Automatizirani sustav upravljanja pumpama za navodnjavanje

Kukrika, Vedran

Undergraduate thesis / Završni rad

2017

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

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:459505

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-21**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
ELEKTROTEHNIČKI FAKULTET

Stručni studij

AUTOMATIZIRANI SUSTAV UPRAVLJANJA PUMPAMA ZA NAVODNJAVANJE

Završni rad

Vedran Kukrika

Osijek, 2017

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1S: Obrazac za imenovanje Povjerenstva za obranu završnog rada na preddiplomskom stručnom studiju

Osijek, 04.09.2017.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za obranu završnog rada na preddiplomskom stručnom studiju

Ime i prezime studenta:	Vedran Kukrika
Studij, smjer:	Preddiplomski stručni studij Elektrotehnika, smjer Automatika
Mat. br. studenta, godina upisa:	A4195, 22.07.2014.
OIB studenta:	96023713112
Mentor:	Doc.dr.sc. Tomislav Keser
Sumentor:	
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Doc.dr.sc. Tomislav Keser
Član Povjerenstva:	Doc.dr.sc. Alfonzo Baumgartner
Naslov završnog rada:	Automatizirani sustav upravljanja pumpama za navodnjavanje
Znanstvena grana rada:	**Procesno računarstvo (zn. polje računarstvo)**
Zadatak završnog rada	Projektirati, izraditi i testirati sustav upravljanja pumpama u poljoprivrednom navodnjavanju. Omogućiti udaljeno upravljanje pumpama i mjerenje relevantnih veličina upravljanog sustava.
Prijedlog ocjene pismenog dijela ispita (završnog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	04.09.2017.

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

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 18.09.2017.

Ime i prezime studenta:	Vedran Kukrika
Studij:	Preddiplomski stručni studij Elektrotehnika, smjer Automatika
Mat. br. studenta, godina upisa:	A4195, 22.07.2014.
Ephorus podudaranje [%]:	6

Ovom izjavom izjavljujem da je rad pod nazivom**: Automatizirani sustav upravljanja pumpama za navodnjavanje**

izrađen pod vodstvom mentora Doc.dr.sc. Tomislav Keser

i sumentora

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

Potpis studenta:

Sadržaj

1. UVOD 1
1.1 Zadatak završnog rada 2
2. AUTOMATIZIRANI SUSTAV ZA UPRAVLJANJE PUMPAMA ZA NAVODNJAVANJE 3
2.1 O automatiziranim sustavima navodnjavanja 3
2.2 Opis procesa i tehničko-tehnoloških zahtjeva sustava 4
2.3 Struktura i funkcionalna građa sustava 5
2.3.1 Upravljačko-nadzorni sustav pumpi za navodnjavanje 7
2.3.2 Centralizirano upravljanje i nadzor 8
2.4 Komunikacija i više-čvorno okruženje 9
2.4.1 Komunikacijsko sučelje i zahtjevi 10
2.4.2 Komunikacijski protokol i karakteristike 12
3. REALIZACIJA SUSTAVA ZA UPRAVLJANJE PUMPAMA ZA NAVODNJAVANJE 13
3.1 Korišteni alati i tehnologije 15
3.2 Građa i karakteristike upravljačkog sustava za pumpe navodnjavanja 19
3.2.1 Struktura i funkcionalnost sustava 19
3.2.2 Senzori 21
3.2.3 Aktuatori 27
3.2.4 Upravljačka jedinica 31
3.2.5 Komunikacijsko sučelje 33
3.3 Shema sustava 37
3.4 Algoritam upravljanja 38
3.4.1 Upravljanje sustavom pumpi 38
3.4.2 Komunikacija i protokol 40
3.4.3 Nadzor, upravljanje i prikaz procesnih parametara 44
3.5 Realizacija sustava 46
3.5.1 Izrada tiskane pločice 46
3.5.2 Sastavljanje sustava 49
3.5.3 Podešavanje sustava 51
3.5.4 Testiranje funkcionalnosti 52
4. TESTIRANJE I REZULTATI 53
4.1 Metodologija testiranja 53
4.2 Rezultati testiranja 54
5. ZAKLJUČAK 56

LITERATURA 57
SAŽETAK 59
ABSTRACT 60
ŽIVOTOPIS 61
PRILOZI 62

1. UVOD

Navodnjavanje podrazumijeva oblik umjetnog povećavanja sadržaja vode u tlu poljoprivrednih površina za optimalan rast i razvoj biljke u cilju ostvarenja što većeg uroda. Navodnjavanje poljoprivrednih kultura je vrlo stara melioracijska mjera i praksa koju su koristile mnoge svjetske civilizacije. Prvi podatci o navodnjavanju poljoprivredne površine zabilježeni su već 6000 godina prije Krista. Po navodnjavanju je poznat egipatski narod, koji je za navodnjavanje koristio obližnju rijeku Nil, dok su u Europi Grci i Rimljani gradili svoje sofisticirane sustave za navodnjavanje. Tehnike dovođenja vode su u usporedbi sa današnjom tehnologijom bile na nižoj razini, ali su učinci bili postizani.

Razvitkom vodenih crpki i otkrićem novih materijala, nakon drugog svjetskog rata, sustav navodnjavanja se naglo razvija. Tako danas imamo 17% obradivih površina koje se navodnjavaju, a na njima se proizvodi gotovo 40% svjetske hrane.

Automatizirani sustav za navodnjavanje podrazumijeva samostalan i zaokružen sustav koji na temelju informacija sa senzora (u polju) signalizira kada i koliko navodnjavati kako bi se određenoj poljoprivrednoj kulturi osigurala optimalna količina vode. Potrošnja optimalne količine vode u tlu donosi dvostruku korist: povećanje biološkog potencijala poljoprivredne kulture uz minimalnu potrošnju vode za navodnjavanje.

1.1 Zadatak završnog rada

Zadatak ovog završnog rada je projektirati, izraditi i testirati sustav upravljanja pumpama u poljoprivrednom navodnjavanju. Omogućiti udaljeno upravljanje pumpama i mjerenje relevantnih veličina upravljanog sustava. Sustav je zamišljen tako da na osnovu podataka dobivenih od računala (korisnika) upravlja pumpama koje potom kontroliraju navodnjavanje pojedine kulture. Prednost ovog sustava naspram drugih sustava je što se sastoji od više međusobno nezavisnih kontrolnih jedinica kojima se upravlja putem jedne centralne jedinice povezane sa računalom.

2. AUTOMATIZIRANI SUSTAV ZA UPRAVLJANJE PUMPAMA ZA NAVODNJAVANJE

2.1 O automatiziranim sustavima navodnjavanja

Navodnjavanje je u osnovi uzgojna mjera u biljnoj proizvodnji kojom se tlu dodaju one količine vode potrebne za optimalan rast i razvoj biljke. S obzirom na potrebe i mogućnosti u Hrvatskoj su navodnjavanjem obuhvaćene male površine. Međutim, uzgoj povrća je, posebno u priobalnom dijelu, gotovo neostvariv bez navodnjavanja. Naročito je to izraženo u ljetnim rokovima sjetve odnosno sadnje pri uzgoju povrća.

Sustav za automatsko navodnjavanje i zalijevanje štedi vrijeme koje se obično troši stojeći sa šmrkom u ruci i omogućava da poljoprivredna kultura, travnjak ili posađeno bilje dobije vodu i u odsustvu čovjeka. Pravilno postavljeni i automatizirani sustav navodnjavanja će osim pravilne distribucije vode po poljoprivrednoj kulturi i zemljištu donijeti i uštedu novaca. Da bi to bilo moguće potrebno je pravilno postaviti i održavati sustav za navodnjavanje. U nastavku nabrojano je nekoliko smjernica koje treba slijediti kako bi se ostvario optimalni učinak sustava za navodnjavanje:

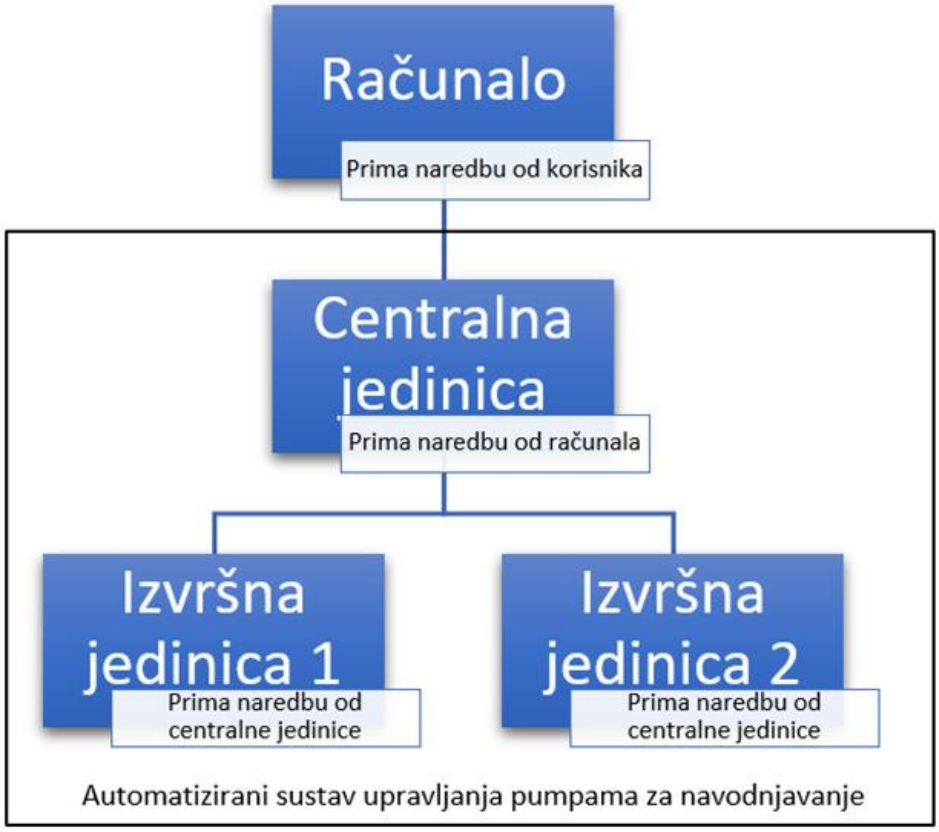
- Pravilna raspodjela biljaka i raslinja po zonama
- Ujednačena podjela vode unutar zona
- Smanjit neželjena rasipanja (umjesto jednog duljeg vremena distribucije vode, bolje je zalijevanje podijeliti u dva kraća intervala)
- Zalijevati u vremenu od 17,00 – 22,00 sata kada je sunce slabije i vjetar mirniji
- Jutarnja polijevanja nisu efikasna zbog pojačanog hlapljenja
- Kasno večernja polijevanja donijet će opasnost od gljivičnih oboljenja od kojih stradavaju i travnjaci
- Pravilnim programiranjem upravljati količinom vode u određenom vremenu
- Regulirati i modificirati navodnjavanje ovisno o količinama padalina
- Navodnjavati samo površine koje to zahtijevaju, što se postiže pravilnim određivanjem kuta zalijevanja na prskalicama
- Gdje je moguće koristiti cijev kap na kap i mikro irigaciju kako bi se voda dovela neposredno do korijena biljaka
- Vršiti rutinske provjere cijelog sustava
- Opredijeliti se za potpunu automatizaciju jer samo tako postiže se visoki učinak sustava i znatne uštede.

