

UNIVERZA V MARIBORU
FAKULTETA ZA ELEKTROTEHNIKO,
RAČUNALNIŠTVO IN INFORMATIKO

Miha Založnik

**AVTOMATIZIRANA PRETOČNA ENOTA Z NADZOROM
PREKO PAMETNEGA TELEFONA**

Diplomsko delo

Maribor, september 2017

AVTOMATIZIRANA PRETOČNA ENOTA Z NADZOROM PREKO PAMETNEGA TELEFONA

Diplomsko delo

Študent: Miha Založnik

Študijski program: Visokošolski študijski program
Elektrotehnika

Smer: Elektrotehnika

Mentor: doc. dr. Iztok Kramberger

Lektorica: Maja Špes, mag. sloven. (UN)



Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko
Smetanova ulica 17
2000 Maribor, Slovenija



Številka: E1040630

Datum in kraj: 13. 04. 2017, Maribor

Na osnovi 330. člena Statuta Univerze v Mariboru (Statut UM – UPB 11, Ur. l. RS, št. 44/2015)
izdajam

SKLEP O ZAKLJUČNEM DELU

1. **Mihi Založniku**, študentu študijskega programa prve stopnje VS ELEKTROTEHNIKA, smer Elektronika, se dovoljuje izdelati zaključno delo.
2. Tema zaključnega dela je pretežno s področja Inštituta za elektroniko in telekomunikacije.
3. **MENTOR:** doc. dr. Iztok Kramberger
4. **Naslov zaključnega dela:**
AVTOMATIZIRANA PRETOČNA ENOTA Z NADZOROM PREKO PAMETNEGA TELEFONA
5. **Naslov zaključnega dela v angleškem jeziku:**
AUTOMATED FLOW UNIT WITH SMART PHONE CONTROL
6. Rok za izdelavo in oddajo zaključnega dela je 30. 09. 2017. Zaključno delo je potrebno izdelati skladno z "Navodili za izdelavo zaključnega dela" in ga v treh izvodih (dva trdo vezana izvoda in en v spiralo vezan izvod) oddati v pristojnem referatu članice. Hkrati se odda tudi izjava mentor-ja/-ice (in morebitnega somentor-ja/-ice) o ustreznosti zaključnega dela.

Pravni pouk: Zoper ta sklep je možna pritožba na Senat članice v roku 10 delovnih dni od dneva prejema sklepa.

Dekan:
red. prof. dr. Borut Žalik



Obvestiti:

- kandidata,
- mentor-ja/-ico,
- odložiti v arhiv.

ZAHVALA

Zahvaljujem se mentorju, doc. dr. Iztoku Krambergerju, za pomoč in vodenje pri opravljanju diplomskega dela.

Posebna zahvala gre mojim staršem, ki so me podpirali vsa leta šolanja, ter moji puncici, ki me je podpirala in mi stala ob strani.

Zahvaljujem se tudi Oliverju Milinčiču, ki me je podpiral in mi pomagal s svojim znanjem ter vsem prijateljem, ki so me vzpodbujali in mi pomagali pri dokončanju študija in diplomske naloge.

AVTOMATIZIRANA PRETOČNA ENOTA Z NADZOROM PREKO PAMETNEGA TELEFONA

Ključne besede: prečrpavanje, bluetooth HC-06, Arduino, črpalka

UDK: 621.395.721.5:004.777(043.2)

POVZETEK

V diplomski nalogi opisujemo nadzorno enoto, ki bo vinarjem prihranila čas pri pretakanju mošta ali vina. Nadzorna enota je na eni strani sestavljena iz elektro omarice in dveh stikal – pretočnega stikala za merjenje količine tekočine in nivojskega stikala za nadziranje vklopa in izklopa črpalke. Oboje komunicira preko kabla s platformo Arduino. Na drugi strani preko bluetooth HC-06 modula pošiljamo odčitane podatke s senzorjev na aplikacijo na telefonu, kjer lahko spremljamo količino tekočine v cisternah. Nadzorna enota je splošen model in ne velja samo za črpalko, ki smo jo uporabili pri projektu. Važno je le, da je črpalka enofazna.

Key words: pumping, bluetooth, HC-06, Arduino platform, pump

UDK: 621.395.721.5:004.777(043.2)

ABSTRACT

The graduation thesis describes a control unit that will save time for wine producers when pumping must (a wine making term which means unfermented grapes, in nonprofessional's terms), or wine. On the one hand, the control unit is composed of an electrical box and two switches. First, a liquid flow switch for measuring the liquid level and the second, a level switch for controlling the turning on and off of the pump. Both switches communicate via cable with the Arduino platform. On the other hand, bluetooth HC-06 module sends data readings from switches into the app on the phone, where the amount of liquid in the tank is monitored. The control unit is a generic model and does not apply only to the pump, which was used in the project. It is also, of utmost importance that the pump is single-phased.

KAZALO

1	UVOD	2
2	PREGLED TEHNOLOGIJ	4
2.1	SENZORJI PRETOKA	4
2.1.1	Nivojsko stikalo	4
2.1.2	Pretočno stikalo	4
2.2	MIKROKRMILNIK	6
2.2.1	Mikrokrmilnik	6
2.2.2	Napajanje	7
2.3	BREŽIČNA KOMUNIKACIJA	8
2.4	ČRPALKE	11
3	ZASNOVA IN IZVEDBA SISTEMA	13
3.1	STROJNA OPREMA SISTEMA	13
3.2	RAZVOJNO OKOLJE	16
3.2.1	Arduino razvojno okolje	16
3.2.2	Android razvojno okolje	17
3.3	VGRAJENA PROGRAMSKA OPREMA	18
3.4	BREŽIČNA POVEZAVA	22
3.5	MOBILNA APLIKACIJA	23
4	MERITVE IN REZULTATI	28
5	PROTOTIP	35
6	SKLEP	37
7	VIRI IN LITERATURA	38

KAZALO SLIK

Slika 2.1: Merilnik pretoka.....	5
Slika 2.2: Mikrokrmilnik Arduino ProMini. Vir: [2].	6
Slika 2.3: Shema napajalnega dela za elemente.	7
Slika 2.4: a) 2D pogled podnožja releja (PCB) b) SCH slika releja	8
Slika 2.5: Bluetooth komunikacijski modul HC-06.	9
Slika 2.6: Shema brezžičnega komunikacijskega modula HC-06.	10
Slika 2.7: Uporabljena črpalka Rover pompe NOVAX M25.....	12
Slika 3.1: Shema elektro omarice.	14
Slika 3.2: Vtičnici in stikalo za vklop nadzorne enote.	15
Slika 3.3: a) NE vrat s tranzistorji, b) Plovec, nameščen na krnico.	16
Slika 3.4: Primer osnovnega programa HelloWorld.	17
Slika 3.5: Definicije konstant in spremenljivk.	18
Slika 3.6: Diagram poteka programa.	19
Slika 3.7: Prikaz nastavljanja željene količine.	20
Slika 3.8: Del kode, kjer nastavljamo količino.....	21
Slika 3.9: Koda preprostega števca in koda za izpis količine tekočine.	21
Slika 3.10: Povezava bluetooth.	22
Slika 3.11: Čakanje in branje.....	22
Slika 3.12: Podatki, ki jih prejme telefon.	23
Slika 3.13: Diagram poteka za Bluetooth povezavo.....	24
Slika 3.14: Prikaz seznama seznanjenih naprav.	25
Slika 3.15: Izpis, če nadzorna enota ni aktivna.	26
Slika 3.16: Zgled MainActivity.java.	27

Slika 4.1: Prikaz naraščanja frekvence.....	28
Slika 4.2: Del spektra pri merjenju.....	31
Slika 5.1: Plovec, nameščen na krnico.....	36

KAZALO TABEL

Tabela 4.1: Rezultati prečrpavanja z Rover črpalko.....	29
Tabela 4.2: Rezultati Rover črpalke med aplikacijo in izmerjeno količino.....	30
Tabela 4.3: Rezultati neznane črpalke med aplikacijo in izmerjeno količino.....	32
Tabela 4.4: Rezultati samodejnega pretoka.....	32

KAZALO ENAČB

Enačba 1.1: Senzorji pretoka.....	5
Enačba 1.2: Senzorji pretoka.....	6
Enačba 4.1: Meritve in rezultati.....	29
Enačba 4.3: Meritev in rezultati.....	30
Enačba 4.3: Meritev in rezultati.....	33

UPORABLJENE KRATICE

UZ	ultrazvočni senzor
IEC	Mednarodna komisija za elektrotehniko
C++	programski jezik
USB	univerzalno serijsko vodilo
FPGA	integrirano vezje, ki se lahko programira naknadno
LCD	zaslon iz tekočih kristalov
C	nizko nivojski programski jezik
Java	programski jezik za objektno programiranje
2D	dvodimenzionalnen pogled
3D	tridimenzionalen pogled
FTDI	<i>Future Technology Devices International</i>
TTL	tranzistorsko-tranzistorska logika (angl. <i>Transistor Transistor Logic</i>)
UART	univerzalni asinhroni sprejemnik – oddajnik (angl. <i>Universal Asynchronous Receiver Transmitter</i>)
PCB	tiskano vezje
INOX	nerjavno jeklo na osnovi železa
REED	reed rele, rele s hermetičnim kontaktnikom
UUID	univerzalno unikatni indentifikator
I/O	vhodno/izhodni

UPORABLJENI SIMBOLI

CO_2	ogljikov dioksid
V	volt, enota za merjenje električne napetosti
kHz	kilo hertz, enota za frekvenco
GHz	giga hertz, enota za frekvenco
MHz	mega hertz, enota za frekvenco
pH	merilo za koncentracijo hidroksidnih ionov v raztopini (kislost, alkalnost)
ω	frekvenca
k	konstanta vsebuje vse lastnosti merilnika pretoka
V	volumen
β	gostota magnetnega polja
Q	elementarni naboj
V	hitrost elektrona

1 UVOD

V diplomski nalogi smo se osredotočili na težavo manjših kleti ob prečrpavanju mošta ali vina. Poudariti velja, da je uporaba opreme za prečrpavanje mošta in vina minimalna skozi celotno leto, zato vlaganje v tehnološko naprednejše rešitve za majhne vinogradnike predstavlja velik finančni zalogaj. V primeru investicije, bi se le-ta povrnila v daljšem obdobju, česar pa si večina majhnih vinogradnikov ne more privoščiti. Ponekod so delovni prostori in prostori za hrambo vina majhni in razdeljeni na več lokacij. Ena oseba ne more nadzorovati procesa prečrpavanja in hkrati izvajati ostala potrebna opravila, ki jih moramo opraviti v prvi fazi spravila vina oziroma pridelka. V okviru diplomskega dela predstavimo avtomatizirano prečrpavanje mošta, ki bo vinarjem omogočilo enostavnejše prečrpavanje in nadzorovanje procesa.

