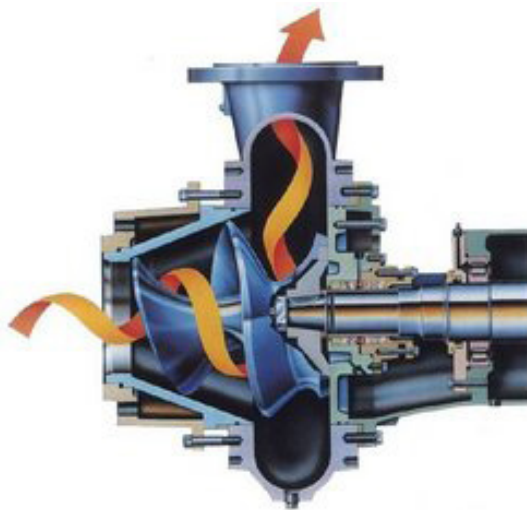
3.1.3 Пумпе

За процесе претакања и рецикулације течности из другог у први суд, због очигледне висинске разлике, било је потребно коришћење пумпи.

За ове намене у прехранбеној индустрији користе се завојне или перисталтичке пумпе.

1. Завојне пумпе

Завојне пумпе или запреминске пумпе раде на принципу завојног вретена које текући медијум погуравају од улаза пумпе ка излазу. Раде на веома малим обртајима (око 200 о/мин) чиме одржавају стабилну структуру текућине. Ове пумпе имају самоусисну карактеристику тако да не захтевају улазни притисак. Такође, предност ових пумпи је у томе што флуид који протиче кроз њих нема контакта са подмазивајућим средствима покретних делова пумпе. Мана ове пумпе је цена, тако да је њихово коришћење било немогуће у овом пројекту. Цена једне овакве пумпе износи око >2000 евра. Слика и принцип рада ове пумпе је приказан на слици 3.1.3.1.

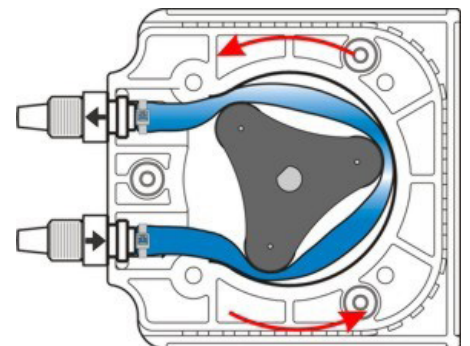


Слика 3.1.3.1 - Завојна пумпа

2. Перисталтичке пумпе

Ове пумпе су веома карактеристичне због начина функционисања. Оне представљају такозване, бесконтактне пумпе. Раде на вакумском принципу и користе се у прехранбеној као и у фармацеутској индустрији. Принцип рада ове пумпе и њен облик је приказан на слици 3.1.3.2.

Наравно, тржиште за проналазак ових пумпи мањег капацитета и запреминске моћи, не постоји на територији Србије. Постоје само веће пумпе за специјалне намене чије су цене абнормално велике. Суштина, наспрот једноставном принципу рада ове пумпе, зашто је цена тако висока је та да, такозвано перисталтичко црево које се налази унутар ње је направљено од специјалног композитног материјала који одржава своју структуру и еластичност веома дуго. Поготово, када постоје велике осцилације у температури флуида, цене иду још више. Довољно је рећи да обично силиконско црево може да издржи тек неколико минута у току рада пумпе, а да притом не изгуби еластичност и структуру.



Слика 3.1.3.2 - Принцип рада перисталтичке пумпе

То су све били разлози зашто ни ова пумпа није могла да дође у обзир за коришћење у овом пројекту.

3. Центрифугалне пумпе

Принцип рада ових пумпи је вероватно један од најчешћих принципа претакања течности помоћу пумпи. Користе се за извлачење воде из бунара (потапајуће пумпе), муљне пумпе, код циркулације воде у систему централног грејања итд. Мана ових пумпи је у томе што су осетљиве на промене улазног притиска и постојање ваздушних чепова. Цена ових пумпи је незамисливо нижа од претходно наведених.

Из претходно наведених разлога, избор пумпи које ће бити уграђене у овај систем је сведен на само једну, а то је центрифугална пумпа.

Пумпе које су коришћене у овом пројекту су пумпе марке “Веко”, иначе то су пумпе за истакање течности из машина за веш.

Спецификација:

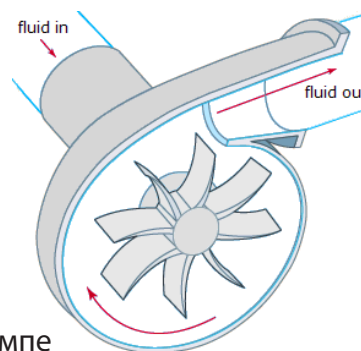
- Снага: 60 W
- Радни напон: 220 V
- Тип: Магнетна
- Проток: 10 L/min
- Максимална висина подизања: 2.5 метара

На слици 3.1.3.3 се налазе пумпе које су коришћене у овом пројекту.

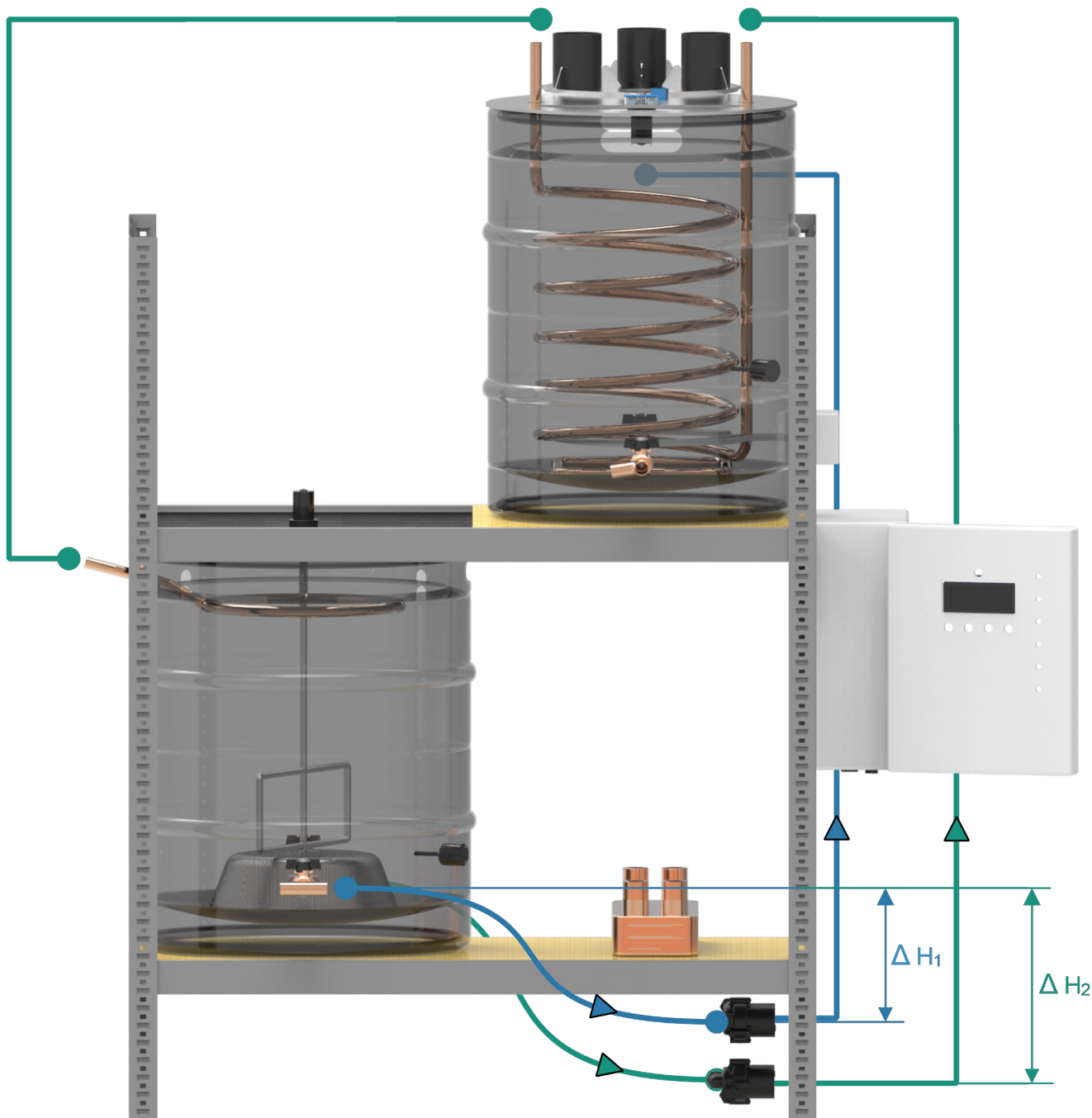


Слика 3.1.3.3 - Пумпе коришћене у овом пројекту

Позиција пумпи у склопу је веома битна, у овом случају мисли се на њихову висинску позицију. Пошто ове пумпе захтевају улазни притисак да би радиле, јер немају самоусисну карактеристику, суд из кога се истаче флуид мора да се налази изнад. На овај начин ствара се хидростатички притисак на улазу у пумпу. Принцип рада ове пумпе је приказан на слици 3.1.3.4.



Слика 3.1.3.4 - Принцип рада центрифугалне пумпе



Слика 3.1.3.5 - Позиција центрифугалних пумпи у односу на суд за укомљавање

На слици изнад је приказана висинска разлика између суда за укомљавање слада и пумпи чиме се постигао улазни притисак у пумпе.

3.1.4 Црева за повезивање судова

За претакање течности из суда у суд су коришћена силиконска-провидна црева. Разлог коришћења ових црева је што она омогућавају визуелно праћење протока флуида. На тај начин било је могуће увидети разлог настанка проблема у процесу претакања, што је честа појава код коришћења центрифугалних пумпи (појава ваздушног чепа, нагомилавање крупних материја у систем, итд.). Још један разлог зашто су ова црева коришћена је тај што је њихова цена приступачна, као и једноставан начин монтаже за разлику од коришћења бакарних цеви.

Мана ових цеви је што се деформишу под утицајем температуре, што доводи до смањења улазног притиска у пумпе, преламања, тј. смањења проточног потенцијала.

За повезивање пумпи и другог суда коришћена су непровидна ребраста црева. Коришћена су због великог улазног и излазног пречника прикључака на пумпи. Наравно, за повезивање истих на прикључке другог суда коришћене су бакарне цеви за редуковање пречника. На слици 3.1.4.1 су приказана коришћена црева.



Слика 3.1.4.2 - Изглед система са коришћеним цревима

Да би се побољшао квалитет крајњег производа као и његова бистрина, накнадно је уграђен фини механички филтер. Овај филтер у себи има уложак од прохромског плетеног платна са окнима величине неколико десетих делова милиметра, тиме се скоро у потпуности филтрирала течност која је пролазила кроз њега. Филтер је постављен на цреву које повезује други и први суд, тако да приликом претакања комине у први суд она сеfino филтрира. Изглед овог филтера је дат на слици 3.1.4.3.



Слика 3.1.4.3 - Фини механички филтер

3.1.5 Електро ормар

Можда најбитнији део овог пројекта представља **електро ормар**. До пројектовања његовог изгледа је дошло тек након прикупљања свих компонента за процес управљања укључујући и идејно решење о начину управљања системом. Идејно решење представља аутоматски програм који управља сваким делом процеса прављења пива уз могућност параметризације од стране корисника. У систем је такође имплементиран и мануални мод, због могућих проблема у контроли процеса. На основу ових података могло се почети са моделирањем електро ормара.

Габаритне мере ормара су узете на основу потребних компоненти које треба да се нађу унутар њега. Тако је усвојен облик и величина кућишта електро ормара величине 250 x 330 x 130 mm произвођача Elmark. Кућиште је израђено од квалитетне пластике беле боје. Класа механичке заштите је IP65. Што значи да је кућиште херметички затворено, потпуно изоловано од прашине и влаге. На слици 3.1.5.1 је приказан изглед кућишта.

Обзиром да је све што се тиче управљања било разјашњено, знало се да је за контролу компонента потребна визуелна комуникација са корисником у виду LED диода и LCD екрана. Да би било могуће мењање параметара система у току рада и управљање компонентама у мануалном моду, било је потребно инсталирање тастера. Број тастера је усвојен на основу броја компоненти које су биле актуатори у овом систему. А то су две пумпе, грејач и мешач. Механизам додавача хмеља је изостављен, јер његово покретање у мануалном моду је непотребно. Тако се дошло до 4 тастера. Процес такође захтева, иако је аутоматски, присуство корисника и његову правовремену реакцију (отварање/затварање вентила, програмске потврде притиском на тастер, итд.). У ту сврху било је потребно уградити пиезо звучник, што кориснику даје простор да се удаљи од система и буде звучно обавештен о тренутном стању процеса. Распоред отвора за тастере, LED диоде, пиезо звучник и LCD екран је био моделиран и касније изрезан у вратима кућишта електро ормара.

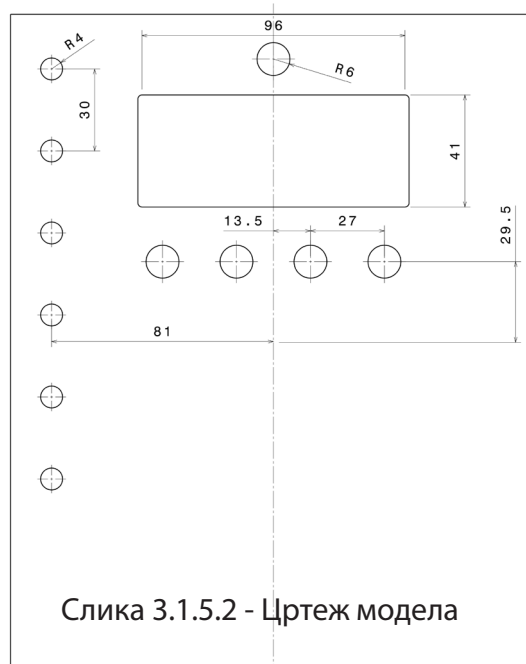
Прављење отвора на вратима електро ормара је одрађено помоћу глодалице *Roland MDX - 40* у оквиру фирме *Stublina*. Радни модел који је увезен у програм *Roland Modela Player 4* је приказан у виду радионичког цртежа на слици 3.1.5.2.

Цртеж показује позиције тастера (4 отвора испод позиције отвора LCD екрана), позицију пиезо звучника (отвор изнад отвора за LCD екран) и позиције за 5 LED диода. Да би се отвори тачно позиционирали као на цртежу било је потребно наћи референтну тачку на вратима електро ормара, а то је био центар врата на коме се налазио отисак уливне дизне од процеса бризгања комада.

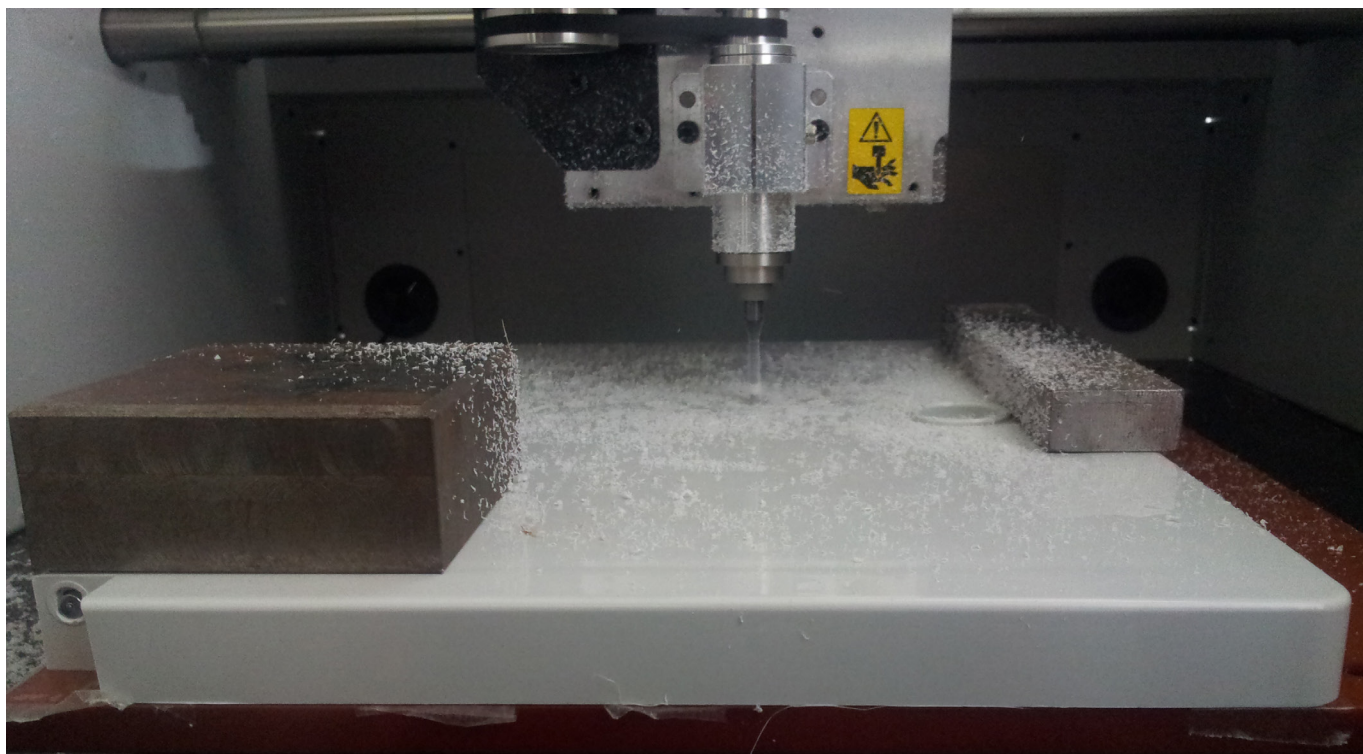
Изглед процеса прављења отвора помоћу горе поменуте глодалице се налази на слици 3.1.5.3.



Слика 3.1.5.1 - Кућиште електро ормара



Слика 3.1.5.2 - Цртеж модела



Слика 3.1.5.3 - Процес резања отвора на вратима електро ормара

Након резања отвора на вратима електро ормара уграђене су претходно наведене компоненте. Све компоненте на вратима ормара су купљене у фирми *ProElectronic* са седиштем у Нишу. У то спадају LED диоде и њихова кућиштима, тастери, LCD екран и пиезо звучник. На сликама испод се налази изглед тих компонената и изглед врата ормара након уграђивања истих.



Слика 3.1.5.4 - Компоненте уграђене у врата електро ормара



Слика 3.1.5.5 - Врата електро ормара

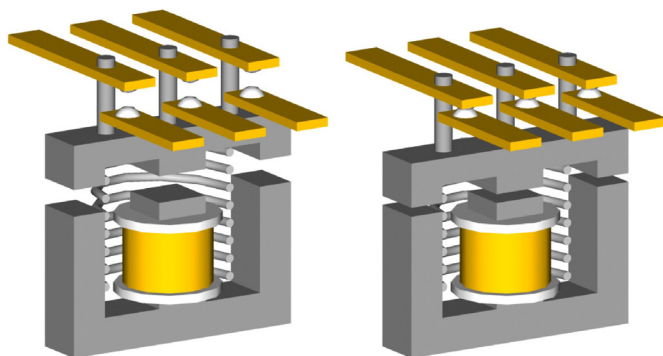
3.1.5.1 Распоред компоненти унутар ормара

Компоненте које се налазе унутар ормара су већ описане у оквиру другог поглавља овог рада. Једина компонента која није описана у оквиру другог поглавља је контактор. Он је коришћен за сигурнију манипулацију јаких струја у оквиру електро ормара. Контактори су коришћени за покретање компоненти чији је извор напајања наизменична струја, а то су:

- Грејач
- Пумпа 1 - Процес рецикулације
- Пумпа 2 - Процес претакања

Коришћени контактори су приказани на слици 3.1.5.1.1.

Контактор ради на сличном принципу као и релеј. У себи има индуктивни калем, који под напоном ствара магнетно поље које покреће полугу и на тај начин ствара контакт између извора напајања и компоненте коју покреће то напајање. За разлику од релеја, контактор у исто време остварује контакт између више редно одвојених веза. На слици испод је приказана шема функционисања контактора.