2.2 Opis procesa i tehničko-tehnoloških zahtjeva sustava

Ovaj sustav automatiziranja upravljanjem pumpa za navodnjavanje je zamišljen tako da se sastoji od jedne centralne jedinice i više izvršnih jedinica. Važno je napomenuti da su izvršne jedinice međusobno nezavisne jedna od druge, i sve su bežično povezane sa centralnom jedinicom. Pošto veličina poljoprivredne površine varira, bežična komunikacija među jedinicama se čini kao najbolji način komuniciranje. najveća procijenjena udaljenost između centralne i izvršne jedinice bi iznosila oko 1 Km. Zbog toga potrebno je izabrati pravilnu radio-frekventnu komponentu za komunikaciju. Izvršne jedinice koje će se ugrađivati na polje moraju biti što je moguće manjih dimenzija kako ne bi smetale pri rastu pojedine kulture koju će navodnjavati. Zbog vremenskih utjecaja na poljoprivrednoj površini (kiša, temperatura, vlaga, nečistoće, vjetar, i sl.) sve jedinice moraju biti otporne na vremenske uvjete.

2.3 Struktura i funkcionalna građa sustava

Kao što je u prethodnom poglavlju spomenuto, sustav će funkcionirati u dvije cjeline: centralna i izvršna jedinica. Svaka od te dvije jedinice se ponaša kao zasebni sklop sa mogućnošću bežičnog komuniciranja sa ostalim jedinicama s kojima je povezana. Shematski prikaz organizacije sustava može se vidjeti na slici 2.1. Količina izvršnih jedinica koje se mogu bežično spojiti sa centralnom jedinicom ograničena je na 511. Na shematskom prikazu su prikazane samo 2 izvršne jedinice.



Slika 2.1: Shematski prikaz organizacije sustava.

Centralna jedinica se sastoji samo od bežičnog modula i mikrokontrolera povezanog sa računalom i ona zapravo služi kao „poveznica“ bežične komunikacije između izvršnih jedinica i računala. Za razliku od nje, izvršna jedinica sastoji se od mikrokontrolera, bežičnog modula, senzora i upravljačke jedinice za motore.

Princip rada sustava je sljedeći: centralna jedinica putem USB komunikacijskog sučelja prima podatke poslane određenim redoslijedom. Raščlanjuje ih i prosljeđuje izvršnim jedinicama. Komunikacija između jedinica je dvosmjerna, što znači da podatci mogu putovati iz centralne jedinice u izvršnu i obrnuto. Isto važi i za komunikaciju između računala i centralne jedinice.

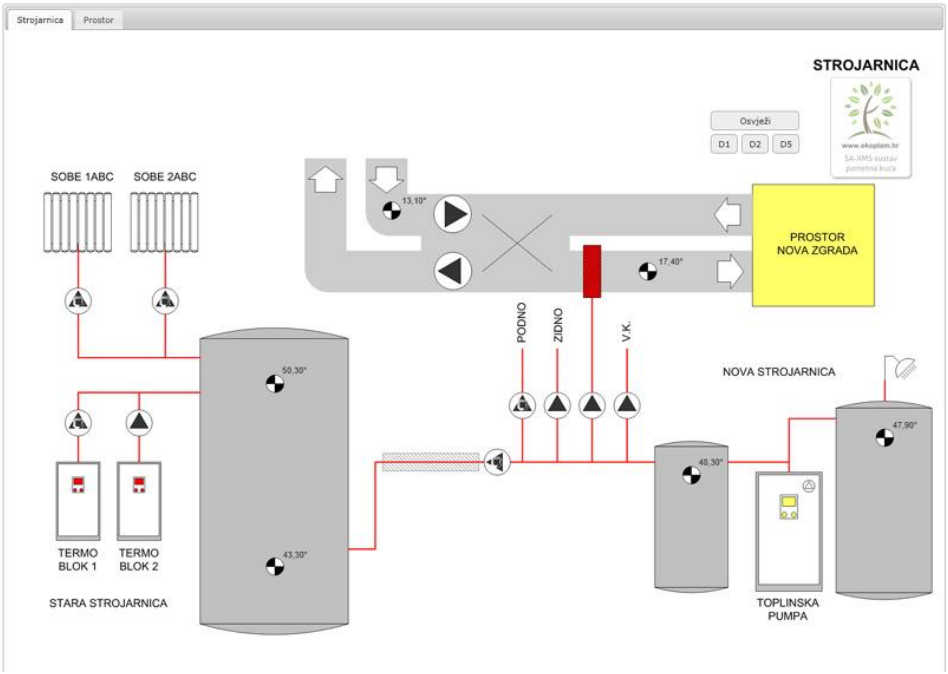
Za primjer, za očitanje temperature zraka kod određene kulture, računalo šalje centralnoj jedinici zahtjev za temperaturu na izvršnoj jedinici 1, centralna jedinica 1 prosljeđuje bežičnom komunikacijom zahtjev na izvršnu jedinicu 1. Izvršna jedinica prima zahtjev, očitava temperaturu i šalje je u obliku odgovora centralnoj jedinici, centralna jedinica ga obrađuje i prosljeđuje računalu. Isti je princip komunikacije i za upravljanje pumpama. Kako bi računalo znalo s kojom izvršnom jedinicom komunicira, svaka izvršna jedinica prilikom slanja odgovora, uz odgovor doda je i svoj digitalni „potpis“ u obliku svog rednog broja.

2.3.1 Upravljačko-nadzorni sustav pumpi za navodnjavanje

Upravljačko-nadzorni sustav pumpi za navodnjavanje je korisničko sučelje i nalazi se na računalu, njime upravlja korisnik navodnjavanja. To je zapravo računalni softver koji predočuje korisniku trenutno stanje sustava za navodnjavanje, odnosno dozvoljava korisniku da iščitava i unosi parametre u sustav.

Svaki dio je u softveru predstavljen svojim simbolom i položajem u sustavu navodnjavanja. Ispod svakog simbola pišu bitni prethodno očitani parametri pojedinog djela sustava. Pojedini parametri (poput protoka pumpe, tj. količine navodnjavanja) se mogu promijeniti ili postaviti u drugo stanje, dok se parametri poput temperature ili protoka mogu samo prikazati korisniku, tj. korisnik ih može očitati.

Ovakav način prikazivanja, očitavanja i postavljanja parametara objedinjuje sve izvršne jedinice na jednom mjestu pred korisnikom i daje korisniku jednostavniji prikaz cijelog sustava. Korisnik ne mora poznavati način rada pojedinih dijelova sustava ili organizaciju sustava kako bi mogao upravljati njime. Dakle, može se reći kako upravljačko-nadzorni sustav pumpi za navodnjavanje uvelike pojednostavljuje korisnikovo korištenje sustava. Na slici 2.2 prikazan je primjer snimke zaslona upravljačko – nadzornog sustava nekog procesa.



Slika 2.2: Primjer snimke zaslona upravljačko – nadzornog sustava nekog procesa.

Snimka zaslona upravljačko nadzornog sustava automatiziranog upravljanja pumpama za navodnjavanje prikazana je u prilogu P9.5.

2.3.2 Centralizirano upravljanje i nadzor

Centralni nadzor i upravljanje sustavima obično se dijeli u dvije grupe:

1) Centralni nadzor i upravljanje sigurnosnim sustavima u koje pripadaju sustav za dojavu požara, sustav plinodetekcije te sustavi tehničke zaštite.

2) Centralni nadzor i upravljanje automatizacijom cjelokupnog sustava, gdje se najčešće automatiziraju termotehničko-strojarske instalacije i elektrotehničke instalacije. U ovu grupu bi spadao sustav automatiziranog upravljanja pumpama za navodnjavanje.

Sve sustave prve grupe moguće je integrirati u jedan centralni nadzorni sustav. Centralni uređaji na koje su spojeni svi elementi svakog pojedinog sustava, spajaju se preko odgovarajućeg mrežnog uređaja (*engl. gateway* –mrežni uređaj) na centralno nadzorno računalo. Na njemu se prikazuju grafičke informacije (situacije, tlocrti, presjeci, sheme itd.) objekta, sustava, komponente sustava, sukladno izvedenom stanju projekta. Na taj način, može se precizno odrediti na kojem djelu sustava, u kojoj prostoriji odnosno sobi, se pojavio alarm, odnosno koji javljač je u alarmu, grešci, prekidu ili sl.

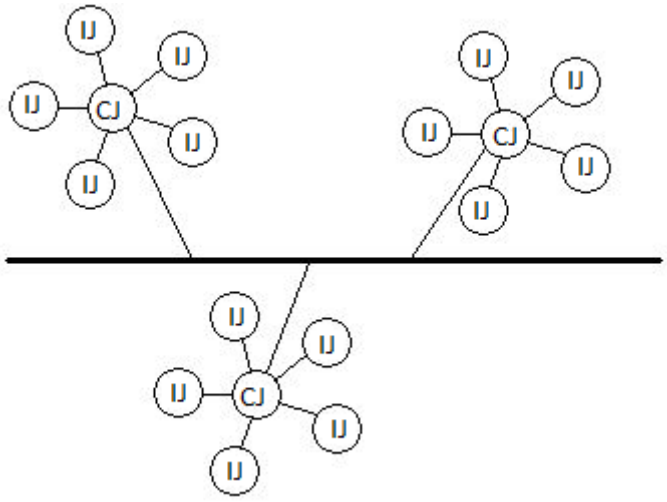
Bitno je naglasiti da je komunikacija u takvim sustavima centralnog nadzora dvosmjerna, što znači da osim prihvata različitih informacija (alarma, predalarma, detekcije anomalija u sustavu, grešaka na sustavu itd.) može se obaviti i upravljanje (potvrđivati i poništiti inicirane događaje, obavljati uključenja i isključenja sekcija, zona ili elemenata i sl.) U takvom sustavu moguće je i podesiti razinu ovlasti svakog operatera, tako da se može omogućiti potvrđivanje i poništenje događaja, ali i zabraniti bilo kakva mogućnost uključenja/isključenja elemenata, zona ili sekcija te upravljanje izvršnim funkcijama. Lokalne jedinice sustava moguće je povezati putem lokalne računalne mreže s centralnom nadzornom stanicom ali i na udaljene klijentske stanice putem interneta. Svaki od sustava obično se prilagođava zahtjevima individualnog korisnika.

Centralni nadzor i upravljanje sustavima automatizacije, odnosno druge grupe sustava spomenute na početku odlomka karakteristično je za svaki pojedini sustav. U to ulazi prikupljanje i obrada informacija na temelju kojih se provode akcije, dok se podaci arhiviraju i vizualiziraju prema potrebama korisnika.

2.4 Komunikacija i više-čvorno okruženje

Računalna mreža je skupina dva ili više međusobno povezanih računala koji dijele neke resurse (podatke, sklopovlje, programe...). Računala se smatraju povezanima ako mogu razmjenjivati informacije. Čvorovi računalne mreže u međusobnoj komunikaciji koriste komunikacijske protokole.