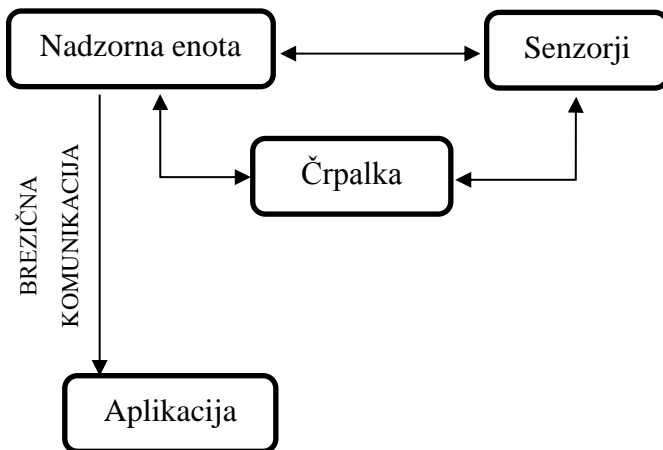
V večjih kletah imajo samodejne sisteme, ki se vodijo in nadzorujejo preko računalnikov, tabličnih računalnikov ali pametnih telefonov, s pomočjo aplikacij kot je na primer Smart Wine. S tem sistemom lahko nadzorujemo:

- zaporne ventile,
- pretok vina,
- črpalke,
- temperaturo v cisternah,
- pH vrednosti itd.

S telefonom imamo možnost nadzorovanja ohlajanja ali ogrevanja tankov v času fermentacije, možnost nadzorovanja mešanja mošta, ko se dodajajo kvasovke, možnost odpiranja ventilov za sproščanje CO_2 ali samo odpiranje ventilov v času polnjenja. Ti ventili in senzorji so fiksno nameščeni na cisterne, zbiralnike in črpalke [5].

Pri projektu smo naredili sistem nadzorovanja prečrpavanja mošta, ki bo dostopen tudi manjšim kletarjem, saj je cenovno ugoden in enostaven za uporabo. Uporaba tega manjšim kletarjem in drugim potencialnim uporabnikom ne bo predstavljala znatnega dodatnega stroška.

Osredotočili smo se na pretakanje mošta v času stiskanja. Celotno vezje smo najprej razbili na štiri bloke, kot prikazuje Slika 1.1.



Slika 1.1: Osnovna shema delovanja celotnega sklopa

(Vir: Lasten vir, 18. 6. 2016)

Mošt ob stiskanju grozdja izteče v zbirno posodo, ki jo je potrebno opremiti s primernimi stikali za nadzor nivoja. Črpalka za prečrpavanje mošta se vklopi in izklopi avtomatsko glede na nivo nivojskega ali pretočnega stikala, ki ga nadzira nadzorna enota. Preko nadzorne enote pridobivamo informacije o količini tekočine na telefon, kjer preko aplikacije spremljamo količino mošta v cisterni.

2 PREGLED TEHNOLOGIJ

2.1 Senzorji pretoka

Senzor je element, ki se uporablja kot pretvornik različnih veličin v električni izhodni signal kot rezultat merjenja. Poznamo različne vrste senzorjev, ki jih uporabljamo v raznovrstne namene. Pri našem prototipu smo uporabili nivojsko ali plovno stikalo in ultrazvočni senzor.

2.1.1 Nivojsko stikalo

Nivojska stikala s plovcem se uporabljajo za nadzor nivoja tekočin. Uporabljajo se za nizko in srednje viskozne materiale in za kontrolo minimalnih ter maksimalnih nivojev tekočine. Mi smo uporabili magnetno nivojsko stikalo, izdelano iz inox jekla z vgrajenimi stikali. V našem primeru imamo stikalo z dvema plovčema, kajti ta plovec je sestavni del novih stiskalnic, ki imajo vklop in izklop že vgrajene. V našem primeru smo uporabili samo spodnji plovec. Plovec deluje s pomočjo REED stikala. Ko se nivo tekočine dvigne, se plovec približa REED stikalu in ta povzroči preklon. V našem primeru vklop ali izklop črpalke [21].

2.1.2 Pretočno stikalo

Potreba po merjenju pretoka se pojavi vselej, ko se transportirajo tekočine (olje, mleko, vino, voda ipd.). Pretok v večini poteka v zaprtih ceveh, le redko po odprtih kanalih. Merilniki pretoka so naprave, ki merijo količino tekočine, plina ali hlapov, ki prehajajo skozi njih. Nekateri merijo pretok kot količino tekočine, ki poteka skozi merilnik pretoka v časovnem obdobju, drugi merilniki pa merijo skupno količino tekočine, ki je prešla skozi merilnik pretoka.

Meritve pretoka ne sodijo med najbolj zanesljive meritve, po podatkih našega pretočnega stikala prihaja do 3 % odstopanja.



Slika 2.1: Merilnik pretoka

Vir: [18]

Turbinski merilniki delujejo na principu vrtilne frekvence rotorja, ki je vgrajen v notranjosti merilnika. Vrtilna frekvenca je proporcionalna volumnskemu pretoku fluida.

$$V = k \times \omega \quad (1.1)$$

Tu je:

ω – frekvenca (s^{-1})

k – konstanta vsebuje vse lastnosti merilnika pretoka ($\frac{m^3}{s}$)

V – volumen (m^3)

Merilniki pretoka torej delujejo na principu Hallovega pojava, ki temelji na vplivu med gibljivimi nosilci naboja v prevodniku in z zunanjim magnetnim poljem. V primeru kovin so nosilci elektroni. Kadar se elektron giblje skozi magnetno polje, nanj deluje sila, ki je pravokotna na smer polja [7].

$$F = qv \times \beta \quad (1.2)$$

Tu je:

F – sila (N)

q – elementarni naboj (C)

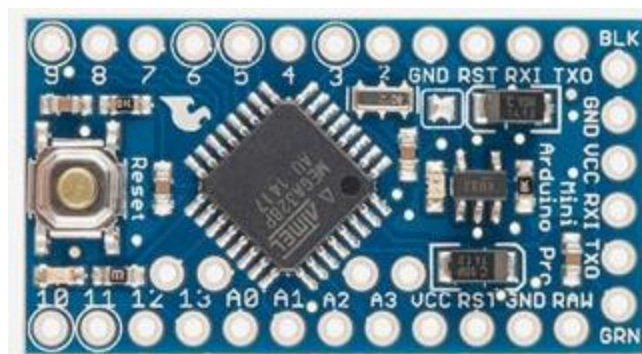
v – hitrost elektrona ($\frac{m}{s}$)

β – gostota magnetnega polja (T)

2.2 Mikrokrmilnik

2.2.1 Mikrokrmilnik

Projekt je zasnovan na Arduino razvojni ploščici, ki je enostavna za uporabo in zelo fleksibilna ter pogosto uporabljena platforma. Za krmilnik Arduino se uporablja odprtokodna platforma, ki je zasnovana na PCB plošči z mikrokrmilnikom in s posebnim čipom za serijsko komunikacijo FTDI TLL, ki poteka preko USB-ja. USB služi tudi za napajanje naprave. Mikrokrmilnik Arduino se programira v programu Arduino IDE, o katerem več v nadaljevanju. V našem primeru smo uporabili verzijo Arduino Pro Mini, kot prikazuje Slika 2.1, ki temelji na procesorju Atmel ATmega328. Ta procesor deluje na 3,3 V z 8 MHz ter 5 V z 16 Mhz. V našem primeru smo uporabili 5 V [9], [3], [27].



Slika 2.2: Mikrokrmilnik Arduino ProMini

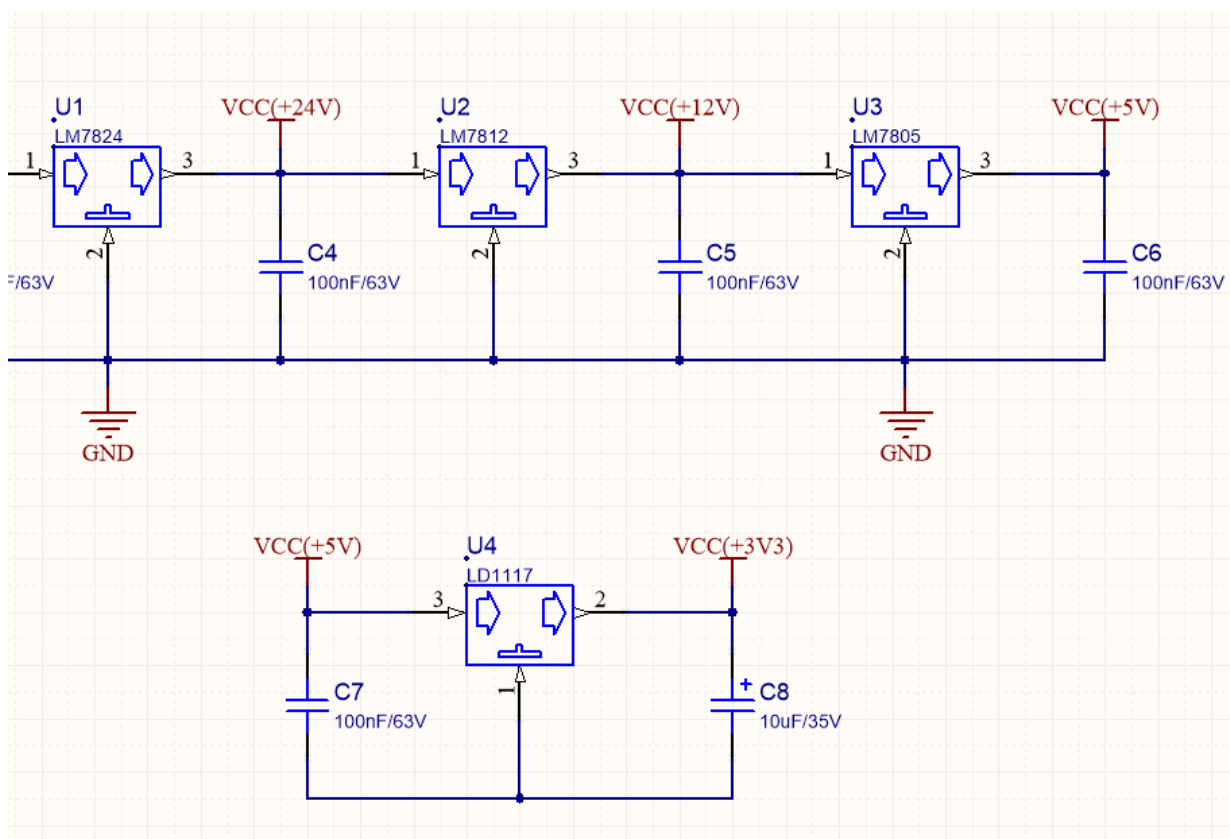
Vir: [2]

2.2.2 Napajanje

Celotni napajalni del smo narisali v Altium Designer, ki je programsko orodje za načrtovanje elektronskih vezij z dolgoletno tradicijo. Prvotno je bilo narejeno za načrtovanje PCB vezja, nakar so se začeli ukvarjati z oblikovanjem tiskanih vezij, FPGA in integriranih vezij [1].