Слика 3.1.5.1.2 - Шема функционисања контактора



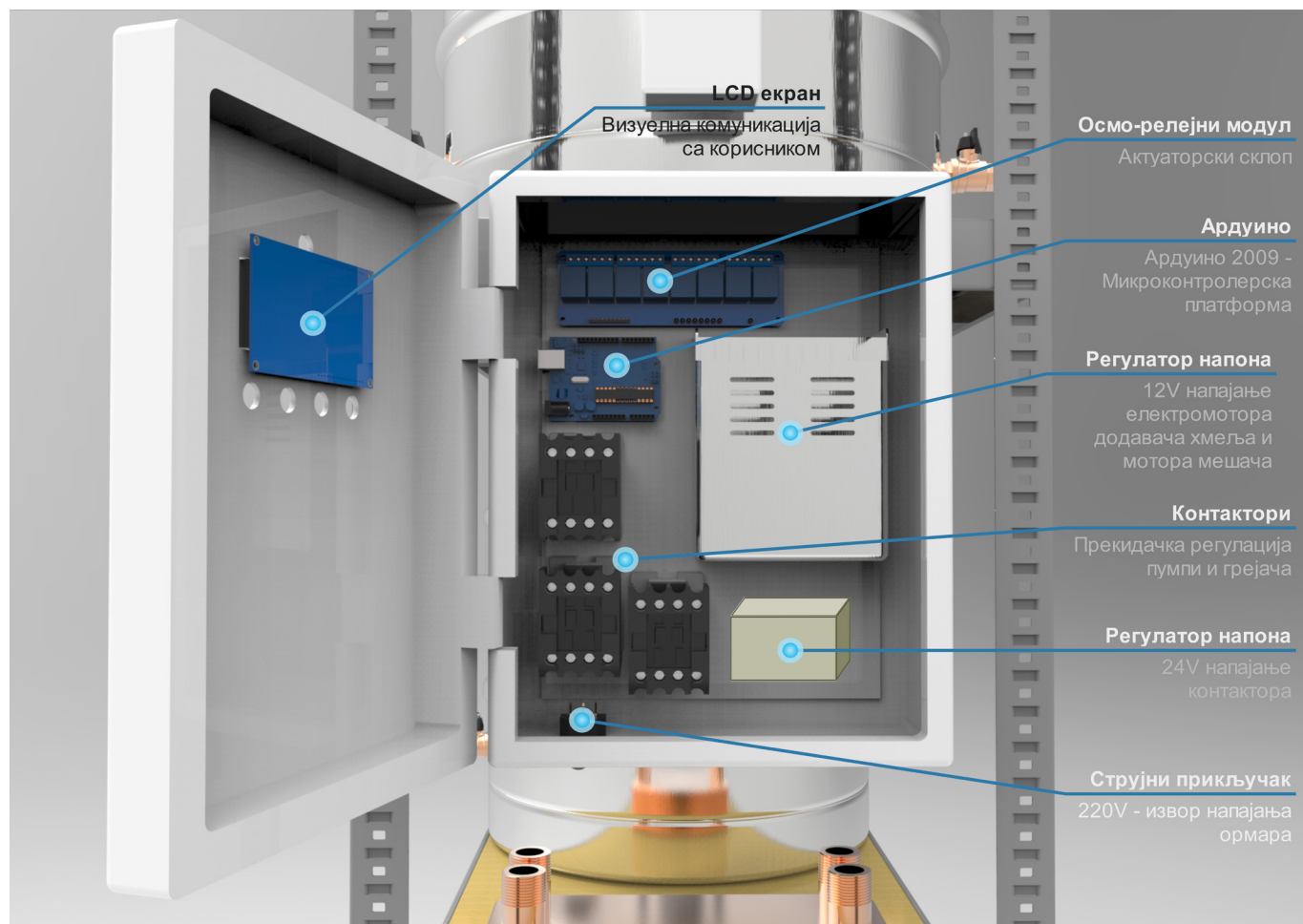
Слика 3.1.5.1.1 - Контакттор марке "Iskra"

Са претходним описом су објашњене и презентоване све компоненте које ће се наћи у унутрашњости електро ормара.

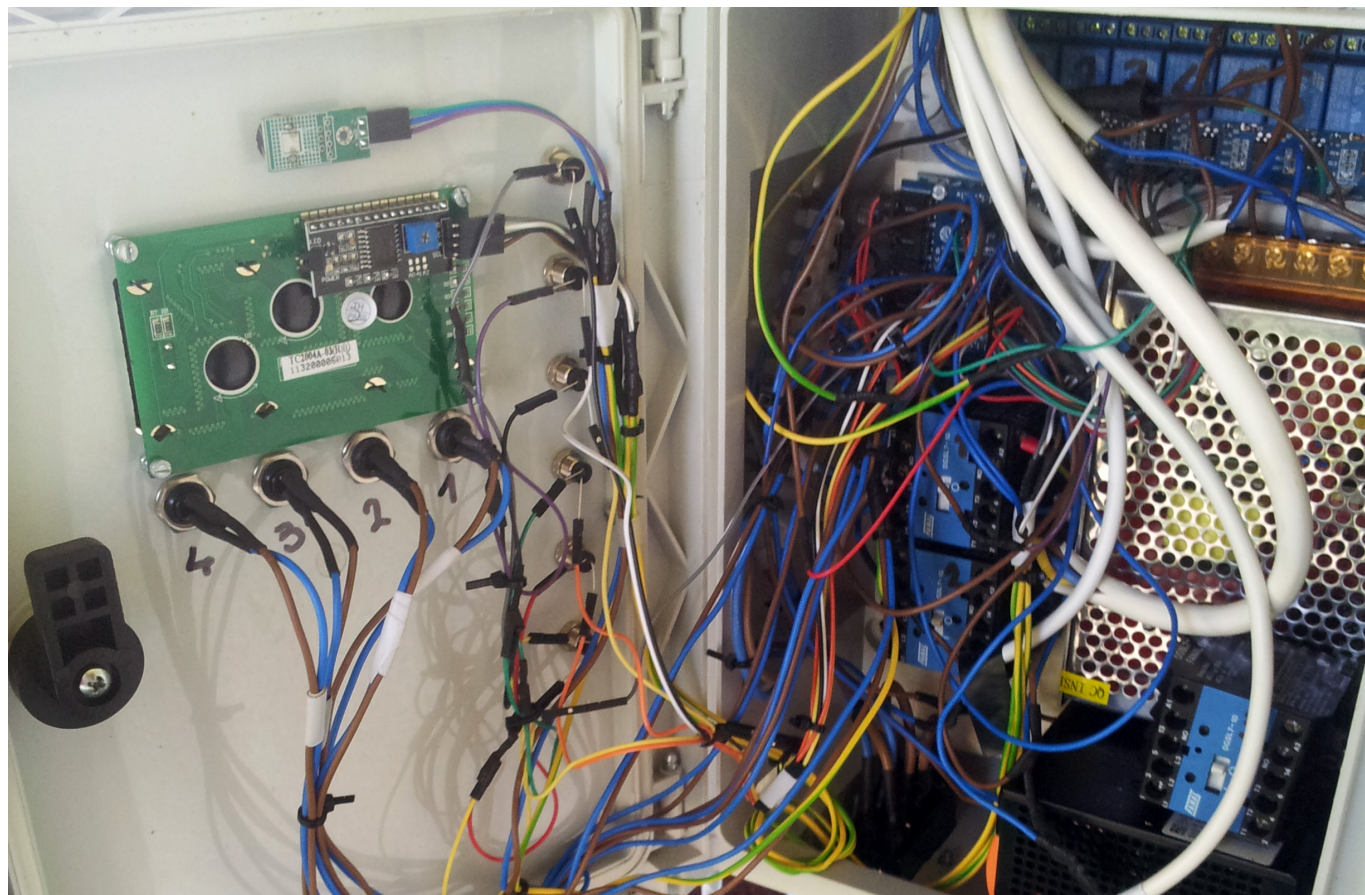
Распоред компонената унутар ормара је одређен према усвојеним позицијама уводника за каблове који се налази са горње стране кућишта ормара и позицијом извора напајања од 220V који се налази са доње стране ормара. Тиме је релејски склоп позициониран у горњем делу, напајања једносмерне струје од 12V и 24V су постављена дуж десне стране, Ардуино је постављен непосредно испод релејског склопа и контактори дуж леве стране ормара. На слици 3.1.5.1.3 је приказан моделиран изглед унутрашњости ормара.

Из очигледних разлога, релејски склоп је морао бити постављен близу отвора за уводник каблова, чиме се постигло лакше повезивање периферних компонената за њега. Напајања су постављена дуж леве стране ормара (једно испод другог) због лакшег довођења наизменичне струје и повезивања са релејима. Обзиром на начин повезивања ардуина и релејског склопа (преко џампер каблова који имају малу дужину) Ардуино се налази одмах испод. Контактори се директно повезују са извором напајања наизменичне струје, одмах након осигурача, па се налазе дуж леве стране поред стезаљки за каблове.

На слици 3.1.5.1.4 се налази коначан изглед електро ормара са свим компонентама и уграђеним кабловима.



Слика 3.1.5.1.3 - Моделиран изглед унутрашњости електро ормара

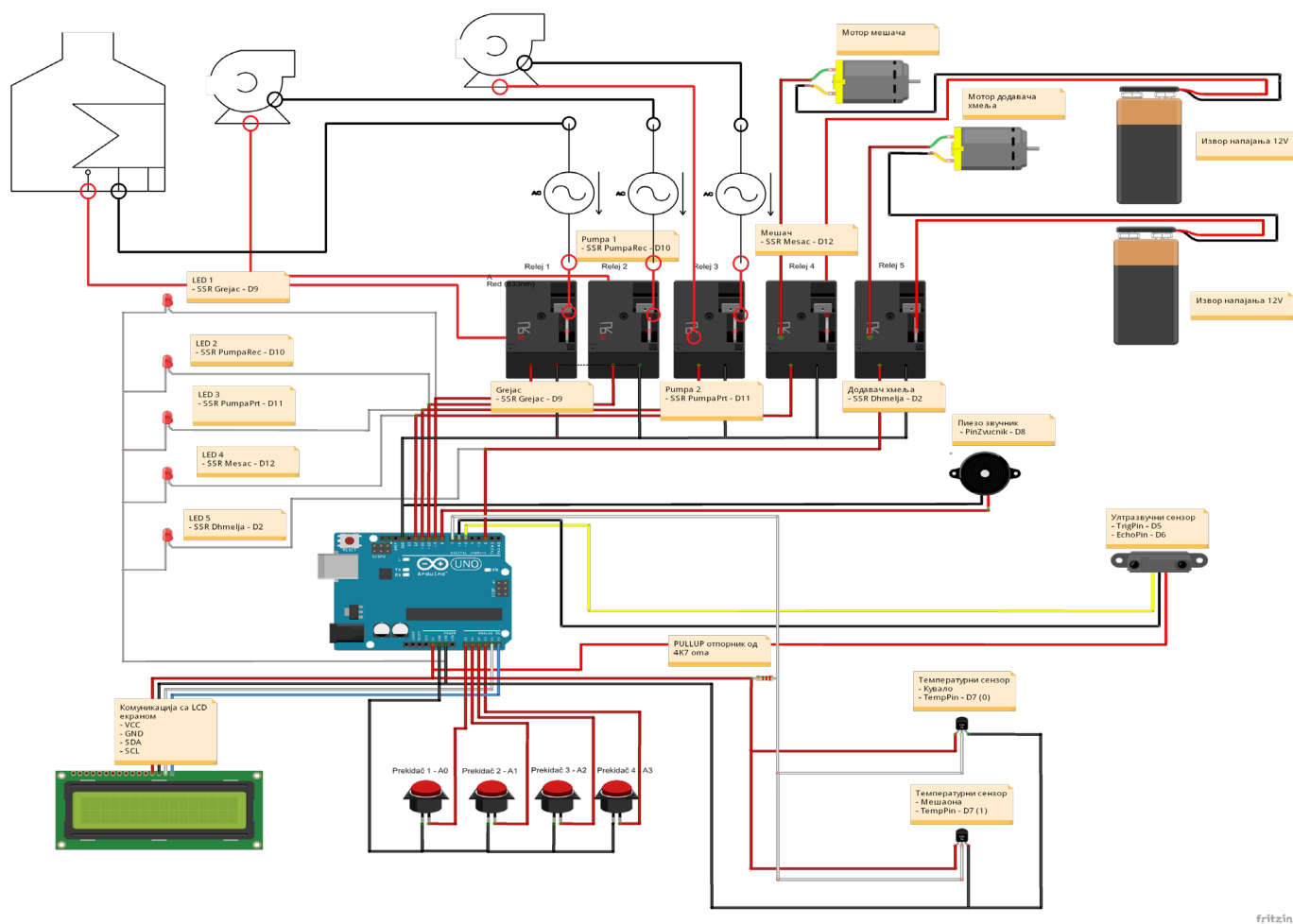


Слика 3.1.5.1.4 - Стваран изглед електро ормара и његове унутрашњости

3.1.5.2 Електро шема система

Електро шема повезивања компоненти са ардуином је урађена помоћу програма *Fritzing*. Овај програм је намењен за визуелно представљање компоненти заједно са количином конекција потребних за повезивање истих са ардуином. На слици 3.1.5.2.1 је приказана електро шема система.

У главном шематском приказу система је изостављена веза између релеја и контактора за контролисање уређаја чији је извор напајања наизменична струја, као и напајање калема контактора. Изостављена је зато што би представљање ових спрега само додатно закомпликовало шему, а њихово постојање у склопу је само из сигурносних разлога.



Слика 3.1.5.2.1 - Главна електро шема система

Спецификација улазно/излазних сигнала:

- | | |
|------------------------------------|---|
| 1. A0 - Тастер 1 | 9. D5 - TriggerPin / Ултразвучни сензор |
| 2. A1 - Тастер 2 | 10. D6 - EchoPin / Ултразвучни сензор |
| 3. A2 - Тастер 3 | 11. D7 - Температурни сензори |
| 4. A3 - Тастер 4 | 12. D8 - Пиезо звучник |
| 5. A4 - SCL | 13. D9 - Релеј грејача |
| 6. A5 - SDA | 14. D10 - Релеј пумпе за рецикулацију |
| 7. D2 - Релеј додавача хмеља | 15. D11 - Релеј пумпе за претакање |
| 8. D3 - Микротастер додавача хмеља | 16. D12 - Релеј мотора мешача |

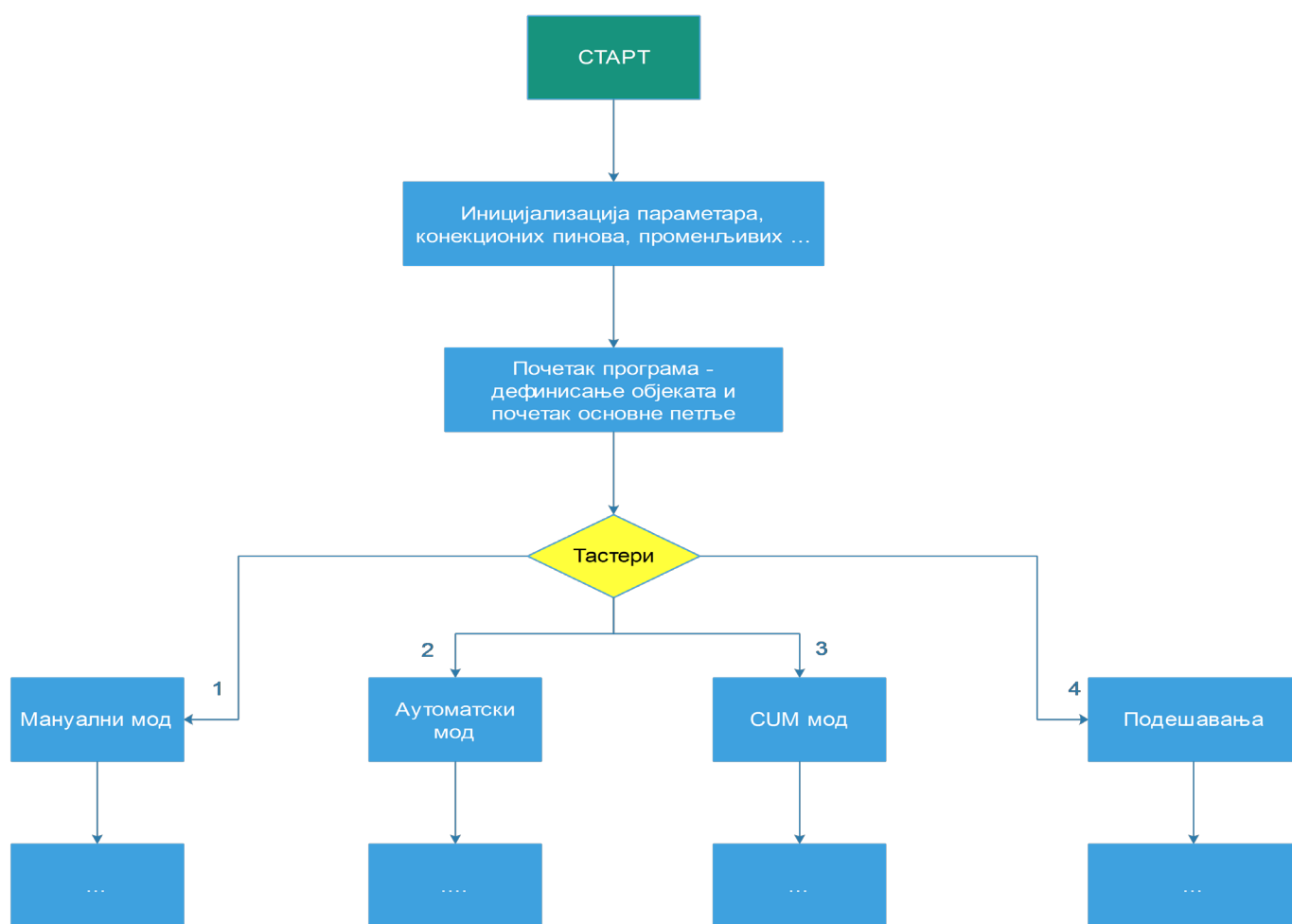
4. Алгоритам програма и повратне спреге

У оквиру овог поглавља ће бити представљен алгоритам рада програма и спрега између програма и физичких компоненти у оквиру система.

Основни алгоритам рада програма је био референтна тачка за израду система, тако да је и први настао. Сваки део основног алгоритма ће бити посебно објашњен кроз шематски приказ рада програма преко блок дијаграма и њихову функционалну примену у систему.

4.1 Алгоритам рада система

Основни алгоритам рада програма је приказан преко блок шеме на слици испод (слика 4.1.1).



Слика 4.1.1 - Основни алгоритам рада система

Као сваки почетак програма код компоненти са архитектуром микроконтролера и овај почиње са иницијализацијом параметара, унетих библиотека, конекционих пинова, дефинисаних променљиви, иницијалним вредностима променљиви и пинова, итд..

Овај систем има многобројне дефинисане библиотеке због такође многобројних протокола комуникације, многих функција које су дефинисане у оквиру њих за олакшавање рада са променљивима, такође много променљивих које су морале да постоје да би параметризација система била могућа као и рачунање времена. На претходној страни је приказана спецификација улазно/излазних променљиви на коју ћу се често позивати.

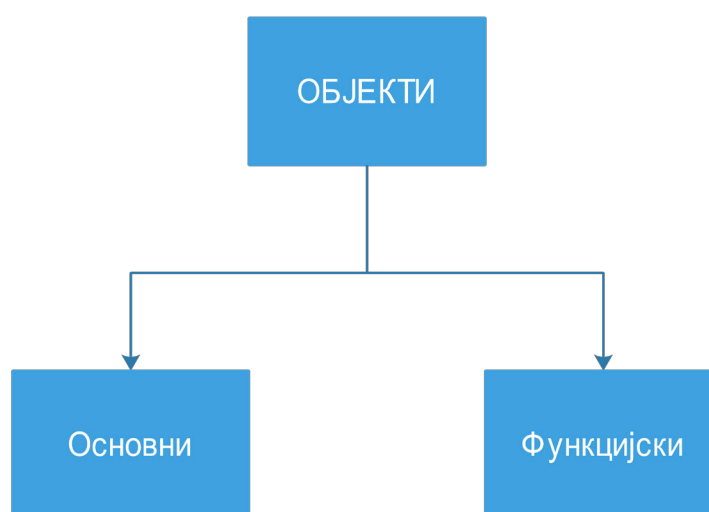
Програм је сувише компликован да би се свео само на једну петљу, тј. било је немогуће, зато се дизајнирању програма приступило преко објектно-оријентисаног програмирања.

Решавање проблема парадигмом објектно-оријентисаног програмирања, је врло слично људском начину размишљања и решавању проблема. Састоји се од идентификовања објеката и постављање објеката који ће се користити у одговарајућу секвенцу за решење одређеног проблема. Ради се о дизајну објеката чија ће понашања као јединица и у њиховој међусобној интеракцији, решити одређени проблем. Интеракција између објеката се састоји у размени порука, где одређена порука усмерена према одређеном објекту, покреће уграђене операције у том објекту, чиме се решава део обично ширег и сложенијег проблема. Уопштено гледано, објектно-оријентисано решавање проблема се састоји из четири корака:

1. Идентификовање проблема
2. Идентификовање објеката који су потребни за његово решење
3. Идентификовање порука које ће објекти међусобно слати и примати
4. Креирање секвенце порука објектима, које ће решавати проблем или проблеме

Тако се дошло до низа објеката, тј. потпрограмских делова, који служе за решавање низа релативно комплексних спрега и прорачуна у оквиру система. Овакав приступ је омогућио стварање вишестраног програма који је могао бити смештен у мали меморијски простор, а опет имао веома добар одзив у интеракцији корисника и сензора са њим.

Пре почетка основне петље приказане на слици 4.1.1, дефинисани су потпрограми, тј. објекти, хијерархија њих се разликује и према намени и према функцији у комплетном програму. Из претходно наведеног разлога они су подељени у две групе:



Слика 4.1.2 - Подела објеката

Основне објекте представљају одвојене целине којих има четири, што се може видети на слици 4.1.1. Они су пред корисником након покретања основне петље, и дефинисани су као:

- void **manualmode()**
- void **automode()**
- void **cipmode()**
- void **setupmenu()**

Коришћени су енглески називи у програму за дефинисање појединих променљивих, објеката и других компонената програма због лакшег референцирања и програмирања. Функцијске објекте представљају објекти нижег хијерархијског ранга, јер су они коришћени у сврхе дефинисања одређених параметара, рачунања вредности прикупљених са сензора, решавање функција на

основу параметара, итд.. Функцијске објекте представљају:

- void **Core()**
- void **beep**(unsigned char **PinZvucnik**, int **frequencyInHertz**, long **timeInMilliseconds**)
- void **welcomescreen()**
- void **mainmenu()**
- void **formattime**(int **hours**, int **mins**, int **secs**, char **time[]**)
- void **formattimeseconds**(long **secs**, char **time[]**)
- void **calcstrike()**
- void **uss()**

Прво ћу објаснити разлог постојања функцијских објеката, из разлога што се они појављују и позивају у основним објектима.