Sustav automatiziranog upravljanja pumpama za navodnjavanje spojen je u zvjezdastu topologiju komunikacije. Ta zvjezdasta topologija korištenjem više centralnih jedinica u sustavu na posljetku može prerasti u stablastu topologiju. Na slici 2.3 prikazana je stablasta topologija.



Slika 2.3: Stablasta topologija.

Ova topologija organizirana je na način da se u središtu komunikacije nalazi centralna jedinica povezana bežičnim putem sa izvršnim jedinicama, a putem kabla sa računalom odnosno računalnom sabirnicom. Izvršne jedinice komuniciraju samo sa centralnim jedinicama, a centralne jedinice komuniciraju i sa računalom i sa izvršnim jedinicama. Računalo može putem sabirnice komunicirati samo sa centralnim jedinicama.

Prednosti ove topologije su: jednostavna instalacija i umrežavanje; bez smetnji za mrežu kada se spajaju odnosno odspajaju uređaji; jednostavno dijagnosticiranje problema u mreži. S druge strane, glavna mana ove topologije je to što ukoliko dođe do kvara centralne jedinice, sve izvršne jedinice spojene na nju ostaju bez komunikacije i na taj način postaju neupotrebljive.

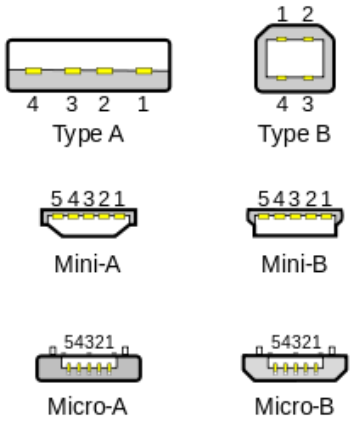
2.4.1 Komunikacijsko sučelje i zahtjevi

Sučelje je mjesto ili dio uređaja gdje se vrši radnja između čovjeka i stroja ili između dva stroja. Cilj komunikacijskog sučelja je što efikasnije obaviti radnju između dvije komunikacijske cjeline. Sučelje korišteno za komunikaciju između računala (čovjeka) i sustava automatiziranog upravljanja pumpama za navodnjavanje je USB sučelje.

USB (*engl. Universal Serial Bus, USB* – Univerzalna serijska sabirnica) je tehnološko rješenje za komunikaciju računala s vanjskim uređajima pri čemu se podatci razmjenjuju serijski relativno velikom brzinom. USB je zamijenio razna dotadašnja serijska i paralelna sučelja na računalima. Sedam vodećih svjetskih računarskih kompanija – Compaq, DEC, IBM, Intel, Microsoft, NEC i Nortel – započelo je razvoj USB-a 1994. godine s ciljem da se bitno olakša povezivanje računala s vanjskim uređajima u odnosu na dotadašnja sučelja, pojednostavi softverska konfiguracija tih uređaja te istovremeno poveća brzina komunikacije.

Cilj tehnologije USB jest rasterećivanje glavne sabirnice računala od posebnih kartica za proširenje, kao i olakšavanje umetanja i odvajanja vanjskih uređaja (plug-and-play) bez potrebe za ponovno pokretanja računala (hot swapping).

USB je asimetričnog dizajna i sastoji se od poslužitelja (*engl. host* - poslužitelj) i više jedinica koje se priključuju na poslužitelj kao grane stvarajući zvjezdastu strukturu. Međutim, na poslužitelj se može priključiti i poseban koncentrator (*engl. hub* - koncentrator) na kojeg se mogu priključivati druge grane što stvara stablastu strukturu. Kod USB-a je moguće imati do pet razina grananja po svakom kontroleru poslužitelju te je moguće priključiti ukupno do 127 uređaja, umanjeno za svaki koncentrator koji je priključen na isti poslužitelj. Standardi USB 1.x i 2.0 koriste kabel s dvije bakrene parice, od kojih se jedan par koristi za napajanje a drugi za podatke. Podatkovna parica je upletena kako bi se smanjio utjecaj interferencije i šuma.



Slika 2.4: A i B tip USB konektora, [14].

Konektori za USB (slika 2.4) su izvorno izvedeni u dvije osnovne inačice – A i B. Tip A najčešće se koristi na uređajima s kojih se podatci šalju prema računalu, kao što su miš, tipkovnica, digitalni foto-aparat itd. Konektor tipa B najčešće se koristi na uređajima na koje se podatci šalju s računala, poput pisača. Zbog potrebe za spajanjem uređaja koji teže smanjivanju dimenzija (npr. foto-aparati i mobiteli) kasnije su standardizirani konektori Mini-A i Mini-B dimenzija 6,8 x 3 mm, a još kasnije Micro-A i Micro-B dimenzija 6,85 x 1,8 mm. Uz smanjenje dimenzija dodan je još jedan pin za identifikaciju tipa konektora o kojemu se radi: za tip A on je spojen na uzemljenje, dok je kod tipa B od spojen. Životni vijek konektora tipa A i B procjenjuje se na oko 1500 uključivanja/isključivanja, dok za tipove Mini iznosi oko 5.000, a za tipove Micro oko 10.000.

Standardi 1.x definiraju napajanje od 5 V ± 5% preko jedne žice. U standardu 2.0 donja granica je spuštena na 4,4 V dok je gornja granica ostala na 5,25 V. Jedinično strujno opterećenje iznosi 100 mA, dok maksimalno opterećenje iznosi pet takvih opterećenja – 500 mA. U standardu USB 3.0 jedinično opterećenje povećano je na 150 mA a maksimalno opterećenje na 900 mA.

U tablici 2.1 prikazane su maksimalne brzine prijenosa koje je moguće postići pojedinom inačicom USB sučelja.

Tablica 2.1: Brzine prijenosa podataka USB sučeljem, [14].

Oznaka	Brzina prijenosa	Podržano od verzije standarda
Low Speed	1,5 Mbit/s	USB 1.0
Full Speed	12 Mbit/s	USB 1.1
HiSpeed	480 Mbit/s	USB 2.0
SuperSpeed	4,8 Gbit/s	USB 3.0

2.4.2 Komunikacijski protokol i karakteristike

Komunikacijski protokol je skup jednoznačno određenih pravila za razmjenu informacija između dva entiteta na mreži koji uključuje sintaksu informacije, semantiku informacije te pravila za razmjenu informacije. Entitet može biti računalo, program, sigurnosna kamera, senzor i sl.

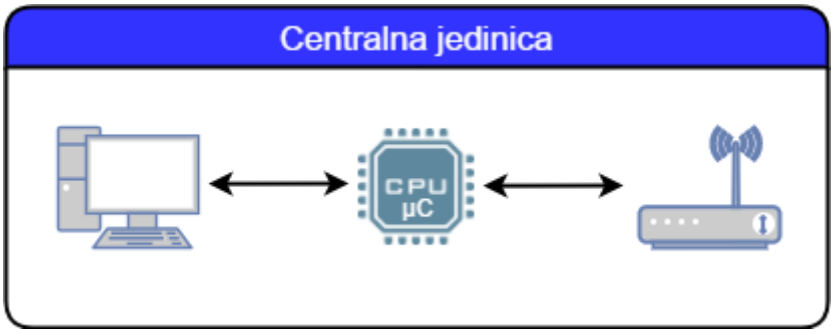
Računalo putem USB sučelja šalje mikrokontroleru pakete, koji su dogovoreni internim standardom (komunikacijskim protokolom), podataka koje on raščlanjuje i prosljeđuje određenim izvršnim jedinicama. Struktura paketa je sljedeća:

[CA|CM|b1|b2|b3|b4]

Paket se sastoji od ukupno 6 bajta podataka. Prvi bajt pod nazivom CA označava adresu klijenta koju računalo želi kontaktirati, ili adresu klijenta s kojega dolazi povratna informacija (podatak). Drugi bajt pod nazivom CM označava komandu koju računalo želi potaknuti, na primjer to može biti komanda za očitanje temperature, za pokretanje ili za ustavljanje pumpe ili slično. Preostala 4 bajta su rezervirana za slanje nekih vrijednosti. Vrijednost ta 4 bajta prilikom komunikacije računalo – izvršna jedinica je 0 iz razloga što računalo ne postavlja nikakve vrijednosti. Osim u slučaju kada računalo postavlja vrijednost protoka PID regulatoru, u tom slučaju računalo kroz ta 4 bajta šalje podatak o iznosu protoka izvršnoj jedinici. Osim tada, ta 4 bajta se još koriste kada izvršna jedinica šalje tražene očitane podatke nazad u računalo, na primjer kada se vraća vrijednost očitane temperature. Detaljniji opis komunikacijskog protokola između računala i centralne jedinice nalazi se u poglavlju 3.4.2.

3. REALIZACIJA SUSTAVA ZA UPRAVLJANJE PUMPAMA ZA NAVODNJAVANJE

Sustav je realiziran pomoću centralne jedinice i izvršne jedinice. Centralna jedinica, kao što je ranije spomenuto, povezana je sa računalom putem USB komunikacijskog protokola. Ona služi kao poveznica računala sa izvršnim jedinicama. Ona se sastoji od mikrokontrolera te komunikacijskog modula (prikazano na slici 3.1).



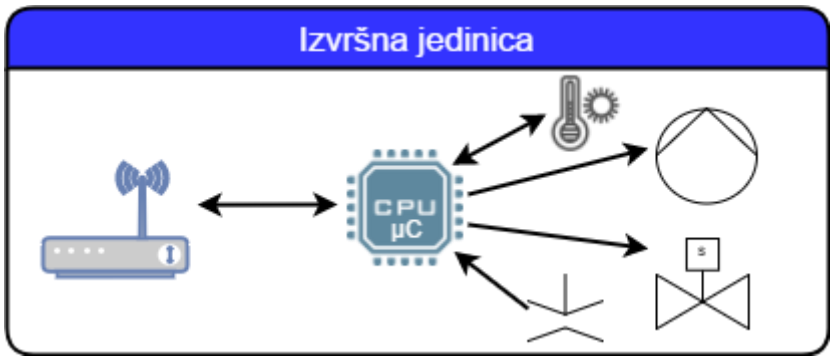
Slika 3.1: Blokovski prikaz centralne jedinice.

Komunikacijski (bežični) modul koji koristi centralna jedinica za komunikaciju sa izvršnim jedinicama, komunicira sa mikrokontrolerom putem SPI protokola. Računalo, s druge strane, ne podržava komunikaciju SPI protokolom pa se zbog toga koristi mikrokontroler kako bi „preuzeo“ podatke s jedne strane i „predao“ ih drugoj strani, njima prihvatljivim protokolima.

Kao mikrokontroler se koristi arduino platforma, odnosno mikrokontroler ATmega328P tvrtke ATMEL. Za mrežnu komunikaciju između jedinica se koristi čip tvrtke Nordic Semiconductor pod nazivom nRF905.