Ta tiskanina zadostuje za naše trenutne in nadaljnje potrebe. Naredili smo štiri različne napetostne nivoje na tiskanini, in sicer 3,3 V za napajanje bluetooth modula, 5 V za napajanje mikrokrmilnika in pretočnega senzorja. Poleg omenjenega imamo še 12 V in 24 V napajanje, v primeru, če bomo sistem kdaj razširili (Slika 2.3).

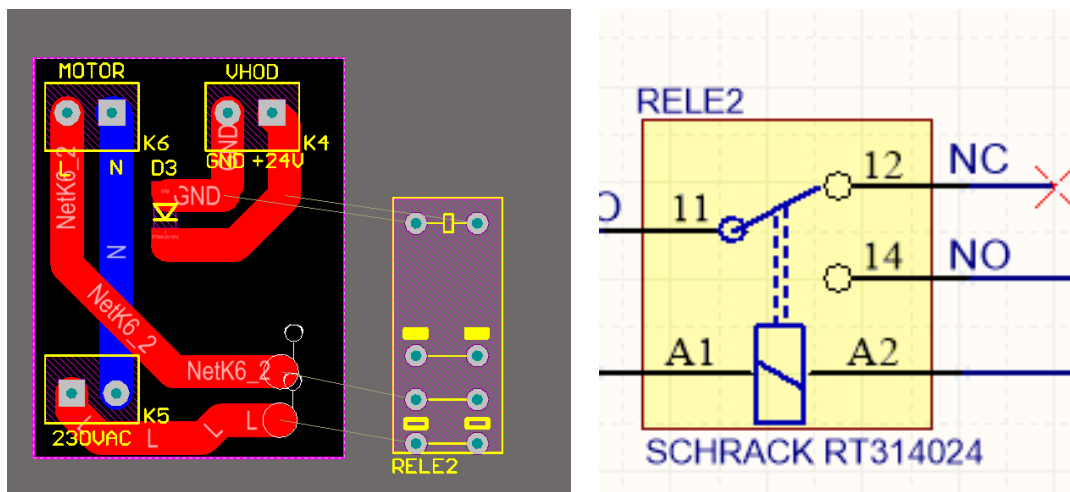
Celoten sistem se napaja preko nadzorne enote. Nadzorno enoto priključimo na 230 V, kjer preko napajalnika napajamo črpalko. Na izhodu je vtičnica, kamor se priklopi črpalka. Za napajalni kabel in vtičnico smo uporabili IEC C13 in IEC C14 po standardu IEC 60320 [26].



Slika 2.3: Shema napajalnega dela za elemente

Vir: Lasten vir, 7. 8. 2017

Pri risanju tiskanega vezja, smo si na računalniku morali ustvariti posebno mapo, kjer smo imeli shranjene podatke komponent, v primeru, da bomo kdaj odpirali projekt na drugem računalniku. Kot primer bomo prikazali zunanji rele (Slika 2.4). Ker za zunanji rele, na katerega sta vezana črpalka in plovno stikalo, ni bilo knjižnice, smo poiskali podobno, z enakimi dimenzijami, s priključki in z enakim številom nogic.



Slika 2.4: a) 2D pogled podnožja releja (PCB)

b) SCH slika releja

Vir: Lasten vir, 7. 8. 2017

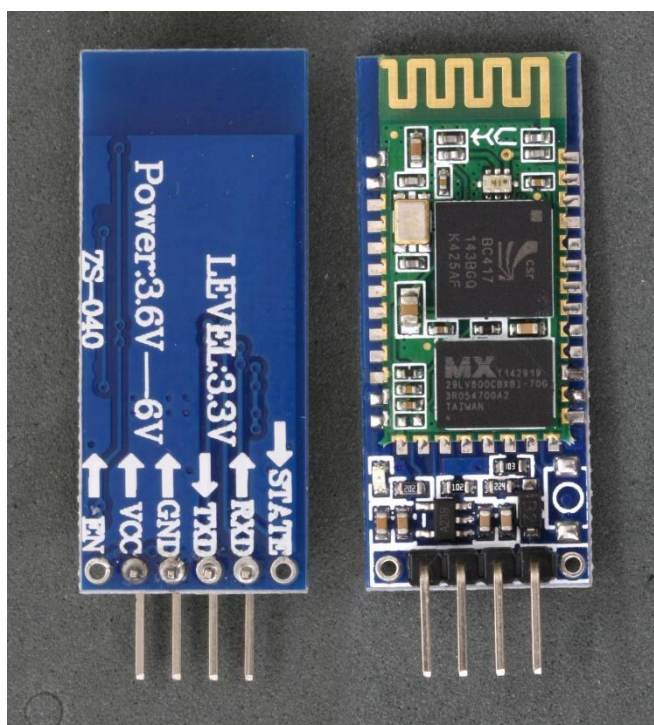
2.3 Brezžična komunikacija

Eden izmed naših ciljev je bil, da uporabniku nadzornega sistema olajšamo delo na najbolj preprost način. Po naši presoji je to uporaba komunikacije preko pametnega telefona. Hkrati uporabnik ne potrebuje nenehnega nadzora, saj nadzor poteka preko sensorja, katerega upravlja program. Ker imamo v kletnih prostorih veliko mokrih površin, kjer ne moremo dopuščati, da imamo povezavo z nadzorno enoto napeljana s kablom, in zaradi veliko višinskih razlik med nadzorno enoto in vrhom cisterne, smo se odločili, da za naše potrebe uporabimo tiskanino za brezžično komunikacijo, in sicer bluetooth tehnologijo, ki je sposobna sprejemati ukaze preko amplitudno-modulirane frekvence 27 MHz. Za daljinsko vodeno enoto smo se odločili za pametni telefon Samsung Galaxy S4 Active, ki je

tudi vodoodporen, saj imajo kletarji večino časa mokre roke in bi zaradi kontakta z vodo v nasprotnem primeru lahko prišlo do kakšnih napak. Naš pametni telefon vsebuje po meri narejeno aplikacijo, ki smo jo napisali posebej za ta projekt. Za bluetooth komunikacijo smo izbrali HC-06 (Slika 2.5).

Tehnične lastnosti Bluetooth modula (HC-06):

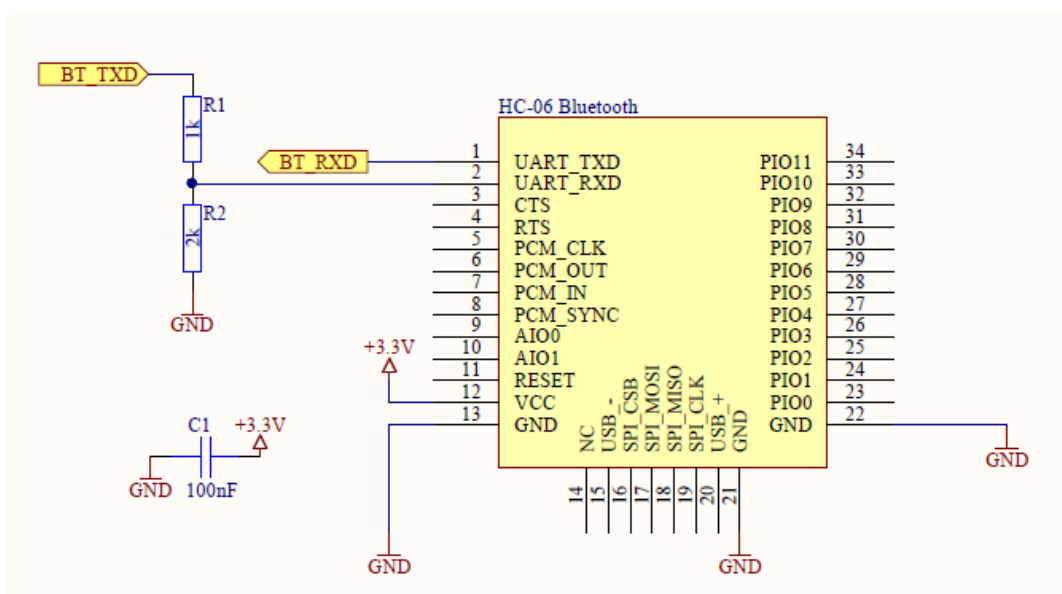
- napajanje 3,6 V–6 V,
- vgrajena 2,4 GHz antena,
- komunikacija skozi UART, SPI ali USB vrata,
- majhne dimenzije 31 × 15 × 2 mm,
- 8 Mbit bliskovni polnilnik [13].