4.1.1 Функцијски објекти система

4.1.1.1 Void Core()

Овај функцијски објекат служи као модул за добијање и дефинисање вредности на екрану добијених са дигиталних сензора температуре и прорачуна укупног протеклог времена од покретања система. Он користи претходно дефинисане променљиве којима додељује вредности на основу промена стања сензора температуре у времену, као и промена у оквиру објекта **formattimeseconds()**. Изглед кода који се налази у оквиру овог објекта је приказан на слици 4.1.1.1.1.

```
void Core()
{
    sensors.requestTemperatures();

    tempsenzor1 = sensors.getTempCByIndex(0), 1;
    lcd.setCursor(5, 3);
    lcd.print(tempsenzor1);

    tempsenzor2 = sensors.getTempCByIndex(1), 1;
    lcd.setCursor(15, 3);
    lcd.print(tempsenzor2);

    lcd.setCursor(0, 3);
    lcd.print("TK ");
    lcd.setCursor(3, 3);
    lcd.write(1);
    lcd.setCursor(7, 3);
    lcd.print((char)223); // karakter oznake stepena
    lcd.print("C");

    lcd.setCursor(10, 3);
    lcd.print("TM ");
    lcd.setCursor(13, 3);
    lcd.write(1);
    lcd.setCursor(17, 3);
    lcd.print((char)223); // karakter oznake stepena
    lcd.print("C");

    while(millis() - previous_millis_value >= 1000)
    {
        cumulativeSeconds++;
        previous_millis_value += 1000;
    }
    formattimeseconds(cumulativeSeconds, totaltime);

    if (totaltime != lasttotaltime) // ispisuje ukupno proteklo vreme na ekranu
    {
        lcd.setCursor(0, 2);
        lcd.print("TOT:");
        delay(100);
        lcd.setCursor(5, 2);
        lcd.print(totaltime);
        lasttotaltime = totaltime;
    }
}
```

Слика 4.1.1.1.1 - Објекат Void Core()

4.1.1.2 Void beep(параметри)

Овај функцијски објекат се користи за генерисање звучног сигнала преко пиезо компоненте. За разлику од претходног он је параметризован, тј. параметарски је могуће мењати тон звука и његово трајање приликом позивања овог објекта. Његова синтакса је приказана испод:

void beep(unsigned char PinZvucnik, int frequencyInHertz, long timeInMilliseconds)

Приликом позивања уместо интелектуалног променљиве **frequencyInHertz** се унесе жељена фреквенција осциловања напонског сигнала, и уместо **long** променљиве **timeInMilliseconds** се унесе дужина трајања сигнала. Синтакса позивања функцијског објекта у програму биће приказана у даљем тексту. Изглед кода у оквиру овог објекта је приказан на слици 4.1.1.2.1.

```
void beep (unsigned char PinZvucnik, int frequencyInHertz, long timeInMilliseconds) // Funkcija za proizvodnju zvuka :)
{
    int x;
    long delayAmount = (long)(1000000/frequencyInHertz);
    long loopTime = (long)((timeInMilliseconds*1000)/(delayAmount*2));
    for (x=0;x<loopTime;x++)
    {
        digitalWrite(PinZvucnik,LOW);
        delayMicroseconds(delayAmount);
        digitalWrite(PinZvucnik,HIGH);
        delayMicroseconds(delayAmount);
    }
}
```

Слика 4.1.1.2.1 - Функција void beep()

4.1.1.3 Void welcomescreen()

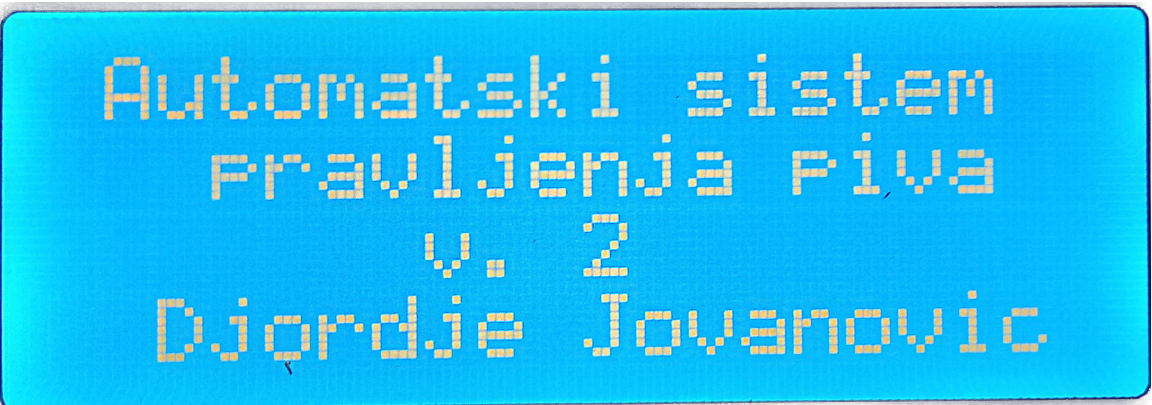
Као што и сам назив овог објекта каже, његова сврха је приказивање иницијалног екрана приликом покретања система. Он приказује претходно дефинисан текст у оквиру овог објекта, нема никакве прорачунске активности, позива број верзије коју повлачи из иницијалног дела овог програма и на крају позива објекат void beep(). Облик кода у оквиру овог објекта је приказан на слици 4.1.1.3.1.

```
void welcomescreen()
{
    lcd.clear();
    lcd.setCursor(1, 0);
    lcd.print("Automatski sistem");
    delay(50);
    lcd.setCursor(3, 1);
    lcd.print("pravljjenja piva");
    delay(50);
    lcd.setCursor(7, 2);
    lcd.print("v.");
    delay(50);
    lcd.setCursor(10, 2);
    lcd.print(verzija);
    delay(50);
    lcd.setCursor(2, 3);
    lcd.print("Djordje Jovanovic");

    beep(PinZvucnik,2093,500); //C: ton C (C7 ton je povezan sa linkom navedenim pre funkcije za proizvodnju zvuka) traje 500ms
}
```

Слика 4.1.1.3.1 - Функција void welcomescreen()

У оквиру ове функције је показано како се позива објекат void beep() и како се он параметарски дефинише. На слици 4.1.1.3.1 је приказан изглед LCD Екрана након укључивања система.



Automatski sistem
Pravljenja Fiva
v. 2
Djordje Jovanovic

Слика 4.1.1.3.1 - Изглед LCD екрана након укључивања система

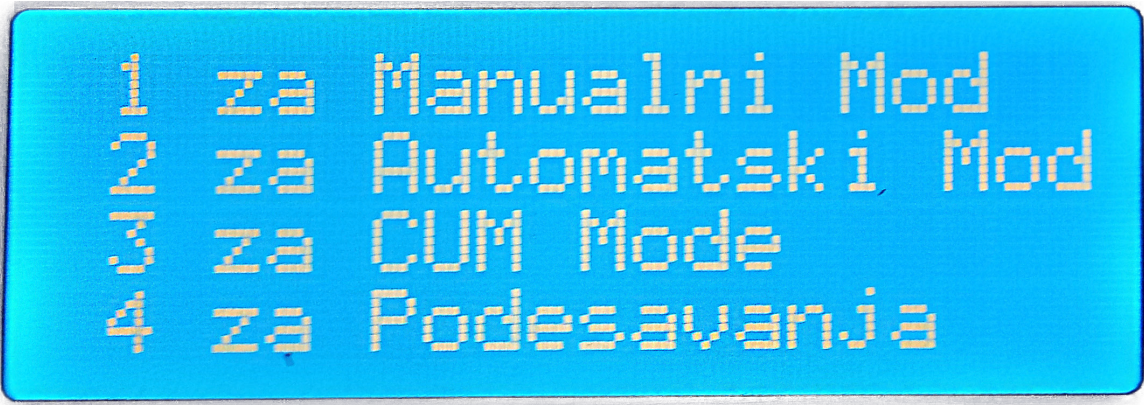
4.1.1.4 Void mainmenu()

Ова функција служи за исписивање текста који везан за иницијални изглед екрана који кориснику пружа избор између могућих основних објеката. На слици 4.1.1.4.1 је приказан изглед кода који се налази у оквиру овог објекта.

```
void mainmenu()  
{  
    lcd.clear();  
    delay(50);  
    lcd.setCursor(1, 0);  
    lcd.print("1 za Manualni Mod");  
    delay(50);  
    lcd.setCursor(1, 1);  
    lcd.print("2 za Automatski Mod");  
    delay(50);  
    lcd.setCursor(1, 2);  
    lcd.print("3 za CUM Mode");  
    delay(50);  
    lcd.setCursor(1, 3);  
    lcd.print("4 za Podesavanja");  
    delay(50);  
}
```

Слика 4.1.1.4.1 - Void mainmenu()

Након иницијализације система, он улази у while() петљу и док чека интеракцију од корисника путем тастера, на слици 4.1.1.4.2 се налази изглед LCD екрана.



1 za Manualni Mod
2 za Automatski Mod
3 za CUM Mode
4 za Podesavanja

Слика 4.1.1.4.2 - Изглед LCD екрана након иницијализације

4.1.1.5 Void formattime(параметри и променљиве)

Овај функцијски објекат служи за форматирање времена у формату сати : минути : секунде. Формат се конвертује у карактере и тако се представља на екрану. На слици испод је приказан облик кода у оквиру ове функције (слика 4.1.1.5.1).

```
void formattime(int hours, int mins, int secs, char time[]) { // formatiranje vremena u string u formi sati, miinuti, sekunde

    if (secs>60) {
        secs=secs-60;
        mins++;
    }
    if (mins>60) {
        mins=mins-60;
        hours++;
    }
    time[2]=': ';
    time[5]=': ';
    time[0]=48+(hours / 10);
    time[1]=48+(hours % 10);
    time[3]=48+(mins / 10);
    time[4]=48+(mins % 10);
    time[6]=48+(secs / 10);
    time[7]=48+(secs % 10);
}
```

Слика 4.1.1.5.1 - Функција void formattime(parametri)

Ова функција поседује параметре и променљиве помоћу којих се касније може исписивати време на екрану.

4.1.1.6 Void formattimeseconds(променљиве)

Овај објекат поседује променљиве помоћу којег се такође форматира време, и у исто време повлачи функцију void **formattime**(long secs, char time[]). На слици испод се види код у оквиру ове функције (слика 4.1.1.6.1).

```
void formattimeseconds(long secs, char time[]) { // formatiranje vremena u string u sekundama
    int tempsecs = secs % 60;
    int tempmins = (secs / 60) % 60;
    int temphours = secs / 3600;
    formattime(temphours,tempmins,tempsecs,time);
}
```

Слика 4.1.1.6.1 - Функција void formattimeseconds(long secs, char time[])

Претходне две функције се користе за рачунање времена чији је корен функција **millis()**, а она користи унутрашњи осцилатор за генерисање времена у милисекундама. Позивање ове две функције је спрегнуто са функцијом void **Core()**, чиме она позива функцију void **formattimeseconds**(long secs, char time[]), а претходна позива функцију void **formattime**(int hours, int mins, int secs, char time[]).

4.1.1.7 Void calcstrike()

У случају да је један од постојећих параметара подешавања система постављен на нулу, а то је конкретно параметар жељене страјк температуре или почетне темпаратуре у кувалу, која служи за процес укомљавања, ова функција користи једначину за прорачунавање температуре која ће

бити у кувалу да би се добила жељена температура у процесу укомљавања. Функција користи, као улазну променљиву, температуру празног мешача чиме се дефинише температура јечма и предефинисане параметре из EEPROM меморије. Предефинисани параметри су:

- Температура укомљавања
- Запремински однос воде и јечма
- Фактор температурног поремећаја система, тј. губитак температуре у процесу претакања

На слици испод је приказана једначина у оквиру `void calcstrike()` функције (слика 4.1.1.7.1).

```
void calcstrike()    // koristi se za racunanje strajk temperature
{
    int tempofgrain = tempsenzor2; // Temperatura јечма (praznog bureta), ova velicina je uzeta sa temperaturnog senzora mesaca

    strajktemp = (ukotemp + 0.4 * (ukotemp - tempofgrain) / odnosvij) + sisfaktor; // Fункција за izracunavanje strajk temperature vode
}
```

Слика 4.1.1.7.1 - Функција `void calcstrike()`

4.1.1.8 Void uss()

Ова функција се користи за добијање раздаљине од носача подсклопа кувала до површине воде коришћењем ултразвучног сензора. Функција која се повлачи из основне Arduino библиотеке за добијање сигнала са сензора овог типа је `pulseIn()`. У другом поглављу је објашњено како ради ултразвучни сензор, овде ће бити само приказан коначни код. Он је приказан на слици 4.1.1.8.1.

```
void uss()
{
    digitalWrite(trigpin, LOW); //
    delayMicroseconds(5);      // AKTIVIRANJE "ULTRASONIC" SENZORA
    digitalWrite(trigpin, HIGH); //
    delayMicroseconds(10);      //
    digitalWrite(trigpin, LOW); //

    trajanje = pulseIn(echopin, HIGH, 5000); // OBRADA SIGNALA SA "ULTRASONIC" SENZORA
    razdaljinauss = (trajanje/2) / 29.1;      //

    if (razdaljinauss <= 34 || razdaljinauss <= 0){

        // Oduzeti od poslednjeg citanja signala
        total -= citanje[index];
        // Ocitanje sa senzora
        citanje[index] = razdaljinauss;
        // Dodato ocitanje totalu
        total += citanje[index];
        // Napredovanje index promenljive do sledece pozicije u nizu
        index = index + 1;

        srvrednostuss = total / brcitanja; // Racunanje srednje vrednosti
    }
    else {

        razdaljinauss = 00; // prijavljivanje greska
        srvrednostuss = 00; // prijavljivanje greska
    }

    if (srvrednostuss != srvrednostussp) // smanjenje frekvencije updateovanja na displej
    {
        srvrednostussp = srvrednostuss;
    }

    if (index >= brcitanja) // vracanje na pocetak racunanja srednje vrednosti razdaljine
    {
        index = 0;
    }
}
```

Слика 4.1.1.8.1 - Функција `void uss()`

Сигнал код оваквих врста сензора може да буде у неким случајевима стохастички, уведено је рачунање средње вредности на основу последњих десет улазних величина чиме се сигнал довео у неки функционалан облик. Инерција промене је повећана, чиме се постигла континуалност и стабилност вредности очитаних са овог сензора.

4.1.2 Основни објекти система

У оквиру овог потпоглавља неће бити описан основни објекат `void cip()`, јер је он само дефинисан као могући избор, али због ограничења у меморијском простору овог микроконтролера је изостављен. Наравно, програмски код за овај део комплетног алгорита постоји, тај део кода је намењен за дезинфекцију и чишћење система након процеса кувања пива (енг. **Cleaning In Place**).

4.1.2.1 Void manualmode()

Овај објекат је један од избора који је на располагању кориснику овог система. Он представља мануелну или ручну контролу над актуаторима у оквиру система. Уз то, он има надзор над температурама у оба суда, укупно протекло време од стартовања система, и количини течности у суду за кување. Овај мод или објекат се користи у случају избијања проблема код рада компоненти, чиме се процес прављења пива наставља, а сигурност преосталих компоненти не угрожава. Изглед кода у оквиру овог основног објекта неће бити приказан због његове величине, уместо тога биће приказан изглед LCD екрана приликом рада система у овом моду (слика 4.1.2.1.1).



Слика 4.1.2.1.1 - Мануелни мод

4.1.2.2 Void setupmenu()

Основни објекат `setupmenu()` приказује начин на који је параметризован овај систем. Представља одвојену целину која је задужена за снимање одређених величина у EEPROM меморију, који ће касније бити коришћени у аутоматском моду. Ове величине су параметри система. Параметри су усвојени у односу на изглед система укључујући захтеве технологије производње пива. Постоји 21 параметар од кога зависи понашање овог система у аутоматском моду, а то су:

- | | |
|-----------------------|---------------------------|
| 1. Тест мод | 6. СИС фактор |
| 2. Одбројавање | 7. Однос воде и јечма |
| 3. Провера нивоа | 8. Температура покапавања |
| 4. Страјк температура | 9. Време утакања |
| 5. Температура комине | 10. Време укомљавања |

11. Време утакања ЗВП	17. Време кувања
12. Рад мешача са ЗВП	18. Додавање хмеља 1
13. Време мировања М	19. Додавање хмеља 2
14. Претакање и рецикулација	20. Додавање хмеља 3
15. Време пуњења К	21. Време хлађења
16. Температура кувања	

Сваки од ових параметара игра важну улогу у процесу производње пива.

1. Тест мод - EEPROM.read(0):

Он представља избор између мануелног прескакања корака и потпуног аутоматског мода. Овај параметар пружа прилику кориснику да провери све параметре које је унео у систем, прескачући поједине кораке да би тиме уклонио евентуалне грешке у параметрима. Када је величина овог параметра постављена на 0, тест мод је искључен, а ако је постављен на било који број који није 0, онда је укључен. Прескакање између корака се врши притиском тастера 2 на поклопцу електро ормара, када је укључен тест мод.

2. Одбројавање - EEPROM.read(1):

Мерна јединица овог параметра је време. Ако је овај параметар постављен на нулу, одбројавања неће ни бити, а ако је постављен на било који број који није нула, нпр. 1, одбројавање ће трајати тај број пута 10, чиме се добија корак од 10 минута.

3. Провера нивоа - EEPROM.read(2):

Овај параметар управља провером нивоа воде у суду преко ултразвучног сензора. Уколико је укључен, тј. постављен на 1, провера нивоа ће бити спроведена у аутоматском моду, а ако је постављен на нулу, провере нивоа неће ни бити.

Напомена: вредности морају бити или 1 или 0, иначе се програм неће покренути!

4. Страјк температура - EEPROM.read(3)

Овај параметар дефинише температуру воде која ће се мешати са јечмом у процесу укомљавања. У случају да је овај параметар постављен на 0, израчунавање ове величине ће се покренути у аутоматском моду преко функцијског објекта calcstrike(). Мерна јединица ове величине, тј. параметра је целзијус. Пример: уколико се унесе број 80, то значи да ће температура воде бити 80 степени целзијуса.

5. Температура комине - EEPROM.read(4)

Овај параметар дефинише температуру комине, тј. температуру воде унутар мешача за време процеса укомљавања. Овај параметар се користи и за израчунавање страјк температуре у оквиру функцијског објекта calcstrike(). Мерна јединица је целзијус.

6. СИС фактор - EEPROM.read(5)

Овај параметар представља фактор поремећаја система. У овом случају се односи на температурни губитак у процесу претакања из кувала у суд за укомљавање. Овај параметар фигурише у формули за израчунавање страјк температуре и представља величину која је добијена емпиријским путем. Фактор је уведен да би систем био прилагодљивији променама које могу да се десе (мењање цеви, судова, брзине протока, итд.). Мерна јединица је целзијус.

7. Однос воде и јечма - EEPROM.read(6)

Параметар представља однос запремине воде и јечма у суду за укомљавање. Величина овог параметра је условљена технологијом производње пива.

8. Температура покапавања - EEPROM.read(7)

Овај параметар представља захтевану температуру воде која долази из кувала у процесу покапавања. Покапавање је финални део процеса укомљавања и служи за максималну екстракцију шећера и беланчевина из сладовине. Температура течности за процес покапавања је до 20 процената виша од температуре у процесу укомљавања.

9. Време утакања - EEPROM.read(8)

Овај параметар представља време утакања течности у мешаону из кувала при чему је температура течности једнака страјк температури. Ово је прво претакање у процесу прављења пива. Мерна јединица овог параметра је минут.

10. Време укомљавања - EEPROM.read(9)

Параметар представља време трајања процеса укомљавања. Мерна јединица овог параметра је минут.

11. Време утакања ЗВП - EEPROM.read(10)

Овај параметар дефинише време трајања процеса утакања воде загрејане на температуру покапавања. Процес се дешава након примарног процеса укомљавања. Мерна јединица овог параметра је минут.

12. Рад мешача са ЗВП - EEPROM.read(11)

Параметар представља временски интервал рада мешача након процеса додавања загрејане воде покапавања. Мерна јединица овог параметра је минут.

13. Време мировања М - EEPROM.read(12)

Ово је временски интервал када мешач неће радити тј. паузираће пре почетка процеса претакања и рецикулације. Представља процес одлежавања комине. Мерна јединица је минут.

14. Претакање и рецикулација - EEPROM.read(13)

Временски период трајања процеса претакања и рецикулације након што се троп (скупљени јечам) наталожио на дуплом дну. Овај процес служи за максималну екстракцију шећера и беланчевина из сладовине. Мерна јединица овог параметра је минут.

15. Време пуњења К - EEPROM.read(14)

Период претакања воде из мешача у кувало. Ово је емпиријска величина. Мерна јединица овог параметра је минут.

16. Температура кувања - EEPROM.read(15)

Захтевана температура кувања. Мерна јединица је целзијус.

Напомена: Параметар никако не постављати на теоријску температуру кључања. Увек на који степен испод.

17. Време кувања - EEPROM.read(16)

Временски период кувања сладовине (енг. *wort*). Мерна јединица је минут.

18. Додавање хмеља 1 - EEPROM.read(17)

Тачно дефинисано време додавање прве дозе хмеља. Обично је на самом почетку и износи 2/3 укупне запремине хмеља који ће бити коришћен у процесу. Мерна јединица је минут.

19. Додавање хмеља 2 - EEPROM.read(18)

Тачно дефинисано време додавање друге дозе хмеља. Обично је петнаест минута пред крај процеса кувања и износи половину укупне запремине последње две дозе. Мерна јединица је

минут.