Centralna jedinica još na sebi ima tipkalo koje kada se pritisne omogućuje spajanje novih izvršnih jedinica. Isto to tipkalo se mora pritisnuti i na izvršnoj jedinici, te one onda međusobno razmjenjuju podatke i centralna jedinica upisuje adresu izvršne jedinice u svoju memoriju. Ova mogućnost je dodana kako bi se omogućilo lakše dodavanje izvršnih jedinica, pošto njihov broj svake godine varira. Ovisno koliko kultura ima na pojedinom poljoprivrednom zemljištu, toliko izvršnih jedinica će biti spojeno na centralnu jedinicu.

Ideja automatiziranog upravljanja pumpama za navodnjavanje je potpuno automatizirano zalijevanje poljoprivredne površine. Osim toga, automatizacijom se može nadomjestiti potreba za ljudskom radnom snagom. Nova radna snaga su strojevi i upravljački sklopovi. Upravo izvršna jedinica igra glavnu ulogu u zamjeni ljudi za strojeve. Ona upravlja ventilom za dovod vode, pumpom, senzorima i mrežnim modulom.



Slika 3.2: shematski prikaz slojeva izvršne jedinice.

Blokovski prikaz izvršne jedinice prikazan je na slici 3.2, njezin princip rada sličan je centralnoj jedinci, osim što mikrokontroler nema zadaću samo skupljati i slati podatke, već pored toga prikuplja podatke sa senzora i djeluje na vodenu pumpu ili ventil.

No, za razliku od centralne jedinice, izvršna jedinica mora upravljati robusnim ventilom i pumpom za navodnjavanje za čije je upravljanje potrebna veća snaga nego što nam mikrokontroler može dati na svojim izlazima. Zbog toga izvršna jedinica mora koristiti 2 izvora napajanja, jedan od 5V za napajanje mikrokontrolera, senzora i mrežnog modula, i 24 V za napajanje ventila i pumpe za vodu.

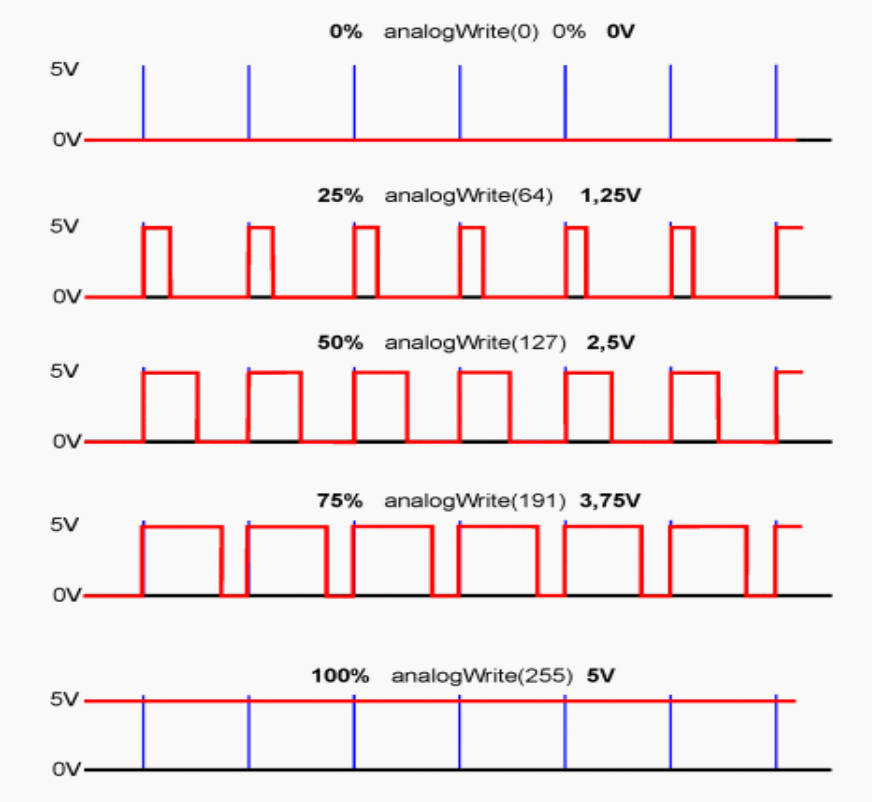
Kao što je prije spomenuto, mikrokontroler na svojim izlazima ne može dati dovoljno veliku struju za upravljanje pumpom i ventilom, mikrokontroler na svojim izlazima može dati maksimalnu struju od 30 mA, a za pokretanje pumpe potrebno je minimalno 1,5A. Iz tog razloga se za pokretanje pumpe i ventila koriste dva N-kanalna mosfeta IRF530. Oni mogu pomoću upravljačkog signala sa mikrokontrolera (5V) upravljati sa naponima do 100V i strujama do 14 A.

Princip rada ove izvršne jedinice temelji se na varijablama koje joj postavi korisnik odnosno centralna jedinica. Za na primjer optimalno navodnjavanje se postavi protok vode kroz cijev koji je potreban, u mjernoj jedinici l/h. Putem PID regulatora pumpa se regulira i postavlja na traženi protok. Povratnu informaciju o protoku joj daje senzor protoka.

3.1 Korišteni alati i tehnologije

Pulsno širinska modulacija, odnosno poznatija kao PWM (*engl. Pulse Width Modulation* – pulsno širinska modulacija), je tehnika kontroliranja analognih strujnih krugova putem digitalnih izlaza sa mikrokontrolera ili nekog drugog upravljača. Napon se dovodi do analognog uređaja kao serija impulsa, a informacija o amplitudi analognog signala se predstavlja širinom (trajanjem) impulsa signala. Ova vrsta kontroliranja ima vrlo veliku primjenu u svim granama elektronike, koristi se pri mjerenju, u komunikaciji, napajanju, kontroli motora kao i u raznim vrstama pretvarača.

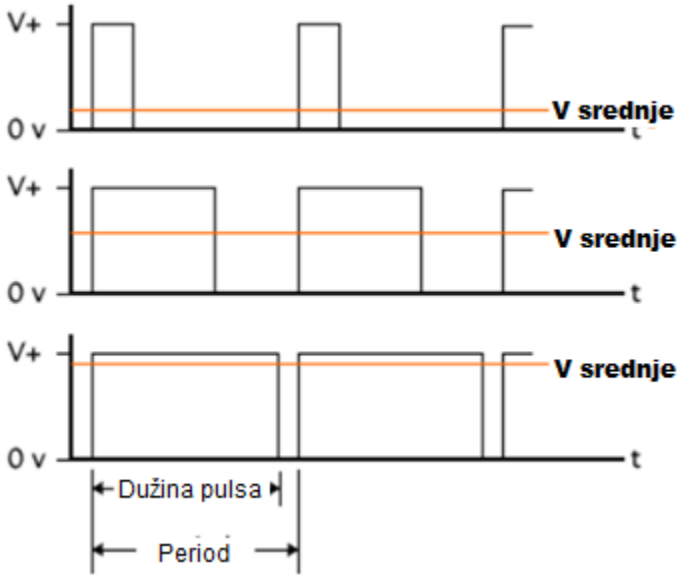
Analogni naponi i struje se mogu koristiti za direktnu kontrolu uređaja, to je jednostavan i jasan način kontrole, ali nije uvijek praktičan i isplativ. Analogna kontrola ima razne mane kao što su dimenzije komponenata, promjenjive karakteristike komponenata tokom vremena, veća potrošnja, veće grijanje, osjetljivost na šum i drugo… Digitalnom kontrolom analognih uređaja vijena sistema opada i potrošnja energije se drastično smanjuje. Mnogi današnji mikrokontroleri imaju ugrađen hardver koji podržava pulsno širinsku modulaciju.



Slika 3.3: PWM modulacija, [9].

Kod PWM modulacije, ciklus trajanja impulsa je konstantan i naprijed određena. Mijenja se trajanje logičke jedinice, odnosno vrijeme u kojem je tokom trajanja ciklusa postavljana logička

jedinica. Na taj način uređaji kojima se upravlja vide samo srednju vrijednost napona (slika 3.4). Različita vremena trajanja impulsa prikazana su na slici 3.3, ciklusi su označeni plavom bojom.

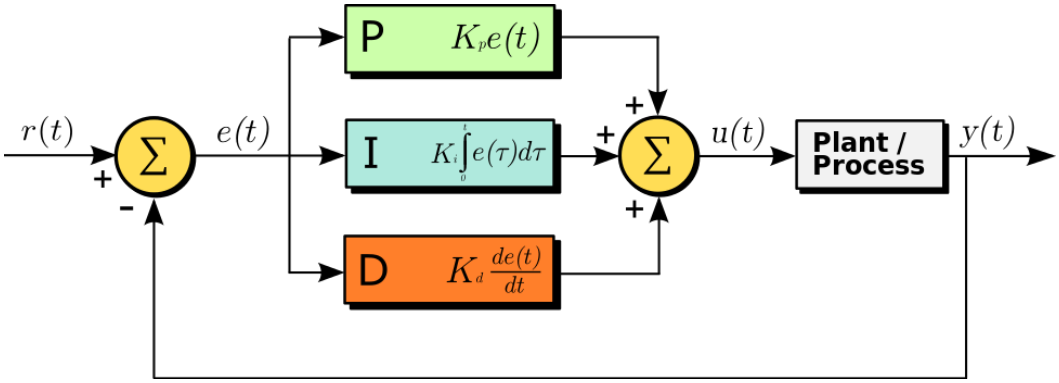


Slika 3.4: Srednja vrijednost napona kod PWM modulacije, [8].

PWM pretvornici koji se najčešće koriste u jednostavnijim mikrokontrolerima su 8-bitni pretvornici. Što bi značilo da se u programu vrijednosti izlaza od 0 do 5 V definiraju brojevima od 0 do 255. Odnosno ako se želi na analognom izlazu pustiti 2,5V u program će biti upisan broj 127. Ova vrsta PWM regulacije može se spojiti sa upotrebom tranzistora ili mosfeta, te na taj način postići će se upravljanje puno većim naponima, strujama i jačim strojevima.

Svaki sistem automatskog upravljanja se odlikuje određenim zakonom ili zakonima upravljanja. Zakon upravljanja predstavlja matematičku ovisnost na osnovu koje upravljački uređaj obrađuje pripadajuće signale i generira odgovarajuće upravljačke signale. U suštini, PID regulator se sastoji od postavljene, poremećajne i izlazne veličine. Njegova svrha je da sa što većom točnošću pokuša izjednačiti izlaznu veličinu nekog sustava sa postavljenom veličinom.

PID regulator sastoji se od 3 vrste regulatora, to su P, I i D regulator. Zbog toga što se PID regulator sastoji od 3 regulatora, za podesiti ima 3 varijable. to su Kp, Kd, Ki. Operacije ta 3 regulatora se zbrajaju i na izlazu dobijemo PID regulator. Važno je napomenuti da regulator ima povratnu vezu, odnosno ima direktan uvid u stanje procesa.