Slika 2.5: Bluetooth komunikacijski modul HC-06

Vir: [13]

Brezžični modul mora od mikrokrmilnika »nadzorne enote« sprejeti po UART serijski povezavi ASCII znake, ki jih obdela in pošlje na pametni telefon in obratno. Na vezju imamo 3,3 V napajanje preko gladilnega kondenzatorja. Komunikacija mikrokrmilnik – brezžični modul poteka preko UART vrat, ta povezava pa poteka preko BT_TXD in BT_RXD. Pri vezavi je bilo treba paziti, saj mikrokrmilnik uporablja 5 V logični nivo, brezžični modul pa 3,3 V, vezava pa je obratna, torej je BT_RXD vezan na TX na mikrokrmilniku, BT_TXD pa na RX. Paziti smo morali, saj BT_RXD lahko sprejme 3,3 V logično nivo, BT_TXD pa lahko vežemo direktno na mikrokrmilnik. Zaradi tega smo naredili delilnik napetosti za BT_RXD linijo, in sicer z dvema uporoma, R1 in R2, v razmerju 1 : 2, kot je razvidno s Slike 2.6, neposredno med BT_RXD in UART_TXD. Tako smo dobili iz 5 V TTL nivo za 3,3 V.



Slika 2.6: Shema brezžičnega komunikacijskega modula HC-06

Vir: Lasten vir, 2. 9. 2017

2.4 Črpalke

Črpalke so delovni stroji za prečrpavanje drozge, mošta in vina. Ločimo več modelov:

- rotacijske črpalke zaradi vrtenja zobnika ustvarjajo vakum, ki sesa tekočino. Zagotavljajo enakomeren pretok;
- ekscentrične črpalke delujejo podobno kot rotacijske črpalke, s tem da imajo te ekscentrično nameščen rotor. Uporabljajo se predvsem za prečrpavanje gostejših materialov, kot so na primer latex, beton, drozga ...;
- polžaste črpalke delujejo tako, da vsebujejo dva ali tri polže, levega in desnega, ki se vrtijo nasproti in tako premikajo tekočino;
- centrifugalne črpalke delujejo tako, da tekočina prihaja aksialno v središče rotorja, nato se smer toka tekočine obrne radialno, v smer, v katero se vrti rotor;
- membranske črpalke delujejo po principu nihanja membrane. Zaradi tega se posledično spreminja prostornina, saj pri nihanju navzdol membrana ustvari podtlak in pri nihanju navzgor nadtlak;
- batne črpalke so najstarejše črpalke in delujejo na principu izpodrivanja tekočine. Zaradi tega prihaja do neenakomernega prečrpavanja tekočine [10], [20], [23], [24].

V našem primeru smo uporabili rotacijsko črpalco (Slika 2.7), ki deluje na osnovi rotacije rotorja. Le-ta je nameščen ekscentrično v ohišje. Za delovanje prototipa ni potrebna prava črpalca. Mi smo jo uporabili, ker imamo trenutno na razpolago samo te črpalke.



Slika 2.7: Uporabljena črpalka Rover pompe NOVAX M25

Vir: [17]

Zahteve za sodobne črpalke so:

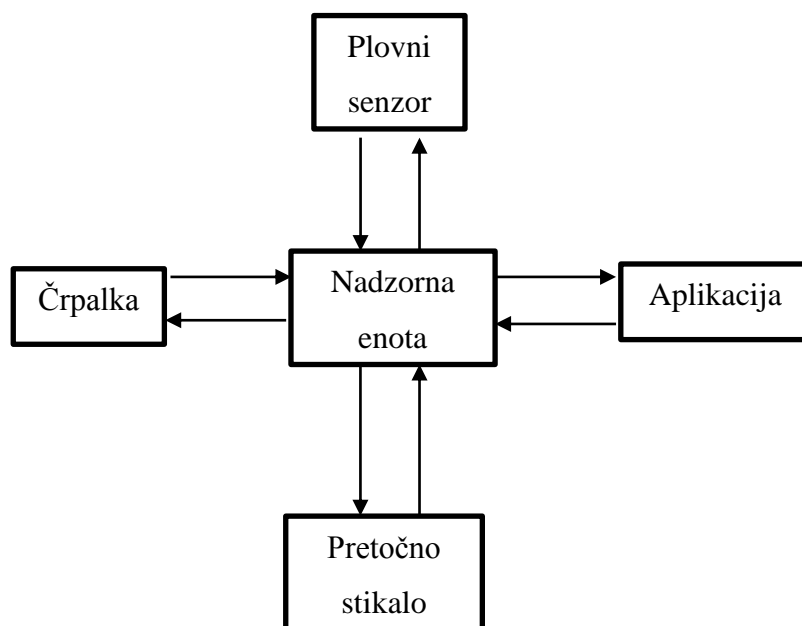
- enakomeren nesunkovit tek in minimalno penjenje vina,
- da dodatno ne poškodujemo drozge,
- uporaba odpornih in vzdržljivih materialov,
- da so enostavne za premikanje in upravljanje,
- da so neobčutljive na delovanje pri prostem teku in
- možnost regulacije pretoka [4].

3 ZASNOVA IN IZVEDBA SISTEMA

Za pravilno delovanje sistema je bilo potrebno razviti programsko opremo, ki bo pravilno komunicirala med nadzorno enoto in pametnim telefonom, vse pridobljene podatke s senzorja ali telefona pravilno obdelala in pravilno reagirala. V nadaljevanju bomo podrobno obravnavali razvojna okolja, ki smo jih uporabljali za pisanje programske kode.

3.1 Strojna oprema sistema

V kletah so vedno vlažna in mokra tla. Da bi sistem zavarovali pred neželjenimi vplivi, smo se odločili, da bomo celotno nadzorno enoto zaprli v elektro omarico in jo dvignili od tal z nogicami. Istočasno smo izpolnili cilj, da je elektro omarica kompatibilna za vse enofazne črpalke. Zaradi tega smo na elektro omarico dodali vtičnico, kjer se priklopi črpalka. Vsi priključki za senzorje so priklučeni s pomočjo konektorjev, tako da so lažje snemljivi. Za vtičnico za črpalko smo uporabili industrijsko s poklopцем, zato da ne more priti v stik z vlago (Slika 3.2). Za napajanje nadzorne enote smo uporabili, kot že omenjeno pri napajalnem delu, vtičnico in kabel, IEC C13 in IEC C14. Slika 3.1 predstavlja shematični prikaz načrta elektro omarice oziroma, kako bi naj potekala komunikacija znotraj elektro omarice.



Slika 3.1: Shema elektro omarice

Vir: Lasten vir, 25. 7. 2017

Ob iztekanju mošta v zbirno posodo, plovni senzor vklopi črpalko, ki začne prečrpavati. Pretočni senzor, ki je nameščen na spodnjem ventilu cisterne, preko nepovratnega ventila s pomočjo črpalke meri pretok vina oziroma mošta v cisterni in preko brezžičnega modula javlja direktno na aplikacijo, nameščeno na pametnem vmesniku. Preko aplikacije lahko nadzorujemo količino tekočine v cisterni ter vklop in izklop sistema, če je potrebno ročno upravljanje. Ko nivojsko stikalo pade nazaj na začetno pozicijo, se črpalka izklopi.

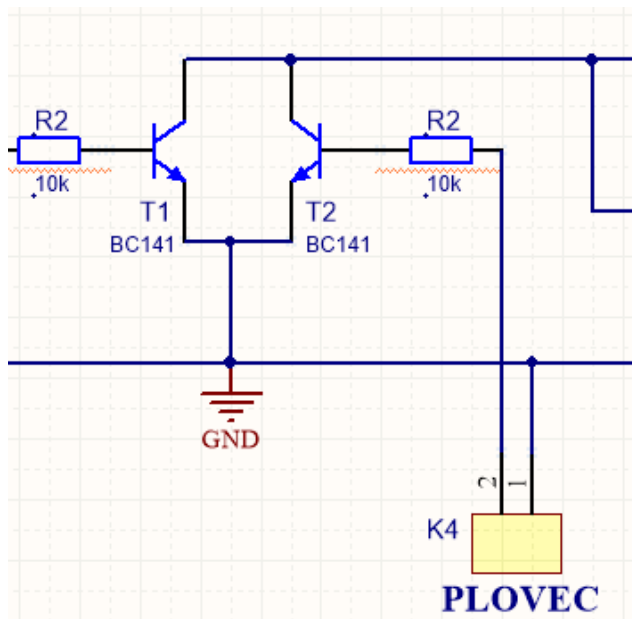
Imamo standardno elektro omarico. Na zgornjem delu elektro omarice smo namestili ročaj za lažji prenos in dva konektorja za priklop senzorjev. Na spodnjem delu smo dodali nogice, da se omarica ne odlaga direktno na tla. V notranjosti omarice smo pritrdili nosilec za napajalnik in dva releja. Napajalnik in releja smo fizično ločili z aluminijasto ploščo, v primeru da bi prihajalo do motenj zaradi preklopa releja, ko se inducira visoka napetost. Na zunanji steni smo pritrdili C14 vtičnico za napajanje, ki ima fizično stikalo za vklop in izklop, kot prikazuje Slika 3.2. K temu smo dodali industrijsko vtičnico s pokrivalom za vlažne prostore, da ne pride do stika z vlago.



Slika 3.2: Vtičnici in stikalo za vklop nadzorne enote

Vir: Lasten vir, 25. 8. 2017

Ker imamo plovni senzor, ki deluje preko REED stikala, smo dodali dva vzporedno vezana tranzistorja BC141, kot prikazuje Slika 3.3, da ustvarimo negirana (NE, angl. *NOT*) logična vrata za pravilno delovanje plovnega stikala. Možno je delovanje preko digitalnih izhodov, kot v našem primeru, kjer vklop in izklop nadzoruje program ali vezano direktno, kjer program nima vpliva na vklop in izklop črpalke.