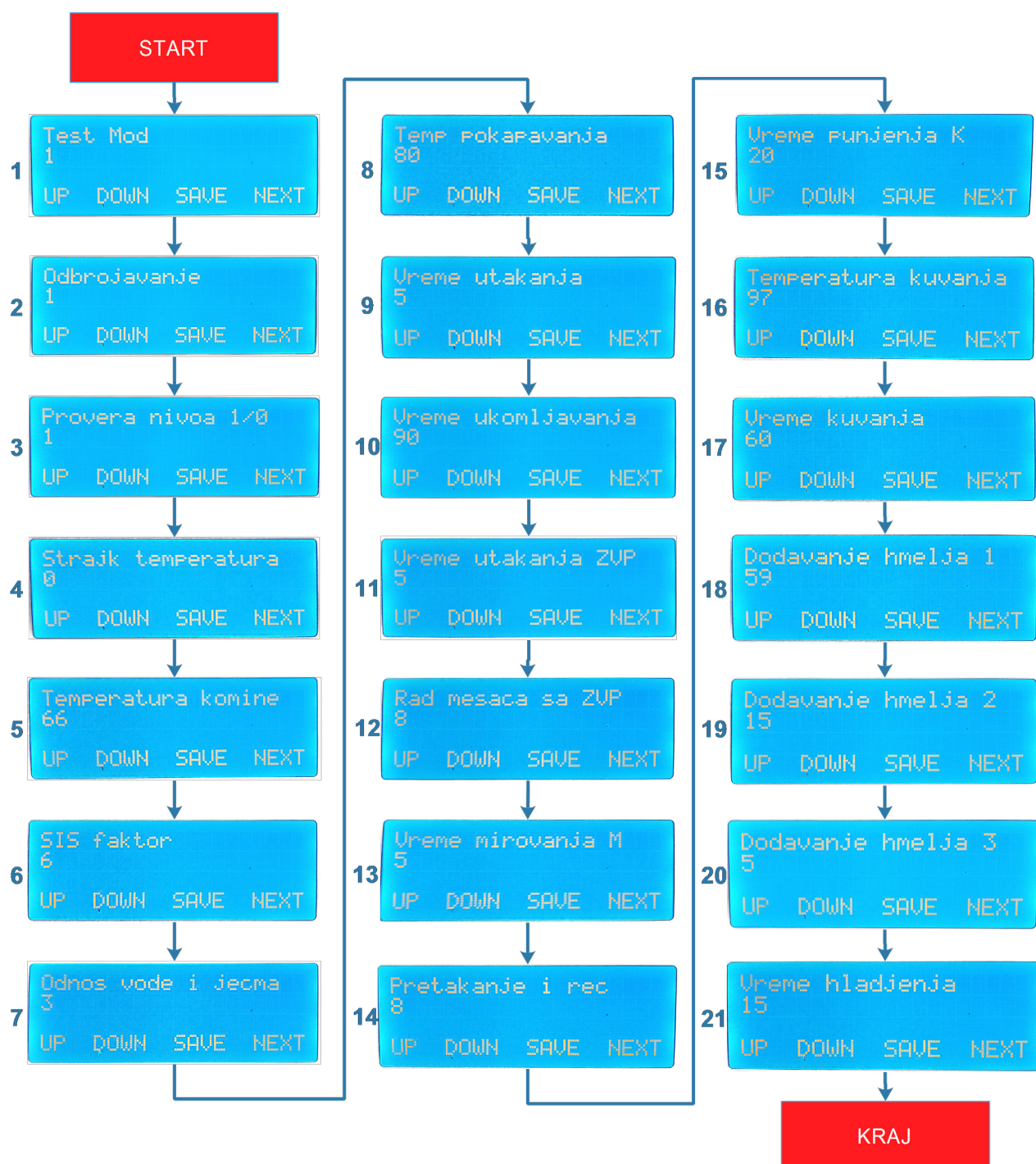
20. Додавање хмеља 3 - EEPROM.read(19)

Тачно дефинисано време додавање треће дозе хмеља. Обично је пет минута пред крај процеса кувања. Мерна јединица је минут.

21. Време хлађења - EEPROM.read(20)

Представља последњу етапу у процесу прављења пива и служи за нагло хлађење пива. Мерна јединица је минут.

Изглед екрана приликом избора у оквиру овог функцијског објекта је приказан сликовним алгоритмом на шеми испод. На сликама се види на који начин функционише процес подешавања.



Слика 4.1.2.2.1 - Сливовни алгоритам функцијског објекта setupmenu()

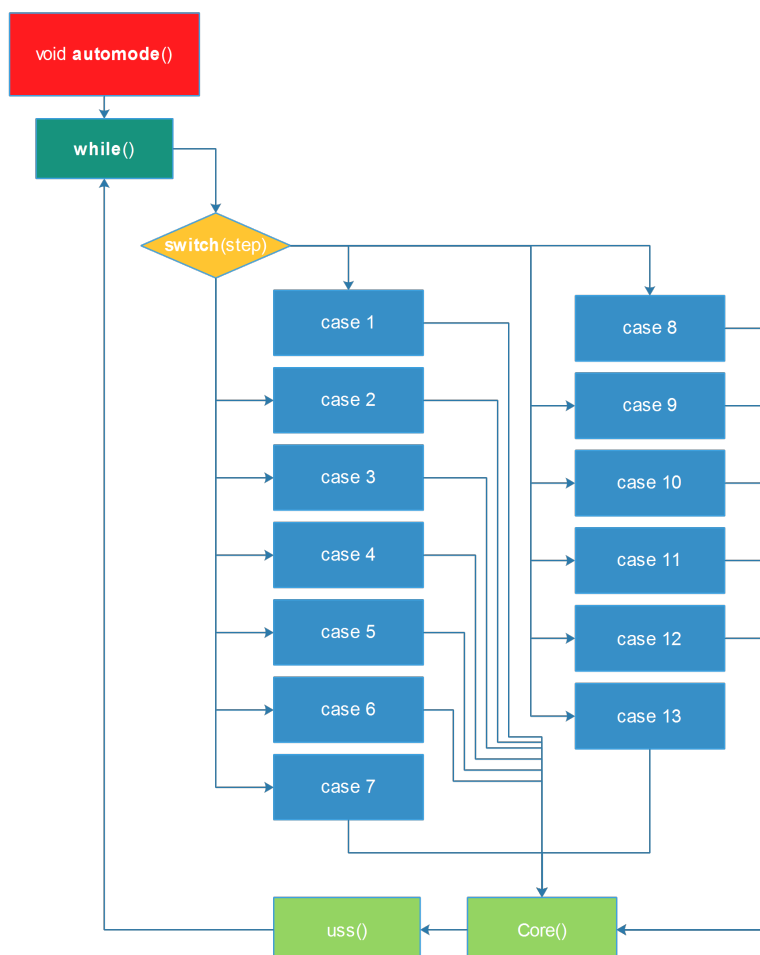
Прва два тастера, почевши са леве стране електро ормара ка десној, се односе на инкрементално повећање, тј. умањење величине параметра приказане изнад њих. Следећи тастер у низу служи за снимање дефинисане величине у EEPROM меморију и последњи тастер служи за прелажење на следећи параметар у низу.

4.2 Повратне спреге - аутоматски мод

4.2.1 Void automode()

Аутоматски мод представља најкомпликованији део овог дипломског рада, он је суштина комплетног процеса. Било је потребно много времена да се он доведе у функционално стање, иако је већина процеса временски зависна. Процес се у овом моду одвија аутоматски, из тог разлога је приказан у поглављу повратних спрега.

Да би аутоматски мод био могућ, а обзиром да се процес прављења пива састоји из етапа, моделирању кода у ове намене се приступило преко петљи за задржавање, тј. **while()** петљи. Уз ову петљу су коришћене функције **switch()**, **step** и **case**, које обезбеђују издвајање делова кода у комплетној **while()** петљи, чиме се постиже константан пролаз петље кроз функцијске објекте система у свакој етапи, уз само једно дефинисање у коду. На слици 4.1.2.2.1 је приказан алгоритам рада овог основног објекта **automode()** у програму. Прелазак из једне у другу петљу, поред тога што је временски условљен, такође је условљен захтеваним условима који требају да се испуне у одређеној етапи. Те услове представљају захтевана температура, ниво течности у суду и позиција одређених компоненти у систему (позиција окретне плоче додавача хмеља, итд.).

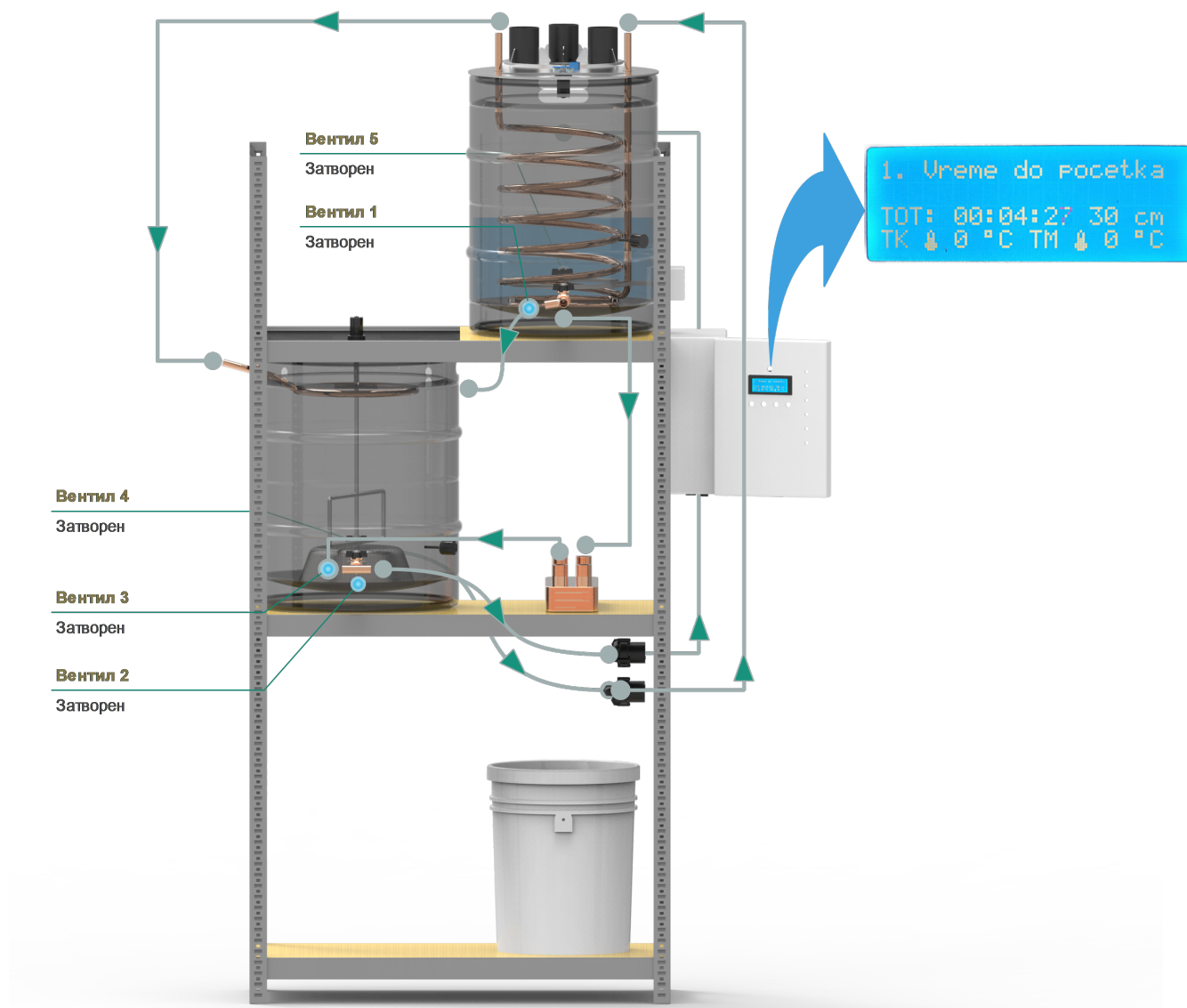


Слика 4.2.1.1 - Алгоритам рада **automode()** функцијског блока

Свака појединачна етапа, или случај (енг. case), има своју функцију у процесу прављења пива.

4.2.1.1 Case 1 - Одбројавање

Прва етапа представља процес одбројавања до почетка процеса прављења пива. Она узима вредност параметра из основног објекта `setupmenu()`, помоћу којег се дефинише период одбројавања. Ова етапа се може прескочити уношењем вредности 0 за параметар "тајмер" у оквиру подешавања, тј. основног објекта `setupmenu()`. На слици 4.2.1.1.1 је приказан изглед система и LCD екрана у етапи 1.



Слика 4.2.1.1.1 - Изглед стања система у првој етапи

Ова етапа се користи када желимо да имамо надзор над запремином воде у суду, као и температуре воде током процеса пуњења система.

На слици су показани путеви кретања флуида кроз црева као и смер кретања флуида, иако у том тренутку флуид не циркулише кроз њих. Ово је урађено да би читалац могао лакше да прати понашање система кроз сваку следећу етапу.

У оквиру слике 4.2.1.1.1 је такође приказано стање вентила. Програмски код ове етапе је приказан на слици 4.2.1.1.2.

```

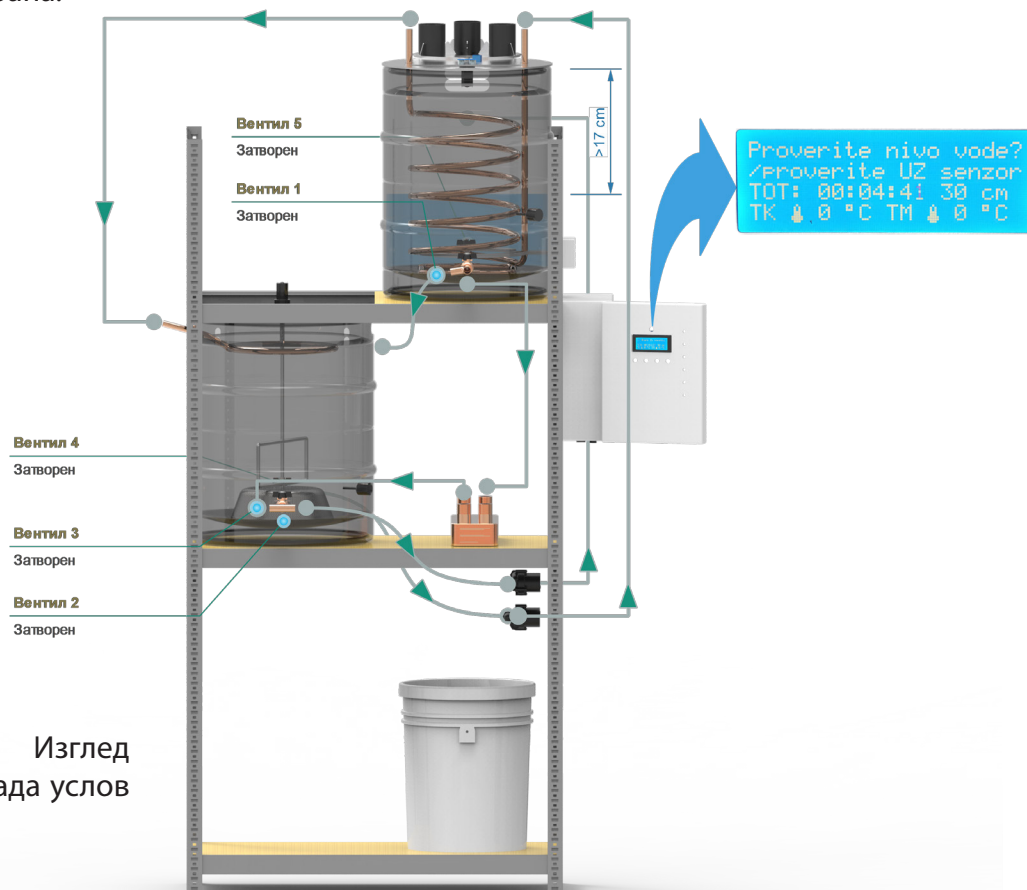
case 1: // Korak 1 - Odbrojavanje do pocetka procesa pravljenja piva
  if (!msgdisplayed1)
  {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("1. Vreme do pocetka");
    msgdisplayed1=1;
  }
  if (!timercalculated) {
    startstep = millis(); // pokreni sat koraka, citaj milisekunde pocevsi od ovog trenutka
    steptime = (unsigned long)tajmer*60*1000; // uzimanje vremena iz EEPROM - a za promenljivu "tajmer" [minuti] i racunanje vremena do pocetka
    stopstep = startstep + steptime; //
    timercalculated = 1; // ne radi ponovo u ovom koraku
  }
  if (millis()>=stopstep) { // periodu odbrojavanja dosao kraj, prelazak na 2. korak
    step++;
    msgdisplayed1=0;
    timercalculated=0; // sledeci korak jos nije izracunat
  }
  if (!(cyclecount % 1) && (tempsenzor1>0) && (tempsenzor2>0)) { // ispisati tekst svaki hiljaditi put kada se prodje kroz ovu petlju //
    steplapsed = (long)(millis()-startstep); // racuna koliko je milisekundi proslo od pocetka koraka
    steplapsedsecs = (int)(steplapsed/1000); // konvertovanje u sekunde (ne bas precizno)
    long tajmerremaining = (stopstep - steplapsed) / 1000; // u sekundama
    formattimeseconds(tajmerremaining,steplapsedtime);
    lcd.setCursor(3, 1);
    lcd.print(steplapsedtime);
    cyclecount++;
  }
}
break;

```

Слика 4.2.1.1.2 - Програмски код етапе 1

4.2.1.2 Case 2 - Провера нивоа воде

Друга етапа провера ниво воде у суду на основу задатих параметара. У случају да је променљива изнад захтеване величине дефинисане параметром, програм пријављује грешку и чека на испуњење услова за почетак процеса прављења пива. Ово је веома важан део процеса прављења пива који обезбеђује почетну запремину течности у суду за кување која служи за комплетан процес прављења пива. На слици 4.2.1.2.1 је приказано стање система у етапи 2 заједно са изгледом LCD екрана.



Слика 4.2.1.2.1 - Изглед система и екрана када услов није испуњен

Ова етапа се такође може заобићи постављањем параметра "checkfloat" у 0. Основна намена овог дела кода је сигурност да не дође до паљења грејача када је суд празан и да ниво течности не буде мањи од захтеване.

На слици 4.2.1.2.1 је приказан изглед екрана услед мањка течности у суду.

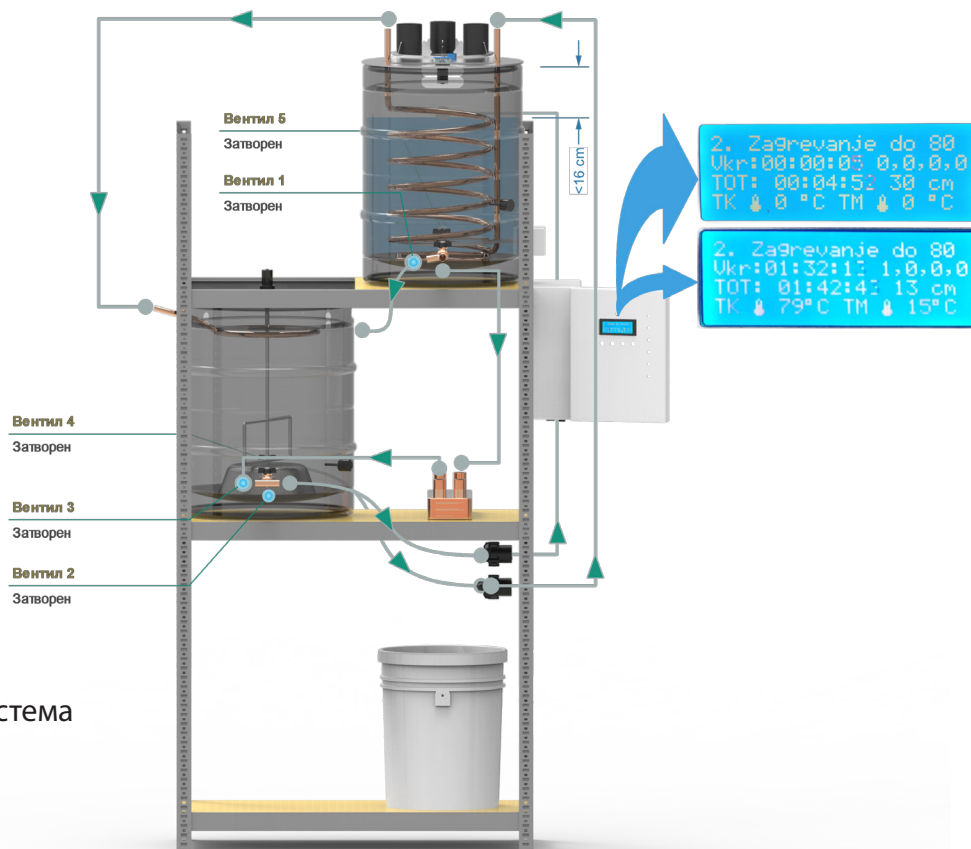
Програмски код ове етапе је приказан на слици 4.2.1.2.2.

```
case 2: // Korak 2 - Proveravanje nivoa vode u sudu (USS)
if (checkfloat == 1)
{
if (!msgdisplayed1 && (svrednostussp >= 17))
{
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Proverite nivo vode?");
lcd.setCursor(0, 1);
lcd.print("/proverite UZ senzor");
msgdisplayed1=1;
}
if (svrednostussp < 24)
{
step++; // predji na korak 3
msgdisplayed1=0;
}
else
{
lcd.setCursor(0, 0);
lcd.print("Proverite nivo vode?");
lcd.setCursor(0, 1);
lcd.print("/proverite UZ senzor");
}
}
break;
```

Слика 4.2.1.2.2 - Програмски код етапе 2

4.2.1.3 Case 3 - Загревање воде

У оквиру овог корака се вода загрева на претходно дефинисану температуру или се израчунава на основу жељене температуре укомљавања. Ако је параметар "strajktemp" једнак нули, позива се функција calcstrike() преко које се израчунава потребна температура у кувалу за процес укомљавања у мешачу.