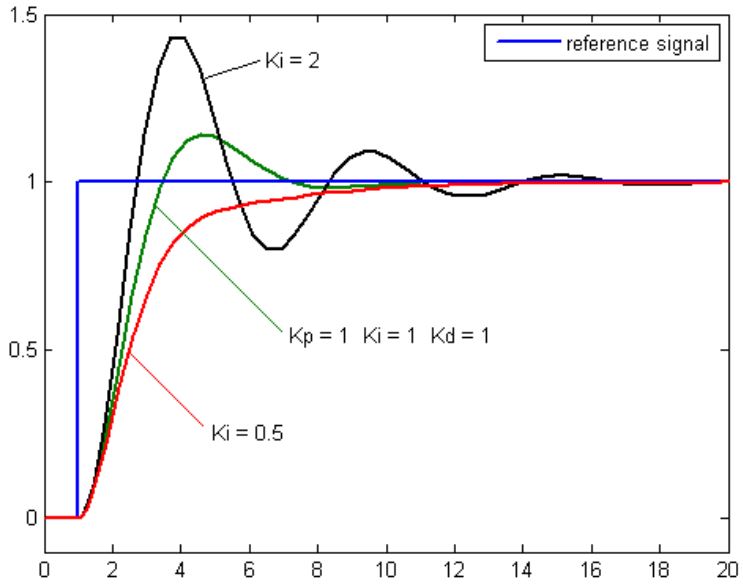


Slika 3.5: PID regulator, [15].

Proporcionalni član djeluje na vrijeme odziva, ali se njegovim povećanjem ne može ukloniti pogreška stacionarnog stanja. Integralni član djeluje na smanjenje pogreške stacionarnog stanja, ali može pogoršati dinamička svojstva sustava (usporiti sustav). Derivacijski dio utječe na povećanje stabilnosti sustava, smanjuje prebačaj i poboljšava karakteristike prijelaznog dijela odziva.

Poveznica između slike 3.5 i automatiziranog sustava upravljanja pumpama za navodnjavanje je sljedeća, proces bi bile pumpe u sustavu i upravljanje protokom, vodeća veličina (r(t)) je postavljeni protok koji želimo postići na izlazu našeg sustava. Povratna veza je senzor protoka spojen na cijev, a PID regulator koji upravlja svim tim veličinama je mikrokontroler. Upravljačka veličina (u(t)) je izlazni PWM signal iz mikrokontrolera. Izlazna veličina (y(t)) je željeni protok na izlazu iz cijevi.

Izlazna karakteristika PID regulatora vidljiva je na grafu 3.1, na njoj ima više izlaznih karakteristika. Svaka karakteristika ima drugačije parametre regulatora (KP, Ki, Kd). Plavom bojom označena je step promjena vodeće veličine. Izbor parametara je vrlo bitan, jer ako se odaberu krivi parametri sustav može otići u nestabilnost. Dakle svaki proces ima vlastite parametre koji zavise o procesu, načinu rada, regulatoru i sl.



Grafikon 3.1: Odzivi PID regulatora za različite parametre, [15].

Na grafikonu 3.1 da se primijetiti da PID regulator ima određeno nadvišenje izlazne veličine dok se odziv ne dovede u stacionarno stanje. Što bi značilo ako se protok namjesti na konstantnu vrijednost od 4,2 l/h, dok PID ne postavi vrijednost na 4,2 l/h, vrijednost može varirati od 4,7 do 3,9 l/h. Ova pojava naziva se prijelazna pojava PID regulatora i njezino trajanje određeno je parametrima PID regulatora i konstantama procesa.

Brzina odziva PID regulatora isto tako je određena parametrima. Što je veća brzina odziva, povećava se trajanje prijelazne pojave i povećava se maksimalno nadvišenje izlazne veličine. Ako se smanji pojačanje, tj., brzina odziva smanjuje se i vrijeme trajanja prijelazne pojave a time i maksimalno nadvišenje.

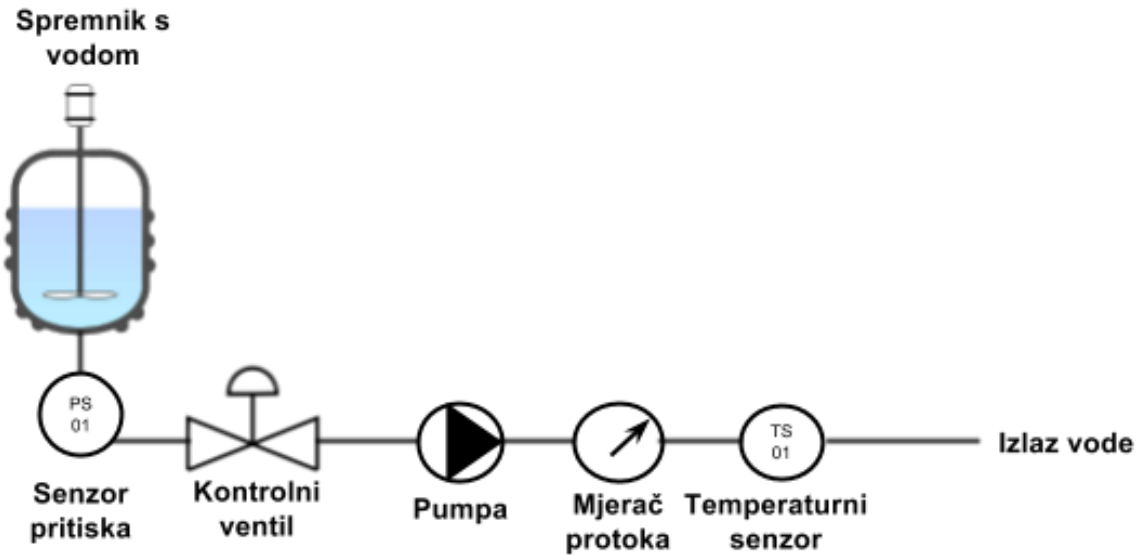
Važno je napomenuti da kod PID regulatora izlazna veličina u stacionarnom stanju nikada neće biti potpuno jednaka vodećoj (postavljenoj) veličini.

3.2 Građa i karakteristike upravljačkog sustava za pumpe navodnjavanja

3.2.1 Struktura i funkcionalnost sustava

U slučaju izvršne jedinice, u izvršne članove spadaju ventil i pumpa. Njihova zadaća je na osnovu signala dobivenog od mikrokontrolera izvršiti neku zadaću. Odnosno, u ovom slučaju izvršiti zadaću kontrole vodenog protoka.

Zadaća mjernih članova je obrnuta, oni daju mikrokontroleru (na posljetku i korisniku) podatke o okolini u kojoj radi, odnosno u ovom slučaju o protoku vode kroz cijev, temperaturi medija (vode) u cijevi, količini vode u spremniku, te temperaturu i vlažnost zraka okoline. Shema spajanja mjernih i izvršnih članova u cijelom sustavu navodnjavanja može se vidjeti na slici 3.6:



Slika 3.6: Shema spajanja mjernih i izvršnih članova u sustavu navodnjavanja.

Spremnik vode može biti spojen iznad sustava ili ispod sustava, ako je spremnik iznad sustava onda je kontrolni ventil od krucijalne važnosti, on sprečava da voda teče kroz sistem dok je pumpa ugašena. Isto tako kada je spremnik gore pumpa mora uložiti manje rada da bi crpila vodu, time štedi električnu energiju. Kada je spremnik ispod sustava onda kontrolni ventil sprečava povrat vode, koja je u sustavu, nazad u spremnik, na taj način se skraćuje vrijeme rada pumpe kada crpi vodu iz spremnika. Razinu vode u spremniku mjerimo senzorom pritiska koji se nalazi direktno ispod njega. Težina vode pritišče membranu senzora koji na osnovu tog pritiska daje analognu izlaznu vrijednost i kasnije se preračunava u količinu vode u spremniku. Što ima više vode u spremniku, voda je teža, pritisak je veći.

Ulogu senzora temperature medija ima senzor temperature DS18B20, utisnut u cijev i na taj način mjeri temperaturu. Proizvođač ovog senzora je tvrtka maxim integrated. Dimenzije ovog senzora su vrlo male, zbog toga je prikladan za ovaj posao. Temperaturni opseg mu je od -55 do +125 °C.

Senzor protoka je senzor koji se montira tako da se prekine cijev te se na tom prekidu stavi senzor protoka. Senzor koji koristimo je model YF-S201. Njegov princip rada kao i ostalih komponenata biti će razjašnjen kasnije u tekstu.

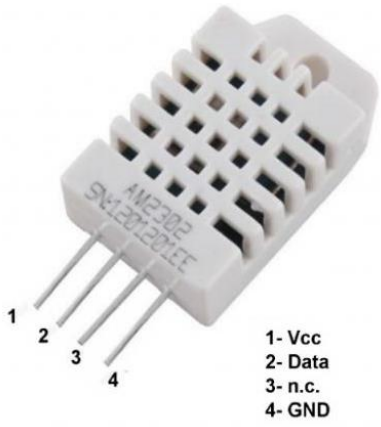
Pored ovih mjernih članova navedenih u vodenom djelu sustava, koristi se još senzor vlažnosti i temperature okoline. On se krije pod nazivom AM2302. U sebi sadrži senzor vlažnosti zraka čija izlazna vrijednost je od 0-100%, te senzor temperature zraka. On se nalazi u zraku i nije spojen sa vodenim dijelom sustava.

Tako korisnik putem izvršne jedinice može dobiti podatke o temperaturi medija (vode) u sistemu navodnjavanja, brzinu okretaja pumpe, protok u cijevima, razinu vode u spremniku, temperaturu okoline te vlažnost zraka okoline.

3.2.2 Senzori

Senzor ili pretvornik je uređaj koji mjeri fizikalnu veličinu (npr. temperature, vlažnosti zraka, tlaka, broj okretaja motora) i pretvara ju u signal pogodan za daljnju obradu (najčešće u električni signal).

AM2302 je jednostavni senzor za mjerenje temperature i vlage zraka. Koristi polimerni kondenzator za mjerenje vlage i termistor za mjerenje temperature zraka iz okoline. Podatke šalje u obliku digitalnog signala na pin od mikrokontrolera. Prednost mu je cijena i jednostavna uporaba, a nedostatak predstavlja vrijeme očitavanja podataka koje iznosi 2 sekunde. Senzor je kalibriran u posebnoj komori, a njegov kalibracijski koeficijent je pohranjen u OTP (One-time programmable) memoriju. Kalibracijski senzor poziva se svaki put prilikom mjerenja kako bi vrijednosti bile pouzdane. Osim niske potrošnje energije, senzor obuhvaća transmisijski domet do 20 metara. Njegov izgled i raspored pinova prikazan je na slici 3.7:



Slika 3.7: Izgled i raspored pinova AM2302 senzora, [6].