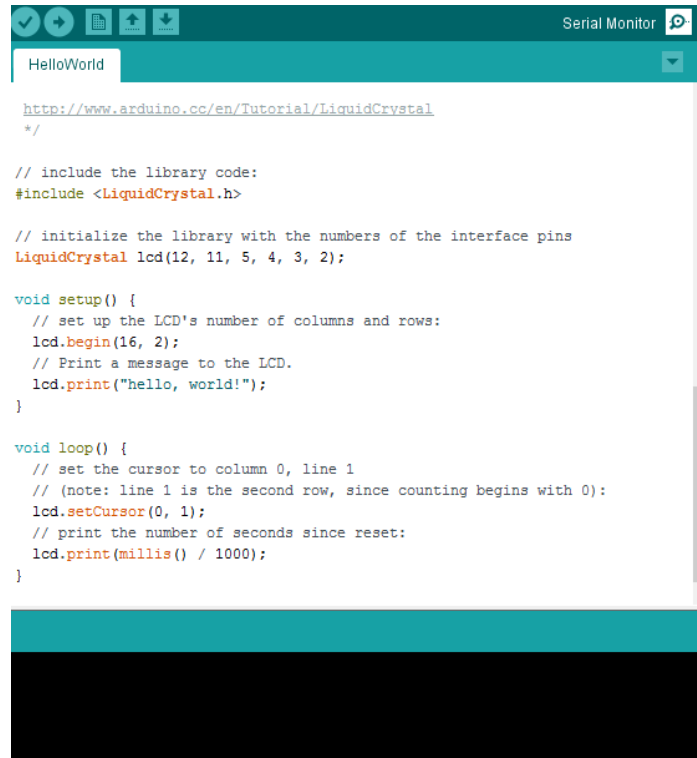
Slika 3.3: a) NE vrat s tranzistorji, b) Plovec, nameščen na krnico

Vir: Lasten vir, 29. 8. 2016

3.2 Razvojno okolje

3.2.1 Arduino razvojno okolje

Programsko kodo smo pisali v odprtokodni platformi v razvjonem okolju Arduino IDE, ki smo ga pisali v C programskem jeziku. Sestavljeno je iz tekstovnega urejevalnika za pisanje programske kode, zavihka za sporočila in orodne vrstice ter funkcije »*Serial monitor*«, ki je namenjena izpisovanju iz UART vrat mikrokrmilnika. Shranjevanje programa ali datoteke se imenuje skica. Nalaganje programov na krmilnik poteka preko USB kabla.



```
Serial Monitor

HelloWorld

http://www.arduino.cc/en/Tutorial/LiquidCrystal
*/

// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis() / 1000);
}
```

Slika 3.4: Primer osnovnega programa HelloWorld

Vir: Lasten vir, 25. 8. 2017

3.2.2 Android razvojno okolje

Android studio je uradno integrirano razvijalno okolje za razvoj Android aplikacij. Omogoča vizualno razvijanje uporabniškega vmesnika po principu povleci in spusti (ang. *drag and drop*). Aplikacija za telefon je bila pisana v programskem jeziku JAVA, ki se razlikuje od C programskega jezika. Uporablja se tudi XML, ta del kode lahko generiramo tudi z grafičnim vmesnikom. Program ima vgrajen tudi simulator »pametni telefon«, s katerim lahko simuliramo delovanje programa ali pa si program naložimo v živo na svoj telefon. Program vsebuje tudi izpis napak med testiranjem aplikacije. Ta funkcija je dobrodošla pri razhroščevanju.

V našem primeru smo uporabili Samsung Galaxy S4 Active z operacijskim sistemom 5.0. Komunikacija je potekala preko USB vrat. Najprej je bilo potrebno telefon pripraviti na sprejemanje kode, torej omogočiti razhroščevalni način.

3.3 Vgrajena programska oprema

Ker je celoten sistem vgrajen v elektro omarico in se nadzoruje preko aplikacije na pametnem telefonu, smo naredili aplikacijo s pomočjo programa Android Studio, kjer imamo preko telefona nadzor nad celotnim sistemom.

Za pravilno delovanje je potrebno realizirati kodo nadzorne enote, ki skrbi za pravilno branje in izpisovanje senzorskih podatkov. Najprej smo naredili diagram poteka, kot prikazuje Slika 3.6.

Na začetku programa je bilo treba definirati spremenljivke. S Slike 3.5 je razvidno, da je bilo potrebno definirati, na katerih pinih imamo priključen plovec, črpalko in pretočni senzor. Potrebno je bilo tudi nastaviti I/O (vhodno/izhodne) priključke in stanje priključkov na mikrokrmilniku. V nadaljevanju imamo *void setup()* funkcijo, ki se kliče samo enkrat in se uporablja za inicializacijo. Nato sledijo različni podprogrami, ki se uporabljajo samo takrat, ko so klicani. Zadnji del programa je *void loop()*, kjer se koda izvaja neskončno.

```
#define PUMP 4

#define FLOWSENSORPIN 2

#define PLOVEC 3

int stanjePlovca = 0;

int volNast = 500;

unsigned long volTrenuten = 0;

bool stanjeCrpalke = false;

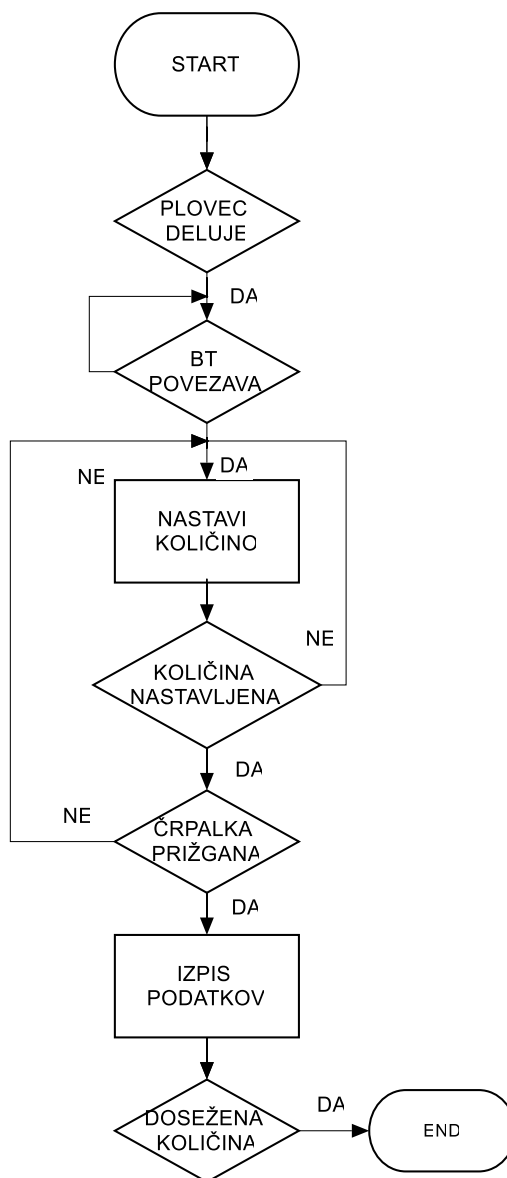
int forceStanje = 0;

bool prejsnjeStanje = false;
```

Slika 3.5: Definicije konstant in spremenljivk

Vir: Lasten vir, 10. 9. 2017

Za lažjo orientacijo, kako bi program potekal, smo izdelali diagram poteka (Slika 3.6), po katerem smo se ravnali v nadaljevanju pri pisanju programa.



Slika 3.6: Diagram poteka programa

Vir: Lasten vir, 25. 8. 2017

Najprej s konektorji priključimo zunanje stikalo in senzorje, kot sta plovec in senzor pretoka, na ohišje nadzorne enote. Preverimo, če plovec, ki deluje kot stikalo, vklopi oz. izklopi črpalko. Program gre nato v zanko in čaka, da se poveže z bluetooth napravo. Če je vse pravilno in se poveže, v nadaljevanju nastavimo željeno količino in pritisnemo vklop ali pustimo, da plovec sam avtomatsko vklopi sistem. Ko pridemo do željene količine, se sistem izklopi.

S pomočjo vodila lahko konstantno spreminjamo željeno količino (Slika 3.7) ali vklopimo in izklopimo sistem glede na nastavljen količino. Da ni potrebe po zasilnem izklopu, lahko samo zmanjšamo količino na vodilu in se pumpa izklopi ali obratno, povečamo in prečrpavanje se nadaljuje.

S pomočjo *Serial.readStringUntil()* preverjamo znake. S pomočjo prekinitvev konstantno preverjamo vsake 1–2 ms, če je bila pritisnjena kakšna tipka. Z *attachInterrupt()* naredimo prekinitvev, nato *Serial.readStringUntil()* preveri, če je bila pritisnjena kakšna tipka. Temu sledi *detachInterrupt()*, konec prekinitvev. Aplikacija pošlje željeno količino na mikrokontroler samo takrat, ko spustimo vodilo (ang. *Seekbar*) *onStartTrackingTouch()*, kot je prikazano na Sliki 3.8.



Slika 3.7: Prikaz nastavljanja željene količine

Vir: Lasten vir, 10. 9. 2017

```

seekbar.setOnSeekBarChangeListener(new
SeekBar.OnSeekBarChangeListener() {
    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {
        // TODO Auto-generated method stub
        setVolumen(seekBar.getProgress()); //ko spustis prst
        nastavi volumen
    }

    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onProgressChanged(SeekBar seekBar, int
progress, boolean fromUser) {
        Log.d("PROGRESS", valueOf(progress));
        number.setText(valueOf(progress));
    }
}

```

Slika 3.8: Del kode, kjer nastavljamo količino

Vir: Lasten vir, 10. 9. 2017

Pri izračunavanju, koliko tekočine je preteklo skozi pretočni senzor, si pomagamo s Hallovim pojavom in s pomočjo frekvence izračunamo količino. V našem primeru smo naredili števec, kot prikazuje Slika 3.9, ki šteje pulze. S pomočjo podatkov senzorja smo izračunali potrebne podatke, podrobneje je opisano pri meritvah.

```

void stevec() {
    pulses++;

    if (pulses == 264.0) {
        pulses = 0;
        volTrenuten++;
    }
}

void sendVolume() {
    Serial.print("Kolicina ");
    Serial.println(float(volTrenuten)+float(pulses/ 264.0));
}

```

Slika 3.9: Koda preprostega števca in koda za izpis količine tekočine

Vir: Lasten vir, 10. 9. 2017

3.4 Brezžična povezava

Povezava med mikrokrmilnikom in telefonom poteka preko *BluetoothServerSocket*, kot prikazuje Slika 3.10, kjer je del kode za povezavo. Telefon konstantno preverja nove povezave oziroma nove naprave, ki so v bližini. Ko le-ta napravo zazna, pošlje *BluetoothSocket*. Na drugi strani prav tako pošlje *BluetoothSocket*, da ohranja povezavo med napravami. Ko imamo enkrat povezavo in če hočemo pošiljati ter sprejemati podatke iz obeh strani, uporabimo *getInputStream()* in *getOutputStream*. Ko telefon pošlje zahtevo za »podatke«, ti podatki povedo nastavljeno količino, stanje plovca in črpalke. Podatek, ki zajema te podatke, predstavlja eno črko. V našem primeru je to velika tiskana črka P, ki se konča z »\n«.

```
if (!isBtConnected) {
    mBluetoothSocket =
    mBluetoothDevice.createInsecureRfcommSocketToServiceRecord(SPP_UUID);
    //create a RFCOMM (SPP) connection
    mBluetoothSocket.connect();
    ConnectSuccess = true;
}
```

Slika 3.10: Povezava bluetooth

Vir: Lasten vir, 10. 9. 2017

Mikrokrmilnik čaka tako dolgo, dokler ni pritisnjen gumb za vklop, oziroma dokler se plovec ne dvigne, torej se pripravi v stanje za vklop (Slika 3.11).

```
if (Serial.available() > 0) {
    detachInterrupt(digitalPinToInterrupt(FLOWSENSORPIN));
    // read the incoming byte:
    String stri = Serial.readStringUntil('\n');
    if (doCommand(stri) != 0) {
        Serial.println("ERROR#");
    }
}
```

Slika 3.11: Čakanje in branje

Vir: Lasten vir, 10. 9. 2017

Na Sliki 3.12 imamo prikazan del kode za izpis podatkov, ko v aplikaciji pritisnemo tipko »podatki«. Vsi ti podatki so podprogrami, ki jih lahko kliče podprogram »sendData«, ki nato vse te podatke posreduje na telefon.