Слика 4.2.1.3.1 - Изглед система и екрана у току етапе 3

Када се испуне сви услови, ниво воде буде у реду, добије се вредност страјк температуре која мора бити достигнута пали се грејач и време корака почиње. На слици 4.2.1.3.1 је приказан изглед система у току етапе 3.

У случају нестабилног рада ултразвучног сензора, параметар "checkfloat" се искључује, тј. поставља на 0, да не би дошло до гашења грејача у току грејања воде у овом кораку. На слици 4.2.1.3.2 је приказан изглед кода за подизање температуре на захтевану, тј. програмски код корака 3.

```
case 3: // Korak 3 - Zagrevanje suda do strajk temperature
  if (strajktemp == 0)
  {
    calcstrike();
  }
  if (!msgdisplayed1)
  {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("2. Zagrevanje do ");
    lcd.setCursor(17, 0);
    lcd.print(strajktemp); //prikazivanje strajk temperature
    delay(50);
    lcd.setCursor(13, 1);
    lcd.print("0,0,0,0");

    delay(50);
    msgdisplayed1=1;
    startstep=millis();
  }
  if (checkfloat == 1)
  {
    if (srvrednostussp < 16)
    {
      digitalWrite(pingrejac, LOW); // paljenje ssr preko pingrejac pina
      lcd.setCursor(13, 1);
      lcd.print("1"); // paljenje grejaca 1 - > upaljen
    }
    else
    {
      digitalWrite(pingrejac, HIGH); // gasenje ssr preko pingrejac pina
      lcd.setCursor(13, 1);
      lcd.print("0"); // gasenje grejaca 0 - > ugasen
    }
  }

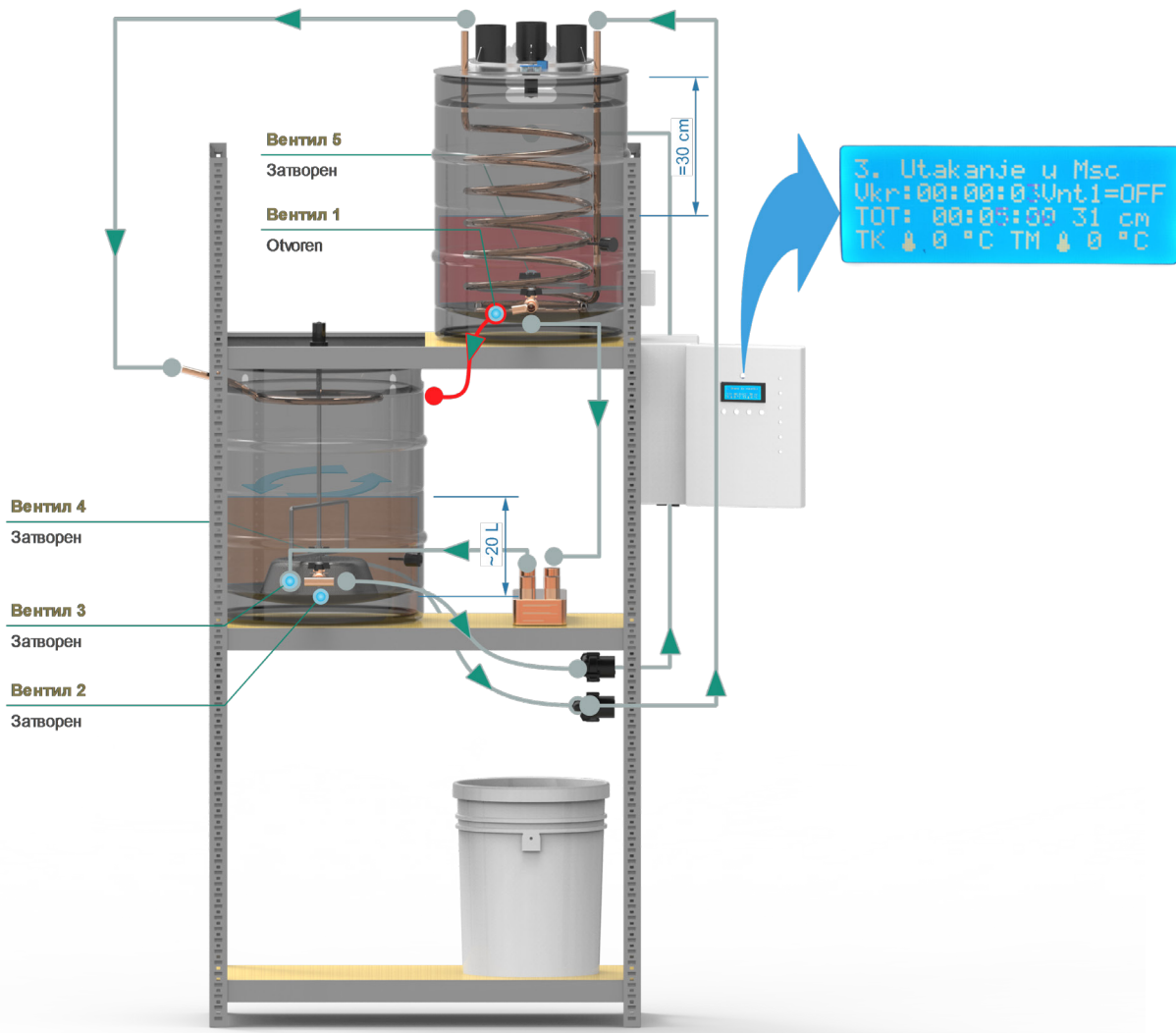
  if (tempsenzor1 >= strajktemp) {
    digitalWrite(pingrejac, HIGH); // Gasenje grejaca
    step++;
    msgdisplayed1=0;
    timercalculated=0; // vreme sledeceg koraka jos nije izracunato
  }
  if (!(cyclecount % 1)) { // ispisati tekst svaki hiljaditi put kada se prodje kroz ovu petlju
    stepelapsed = (long)(millis()-startstep); // racuna koliko je milisekundi proslo od pocetka koraka
    stepelapsedsecs = (int)(stepelapsed/1000); // konvertovanje u sekunde (ne bas precizno)
    formattimeseconds(stepelapsedsecs, stepelapsedtime);
    lcd.setCursor(0, 1);
    lcd.print("Vkr:"); // vreme trajanja koraka u procesu
    lcd.setCursor(4, 1);
    lcd.print(stepelapsedtime); // vreme trajanja koraka u procesu
    cyclecount++;
  }
  break;
```

Слика 4.2.1.3.2 - Програмски код етапе 3

Цифре одвојене запетом у другом реду LCD екрана представљају стање излазних пинова. Респективно оне представљају стање грејача, пумпе за рецикулацију, пумпе за претакање и мотор мешача. Служе као додатна контрола излазних пинова. Приказана су два стања улазно излазних пинова на слици 4.2.1.3.1, када услов са ултразвучног сензора јесте испуњен и када услов није испуњен.

4.2.1.4 Case 4 - Утакање у мешач

У оквиру овог корака се течност пребацује из суда за кување у мешаону при чему се врши сједињавање слада са загрејаном водом. Такође, мешач креће са радом након одређеног времена. Овај корак је временски дефинисан на основу емпиријских података. Куглични вентил за претакање течности из суда у суд се отвара и поставља под углом од 45 степени у односу на почетни положај, чиме се регулише проток, тј. брзина претакања из кувала у мешач. На слици 4.2.1.4.1 је приказано стање система у етапи 4.



Слика 4.2.1.4.1 - Изглед и стање система током етапе 4

Да би количина воде која треба да се преточи у мешаону била што прецизнија, корисник би требао да прати ниво воде преко ултразвучног сензора. Када се ниво воде у кувалу са 14 центиметара спусти на 30 центиметара, треба затворити вентил, иако корак није завршен (тешко је поставити вентил прецизних 45 степени), тиме је преточена количина воде приближно 20 литара. Ово такође представља емпиријски податак, добијен процесом калибрације ултразвучног сензора.

На слици 4.2.1.4.1 се види изглед екрана, одмах након почетка етапе 4. Вентил 1 је затворен. Непосредно након отварања вентила 1 треба притиснути тастер 1, чиме почиње одбројавање у овом кораку, тј. биће прорачунато време трајања корака „Vkr“. У супротном, време трајања овог корака биће бесконачно.

На слици 4.2.1.4.2 се налази изглед кода који је коришћен за контролу овог процеса. Главни параметри који фигуришу у овом кораку су везани за времена и звучну сигнализацију.

```

case 4: // Korak 4 - Pretakanje iz kuvala u mesac
if (!msgdisplayed1)
{
    striketime = stepelapsed; // ovo ce sacuvati vreme trajanja zagrevanja do "strajktmp"
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("3. Utakanje u Msc");
    delay(50);
    lcd.setCursor(12, 1);
    lcd.print("Vnt1=OFF"); // obavestjenje za otvaranje ventila 1, kada se to uradi pritiskom na taster 1 se potvrđuje
    delay(50);
    msgdisplayed1=1;
    startstep = millis(); // pokreni sat koraka, citaj milisekunde pocevsi od ovog trenutka
}

if (!timercalculated) {
    beep(PinZvucnik,2093,500); // Emituje ton C u trajanju od pola sekunde sto oznacava kraj pretakanja, i obavestjenje za otvaranje ventila 1
    if(taster1.uniquePress()){
        lcd.setCursor(12, 1);
        lcd.print("Vnt1= ON");
        steptime = (unsigned long)vremeutakanja*60*1000; // racuna koliko je vremena potrebno za ovaj korak uzimajuci parametar "Vreme utakanja"
        stopstep = startstep + steptime; // racuna kraj ovog koraka u odnosu na ukupno proteklo vreme i uzimajuci u obzir parametar "Vreme utakanja"
        startmashmotor = startstep + (steptime/2);
        timercalculated = 1; // ne radi ponovo u ovom koraku
    }
}

if (millis()>=startmashmotor && !startmashmotorreached && timercalculated == 1)
{
    digitalWrite(pinmesac, LOW); // ukljuciti mesac
    startmashmotorreached=true;
}

if (millis()>=stopstep && timercalculated == 1) { // vreme je isteklo za ovaj korak
    beep(PinZvucnik,2093,500); // Emituje ton C u trajanju od pola sekunde sto oznacava kraj pretakanja, i obavestjenje za zatvaranje ventila 1
    lcd.setCursor(12, 1);
    lcd.print("Vnt1=OFF");
    delay(2000);
    step++;
    msgdisplayed1=0;
    timercalculated=0; // vreme sledeceg koraka jos nije izracunato
}

if (!(cyclecount % 1)) { //ispisati tekst svaki hiljaditi put kada se prodje kroz ovu petlju
    stepelapsed = (long)(millis()-startstep); // racuna koliko je milisekundi proslo od pocetka koraka
    stepelapsedsecs = (int)(stepelapsed/1000); // konvertovanje u sekunde (ne bas precizno)
    formattimesecs(stepelapsedsecs,stepelapsedtime);
    formattime(0,vremeutakanja,0,steptotaltime);
    lcd.setCursor(0, 1);
    lcd.print("Vkr:"); // vreme trajanja koraka u procesu
    lcd.setCursor(4, 1);
    lcd.print(stepelapsedtime); // vreme trajanja koraka u procesu
    cyclecount++;
}
break;

```

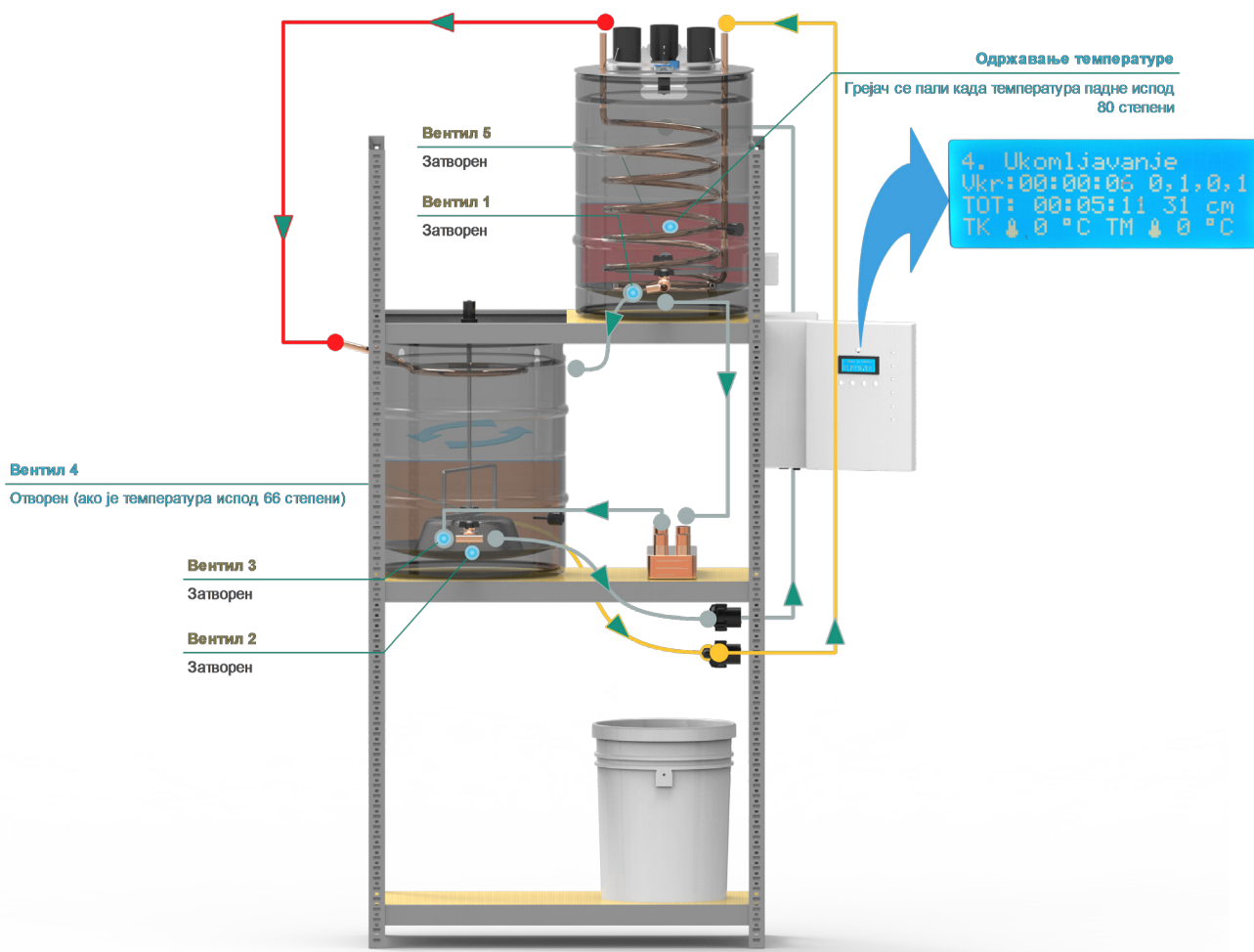
Слика 4.2.1.4.2 - Програмски код етапе 4

4.2.1.5 Case 5 - Укомљавање

Ова етапа је једна од круцијалних етапа у процесу прављења пива. У оквиру ове етапе се врши двострука регулација температуре, тј. регулација температуре посредно у мешаони и директно у кувалу.

Процес укомљавања је процес који траје у просеку 90 минута и за све то време је потребно одржавати температуру комине на 66 степени целзијуса. Ова етапа захтева константно присуство корисника. На слици 4.2.1.5.1 је приказано стање система у току етапе укомљавања.

Као што је у претходним поглављима речено, коришћене су центрифугалне пумпе за процес претакања и рецикулације, тако да је било потребна константна регулација вентила приликом паљења и гашења пумпи од стране оператера, тј. корисника. Процес регулације температуре у суду за кување се врши према условним принципима (енг. bang-bang). Када температура у кувалу падне испод 79 степени, пали се грејач, када температура пређе 81 степен, грејач се гаси. Наведене температурне су добијене из једначина "темпокапавања -1" и "темпокапавања +1". Ова температура тачности се задржава због процеса покапавања и посредне регулације температуре у мешаони.



Слика 4.2.1.5.1 - Регулација температуре у току процеса укомљавања

Када температура у мешаони падне испод 65 степени пали се пумпа за рецикулацију и емитује се звучни сигнал за отварање вентила 4. Пумпа ће да ради све док температура у мешаони не достигне вредност од 67 степени.

Овај систем има неколико сигурносних детаља у коду:

1. Када се деси да се створи ваздушни чеп у систему за рецикулацију, притиском на тастер 4 пумпа ће престати са радом, а код ће ући у while() петљу која ће чекати пуштање тастера 4, када ће пумпа поново почети са радом.
2. У случају пада нивоа воде (ултразвучни сензор показује величину изнад 30 cm или величину једнаку нули - грешка), грејач се неће упалити да не би дошло до прегоревања грејача.

Изглед кода који је имплементиран у ову етапу је приказан на слици 4.2.1.5.2. У коду су издвојене целине које су коментарисане и објашњена је њихова функција у процесу.

Проблеми који се јављају током овог процеса су појава ваздушних чепова услед велике температуре и карбонификације течности у процесу укомљавања. Рад центрифугалних пумпи поспешује сепарацију кисеоника из комине која се претаче, тако да се ваздушни чеп створи иако је систем потпуно затворен. Такође, проблем високо постављеног температурног сензора у суду за кување, доводи до наглог пада температуре, ако ниво течности падне испод висине на којој је постављен температурни сензор, што доводи до константног рада грејача и кључања течности унутар кувала.


```

case 5: // Korak 5 - Mesanje i grejanje vode za pokapavanje
if (!msgdisplayed1)
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("4. Ukomljavaње"); // vreme trajanja koraka u procesu
    delay(50);
    lcd.setCursor(13, 1);
    lcd.print("0,0,0,0"); // POČETNE VREDNOSTI STANJA PINOVA - isključeni = 0
    delay(50);
    msgdisplayed1=1;
}
digitalWrite(pinmesac, LOW); // ukljuci motor mesaca
delay(100);
lcd.setCursor(19, 1);
lcd.print("1");
if ((tempsenzor1>pokaptemp+1) && (tempsenzor1>0))
{
    digitalWrite(pingrejac, HIGH); // UGASI GREJAC
    lcd.setCursor(13, 1);
    lcd.print("0");
}
if (checkfloat == 1)
{
    if (srvrednostussp < 30) // ako je nivo vode u redu, nastavi (ne ispod visine grejaca i senzora temperature)
    {
        if ((tempsenzor1<=pokaptemp-1) && (tempsenzor1>0))
        {
            digitalWrite(pingrejac, LOW); // UPALI GREJAC
            lcd.setCursor(13, 1);
            lcd.print("1");
        }
    }
}
else
{
    if ((tempsenzor1>=pokaptemp-1) && (tempsenzor1>0))
    {
        digitalWrite(pingrejac, LOW); // UPALI GREJAC
        lcd.setCursor(13, 1);
        lcd.print("0");
    }
}
if (!timercalculated) {
    startstep = millis(); // pokreni sat koraka, citaj milisekunde pocevsi od ovog trenutka
    steptime = (unsigned long)vrememesanjal*60*1000; // racuna koliko je vremena potrebno za ovaj korak uzimajući parametar "Vreme mesanja 1"
    stopstep = startstep + steptime; // racuna kraj ovog koraka u odnosu na ukupno proteklo vreme i uzimajući u obzir parametar "Vreme umesanja 1"
    timercalculated = 1; // ne radi ponovo u ovom koraku
}

// if petlja za održavanje temperature vode u mesacu na 66 stepeni pomocu H.E.R.M sistema

if (tempsenzor2 < 65 && msgdisplayed2 == 0){
    beep(PinZvucnik,2093,500); // Emituje ton C (visok) u trajanju od pola sekunde / obavestava da treba otvoriti ventil 4
    digitalWrite(pinpumparec, LOW); // Upaliti pumpu za recirkulaciju //
    lcd.setCursor(15, 1);
    lcd.print("1");
    msgdisplayed2 = 1;
}

while(taster4.isPressed() == true){
    digitalWrite(pinpumparec, HIGH); // Ugasiti pumpu za recirkulaciju //
    lcd.setCursor(15, 1); //
    lcd.print("0"); //
    msgdisplayed2 = 0; //
} //
//
if (tempsenzor2 > 67 && msgdisplayed2 == 1){
    beep(PinZvucnik,523,500); // Emituje ton C (dubok) u trajanju od pola sekunde/ obavestava da treba zatvoriti ventil 4
    digitalWrite(pinpumparec, HIGH); // Ugasiti pumpu za recirkulaciju
    delay(100);
    lcd.setCursor(15, 1);
    lcd.print("0");
    msgdisplayed2 = 0;
}

if (millis()>=stopstep) { // vreme koraka je završeno
    digitalWrite(pingrejac, HIGH); // Grejac ugasiti
    digitalWrite(pinmesac, HIGH); // ugasiti motor mesaca
    digitalWrite(pinpumparec, HIGH); // ugasiti pumpu za recirkulaciju
    mashtempend = tempsenzor2; // uzeti temperaturu na kraju ukomljavaња za kasnije prikazivanje u izvestaju
    step++;
    msgdisplayed1=0;
    timercalculated=0; // tajmer sledeceg koraka jos nije proracunat
}
if (!(cyclecount % 1)) { // ispisati tekst svaki hiljaditi put kada se prodje kroz ovu petlju
    stepelapsed = (long)(millis()-startstep); // racuna koliko je milisekundi proslo od pocetka koraka
    stepelapsedsecs = (int)(stepelapsed/1000); // konvertovanje u sekunde (ne bas precizno)
    formattimesecs(stepelapsedsecs,stepelapsedtime);
    formattime(0,vrememesanjal,0,steptotaltime);
    lcd.setCursor(0, 1);
    lcd.print("Vkr:"); // vreme trajanja koraka u procesu
    lcd.setCursor(4, 1);
    lcd.print(stepelapsedtime); // vreme trajanja koraka u procesu
    cyclecount++;
}
break;