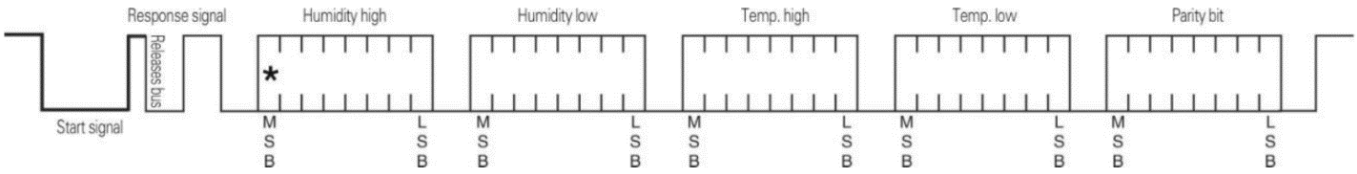
Prvi pin s lijeva spaja se na napajanje (3-5V), drugi pin spaja se na digitalni ulaz Arduino platforme i četvrti tj. zadnji pin spaja se na masu (*engl. ground, GND* - uzemljenje). U odnosu na DHT11 ovaj senzor je precizniji, radi u većem rasponu temperature i vlage, ali je veći i skuplji.

Opće specifikacije senzora:

- Raspon mjerenja relativne vlage od 0 do 100%.
- Raspon mjerenja temperature od -40 do +125°C.
- Moguća odstupanja +-2% za vlagu, +-0.2°C za temperaturu.
- Vrijeme očitavanja svake 2 sekunde.
- Napajanje 3.3 - 6V DC
- Digitalni izlazni signal

AM2302 koristi pojednostavljenu I²C komunikaciju. Inače I²C komunikacija koristi dvije žice za razmjenu podataka, to su SDA (*engl. Serial Data Line* – serijska podatkovna linija) i SCL (*engl. Serial Clock Line* – serijska impulsna linija). No, AM2302 koristi samo jednu žicu za razmjenu, a to je SDA.

SDA linija je inače spojena na +5V putem pull-up otpornika, tako da kada se ne šalju podatci kroz nju napon na njoj je +5V, odnosno logička jedinica. Kada mikrokontroler želi da mu senzor pošalje podatke, trebao bi na SDA liniji postaviti logičku nulu najmanje 800 μs. Taj signal senzor shvaća kao zahtjev za slanjem podataka. Senzor odgovara impulsom tako da postavi logičku nulu na SDA liniju u trajanju od 80 μs i logičku jedinicu koja traje isto tako 80 μs. Taj princip prikazan je na slici 3.8.



Slika 3.8: Komunikacija između mikrokontrolera i senzora, [6].

Nakon signala odgovora sa senzora, senzor šalje 40 bitova podatka, od kojih se 16 bitova koristi za vlažnost zraka, 16 bitova za temperaturu zraka, i 8 bitova za paritetni bit.

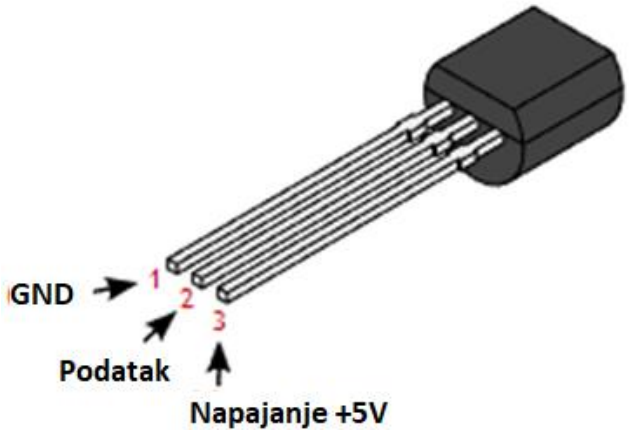
Primjer izračuna vlažnosti zraka i temperature na osnovu primljenih podataka:

Vlažnost zraka: 00000010 10010010 = 0292H (Heksadecimalno)= 2×256 + 9×16 + 2 = 658
=>Vlažnost zraka = 65,8%

Temp. zraka: 00000001 00001101 = 10DH (Heksadecimalno) = 1×256 + 0×16 + 13 = 269
=>Temperatura = 29,6°C

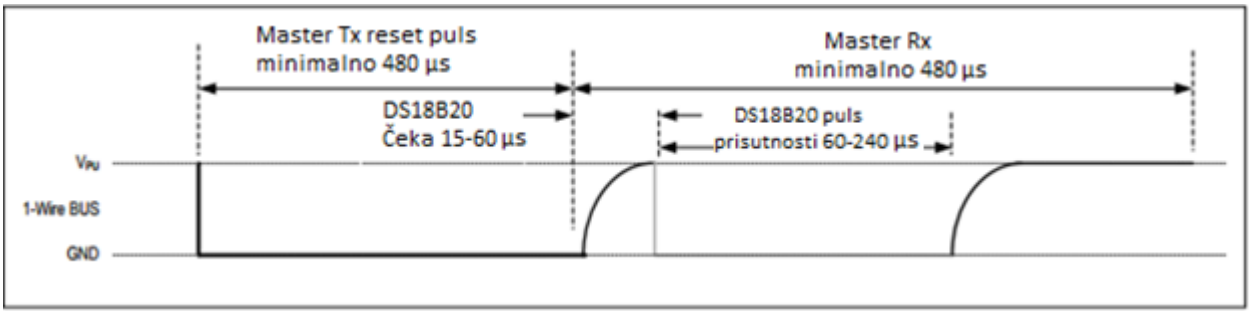
DS18B20 je višenamjenski temperaturni senzor tvrtke maxim integrated. Više namjenski zato što ga je moguće implementirati u razne sustave. On na svom izlazu daje temperaturna očitanja preciznosti 9 do 12 bita. Ima mogućnost programiranja alarma koji se aktivira na korisnički definiranim razinama temperature.

Komunikacija korištena za prijenos temperature između senzora i mikrokontrolera ista je kao i kod prethodno spomenutog senzora AM2302, pojednostavljena I²C komunikacija. Na slici 3.9 prikazan je izgled i raspored pinova temperaturnog senzora DS18B20.



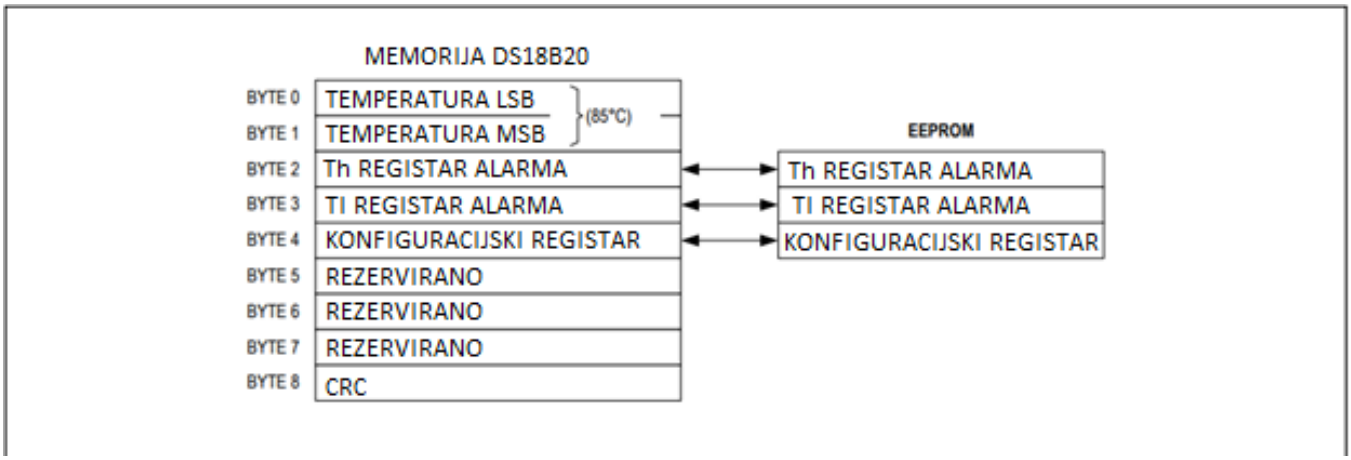
Slika 3.9: DS18B20 temperaturni senzor.

Komunikacija između mikrokontrolera i senzora započinje slanjem zahtjevnog pulsa koji je ujedno i provjera je li spojen senzor na mikrokontroler. Ova sekvenca prikazana je na slici 3.10.



Slika 3.10: Početak komunikacije između senzora i mikrokontrolera, [16].

Vidimo da mikrokontroler postavlja 0 na ulaz senzora u trajanju od 480 mikro sekundi, te postavlja logičku jedinicu ponovno na ulaz senzora. Nakon toga senzor (ako je spojen na sabirnicu) postavlja logičku nulu na sabirnicu, u trajanju od 60 do 240 mikro sekundi, te ponovno vraća logičku jedinicu. Na taj način mikrokontroler ima informaciju da je senzor prisutan na sabirnici te da može započeti očitanje temperature. Ova provjera provodi se prilikom prvog uključenja mikrokontrolera. Prilikom uključenja također u registre senzora mora se upisati konfiguracija (temperatura za alarm, koristi li se temperatura u °C ili °F, itd). Na slici 3.11 prikazana je organizacija registara unutar senzora.



Slika 3.11: organizacija registara unutar senzora, [16].

Memorija senzora sastoji se od 8 bajtova (64 bita) obične SRAM memorije i 3 bajta (24 bita) eeprom memorije. Razlika je što SRAM memorija gubi podatke nakon gubitka napajanja.

Bajt 0 i 1 memorije sadržavaju MSB i LSB temperature, to su dijelovi memorije koji služe samo za iščitavanje (*engl. read-only* – samo za iščitavanje) od strane mikrokontrolera.

Bajt 2 i 3 sadržavaju podatke o nižoj i višoj temperaturi koja aktivira alarm senzora. Kada se podatci upišu u ove memorijske lokacije oni se, upisuju u eeprom memoriju kako bi ostali sačuvan i nakon gubitka napajanja. Upis u eeprom memoriju mora biti aktiviran od strane mikrokontrolera.

Bajtovi 5, 6 i 7 su rezervirani za korištenje od strane procesora unutar senzora i oni se ne mogu konfigurirati niti izmijeniti od strane korisnika.

Bajt 8 sadrži CRC kod za bajtove od 0 do 7.

Ova konfiguracija provodi se samo prilikom prvog uključenja mikrokontrolera odnosno temperaturnog senzora. Nakon toga mikrokontroler može očitavati vrijednosti iz prva dva bajta memorije senzora, obraditi ih i prikazati korisniku.

YF-S201 senzor (slika 3.12), kao što je moguće vidjeti na slici 3.6, se nalazi na vodenoj cijevi skupa sa pumpom. U sebi sadrži kotač sa lopaticama koji se prilikom prolaska vode okreće. U kotaču se nalazi integrirani hall senzor koji na izlazu daje električni puls svaki puta kada se kotač okrene za jedan krug. Hall senzor je zaliven u plastiku i izoliran od utjecaja vode.

Slika 3.12: Izgled senzora protoka YD-S201, [10].

Senzor dolazi sa 3 žice, koje su: crvena (napajanje 5-25V), crna (uzemljenje) i žuta (izlaz iz hall senzora). Brojeći impulse sa izlaza hall senzora može se jednostavno dokučiti količina vode koja protječe kroz senzor. Svaki puls (1 okretaj kotača) je približno 2,25 mililitara vode.