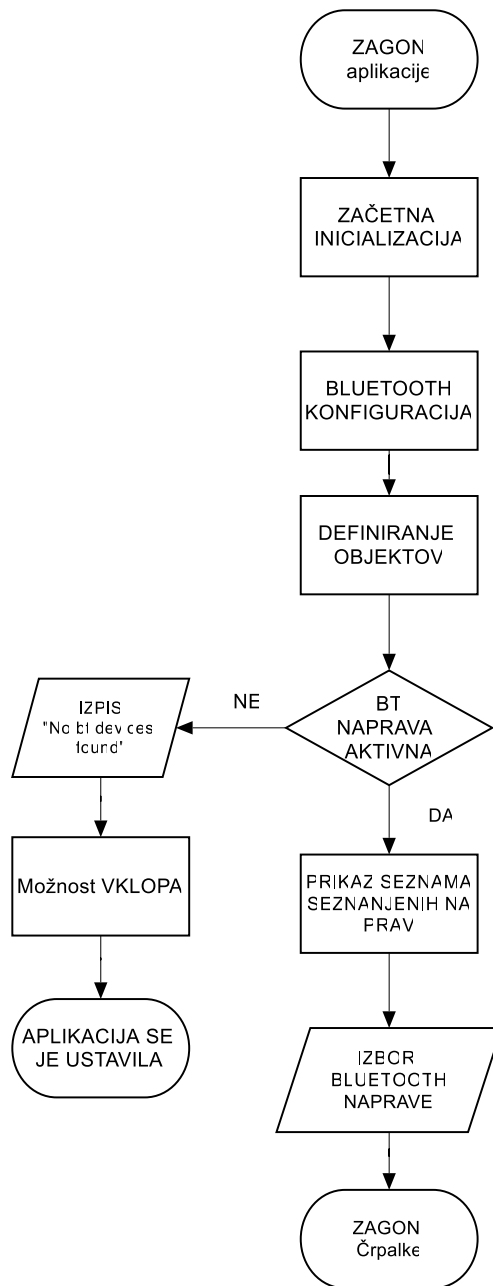
```
void sendData() {  
    sendPump();  
    sendPlovec();  
    sendVolume();  
    sendSetVolume();  
    sendForceStanje();  
    Serial.print("#");  
}
```

Slika 3.12: Podatki, ki jih prejme telefon

Vir: Lasten vir, 10. 9. 2017

3.5 Mobilna aplikacija

Pri izdelavi aplikacije za telefon je bil naš cilj, da bo aplikacija čim bolj preprosta in prijazna za uporabo tudi starejšim uporabnikom. Najprej smo sestavili diagram poteka, Slika 3.13, po katerem smo se orientirali med pisanjem programa. Program Vinoduno je sestavljen iz več datotek, ki so potrebne za pravilno delovanje celotnega sistema. V prvi fazi smo napisali dovoljenje v `AndroidManifest.xml` za uporabo bluetooth naprav v telefonu [6].

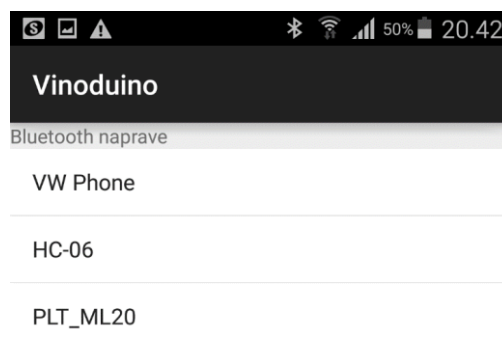


Slika 3.13: Diagram poteka za Bluetooth povezavo

Vir: Lasten vir, 25. 8. 2017

Temu je sledil izgled aplikacije. V prvi fazi je activity_list.xml, ki je potreben za izgled zaslona pri izboru naprav, s katero se povežemo (Slika 3.14), nato pa še sledi activity_main.xml, ki je namenjen za izgled glavnega dela programa (Slika 3.15), kjer imamo vse potrebne tipke in drsnike za delovanje nadzorne enote.

Za delovanje celotnega sistema ima pomembno vlogo tudi MainActivity.java, kjer se nahaja koda za delovanje aplikacije in BTListActivity.java, ki je potrebna za zagon aplikacije. Na začetku poteka inicializacija, ki je sestavljena iz niza uvoženih knjižnic in paketov. Ko se bluetooth konfigurira, dobi poseben UUID naslov, ki se uporabi v nadaljevanju aplikacije. Ko se aplikacija zažene, nas le-ta vpraša, ali je bluetooth naprava fizično vklopljena na telefonu, oziroma če ni ali jo lahko vklopi. Aplikacija nato odpre seznam naprav, s katerim je telefon seznanjen. Če je Bluetooth modul HC-06 na seznamu, ga izberemo, kot prikazuje Slika 3.14. To je bluetooth nadzorna enota in po kliku nas prenese v naslednjo aktivnost MainActivity.java, ki skrbi za celoten program.



Slika 3.14: Prikaz seznama seznanjenih naprav

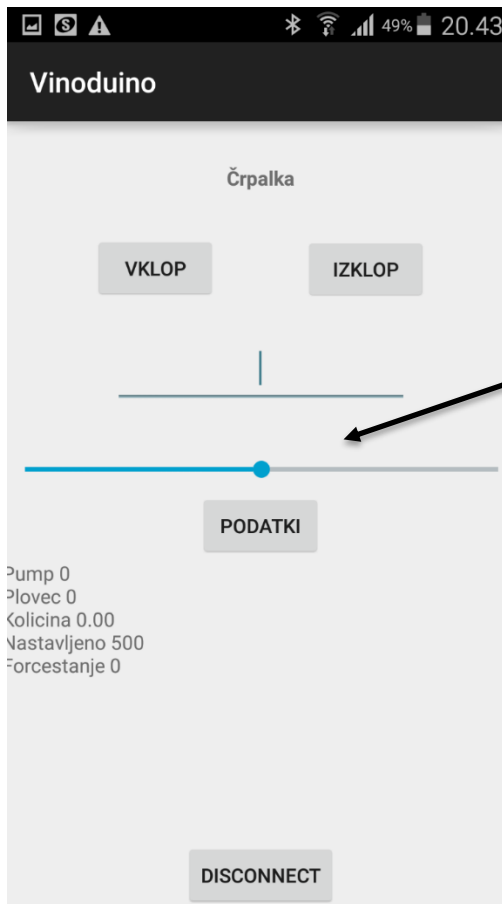
Vir: Lasten vir, 12. 9. 2017



Slika 3.15: Izpis, če nadzorna enota ni aktivna

Vir: Lasten vir, 12. 9. 2017

Ko izberemo Bluetooth modul HC-06 in ko nas ta ponese v naslednjo aktivnost MainActivity.java, kot prikazuje Slika 3.16, se začne uvoz knjižnic, inicializacija paketov ter definiranje objektov in spremenljivk, ki jih potrebujemo. Za bluetooth konfiguracijo, kot že omenjeno, dobimo univerzalno unikatni indentifikator »UUID«, katerega potrebujemo za pravilno delovanje. Da lahko pošiljamo in sprejemamo podatke, ustvarimo povezavo med napravami s pomočjo BluetoothSocket, ki skrbi za konstatno povezavo. Aplikacija konstantno preverja in čaka v glavni zanki, če je bil pritisnjen kakšen gumb in če mora poslati ali sprejeti kakšen podatek.



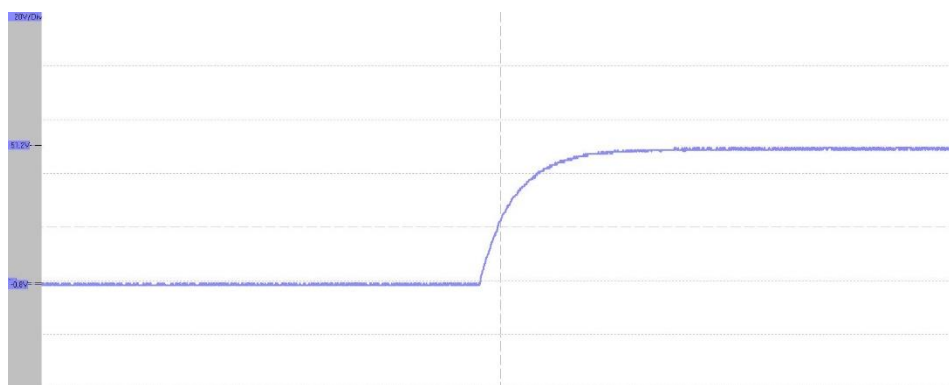
Dršnik za nastavljanje količine.

Slika 3.16: Zgled MainActivity.java

Vir: Lasten vir, 12. 9. 2017

4 MERITVE IN REZULTATI

Pri meritvah smo z osiloskopom izmerili začetek vklopa aplikacije oziroma prečrpavanja. Na Sliki 4.1 je razvidno, kako počasi narašča signal pretočnega senzorja, ko črpalka ni v polnem obratovanju. Nato smo zajeli celotno sliko, ko pretočni senzor obratuje normalno brez zapletov že nekaj časa (Slika 4.2).