```

Паљење мешача

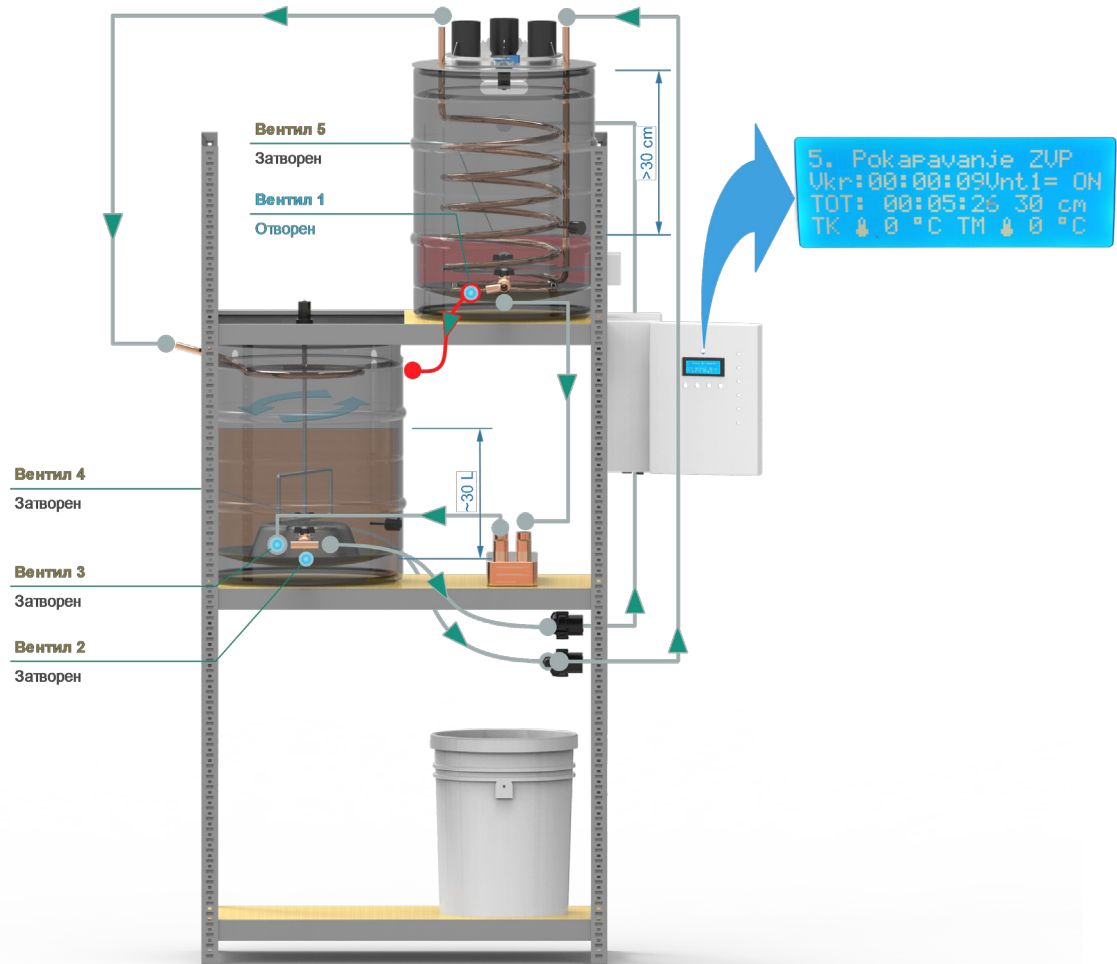
Регулација температуре у кувалу

Регулација температуре
у мешаони

Слика 4.2.1.4.2 - Програмски код етапе 5

4.2.1.6 Case 6 - Покапавање

Процес покапавања служи за максималну екстракцију беланчевина и шећера из комине. Повећањем температуре комине, се постиже мања густина и повећава порозност комине. Овај процес је сличан процесу утакања, с тим што је запремина која се претаче знатно мања од запремине код првог утакања и температура воде износи неких 80 степени. Износи неких 5 до 8 литара.



Слика 4.2.1.6.1 - Изглед система и стање LCD екрана у етапи 6

Овај процес је временски зависан и за разлику од процеса утакања (етапа 4), не може се пратити преко ултразвучног сензора, због акустике кувала, где се након 32 центиметра добијају нелогични и непрецизни резултати са сензора. Тако да се оператер мора ослонити на искуство и процену.

На слици 4.2.1.6.2 је приказан програмски код овог дела процеса прављења пива.

```
case 6: // Korak 6 - Pokapavanje i zagrevanje komine sa vodom za pokapavanje
if (!msgdisplayed1)
{
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("5. Pokapavanje ZUP"); // ime koraka
  delay(50);
  lcd.setCursor(12, 1);
  lcd.print("Vnt1=OFF"); // obavestjenje za otvaranje ventila 1, kada se to uradi pritiskom na taster 1 se potvrđuje
  delay(50);
  msgdisplayed1=1;
  startstep = millis(); // pokreni sat koraka, citaj milisekunde počevši od ovog trenutka
}

digitalWrite(pinmesac, LOW); // ukljuci mesac

if (!timercalculated) { // traženje potvrde da je ventil 1 otvoren pritiskom na taster 1, da bi se vreme koraka racunalo od tog trenutka
  beep(PinZvucnik,2093,500); // Emituje ton C (visok) u trajanju od pola sekunde / obavestava da treba otvoriti ventil 1 i pritisnuti taster 1 (respektivno)
  if (taster1.uniquePress()){
    lcd.setCursor(12, 1);
    lcd.print("Vnt1= ON");
    steptime = (unsigned long)vremeutkZVP*60*1000; // racuna koliko je vremena potrebno za ovaj korak uzimajuci parametar "VremeutkZVP"
    stopstep = startstep + steptime; // racuna kraj ovog koraka u odnosu na ukupno proteklo vreme i uzimajuci u obzir parametar "VremeutkZVP"
  }
}
```

```

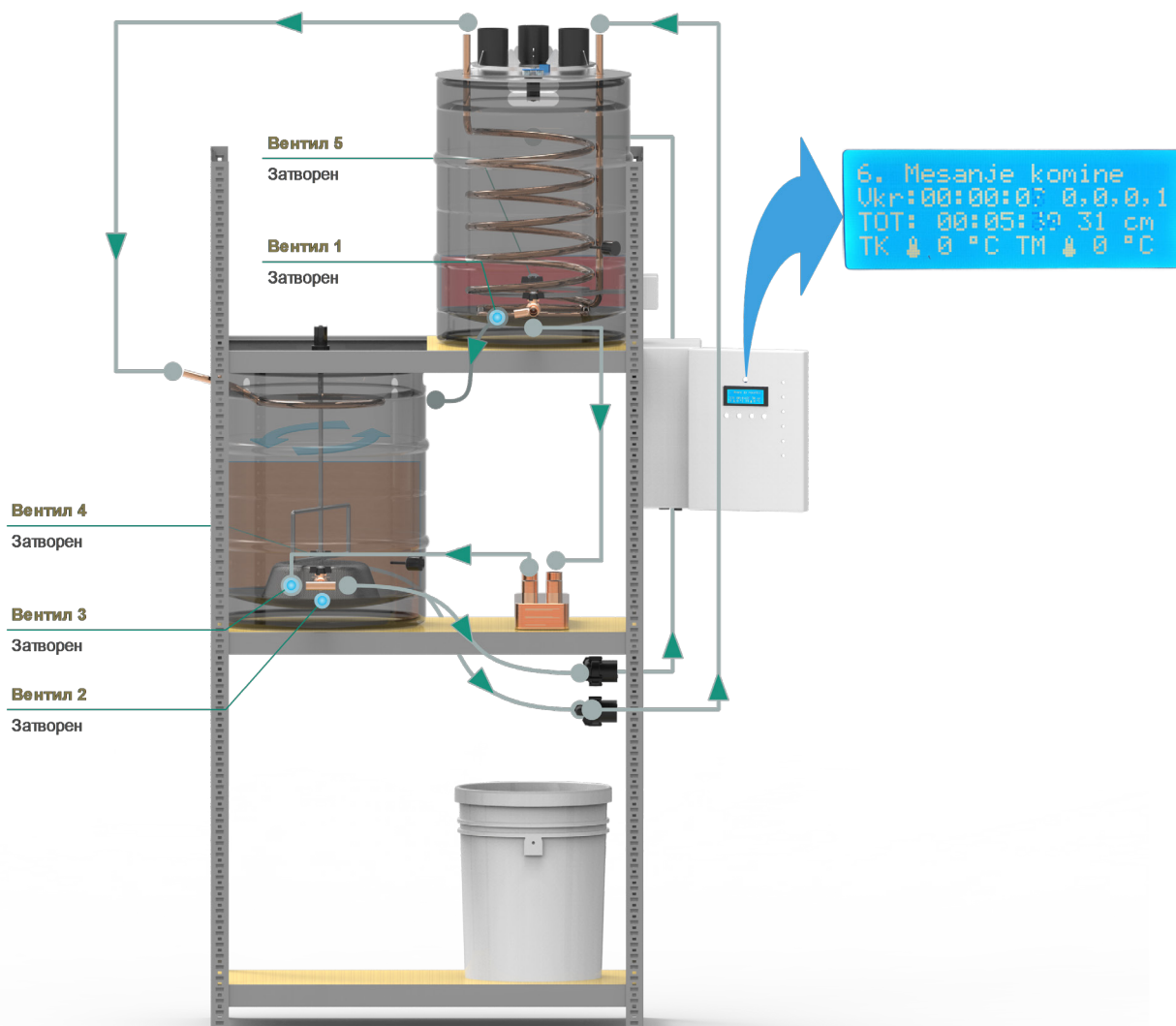
    timercalculated = 1; // ne radi ponovo u ovom koraku
  }
}
if (millis() >= stopstep && timercalculated == 1) { // vreme je zavriseno
  beep(PinZvucnik, 523, 500); // Emituje ton C (dubok) u trajanju od pola sekunde sto oznacava kraj pretakanja, i obavestjenje za zatvaranje ventila 1
  digitalWrite(pinmesac, HIGH); // ugasi mesac
  lcd.setCursor(12, 1);
  lcd.print("Vnt1=OFF");
  step++;
  msgdisplayed1=0;
  timercalculated=0; // vreme sledeceg koraka jos nije izracunato
}
if (!(cyclecount % 1)) { //ispisati tekst svaki hiljaditi put kada se prodje kroz ovu petlju
  stepelapsed = (long)(millis()-startstep); // racuna koliko je milisekundi proslo od pocetka koraka
  stepelapsedsecs = (int)(stepelapsed/1000); // konvertovanje u sekunde (ne bas precizno)
  formattimesecs(stepelapsedsecs, stepelapsedtime);
  formattime(0, vremeutkZVP, 0, steptotaltime);
  lcd.setCursor(0, 1);
  lcd.print("Vkr:"); // vreme trajanja koraka u procesu
  lcd.setCursor(4, 1);
  lcd.print(stepelapsedtime); // vreme trajanja koraka u procesu
  cyclecount++;
}
break;

```

Слика 4.2.1.6.2 - Програмски код етапе 6

4.2.1.7 Case 7 - Мешање након загревања комине

Овај процес је део који је неопходан за уједначавање температуре у комплетном суду за укомљавање након додавања загрејане воде покапавања . Приликом уједначавања температуре, долази и до отпуштања преосталих разграђених шећера из пљевика и крупних зрна брашна. На слици 4.2.1.7.1 се налази изглед система током овог процеса заједно са стањем излазних пинова.



Слика 4.2.1.7.1 - Изглед система током етапе 7

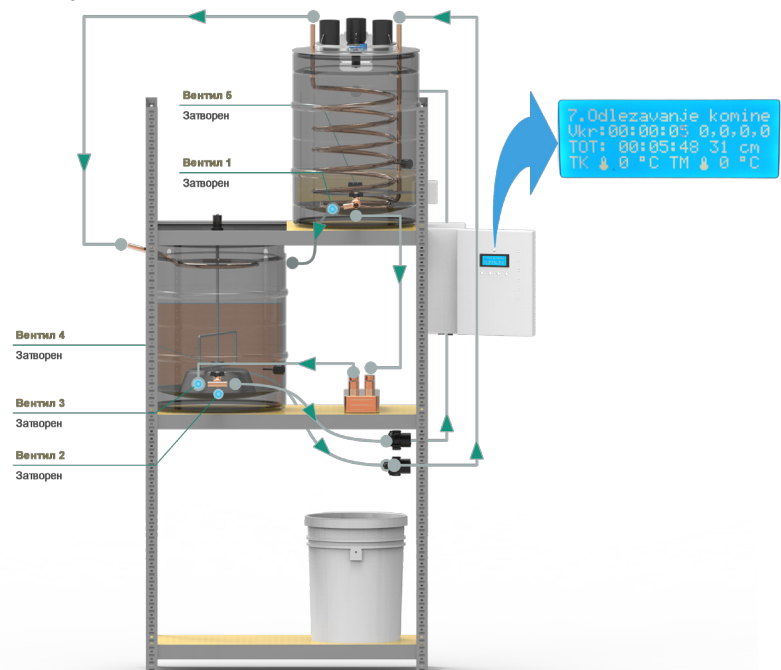
Процес је временски зависан, контролише се само стање излазног пина мешача. На слици 4.2.1.7.2 је приказан програмски код који је коришћен за овај део процеса.

```
case 7: // Korak 7 - Mesanje nakon povecanja temperature u mesaoni
if (!msgdisplayed1)
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("6. Mesanje komine"); // ime koraka
    delay(50);
    lcd.setCursor(13, 1);
    lcd.print("0,0,0,0");
    delay(50);
    msgdisplayed1=1;
}
digitalWrite(pinmesac, LOW); // upali mesac
lcd.setCursor(19, 1);
lcd.print("1"); // paljenje mesaca 1 - > upaljen
if (!timercalculated) {
    startstep = millis(); // pokreni sat koraka, citaj milisekunde pocevsi od ovog trenutka
    steptime = (unsigned long)vrememesanja2*60*1000; // racuna koliko je vremena potrebno za ovaj korak uzimajuci parametar "vrememesanja2"
    stopstep = startstep + steptime; // racuna kraj ovog koraka u odnosu na ukupno proteklo vreme i uzimajuci u obzir parametar "vrememesanja2"
    timercalculated = 1; // ne radi ponovo u ovom koraku
}
if (millis()>=stopstep) { // vreme je isteklo za ovaj korak
    digitalWrite(pinmesac, HIGH); // ugasi mesac
    step++;
    msgdisplayed1=0;
    timercalculated=0; // vreme sledeceg koraka jos nije izracunato
}
if (!(cyclecount % 1)) { //ispisati tekst svaki hiljaditi put kada se prodje kroz ovu petlju
    stepelapsed = (long)(millis()-startstep); // racuna koliko je milisekundi prošlo od pocetka koraka
    stepelapsedsecs = (int)(stepelapsed/1000); // konvertovanje u sekunde (ne bas precizno)
    formattimesecs(stepelapsedsecs,stepelapsedtime);
    formattime(0,vrememesanja2,0,steptotaltime);
    lcd.setCursor(0, 1);
    lcd.print("Vkr:"); // vreme trajanja koraka u procesu
    lcd.setCursor(4, 1);
    lcd.print(stepelapsedtime); // vreme trajanja koraka u procesu
    cyclecount++;
}
break;
```

Слика 4.2.1.7.2 - Програмски код етапе 7

4.2.1.8 Case 8 - Одлеживање комине

Ова етапа у процесу представља део када систем мирује. Сврха ове етапе је у мировању, јер се тиме обезбеђује таложење тропа на дуплом дну суда. Талог има сврху финог филтера. Овај део процеса је припрема за етапу 9, где се врши претакање и рецикулација (нем. *vorlauf*). На слици 4.2.1.8.1 се налази изглед система у току овог процеса.



Слика 4.2.1.8.1 - Одлеживање комине

У коду постоји само временски параметар који износи неких 5 минута, тј. време корака је 5 минута.

На слици испод се налази изглед кода (слика 4.2.1.8.2).

```
case 8: // Korak 8 - Pauziranje nakon mesanja / odlezavanje komine
  if (!msgdisplayed1)
  {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("7.Odlezavanje komine");
    delay(50);
    lcd.setCursor(13, 1);
    lcd.print("0,0,0,0");
    delay(50);
    msgdisplayed1=1;
  }
  if (!timercalculated) {
    startstep = millis(); // pokreni sat koraka, citaj milisekunde pocevsi od ovog trenutka
    steptime = (unsigned long)vrememirM*60*1000; // racuna koliko je vremena potrebno za ovaj korak uzimajuci parametar "vrememirM"
    stopstep = startstep + steptime; // racuna kraj ovog koraka u odnosu na ukupno proteklo vreme i uzimajuci u obzir parametar "vrememirM"
    timercalculated = 1; // ne radi ponovo u ovom koraku
  }
  if (millis()>=stopstep) { // vreme je isteklo za ovaj korak
    step++; // predji na sledeci korak
    msgdisplayed1=0;
    timercalculated=0; // sledeci korak jos nije izracunat
  }
  if (!(cyclecount % 1)) { //ispisati tekst svaki hiljaditi put kada se prodje kroz ovu petlju
    stepelapsed = (long)(millis()-startstep); // racuna koliko je milisekundi prošlo od pocetka koraka
    stepelapsedsecs = (int)(stepelapsed/1000); // konvertovanje u sekunde (ne bas precizno)
    formattimesecs(stepelapsedsecs,stepelapsedtime);
    formattime(0,vrememirM,0,steptotaltime);
    lcd.setCursor(0, 1);|
    lcd.print("Vkr:"); // vreme trajanja koraka u procesu
    lcd.setCursor(4, 1);
    lcd.print(stepelapsedtime); // vreme trajanja koraka u procesu
    cyclecount++;
  }
  break;
```

Слика 4.2.1.8.2 - Изглед кода у етапи 8

4.2.1.9 Case 9 - Претакање и рецикулација

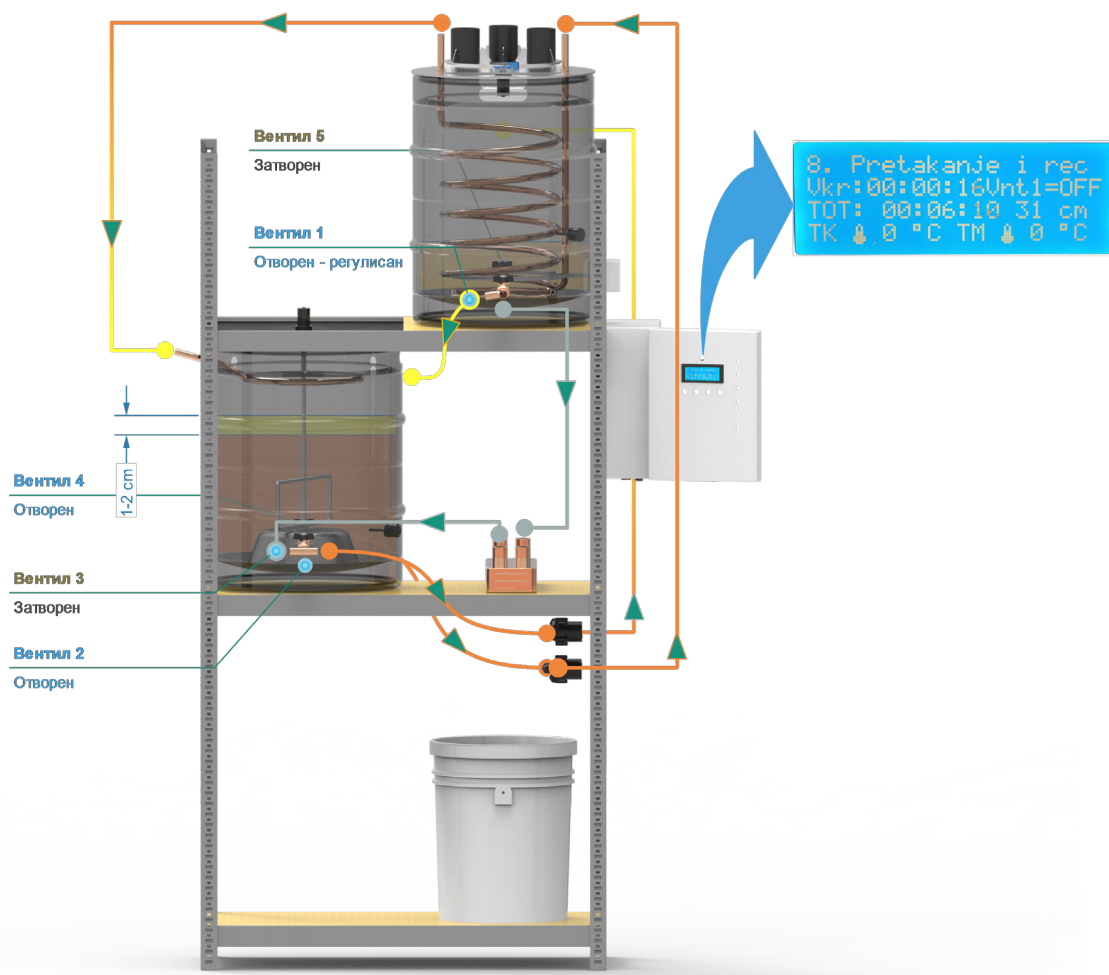
Ова етапа је најсложенија по питању корисникових задужења у току њеног трајања. Такође, овај процес захтева константно надгледање висине течности у мешаони, која треба да износи неких 1 до 2 cm изнад површине тропа. Овај процес се регулише преко кугличног вентила 1 преко којег се повећава и смањује доток течности из кувала у мешаону.