Važno je napomenuti da ovo nije precizan senzor i da točnost ovisi o položaju senzora, količini voda koja protječe, tlaku vode, itd… Preciznost senzora je oko 10% očitane vrijednosti, ali za potrebe ovog sklopa gdje nije potrebna veća preciznost sasvim je dovoljan.

Izlazni signal je kvadratnog oblika, pa ga je lako bilježiti i brojati putem mikrokontrolera. Izračun protoka se vrši prema formuli (6-1):

$$Protok = \frac{Frekvencija\ signala\ [Hz]}{7{,}5}\ \ [{}^{l}/_{min}] \qquad (6\text{-}1)$$

Za potrebe mjerenja razine tekućine u nekom spremniku u sustav je implementiran analogni senzor pritiska. Sa promjenom razine tekućine mijenja se pritisak tekućine na stjenku spremnika, a time i na senzor, odnosno njegove izlazne veličine. Izgled korištenog senzora prikazan je na slici 3.13.

Slika 3.13: Analogni senzor pritiska.

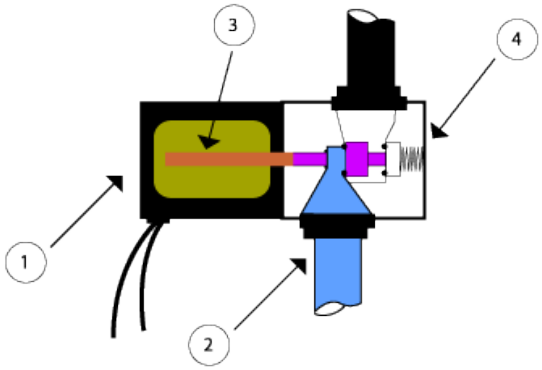
Ovaj senzor u sebi sadrži membranu koja se pod tlakom savija i mijenja svoj otpor. Promjenom tog otpora senzor na izlasku daje određeni napon u iznosu od 0 do 4.5 V. Dakle u shemi se senzor može prikazati promjenjivim otpornikom. Zbog toga ima tri žice. Crvena i crna su korištene za napajanje senzora, a žuta za izlazni signal.

Osnovne karakteristike senzora:

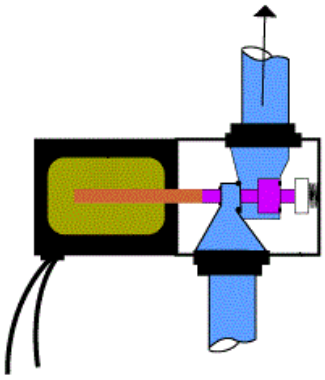
- Napon napajanja: 5V
- Izlazni napon 0.5-4.5V
- Struja napajanja <10 mA
- Radno područje 0-1.2 MPa
- Maksimalni pritisak: 2.4 MPa
- Preciznost mjerenja: +/- 1.5%
- Vrijeme odziva: <2 ms

3.2.3 Aktuatori

Zadaća ventila jest u pravom trenutku uključiti ili isključiti navodnjavanje nekog sektora. Protok vode upravljan je elektromagnetnim ventilom preko izvršne jedinice. Na slikama 3.14 i 3.15 može se vidjeti presjek ventila.



Slika 3.14: Elektromagnetni ventil u zatvorenom položaju, [11].



Slika 3.15: Elektromagnetni ventil u otvorenom položaju, [11].

Na slici 3.14 ventil je u zatvorenom položaju. Njegovi osnovni dijelovi su: solenoid (1), dovodna cijev (2), iglica solenoida (3), opruga (4). Dok je u zatvorenom položaju, opruga drži iglicu solenoida u krajnjem lijevom položaju i zadebljanje na iglici zadržava protok vode. Dolaskom napajanja (12V) na solenoid, stvara se magnetsko polje privlači u svoje magnetsko polje, rezultat toga je pomicanje iglice u desnu stranu i ostvarenje toka vode. Nestankom napajanja opruga vraća iglicu u početni položaj i zaustavlja protok vode.

Ventil ima definirani smjer protoka, pa se prema tome mora poštovati orijentacija ventila prilikom ugradnje u sustav navodnjavanja. Upravljanje solenoidom ne može se vršiti direktnim spajanjem na mikrokontroler, zbog toga se koristi spoj sa sheme 3.1.

Vodena pumpa je jedan od glavnih dijelova ovoga sustava. Njezina zadaća je da na osnovu naredbe od strane mikrokontrolera ispumpava određenu količinu vode iz spremnika vode u polje, bilo putem plastičnih cijevi ili nekih drugih oblika prskalica. Njezin izgled prikazan je na slici 3.16:

Slika 3.16: Izgled vodene pumpe, [17].

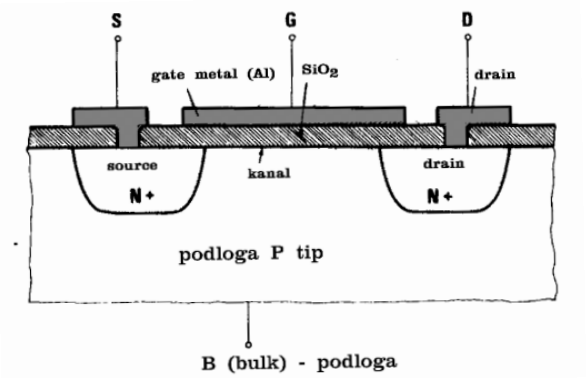
Vodena pumpa se sastoji od lopatica i kućišta. Uslijed rotacije lopatica stvara se razlika tlakova koje pojačavaju protok vode. Ukoliko je voda punu kamenca uslijed kavitacijske erozije strujanjem takve vode po lopaticama dolazi do oštećivanja te loma lopatica odnosno kvara vodene pumpe. Materijali od kojih su izrađene lopatice su većinom neke smjese poli etilena, upravo zbog svoje niske gustoće i mase, čime smanjuju potrošnju električne energije kao i prihvatljive cijene i zadovoljavajućih svojstava.

Važno je napomenuti da pumpa posjeduje IP68 certifikat, što znači da su njezini konektori unutar kućišta i motor u potpunosti vodonepropusni.

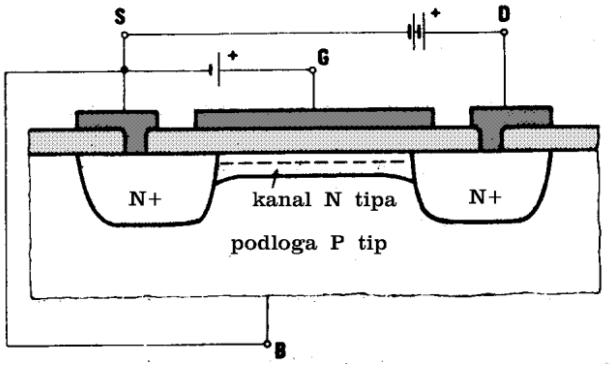
Karakteristike korištene vodene pumpe:

- Promjer cijevi: ½“
- Napon napajanja: 24V
- Struja: 0.75A
- Maksimalni protok: 600 l/h
- Maks. temp. vode: 100°C

MOSFET je upravljačka jedinica koja upravlja radom pumpe. Osim naziva mosfet (*engl. Metal Oxide Semiconductor Fiel Effect Transistor* – metal oksidni poluvodički tranzistor s efektom polja), može se naći naziv tranzistor s efektom polja s izoliranom upravljačkom elektrodom, a proizvode se u dva tipa: s kanalom N tipa i s kanalom P tipa.



Slika 3.17: Struktura MOSFET tranzistora, [18].



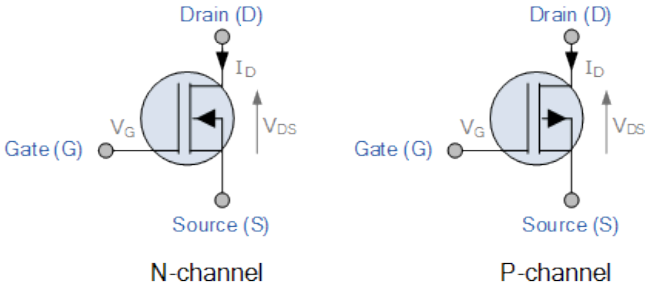
Slika 3.18: Spoj i formiranje kanala na MOSFET tranzistoru, [18].

Na slici 3.17 prikazana je struktura MOSFET tranzistora s efektom polja sa kanalom N tipa. U normalnom radu napon između izvoda (D) i izvora (S) priključen je tako da pozitivan pol dođe na priključak izvoda (D), pa ako je pri tome priključak upravljačke elektrode (G) slobodan, dva N područja i zajednička podloga P tipa predstavljaju dvije PN diode (suprotno orijentirane) između kojih je serijski vezan otpornik. Kako oba spoja ne mogu biti propusno polarizirana, tako se cijeli spoj ponaša kao veliki otpor između izvoda (D) i izvora (S).

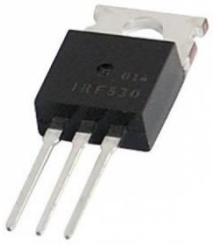
Ako se podloga P tipa poveže na priključak izvora (S), što se može vidjeti na slici 3.18, na PN spoju između izvora i podloge nema napona, a PN spoj između izvoda (D) i podloge ostaje invertno polariziran. Dovođenjem napona na upravljačku elektrodu (G) s pozitivnim potencijalom u odnosu na izvor (S), u podlozi se stvaraju negativni nosioci naboja – elektroni, koji za podlogu P tipa predstavljaju manjinske nosioce. Oni će biti privučeni na metalnu ploču, koja je na pozitivnom potencijalu, ali nju ne mogu stići jer se između ploče i podloge nalazi izolirajući oksid, tako da se oni skupljaju ispod ploče u području između izvoda (D) i izvora (S). Skupljajući se ispod ploče na taj način se formira kanal N tipa kako je prikazano na slici 3.18.

Jakost struje izvoda (D) je to veća što je veći broj nosilaca stvorenih naboja na strani upravljačke elektrode (G), a veći pozitivni potencijal na njoj stvara veći broj nosilaca naboja. Potencijalom na upravljačkoj elektrodi (G) je tako moguće upravljati strujom izvoda (D).

Na slici 3.19 prikazan je simbol za mosfet tranzistore koji se koristi prilikom izrade shema. Lijevo na slici prikazan je N-kanalni mosfet a desno P-kanalni simbol. Na slici 3.20 može se vidjeti fizički izgled mosfet tranzistora korištenog u izvršnoj jedinici za upravljanje pumpama.

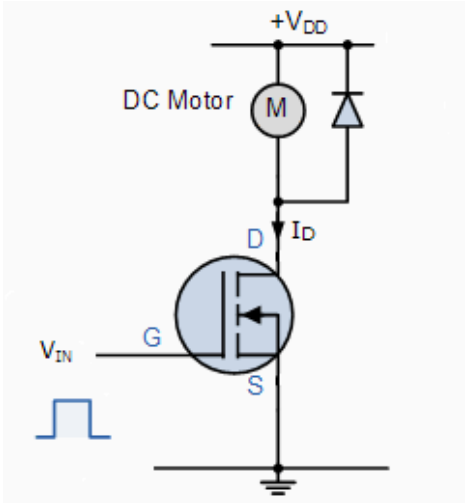


Slika 3.19: Shematski simbol MOSFET tranzistora, [5].