Slika 4.1: Prikaz naraščanja frekvence

Vir: Lasten vir, 19. 9. 2017

Prve meritve smo naredili tako, da smo si pripravili dve merilni posodi, ki sta označeni z merilno skalo. Nato smo vzeli štoparico in ko je začelo prečrpavati, smo začeli štopati čas. Merili smo čas pri 1 minuti, 2 minutah in 3 minutah. Naredili smo daljše preizkuse, v primeru, če bi se pri eni minuti poznal sam zagon črpalke. Črpalko smo imeli polno, tako da smo s tem izničili izgube črpalke.

Tako smo prišli do podatkov, s katerimi smo v nadaljevanju izračunali potrebno frekvenco, ki jo potrebujemo za pretočni senzor. Za ta test smo uporabili črpalko Rover pompe NOVAX M25, ki ima pretok pri normalnem delovanju črpalke od 7 do 45 l/min. Rezultati so prikazani v Tabeli 4.1.

Tabela 4.1.: Rezultati prečrpavanja z Rover črpalko

Čas	Izmerjen čas	Izmerjena količina
1 minuta	1:00,11	24 litrov
2 minuti	2:00,36	48 litrov
3 minute	3:00,24	72 litrov

Tako smo prišli do rezultata, da črpalka na polno obratuje samo s 24 litri na minuto. S tem podatkom smo izračunali frekvenco, ki jo potrebuje senzor za izračun. Enačbo imamo podano pri proizvajalcu.

$$Hz = 5,5 \times Q \pm 3 \% \quad (4.1)$$

$$Hz = 5,5 \times 24l / \text{min} = 132 \text{ Hz} \pm 3 \%$$

Tu je:

Q – pretok tekočine (l/min)

Hz – frekvenca (Hz)

V našem primeru znaša pretok 24 litrov na minuto, torej frekvenca, ki jo potrebujemo, je 264 Hz, saj je izračun podan samo za pol periode.

Zatem smo naredili preizkus prečrpavanja pri 1 minuti, pri 2 minutah in pri 3 minutah, kjer smo ponovno vzeli merilno posodo s 100 litri in črpalko ter zagnali aplikacijo, da je ta prečrpavala. Istočasno smo konstantno primerjali čas. Kot je razvidno iz Tabel 4.1 in 4.2, se količina pri 1 minuti zelo razlikuje, čeprav glede prikazane količine ne toliko. Nato smo naredili meritve še za ostali čas in primerjali s prikazano količino s fizično količino v merilni posodi.

Tabela 4.2: Rezultati Rover črpalke med aplikacijo in izmerjeno količino

Čas	Prikazana količina	Izmerjen čas	Izmerjena količina
1 minuta	15 litrov	0:59,96	14,7 litra
2 minuti	48 litrov	2:00,29	48,4 litra
3 minute	70 litrov	2:59,93	71,8 litra

Kot je razvidno iz Tabele 4.2, so se rezultati razlikovali za toliko, za kolikor nam proizvajalec sam poda informacije o nihanju rezultatov, torej za +/- 3 %. Pri Rover črpalci ni prihajalo do tako velikih razlik, saj smo program umerjali glede na njene podatke in rezultate.

Medtem ko smo dobili podatek, da v eni minuti preteče približno 24,5 litrov, smo s pomočjo osiloskopa zajeli celotni spekter, ki je dolg 1 minuto. Na Sliki 4.2 je viden del spektra. Nato smo prešteli puzle, da smo lahko izračunali, koliko je ena perioda litrov in koliko period je en liter. Prešteli smo 90 period. Zatem smo izračunali, da ena perioda znaša 0,27 litra, kar pomeni, da za 1 liter potrebujemo približno 4 periode.

$$\frac{Q}{X} = \frac{24,5 \text{ l}}{90 \text{ s}^{-1}} = 0,27 \text{ l/periodo} \quad (4.2)$$

Tu je:

Q – pretok količine (l/min)

X – periodni interval (period /min)



Slika 4.2: Del spektra pri merjenju

Vir: Lasten vir, 19. 9. 2017

S Slike 4.2 je razvidno, da najvišja napetost oziroma amplituda znaša 51,2 V. Frekvenca, ki je prikazana, ni konstatna, ampak niha. Zato smo zapisali več vrednosti frekvenc in nato izračunali aritmetično sredino, ki znaša 107,6 Hz.

Nato smo naredili enak preizkus z drugo črpalko, ki ima podobne lastnosti, le da podatki zaradi njene starosti niso poznani. S tem preizkusom smo hoteli izvedeti, če je možno uporabljati različne enofazne črpalke brez velikih sprememb, kot so na novo preračunavanje pretoka in frekvence ter na novo vstavljene frekvence v program.

Kot je razvidno iz Tabele 4.3, se rezultati niso veliko spreminjali. Še vedno so v območju +/- 3 %, kot podaja proizvajalec.

Tabela 4.3: Rezultati neznane črpalke med aplikacijo in izmerjeno količino

Čas	Prikazana količina	Izmerjeni čas	Izmerjena količina
1 minuta	23 litrov	1:00,96	24,2 litra
2 minuti	45 litrov	2:00,07	46,3 litra
3 minute	69 litrov	3:00,17	69,7 litra

Tudi pri večjih količinah ne prihaja do velikih odstopanj, kar nam ustreza pri našem projektu. Ker že cisterne, na katerih se bo ta projekt uporabljal, niso točne. V velikem primeru so vedno večje od 50 do 100 litrov, kar pa naše odstopanje ne bi smelo motiti toliko, da bi prihajalo do izlivov.

Pri testiranju sistema pri 100 litrih in pri 200 litrih, tako da se sistem samodejno izklopi, je prišlo do odstopanja, rezultati so prikazani v Tabeli 4.4.

Tabela 4.4: Rezultati samodejnega pretoka

Nastavljena količina	Izmerjena količina
101 liter	101,5 litra
200 litrov	200,8 litra

Pri projektu smo si med drugim zadali cilj, da smo mobilni, kar pomeni, da lahko gremo tudi v sosednjo klet in nam ni treba biti ves čas v bližini naprave. Test je pokazal, da smo lahko od naprave oddaljeni 35 metrov in še vedno deluje. Prav tako je test pokazal, da lahko gremo v sosednjo klet, ki je oddaljena le 8 metrov, a je vmes 40 cm debela stena.

Naš cilj je bil tudi, da aplikaciji nastavimo količino tekočine, ki mora preteči, nato pa se mora sistem avtomatsko izklopiti. Naredili smo test, kjer smo aplikaciji izdali ukaz, da naj preteče 70 litrov in se nato izklopi. V našem primeru je po informacijah v aplikaciji steklo 70,2 litra,

ko pa smo izmerili dejansko tekočino, ki je pretekla, je ta znašala 69,5 litra. Takšno odstopanje v našem primeru zadošča.

Če bi želeli veliko natančnost, bi morali imeti točno umerjene posode, tako pa so označene z oznakami 100 l, 200 l. Ko smo preverili točnost oznak, torej če je na umerjeni posodi oznaka 100 litrov, če je v posodi dejansko tudi 100 litrov, smo ugotovili, da vrednosti nihajo. Tudi naše meritve niso bile popolnoma natančne, varirale so +/- en liter. Testiranje točnosti oznak na umerjenih posodah pa je pokazalo variiranje od 10 do 15 litrov.

Težave se pojavljajo tudi pri ceveh, saj ima vsaka črpalka svoje cevi. Pri nekaterih so daljše, pri nekaterih krajše, tako da že tukaj lahko prihaja do majhnega odstopanja od željene vrednosti. Veliko tekočine lahko tudi zastane v ceveh.

Imamo namreč različne dolžine cevi od 5 metrov pa do 12 metrov in 3/4" colske cevi (26,9 mm).

$$V = \pi r^2 h \quad (4.3)$$

Tu je:

V – volumen (m^3)

r – polmer cevi (mm)

h – dolžina cevi (m)

π – pi

Primer samo, če imamo skupaj 10 metrov cevi, nam v njih po Enačbi 4.3 ostane 5,6 litrov, zaradi tega lahko prihaja do odstopanj. Tudi, če smo pred začetkom napolnili cevi in črpalko, da bi izničili te izgube.

Že pri samem začetku merjenja, smo vsako črpalko napolnili s tekočino, tako da ko smo začeli z meritvijo, se črpalka ni vrtela v prazno. Do odstopanja pri črpalkah prihaja tudi zaradi količine pretoka.

Delovanje ustreza našim trenutnim zahtevam in olajša delo v kletih. Zavedamo se, da tudi pri cisternah prihaja do odstopanj od vrednosti količine, ki je napisana na njej. Po vsaki meritvi, ko smo izpopolnili program in smo izmerili pretočeno količino, je količina vedno varirala za +/- en liter, ne glede na to, ali je šlo za 50 litrov ali pa za 200 litrov. Kot je bilo že omenjeno, v našem primeru natančnih meritev ne moremo izvesti, ampak za naše zahteve in potrebe sistem deluje dovolj dobro.

5 PROTOTIP

Kako izgleda, ko je črpalka priključena na nadzorno enoto, je razvidno s Slike 5.1, kjer je prikaz sistema v času delovanja. Na Sliki 5.2 je razvidno, kako je nameščen plovec na krnico stiskalnice, kamor izteka mošt. Nosilec za senzor smo morali narediti sami.

Na Sliki 5.3 imamo pretočni senzor nameščen na cisterno preko nepovratnega ventila, da tekočina ne izteka nazaj. Pri tem je lepo razvidna smer, v katero mora biti obrnjen senzor.



Slika 5.1: Črpalka, priključena na nadzorno enoto

Vir: Lasten vir, 12. 9. 2017



Slika 5.1: Plovec, nameščen na krnico

Vir: Lasten vir, 12. 9. 2017



Slika 5.3: Senzor pretoka, nameščen na cisterno

Vir: Lasten vir, 12. 9. 2017

6 SKLEP

Pri diplomski nalogi smo si zadali cilj vinarjem olajšati delo pri prečrpavanju tekočine ter narediti enostavno in varno nadzorno enoto, ki je kompatibilna za vse enosmerne črpalke ter preprosta za uporabo. Uspelo nam je realizirati programski in mehanski del in odpraviti programske ter tehnične napake, na katere smo naleteli med izdelavo prototipa.