Дакле у току овог процеса, преостала чиста вода из кувала се утаче у мешаону, у исто време се и враћа у кувало преко пумпе за претакање, врши се покапавање преко система рецикулације чиме се ситним капљицама разбија површински слој тропа равномерно, из тог разлога је важно задржати ниво течности у суду 1 до 2 центиметра изнад површине тропа.

Током овог процеса преостала чиста вода се сједињује са комином, чиме се врши и испирање тропа. Та вода је такође загрејана (око 60 степени целзијуса).

На почетку процеса је потребно потврдити да је вентил број 1 отворен, притиском на тастер 1, када и почиње одбројавање за овај корак. Пошто је температура и концентрација шећера у течности велика, чиме је и повећана густина воде, сепарација кисеоника приликом претакања центрифугалним пумпама је олакшана. Да се не би створио ваздушни чеп, као и у етапи укомљавања, убачена су технолошка решења за избацивање ваздуха из система за обе пумпе.

Из свих претходно наведених разлога је неопходно да оператер константно визуелно надгледа систем. На слици 4.2.1.9.1 је приказан изглед система у току овог процеса са температурама и смеровима кретања течности.



Слика 4.2.1.9.1 - Претакање и рецикулација (Етапа 9)

Изглед кода који је у овој етапи је приказан на слици 4.2.1.9.2.

```

case 9: // Step 9 - Pretakanje u kuvalo / recirkulacija (Vorlauf)
if (!asgdisplayed1)
{
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("8. Pretakanje i rec");
  delay(50);
  lcd.setCursor(12, 1);
  lcd.print("Vnt1=OFF"); // obvestenje za otvaranje ventila 1, kada se to uradi pritiskom na taster 1 se potvrđuje
  delay(50);
  startStep = millis(); // pokreni sat koraka, citaj milisekunde povepsi od ovog trenutka
  asgdisplayed1=1;
}

digitalWrite(pinpumpeprt, LOW); // ukljuci pumpu za pretakanje u kuvalo
digitalWrite(pinpumparec, LOW); // ukljuci pumpu za recirkulaciju

while (taster3.isPressed() == true) {
  digitalWrite(pinpumpeprt, HIGH); // Ugastiti pumpu za pretakanje //
}
digitalWrite(pinpumpeprt, LOW); // Upaliti pumpu za pretakanje //
// TEHNOLOSKO RESENJE ZA IZBACIVANJE VAZDUHA IZ SISTEMA

while (taster4.isPressed() == true) {
  digitalWrite(pinpumparec, HIGH); // Ugastiti pumpu za recirkulaciju //
}
digitalWrite(pinpumparec, LOW); // Upaliti pumpu za recirkulaciju //
// TEHNOLOSKO RESENJE ZA IZBACIVANJE VAZDUHA IZ SISTEMA

if (!timercalculated && taster1.uniquePress()) {
  beep(PinZvucnik, 2093, 500); // Emituje ton C (visok) u trajanju od pola sekunde / obavestava da treba otvoriti ventil 1 i pritisnuti taster 1 (respektivno)
  lcd.setCursor(12, 1);
  lcd.print("Vnt1= ON");
  stepTime = (unsigned long)vremeprtrec*60*1000; // racuna koliko je vremena potrebno za ovaj korak uzimajuci parametar "vremeprtrec"
  stopStep = startStep + stepTime; // racuna kraj ovog koraka u odnosu na ukupno proteklo vreme i uzimajuci u obzir parametar "vremeprtrec"
  timercalculated = 1; // ne radi ponovo u ovom koraku
}

if (millis() >= stopStep && timercalculated == 1) { // vreme je isteklo
  digitalWrite(pinpumparec, HIGH); // ugasi pumpu za recirkulaciju
  beep(PinZvucnik, 2093, 500); // Emituje ton C u trajanju od pola sekunde sto oznacava kraj pretakanja, i obvestenje za zatvaranje ventila 1
  lcd.setCursor(12, 1);
  lcd.print("Vnt1=OFF");
  mashoutTemp = tempSensor2; // snimi trenutnu temperaturu mesane za kasnije prikazivanje u izveštaju
  step++;
  asgdisplayed1=0;
  timercalculated=0; // vreme sledeceg koraka jos nije izracunato
}

if (!cyclecount & 1) { // ispisati tekst svaki hiljaditi put kada se prodje kroz ovu petlju
  stepElapsed = (long)(millis()-startStep); // racuna koliko je milisekundi proslo od pocetka koraka
  stepElapsedSecs = (int)(stepElapsed/1000); // konvertovanje u sekunde (ne bas precizno)
  formatTimeSecs(stepElapsedSecs, stepElapsedTime);
  formatTime(0, vremeprtrec, 0, stepTotalTime);
  lcd.setCursor(0, 1);
  lcd.print("Vkr:"); // vreme trajanja koraka u procesu
  lcd.setCursor(4, 1);
  lcd.print(stepElapsedTime); // vreme trajanja koraka u procesu
  cyclecount++;
}
break;

```

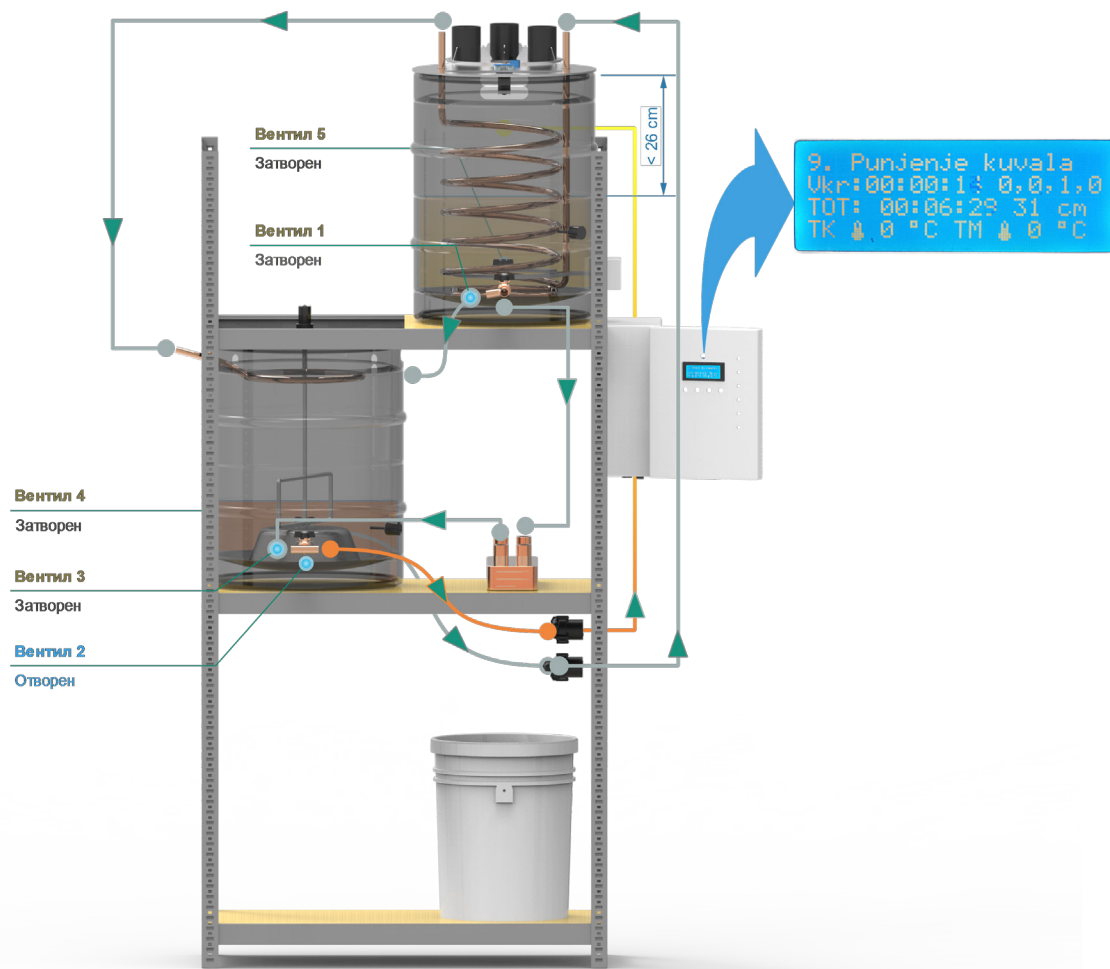
Слика 4.2.1.9.2 - Програмски код етапе 9

4.2.1.10 Case 10 - Пуњење кувала

У овој етапи се течност претаче из мешаоне у кувало. Само вентил 2 је отворен. Грејач ће се упалити након што вредности добијане са ултразвучног сензора буду мање од 26 центиметара. Овај део кода је убачен због смањења временског интервала загревања течности у следећој етапи. Претходно описан процес важи ако је параметар **checkfloat** постављен на 1, у случају да је постављен на 0, грејач ће се упалити након 15 минута од почетка претакања.

Као и код претходних процеса, где се врши претакање и у овом је имплементирано технолошко решење за изbacивање ваздуха из система.

На слици 4.2.1.10.1 се налази изглед система у току овог процеса.



Слика 4.2.1.10.1 - Изглед система у етапи претакања сладовине у кувало

Изглед кода који контролише овај део процеса је приказан на слици 4.2.1.10.2.

```
case 10: // Korak 10 - Punjenje kuvala
  if (!msgdisplayed1)
  {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("9. Punjenje kuvala");
    delay(50);
    lcd.setCursor(13, 1);
    lcd.print("0,0,0,0");
    delay(50);
    msgdisplayed1=1;
  }
  digitalWrite(pinpumpaprt, LOW); // ukljuciti pumpu za pretakanje
  delay(50);
  lcd.setCursor(17, 1);
  lcd.print("1");
```

```

while(taster3.isPressed() == true){
    digitalWrite(pinpumpaprt, HIGH);    // Ugasiti pumpu za pretakanje    //
    lcd.setCursor(17, 1);                //|
    lcd.print("0");                      //
}                                        //
TEHNOLOSKO RESENJE ZA IZBACIVANJE VAZDUHA IZ SISTEM

if (checkfloat == 1)
{
    if (srvrednostussp < 26 && !(srvrednostussp == 00))
    {
        digitalWrite(pingrejac, LOW);    // upali grejac
        delay(50);
        lcd.setCursor(13, 1);
        lcd.print("1"); // paljenje grejaca 1 - > upaljen
        delay(50);
    }
    else {
        digitalWrite(pingrejac, HIGH);    // ugasi grejac
        lcd.setCursor(13, 1);
        lcd.print("0"); // gasenje grejaca 0 - > ugasen
    }
}
else if (checkfloat == 0 && startstep >= 900000) { // upaliti grejac nakon 15 minuta pretakanja (paziti na pretakanje !!!!!)
    digitalWrite(pingrejac, LOW);    // upali grejac
    lcd.setCursor(13, 1);
    lcd.print("1"); // paljenje grejaca 1 - > upaljen
}

if (!timercalculated) {
    startstep = millis(); // pokreni sat koraka, citaj milisekunde pocevsi od ovog trenutka
    steptime = (unsigned long)vremeprtK*60*1000; // racuna koliko je vremena potrebno za ovaj korak uzimajuci parametar "vremeprtK"
    stopstep = startstep + steptime; // racuna kraj ovog koraka u odnosu na ukupno proteklo vreme i uzimajuci u obzir parametar "vremeprtK"
    timercalculated = 1; // ne radi ponovo u ovom koraku
}
if (millis()>=stopstep) { // vreme koraka je isteklo
    digitalWrite(pinpumpaprt, HIGH);    // ugasiti pumpu za pretakanje
    step++;
    msgdisplayedl=0;
    timercalculated=0; // vreme sledeceg koraka jos nije izracunato
}
if (!(cyclecount % 1)) { // ispisati tekst svaki hiljaditi put kada se prodje kroz ovu petlju
    steplapsed = (long)(millis()-startstep); // racuna koliko je milisekundi proslo od pocetka koraka
    steplapsedsecs = (int)(steplapsed/1000); // konvertovanje u sekunde (ne bas precizno)
    formattimesecs(steplapsedsecs,steplapsedtime);
    formattime(0,vremeprtK,0,steptotaltime);
    lcd.setCursor(0, 1);
    lcd.print("Vkr:"); // vreme trajanja koraka u procesu
    lcd.setCursor(4, 1);
    lcd.print(steplapsedtime); // vreme trajanja koraka u procesu
    cyclecount++;
}
break;

```

Слика 4.2.1.10.2 - Програмски код етапе 10

4.2.1.11 Case 11 - Подизање температуре у кувалу

Ова етапа служи за загревање течности у суду до температуре кувања. Издвојена је из разлога што време кувања мора да буде време када се комина налазила на прецизираној температури кувања, а не време које је протекло у загревању исте. Овај процес може да траје и до 2 сата због снаге грејача.

Након што се постигне захтевана температура кувања, програм ће аутоматски прећи на етапу кувања.

Пошто је овај процес једноличан што се тиче изгледа система и физичких промена, изоставићу приказивање ове етапе визуелно, већ ћу само приказати програмски код који је управљао њом.

На слици 4.2.1.11.1 се налази програмски код који управља овом етапом. Ова етапа је једна од ретких која није временски зависна, и трајаће докле год се не достигне задана температура. Из тог разлога је много важно прецизирати температуру коју је могуће достићи, да би програм аутоматски прешао на етапу кувања.

```

case 11: // Korak 11 - Dovedi temperaturu vode do kuvanja
  if (!msgdisplayed1)
  {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("10. Podizanje temp");
    delay(50);
    lcd.setCursor(13, 1);
    lcd.print("0,0,0,0");
    delay(50);
    msgdisplayed1=1;
    startstep=millis();
  }

  digitalWrite(pingrejac, LOW);
  lcd.setCursor(13, 1);
  lcd.print("1"); // paljenje grejaca 1 - > upaljen

  if (tempsenzor1>=kuvtemp) {
    step++;
    msgdisplayed1=0;
    timercalculated=0; // vreme za sledeci korak jos nije izracunato
  }
  if (!(cyclecount % 1)) { // ispisati tekst svaki hiljaditi put kada se prodje kroz ovu petlju
    stepelapsed = (long)(millis()-startstep); // racuna koliko je milisekundi proslo od pocetka koraka
    stepelapsedsecs = (int)(stepelapsed/1000); // konvertovanje u sekunde (ne bas precizno)
    formatmilliseconds(stepelapsedsecs,stepelapsedtime);
    lcd.setCursor(0, 1);
    lcd.print("Vkr:"); // vreme trajanja koraka u procesu
    lcd.setCursor(4, 1);
    lcd.print(stepelapsedtime); // ispisuje vreme trajanja koraka
    cyclecount++;
  }
  break;

```

Слика 4.2.1.11.1 - Програмски код етапе 11

4.2.1.12 Case 12 - Кување

У овој етапи се одвија низ процеса који су од важног значаја за крајњи производ. Као прво ту је регулације температуре, такође, етапа је временски зависна, на основу које се и врши аутоматско додавање хмеља.

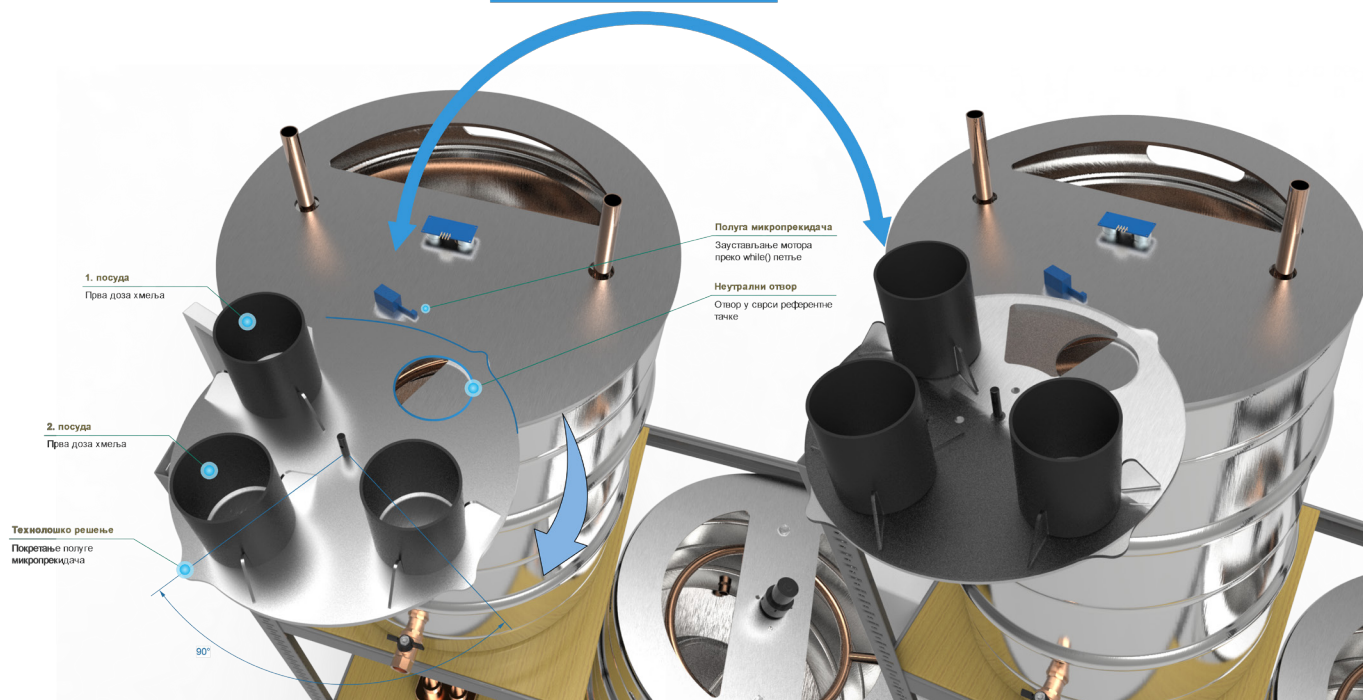
Аутоматско додавање хмеља се врши помоћу претходно описаног подсклопа овог система - подсклоп додавача хмеља. Подсклоп поседује 3 цевасте посуде које су на дну отворене, али само у одређеним позицијама је пут, кроз њих, до комине отворен.

Постоје четири отвора на плочи за додавање хмеља од којих је један неутралан, и означава почетну позицију. На сликама 4.2.1.12.1 је приказана функционалност система додавања хмеља. Сваки отвор на плочи је постављен под углом од 90 степени у односу на осу ротације механизма.

Време трајања овог корака је дефинисано параметром **vremekuv**. Параметри **dodH1**, **dodH2** и **dodH3** су временске одреднице када ће дозе хмеља бити убачене у кувало у току процеса кувања.

Изглед система у току ове етапе је непроменљив, тако да ће само изглед програмског кода бити приказан. На слици 4.2.1.12.2 се налази програмски код. Ова петља је дужа од свих претходно наведених, због утицаја већег броја параметара на њу.