Projekt bi lahko nadgradili tako, da v času stiskanja, ko tekočina oz. mošt steka v zbiralnik, dvigne plovec in s tem vklopi črpalko. V zbiralniku je nivojsko stikalo, ki preko nadzorne enote pošlje signal o vklopu in izklopu črpalke, ki črpa v cisterno, kjer je nameščen pretočni senzor. Projekt bi lahko dodelali tako, da v primeru presežene nastavljene količine pretočnega senzorja, le-ta izklopi črpalko, hkrati pa mora program poslati signal o izklopu stiskalnici in jo postaviti v stanje mirovanja, da tekočina iz zbiralnika ne začne iztekati preko roba. K temu bi morali zaradi varnosti, v primeru, da se zruši sistem, narediti varnostno stikalo. Tega bi namestili na vrhu cisterne, kjer je odprtina za dolivanje. Kot varnostno stikalo bi namestili plovec ali kapacitivni senzor in ga vezali direktno na rele črpalke. Ko bi kapacitivni senzor prišel v stik s tekočino, oziroma ko bi plovec dvignilo, bi se sistem avtomatsko izklopil.

7 VIRI IN LITERATURA

- [1] Altium. Dostopno na: <http://www.altium.com/altium-designer/overview> [28. 7. 2017].
- [2] Arduino. Dostopno na: <https://www.arduino.cc/en/Guide/ArduinoProMini> [4. 7. 2017].
- [3] Arduino. Dostopno na: <https://www.arduino.cc/en/Main/ArduinoBoardProMini> [10. 7. 2017].
- [4] Bavčar, D. *Kletarjenje danes*. Ljubljana: Kmečki glas, 2006.
- [5] Burkert Contromatic Corporation. Dostopno na: <http://www.burkert-usa.com/en/Media/Files/Segments-Applications/Application-Flyers/Smart-Winery-Solutions-Brochure> [27. 7. 2017].
- [6] Developers. Dostopno na: <https://developer.android.com/reference/android/bluetooth/BluetoothSocket.html> [3. 8. 2017].
- [7] Đonlagić, D., *Senzorski sistemi, Zapiski predavanj*. Maribor: Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru, 1999.
- [8] Evans, B.W. *Arduino programming notebook*. San Francisco, California: Creative Commons, 2007. Dostopno na: http://playground.arduino.cc/uploads/Main/arduino_notebook_v1-1.pdf [24. 3. 2016].
- [9] Hrastnik, G. *Alternativna rešitev za nadzor multikopterjev*. Maribor: Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru, 2015.
- [10] Janez Liponik. Dostopno na: <http://janez.liponik.si/files/gradiva-za-predavanje/strojniki-2011/crpalke.pdf> [27. 7. 2017].
- [11] Krkoč, P. *Arduino [na] TM: programirajmo Arduino [na] TM z lahkoto*. Ljubljana: AX elektronika, 2014.
- [12] Mandeljc, P. *Vodenje in nadzor naprav z mikrokrmilnikom Arduino*. Ljubljana: Fakulteta za elektrotehniko, Univerza v Ljubljani, 2016.

- [13] Martyn Currey. Dostopno na: <http://www.martyncurrey.com/arduino-and-hc-06-zs-040/> [10. 8. 2017].
- [14] Monk, S. *Arduino + Android projects for the evil genius: control Arduino with your smartphone or tablet*: New York: McGraw-Hill, cop, 2012.
- [15] Purdum, J. *Beginning C for Arduino: learn C programming for the Arduino and compatible microcontrollers*: New York: Apress: Springer Science+Business Media, cop.,2012.
- [16] Purdum, J., Ph. D. *Beginning C for Arduino*, 2nd Edition: New York: Apress, 2015.
- [17] Rover pompe. Dostopno na: <http://www.roverpompe.com/it/prodotti/elettropompe-novax-/novax-m-230v-50hz/novax-m-25> [10. 9. 2017].
- [18] Seeed. Dostopno na: http://wiki.seeed.cc/G3-4_Water_Flow_sensor/ [7. 7. 2017].
- [19] Seeed, Grow the Difference. 2014. Dostopno na: <http://www.seeedstudio.com/forum/> [24. 3. 2016].
- [20] Študentski svet fakultete za strojništvo. Dostopno na: <http://www.ssfs.si/download/Energetski%20stroji/crpalke.pdf> [27. 7. 2017].
- [21] Tevel. Dostopno na: <http://www.tevel.si/Portals/0/Dokumenti/503%20navodila%20za%20uporabo%20NS1-slovensko.pdf> [15. 8. 2017].
- [22] Warren, J.D., Adams, J., Molle, H. *Arduino Robotics*: New York: Apress, 2011.
- [23] Wikipedija, Altium designer. Dostopno na: https://en.wikipedia.org/wiki/Altium_Designer [28. 7. 2017].
- [24] Wikipedija, Črpalka. Dostopno na: <https://sl.wikipedia.org/wiki/%C4%8Crpalka> [27. 7. 2017].
- [25] Wikipedija, Črpalka. Dostopno na: <https://en.wikipedia.org/wiki/Pump> [27. 7. 2017].
- [26] Wikipedija, IEC 60320. Dostopno na: https://en.wikipedia.org/wiki/IEC_60320 [20. 7. 2017].



Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko

Koroška cesta 46
2000 Maribor, Slovenija



IZJAVA O USTREZNOSTI ZAKLJUČNEGA DELA

Podpisani mentor/-ica : *doc. dr. Iztok Kramberger*
(ime in priimek mentor-ja/-ice)

in somentor/-ica (eden ali več, če obstajajo): _____
(ime in priimek somentor-ja/-ice)

Izjavlja-m/-va/-mo, da je študent/-ka

Ime in priimek: Miha Založnik, ID številka: 1001982398
vpisna številka: E1040630, na študijskem programu: VS-Elektrotehnika

izdelal/-a zaključno delo z naslovom:

AVTOMATIZIRANA PRETOČNA ENOTA Z NADZOROM PREKO PAMETNEGA
TELEFONA _____

(naslov zaključnega dela v slovenskem jeziku)

v skladu z odobreno temo zaključnega dela, navodili o pripravi zaključnih del in mojimi (najinimi/našimi) navodili.

Preveril/-a/-i in pregledal/-a/-i sem/sva/smo poročilo o preverjanju podobnosti vsebin z drugimi deli (priloga) in potrujem/potrujeva/potrjujemo, da je zaključno delo ustrezno.

Datum in kraj: Maribor, 26.9.2017

Podpis mentor-ja/-ice:

Datum in kraj:

Podpis somentor-ja/-ice (če obstaja):

Priloga:

- Poročilo o preverjanju podobnosti vsebin z drugimi deli.



Univerza v Mariboru



Fakulteta za elektrotehniko,
računalništvo in informatiko

IZJAVA O AVTORSTVU IN ISTOVETNOSTI TISKANE IN ELEKTRONSKE OBLIKE ZAKLJUČNEGA DELA

Ime in priimek študent-a/-ke: Miha Založnik

Študijski program: ELEKTROTEHNIKA

Naslov zaključnega dela: AVTOMATIZIRANA PRETOČNA ENOTA Z NADZOROM PREKO PAMETNEGA TELEFONA

Mentor: doc. dr. Iztok Kramberger

Somentor: _____

Podpisan-i/-a študent/-ka Miha Založnik

- izjavljam, da je zaključno delo rezultat mojega samostojnega dela, ki sem ga izdelal/-a ob pomoči mentor-ja/-ice oz. somentor-ja/-ice;
- izjavljam, da sem pridobil/-a vsa potrebna soglasja za uporabo podatkov in avtorskih del v zaključnem delu in jih v zaključnem delu jasno in ustrezno označil/-a;
- na Univerzo v Mariboru neodplačno, neizključno, prostorsko in časovno neomejeno prenašam pravico shranitve avtorskega dela v elektronski obliki, pravico reproduciranja ter pravico ponuditi zaključno delo javnosti na svetovnem spletu preko DKUM; sem seznanjen/-a, da bodo dela deponirana/objavljena v DKUM dostopna široki javnosti pod pogoji licence Creative Commons BY-NC-ND, kar vključuje tudi avtomatizirano indeksiranje preko spleta in obdelavo besedil za potrebe tekstovnega in podatkovnega rudarjenja in ekstrakcije znanja iz vsebin; uporabnikom se dovoli reproduciranje brez predelave avtorskega dela, distribuiranje, dajanje v najem in priobčitev javnosti samega izvirnega avtorskega dela, in sicer pod pogojem, da navedejo avtorja in da ne gre za komercialno uporabo;
- dovoljujem objavo svojih osebnih podatkov, ki so navedeni v zaključnem delu in tej izjavi, skupaj z objavo zaključnega dela;
- izjavljam, da je tiskana oblika zaključnega dela istovetna elektronski obliki zaključnega dela, ki sem jo oddal/-a za objavo v DKUM.

Uveljavljam permisivnejšo obliko licence Creative Commons: _____ (navedite obliko)

Začasna nedostopnost:

Zaključno delo zaradi zagotavljanja konkurenčne prednosti, zaščite poslovnih skrivnosti, varnosti ljudi in narave, varstva industrijske lastnine ali tajnosti podatkov naročnika:

_____ (naziv in naslov
naročnika/institucije) ne sme biti javno dostopno do _____ (datum odloga
javne objave ne sme biti daljši kot 3 leta od zagovora dela). To se nanaša na tiskano in elektronsko obliko zaključnega dela.

Temporary unavailability:

To ensure competition priority, protection of trade secrets, safety of people and nature, protection of industrial property or secrecy of customer's information, the thesis _____ (institution/company name and address) must not be accessible to the public till _____ (delay date of thesis availability to the public must not exceed the period of 3 years after thesis defense). This applies to printed and electronic thesis forms.

Datum in kraj: 26.9.2017, Maribor

Podpis študent-a/-ke:



Podpis mentor-ja/-ice: _____
(samo v primeru, če delo ne me biti javno dostopno)

Ime in priimek ter podpis odgovorne osebe naročnika in žig:

(samo v primeru, če delo ne sme biti javno dostopno)