Изглед доње плоче додавача хмеља



Слика 4.2.1.12.1 - Функционалност подскопа додавача хмеља

```
case 12: // Korak 12 - Kuvanje
if (!msgdisplayed1)
{
    boilttime = stepelapsed; // ova velicina ce snimiti prethodan korak, tj. vreme koje je bilo potrebno da se dostigne temperatura kuvanja
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("11. Kuvanje");
    delay(50);
    lcd.setCursor(13, 1);
    lcd.print("0,0,0,0");
    delay(50);
    msgdisplayed1=1;
}

if (tempsenzor1 >= (kuvtemp + 1)){
    digitalWrite(pingrejac, HIGH);
    lcd.setCursor(13, 1);
    lcd.print("0"); // gasenje grejaca 0 -> UGASEN
}
else if (tempsenzor1 <= kuvtemp){
    digitalWrite(pingrejac, LOW);
    lcd.setCursor(13, 1);
    lcd.print("1"); // paljenje grejaca 1 -> upaljen
}

if (!timercalculated) {
    startstep = millis(); // pokreni sat koraka, citaj milisekunde pocevsi od ovog trenutka
    steptime = (unsigned long)vremekuv*60*1000; // racuna koliko je vremena potrebno za ovaj korak uzimajuci parametar "vremekuv"
    stopstep = startstep + steptime; // racuna kraj ovog koraka u odnosu na ukupno proteklo vreme i uzimajuci u obzir parametar "vremekuv"

    vremedodH1 = (unsigned long)dodH1*60*1000; //
    vremedodH2 = (unsigned long)dodH2*60*1000; // Vremena dodavanja hmelja
    vremedodH3 = (unsigned long)dodH3*60*1000; //

    timercalculated = 1; // ne radi ponovo u toku ovog koraka
    msgdisplayed2 = 0;
}

if ((millis())>=(vremedodH1+startstep)) && (millis())<=(vremedodH2+startstep) && msgdisplayed2 == 0)
{

```

Слика 4.2.1.12.2 - Програмски код етапе 12 (1. део кода)

```

    lcd.setCursor(0, 0);
    lcd.print("1.Dodavanje");
    lcd.setCursor(13, 0);
    lcd.print((int)dodH1);
    lcd.setCursor(15, 0);
    lcd.print("min");
    beep(PinZvucnik,1760,500); //C: reprodukuje ton A6 u trajanju od pola sekunde
    while(!(mikrotaster1.uniquePress()))
    {
        digitalWrite(dhmelja, LOW);
    }
    digitalWrite(dhmelja, HIGH);
    msgdisplayed2 = 1;
}
if ((millis())>=(vremedodH2+startstep) && (millis())<=(vremedodH3+startstep) && msgdisplayed2 == 1 )
{
    lcd.setCursor(0, 0);
    lcd.print("2.Dodavanje");
    lcd.setCursor(13, 0);
    lcd.print((int)dodH2);
    lcd.setCursor(15, 0);
    lcd.print("min");
    beep(PinZvucnik,1760,500); //C: reprodukuje ton A6 u trajanju od pola sekunde

    while(!(mikrotaster1.uniquePress()))
    {
        digitalWrite(dhmelja, LOW);
    }
    digitalWrite(dhmelja, HIGH);
    msgdisplayed2 = 0;
}
if ((millis())>=(vremedodH3+startstep) && msgdisplayed2 == 0)
{
    lcd.setCursor(0, 0);
    lcd.print("3.Dodavanje");
    lcd.setCursor(13, 0);
    lcd.print((int)dodH3);
    lcd.setCursor(15, 0);
    lcd.print("min");
    beep(PinZvucnik,1760,500); //C: reprodukuje ton A6 u trajanju od pola sekunde

    while(!(mikrotaster1.uniquePress()))
    {
        digitalWrite(dhmelja, LOW);
    }
    digitalWrite(dhmelja, HIGH);
    msgdisplayed2 = 1; // vremena dodavanja hmelja se racunaju od pocetka koraka ;)
}
// kraj koda za dodavanje hmelja
if (millis())>=stopstep) { // vreme ovog koraka je isteklo
    digitalWrite(pingrejac, LOW);
    step++;
    msgdisplayed1=0;
    timercalculated=0; // vreme sledeceg koraka jos nije izracunato
    digitalWrite(pingrejac, HIGH);
    while(!(mikrotaster1.uniquePress())) // vracanje dodavaca hmelja u pocetnu poziciju
    {
        digitalWrite(dhmelja, LOW);
    }
    digitalWrite(dhmelja, HIGH);
}
if (!(cyclecount % 1)) { // ispisati tekst svaki hiljaditi put kada se prodje kroz ovu petlju
    stepelapsd = (long)(millis()-startstep); // racuna koliko je milisekundi proslo od pocetka koraka
    stepelapsdsecs = (int)(stepelapsd/1000); // konvertovanje u sekunde (ne bas precizno)
    formattimeseconds(stepelapsdsecs,stepelapsdtime);
    formattime(0,vremekuv,0,steptotaltime);
    lcd.setCursor(0, 1);
    lcd.print("Vkr:"); // vreme trajanja koraka u procesu
    lcd.setCursor(4, 1);
    lcd.print(stepelapsdtime); // vreme trajanja koraka u procesu
    cyclecount++;
}
break;

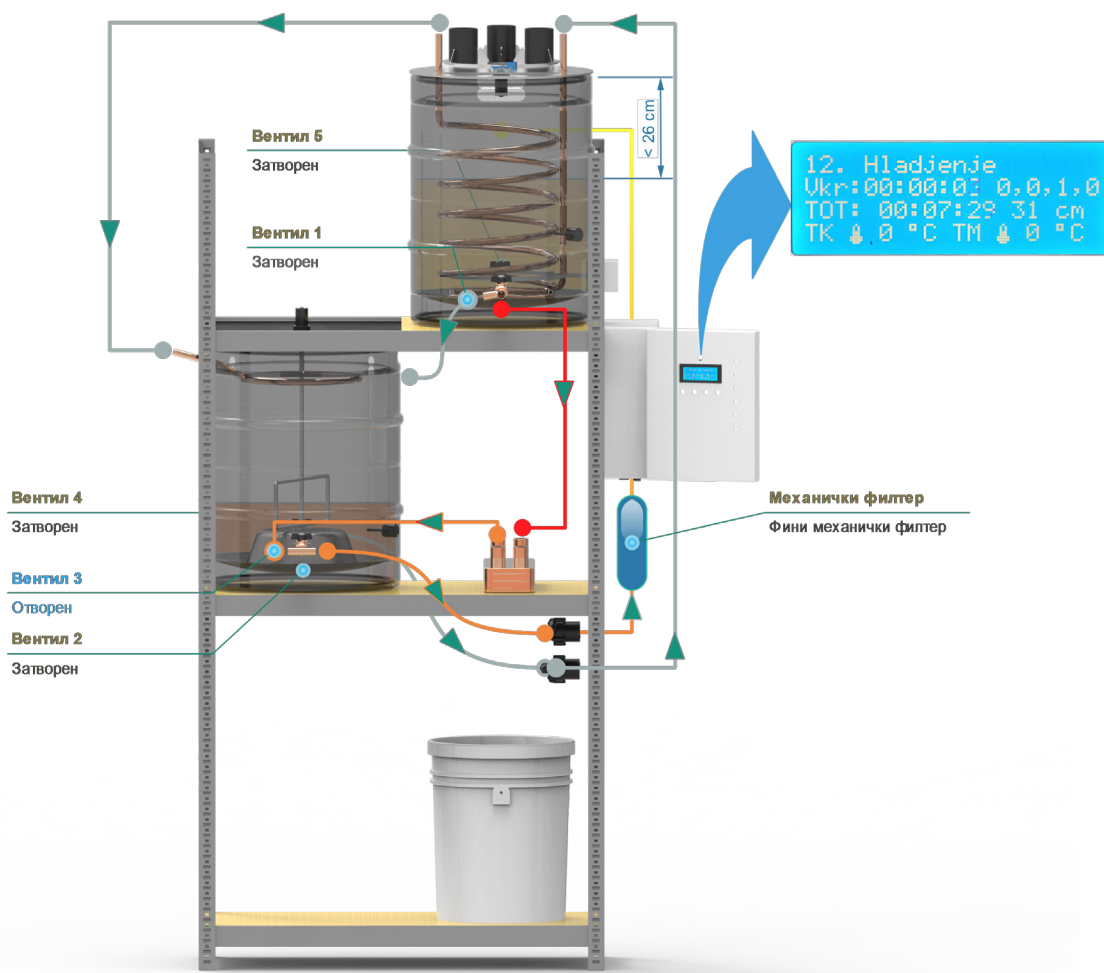
```

Слика 4.2.1.12.2 - Програмски код етапе 12 (2. део кода)

4.2.1.13 Case 13 - Хлађење и филтрација

У овом процесу се течност из кувала пребацује у размењивач топлоте, где се хлади и поново враћа у суд. Овакав начин хлађења није ефикасан колико би могао да буде, али да би се течност што више исфилтрирала што процесом рецикулације кроз размењивач и поновним утакањем кроз закривљени прикључак, што проласком течности кроз фини филтер, овај начин је и усвојен. Процес је временски зависан и требало би да траје неких 15 до 20 минута.

На слици 4.2.1.13.1 је приказан систем током ове етапе.



Слика 4.2.1.13.1 - Изглед система током процеса хлађења и филтрације

Суштина ове етапе је хлађење и спречавање стварања непожељних микроорганизама у пиву, а оптимална температура за стварање оваквих организама је између 50 и 30 степени, зато је много важна брзина хлађења. Коришћењем овог начина хлађења се постигло брзо хлађење пива и додатна филтрација.

На слици 4.2.1.13.2 се налази програмски код.

```
case 13: // Korak 13 - Hladjenje i filtracija
  if (!msgdisplayed1)
  {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("12. Hladjenje");
    delay(50);
    lcd.setCursor(13, 1);
    lcd.print("0,0,0,0");
    delay(50);
    msgdisplayed1=1;
  }
  digitalWrite(pinpump, LOW);
  lcd.setCursor(17, 1);
```

Слика 4.2.1.13.2 - Програмски код етапе 13 (1. део кода)


```

lcd.print("1");

while(taster3.isPressed() == true){
    digitalWrite(pinpumpaprt, HIGH); // Ugastiti pumpu za pretakanje //
    lcd.setCursor(17, 1); //
    lcd.print("0");
} //
digitalWrite(pinpumpaprt, LOW); // Upaliti pumpu za pretakanje // TEHNOLOSKO RESENJE ZA IZBACIVANJE VAZDUHA IZ SISTEMA
lcd.setCursor(17, 1);
lcd.print("1");

if (!timercalculated) {
    startstep = millis(); // pokreni sat koraka, citaj milisekunde pocevsi od ovog trenutka
    steptime = (unsigned long)hladtemp*60*1000; // racuna koliko je vremena potrebno za ovaj korak uzimajuci parametar "hladtemp"
    stopstep = startstep + steptime; // racuna kraj ovog koraka u odnosu na ukupno proteklo vreme i uzimajuci u obzir parametar "vremetemp"
    timercalculated = 1; // ne radi ponovo u ovom koraku
}

if (millis()>=stopstep) { // vreme ovog koraka je zavrшено
    step++;
    msgdisplayedl=0;
    timercalculated=0; // vreme sledeceg koraka jos nije izracunato
}

if (!(cyclecount % 1)) { // ispisati tekst svaki hiljaditi put kada se prodje kroz ovu petlju
    stepelapsed = (long)(millis()-startstep); // racuna koliko je milisekundi proslo od pocetka koraka
    stepelapsedsecs = (int)(stepelapsed/1000); // konvertovanje u sekunde (ne bas precizno)
    formattimeseconds(stepelapsedsecs,stepelapsedtime);
    lcd.setCursor(0, 1);
    lcd.print("Vkr:"); // vreme trajanja koraka u procesu
    lcd.setCursor(4, 1);
    lcd.print(stepelapsedtime); // vreme trajanja koraka u procesu
    cyclecount++;
}
break;

```

5. Прављење пива

Оно што долази на крају комплетног процеса развоја, пројектовања и израде система је прављење пива. Избор састојака, њихове количине и односа се може направити преко многобројних интернет калкулатора. Од тога коју врсту пива желимо да добијемо зависи избор компоненти за његову израду. Пошто постоје три основна састојка пива, важно је знати њихове одлике, карактеристике и намену. Та три састојка су:

- Слад
- Хмељ
- Пивски квасац

1. Слад

Систем је дизајниран за прављење пива из житаричног слада, а не њиховог екстракта. Слад може бити направљен од било које врсте житарица, у то спадају јечам, пшеница, пиринач, итд. Слад чини 95 % укупне тежине састојака пива.

Основна одлика слада је његова боја. Мерна јединица је EBC (European Brewery Convention). Што је мањи број ове величине, то је пиво светлије. На слици 5.1 се види табела боја пива усвојена према европској пиварској конвенцији.

Карактеристика слада је укус који даје крајњем производу и када се употребљава (код светлих или код тамних пива). Количина екстракта која се може добити из слада је вредност која директно утиче на алкохолну вредност пива и то представља његову **намену**. Она може да буде основна или зачинска. Основни слад представља слад из којег се може извући већа количина екстракта, а зачински су они који више утичу на укус и боју пива, у односу на количину екстракта који дају комини. Мерна јединица количине алкохола је ABV (енг. Alcohol by Volume), и изражава се у процентима.

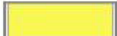
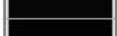
2. Хмељ

Хмељ је састојак који првенствено утиче на горчину пива, а затим на његову арому и мирис. Додаје се у малим количинама у прецизно одређеним интервалима у току процеса кувања пива. Мерна јединица за горчину је IBU (енг. International Bittering Unit). Горчина се мери са спектрофотометром, али се углавном горчина пива искуствено процењује.

3. Пивски квасац

Квасац је главни покретач процеса ферментације и утиче на процес флокулације беланчевина. Главна карактеристика пивских квасаца је температура на којој подстичу ферментацију, тако да избор њега зависи од боје пива које се прави, обзиром да температура ферментације светлих пива је око 22 степена целзијуса док је код тамних нешто виша. Количина квасца према укупној запремини састојака је неких 0.2 %.

Рецепти за прављење пива се могу састављати према жељи или усвајати са разних форума на интернету. Рецептe које сам ја користио сам сам направио према подацима које сам поседовао (искуства других, препоруке дистрибутера, интернет калкулатори, итд.).

	colours	EBC
pale		4
light blond		6
blond		8
yellow		12
gold		16
		20
amber		26
		33
copper		39
dark copper		47
		57
brown		69
dark brown		79
		139

Слика 5.1 - Боје пива

5.1 Прибор

Оно што је неопходно у процесу прављења пива је прибор за мерење одређених величина током процеса кувања пива. Као основни део прибора ту је мензура, запремине 250 ml. Она служи за процену боје пива, мерење густине течности и као дозер за почетно разлагање пивског квасца. На слици 5.1.1 се налази комплетан прибор потребан у процесу прављења пива.



Слика 5.1.1 - Прибор за мерење

Ареометар је мерни инструмент који служи као мерач густине течности. На основу измерене величине се добија, тзв. почетна гравитација (енг. Original Gravity) и крајња гравитација (енг. Final Gravity). Преко ових величина се рачуна концентрација екстракта јечма и количина алкохола у коначном у пиву. Следећи мерни инструмент који је потребан је живин термометар. Он служи за мерење температуре пива пре затварања ферментора, да би се знала температура на коју је охлађено младо пиво. Има специјалан ергономски облик да би могао да се окачи на ивицу суда у коме ће се одвијати процес ферментације. Посуда као четврти део прибора, служи за почетно растварање квасца пре убацивања у ферментор.

5.2 Рецепт 1

Тип овог пива је American Pale Ale (APA). Укупна запремина бачве је 42 L (односи се на укупну количину воде), а коначна запремина је износила 26 L (запремина после кувања). Иначе, ово је пробна рецептура и прво пиво које је добијено из овог система.

Састојци:

- 5,5 kg јечмени слад Pilsen 3 EBC

-
- 0,3 kg јечмени слад Cara Ruby 50 EBC
 - 30 g хемљ Amarilo
 - пивски квасац (7 g Safeale US-05)

Параметри система:

- Страјк температура = 80 ° C
- Температура укомљавања = 66 ° C
- Температура покапавања = 80 ° C
- Температура кувања = 99 ° C
- Време укомљавања = 1 сат и 30 минута
- Време кувања = 1 сат

Циљани резултат:

- OG: 1.050
- FG: 1.014
- ABV: 4.5 %
- IBU: 20
- EBC: 15,65

Комплетан процес је трајао 6 сати и 40 минута. Почетна гравитација пива није била као што је циљано износила је 1.045, док је коначна била 1.014. Због нешто ниже почетне гравитације, количина алкохола по јединици запремине је износила 4.06%. Што се тиче горчине и боје, они су према претходно поменутих показатељима били у реду, тј. постигнут је циљ.

5.3 Рецепт 2

Тип и овог пива је American Pale Ale (APA). Укупна запремина бачве је 42 L, а коначна запремина је износила 30 L. Ово је друга успешно произведена количина пива. Сврха овог рецепта је да покаже дејство хмеља на укус пива.

Састојци:

- 5 kg јечмени слад Pilsen 7 EBC
- 0,5 kg јечмени слад Cara Ruby 50 EBC
- 0,5 kg јечмени слад Biscuit 40 EBC
- 100 g хемљ Bravo
- пивски квасац (7 g Safeale US-05)

Параметри система:

- Страјк температура = 0
- Температура укомљавања = 66 ° C
- Температура покапавања = 79 ° C
- Температура кувања = 97 ° C
- Време укомљавања = 1 сат и 30 минута

-
- Време кувања = 1 сат

Циљани резултат:

- OG: 1.046
- FG: 1.013
- ABV: 4.33 %
- IBU: 98.62
- EBC: 19.45

Комплетан процес је трајао 6 сати. Све је испало како је планирано, с тим да превелика количина хмеља је прилично повећала горчину пива и тиме показала неки вид крајности како хмељ као састојак утиче на укус пива. Превелика горчина и мутност овог пива је утицала на сврху овог екперимента, тј. да он служи само као једна од референтних тачака која треба да води пивара до циља.

5.4 Рецепт 3

Тип и овог пива је American Pale Ale (APA). Укупна запремина бачве је 42 L, а коначна запремина је износила 30 L. Ово је трећа успешно произведена количина пива. Циљ овог рецепта је показивање утицаја зачинског јечма на боју пива.

Састојци:

- 5 kg јечмени слад Pilsen 7 EBC
- 1 kg јечмени слад Cara Ruby 50 EBC
- 55 g хемљ Bravo
- пивски квасац (7 g Safeale US-05)

Параметри система:

- Страјк температура = 0
- Температура укомљавања = 66 ° C
- Температура покапавања = 80 ° C
- Температура кувања = 98 ° C
- Време укомљавања = 1 сат и 30 минута
- Време кувања = 1 сат

Циљани резултат:

- OG: 1.043
- FG: 1.011
- ABV: 4.1 %
- IBU: 69.52
- EBC: 23.08

Комплетан процес је трајао 6 сати и 15 минута. Сви жељени циљеви су постигнути, а првенствено изразита карамел боја. Грочина је у реду, с тим да пиво има нешто мање алкохола у себи, али то је опет ствар укуса и жеље пивара.



Цена комплетне количине састојака за сваки од претходно наведених рецепата износи оквирно 2.500 динара. Процењена ефикасност система је 65 %, а то се односи на то у којој мери су циљани резултати испуњени.

6. Закључак

Крајњи резултат овог вишемесечног пројекта је готов производ - пиво. Оно што је и током овог рада могло да се закључи је да квалитет готовог производа не зависи само од пројектованог система већ и од оператера што будућим корисницима даје слободу експериментисања.

Сва претходно објашњена поглавља су само део путање развоја програмабилног аутоматског система мале пиваре. Сви захтеви који су постојали на почетку су испуњени, с тим да систем није у потпуности аутоматизован. Степен поновљивости процеса је велика. Ефикасност система није као што је очекивана (нешто је мања), због коришћених компонената.

Овај мастер рад представља један истраживачки пројекат који је обезбедио довољно знања читаоцу да овај систем направи и пусти у рад, што је један од његових циљева. Све компоненте су доступне у слободној продаји у нашој земљи, а њихова цена није превисока.

Нешто што ми је много помогло у овом раду је мој претходни пројекат и завршни рад „Програмирање PIC микроконтролера за хидропонијски систем“, из којег сам понео доста знања из сфере програмирања и електронских компонената, начина управљања и планирања једног пројекта. Такође, предмети који су ми много помогли током овог пројекта су „Брза израда прототипова“, „Програмски језици“ и многобројне литературе везане за сличне сфере машинства.

Пиварска индустрија цвета у свету последњих неколико година, а погонско гориво стиже из микро-пивара налик овој широм света, као и наменских барова. Број пивара се током 5 година удвостручио, а индустрија заједно са увозницима запошљава више од два милиона људи и конкретно у буџет америчке државе уплаћује око 49 милијарди долара годишње. Све је популарнија идеја о прављењу пива директно у баровима и увођење точилица које ће точити домаће пиво, што утиче на раст потражње специјалних уређаја - микро-пивара. Иначе, микро-пиваре ентузијаста и “домаћа пивска радиност” последњих годину две се боре да пронађу своје место и у Србији. За производњу пива је потребан и јечам и хмељ, а обзиром да мале пиваре користе три пута више сировина од великих, подухват прављења једне микро-пиваре би значао и подстицање производње хмеља и јечма од стране пољопривредника, што би и за њих значило више посла и више новца.

Процес прављења и стварања овог система је за мене био задовољство, као и уживање у резултату. Комплетан систем је претрпео низ модификација које су мене коштале доста времена и труда за налажење решења, укључујући и новац. На крају све то је допринело стицању велике количине знања из сфера привреде, машинства и економије.

Надам се да сам са овим радом обезбедио потребно предзнање за све колеге који су заинтересоване за ову сферу производње и заинтересовао оне које у будућности планирају да развију и спроведу неки свој пројекат у дело.

7. Литература

- [1] „Rapid Prototyping – брза израда прототипова“, др Ненад Грујовић, 2005
- [2] Andreas Gebhardt, “Rapid Prototyping”, Hanser, 2000
- [3] Милићев Д.: Објектно оријентисано програмирање на језику С++, Микрокњига, Београд, 1991.
- [4] Хенсен А.: Програмирање на језику С, Микрокњига, Београд, 1991.
- [5] „Getting Started With Arduino 2nd Edition“, Massimo Banzi, Maker Media, Inc October 2011
- [6] „Технологија пива“; <http://sr.wikipedia.org/sr/%D0%9F%D0%B8%D0%B2%D0%BE> (4. мај 2014.)
- [7] „Brewbot, Automating the Home brewing Process“; <http://www.brewbot.ca/> (14. јун 2014.)
- [8] „Migros“ - енолошко посуђе; <http://www.migros-kg.co.rs/> (6. мај 2014.)
- [9] „Мајстор за пиво“ - прибор и састојци; <http://www.majstorzapivo.com/> (10. јул 2014.)
- [10] „Perfom“ - перфорирана сита за прехранбену индустрија; http://www.perfom.rs/html/proizvo-di/srp_perforirana_sita_i_delovi.html (20. август 2014.)
- [11] „DIGIStore“ - Електронске компоненте; www.digistore.rs/webshop/ (22. септембар 2014.)
- [12] „Brewer’s Friend“ - пиварски калкулатори; <http://www.brewersfriend.com/stats/> (8. јун 2014.)
- [13] „Proelectronic“ - електронске компоненте; <http://www.proelectronic.rs/> (17. јул 2014.)
- [14] „Arduino“ - библиотеке; <http://arduino.cc/> (22. септембар 2014.)
- [15] „Handbook of Brewing: Processes, Technology, Markets“ <http://onlinelibrary.wiley.com/book/10.1002/9783527623488> (26. јул 2014.)