Slika 3.20: Izgled MOSFET tranzistora IRF530, [5].

U sustavu upravljanja pumpama za navodnjavanje, pumpe se napajaju sa 24V i 1A, što je mnogo više nego što mikrokontroler može dati na svojim izlazima (5V i 30mA). Prije je u tekstu spomenuto da se taj problem rješava upotrebom mosfeta. Shema spajanja mosfeta, mikrokontrolera i motora je prikazana na shemi 3.1.



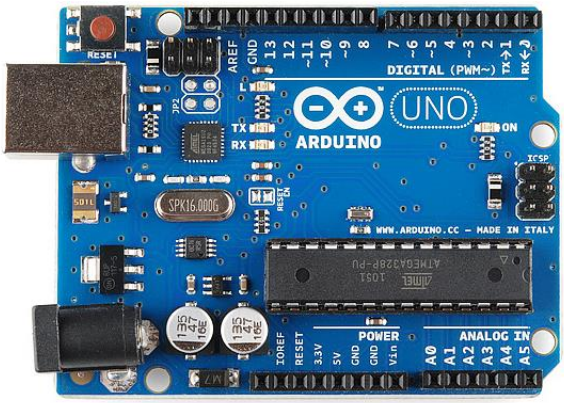
Shema 3.1: Spajanje N-kanalnog MOSFET tranzistora sa mikrokontrolerom i motorom, [5].

Na ovaj način postignuto je upravljanje naponom od 0 do 24 V sa naponom od 0 do 5V, to se može smatrati nekom vrstom pojačanja. Tako se postiže upravljanje sa motorom pumpe i zavojnicom elektromagnetnog ventila. Shema je vrlo jednostavna, sastoji se od N – kanalnog mosfeta i jedne diode. Ovim postignuto je analogno upravljanje brzinom motora, odnosno pumpe, uz manje troškove nego da je upravljanje realizirano analognim sklopovima.

Spajanje pumpe i elektromagnetnog ventila ovom shemom omogućeno je njihovo upravljanje PWM signalom sa mikrokontrolera. Ovaj mosfet može izdržati struje do 14 A i napone do 100 V. To u potpunosti zadovoljava uvjete za upravljanjem elektromagnetnog ventila i pumpe.

3.2.4 Upravljačka jedinica

Arduino je ime za otvorenu računarsku i softversku platformu koja omogućava dizajnerima i konstruktorima stvaranje uređaja i naprava koje omogućava spajanje računala s fizičkim svijetom tj. stvaranje interneta stvari. Arduino je stvorila talijanska tvrtka SmartProjects 2005. rabeći 8-bitne mikrokontrolere Amtel AVR, da bi stvorili jednostavnu malu i jeftinu platformu s kojom bi mogli lakše povezivati računala s fizičkim svijetom. Dizajneri su izabrali ime Arduino po imenu kafića u kojem su se dizajneri sastajali kada su stvarali projekt.



Slika 3.21: Arduino UNO, [1].

Kako je Arduino platforma open-source tipa, odnosno dozvoljeno je njezino dijeljenje i preuređivanje u svrhu kreiranja novih platforma koje su međusobno kompatibilne, tako da su razvojem nastale još mnoge inačice razvojnih okruženja baziranih na Arduino platformi. Jedna od njih je i hrvatska verzija Arduina – Croduino razvijena od strane e-radionice. Arduino platforma je razvijena u nekoliko inačica: UNO, MEGA, NANO, LEONARDO, MICRO, itd…

Srce Arduina jesu mikrokontroleri. Mikrokontroler je malo računalo sadržano na jednom integriranom sklopu. Arduino okruženje najčešće koristi 8 bitne mikrokontrolere koje proizvodi tvrtka ATMEL. Najrasprostranjeniji model je ATMEGA328P koji se koristi na osnovnoj Arduino prototipnoj pločici koja je prikazana na slici 3.21.

Mikrokontroler ima određen broj izvoda kojima je moguće upravljati pomoću programa kojega korisnik napiše na računalu i koji se onda izvodi na samom mikrokontroleru. Programiranje mikrokontrolera se obavlja u prilagođenoj verziji programskog jezika C++. Ta prilagođena verzija za pisanje programa razvijena je od stane arduina i naziva se arduino IDE. Ovaj softver je open source tipa i radi na svim platformama – Windows, Linux i Mac operativnom sustavu. Za prebacivanje softvera sa računala na mikrokontroler koristi se USB veza sa računalom.

AT mega 328p vrlo je popularan mikrokontroler, a sastavni je dio Arduino UNO-a. U sustavu automatiziranog upravljanja pumpama za navodnjavanje on se koristi u arduino razvojnoj pločici. Ima 28 pinova od kojih je 14 digitalnih ulaza ili izlaza, od čega 6 PWM izlaza i 6 analognih ulaza. U izvršnoj jedinici sustava se koriste svi navedeni pinovi što ukazuje na činjenicu kako je mikroprocesor u potpunosti iskorišten. Karakteristike arduino uno razvojne pločice i njezinog mikrokontrolera dane su u tablici 3.1.

Tablica 3.1: Karakteristike razvojne ploče arduino UNO, [1].

Mikrokontroler	ATmega328P
Radni napon	5 V
Ulazni napon (preporučeno)	7-12 V
Ulazni napon (granica)	6-20 V
Digitalni I/O pinovi	14 (od kojih su 6 PWM)
PWM izlazni pinovi	6
Analogni ulazni pinovi	6
Maksimalna struja po pinu	20 mA
Maksimalna struja za 3.3V pin	50 mA
Flash memorija	32KB (0.5KB bootloader)
SRAM	2KB
EEPROM	1KB
Radni takt	16 MHz

Mikrokontroler radi na naponskoj razini od 5V pa treba paziti prilikom spajanja vanjskih elemenata da naponska razina ne prelazi radnu naponsku razinu, kako bi se izbjegla oštećenja.

3.2.5 Komunikacijsko sučelje

Primopredajni mrežni modul nRF905 je najvažnija komponenta, i nalazi se u svakoj jedinici. Omogućava komunikaciju i prijenos podataka među jedinicama, a neke njegove posebne mogućnosti iskorištene su u svrhu manjeg opterećivanja mikrokontrolera. Kratki opis modula i izgled modula prikazani su izvatkom iz službene tehničke dokumentacije proizvođača nordic semiconductor na slici 3.22 i kroz cijelo poglavlje.

Slika 3.22: Izgled modula, [3].

Osnovne značajke modula:

- Malo pakiranje čipa, 32-pinsko pakovanje (32L QFN 5x5mm).
- Napon napajanja između 1,9 i 3.6V
- Moguć rad na više kanala
- Vrijeme prebacivanja između kanala <650 μs
- Promjenjiva izlazna snaga do 10 dBm
- Detekcija dolaznog paketa
- Automatski CRC generator
- Niska potrošnja struje prilikom slanja (do 9 mA)

Modul ima niži napon napajanja (1.8 do 3.6 V) od mikrokontrolera (5V). Prednost nižeg napona napajanja je manja potrošnja energije, no nedostatak je što se za spajanje ovog modula sa mikrokontrolerom mora koristiti naponsko dijelilo radi prilagodbe naponske razine.

Tablicama (Tablica 3.2 i 3.3) prikazana je potrošnja modula pri različitim načinima rada. Uz to, modul ima programibilno izlazno RF pojačalo koje može izračiti snagu na anteni do +10dBm što pokriva zahtjev za dometom od 1 km.

Tablica 3.2: Osnovne karakteristike nRF905, [3].

Parametar	Vrijednost	Mjerna jedinica
Minimalni napon napajanja	1.9	V
Maksimalna snaga odašiljanja	10	dBm
Brzina prijenosa	50	kbps
Potrošnja struje pri odašiljanju podataka (-10dBm snaga odašiljanja)	9	mA
Potrošnja struje prilikom primanja podataka	12.5	mA
Temperaturni opseg	-40 do +80	°C
Potrošnja struje u mirovanju	2.5	μA

Tablica 3.3: Potrošnja struje za razne načine rada modula, [3].

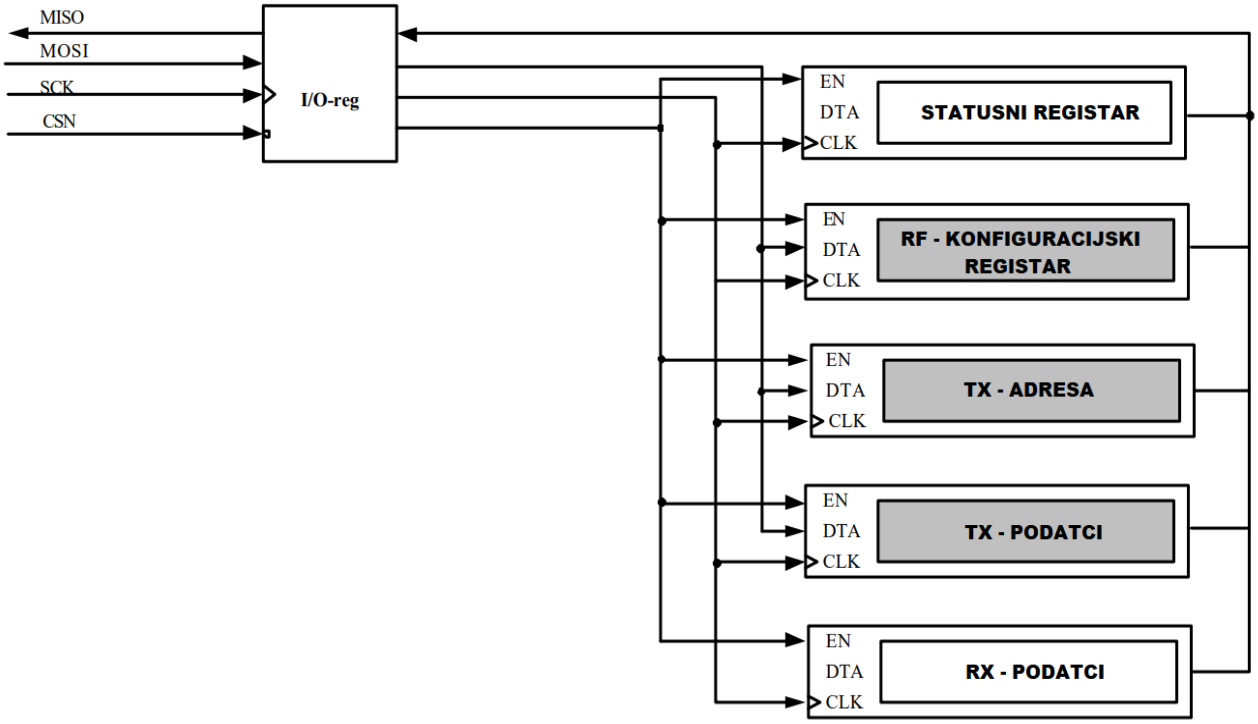
Postavke napajanja	Izlazna snaga	Potrošnja struje
00	-10dBm	9.0 mA
01	-2 dBm	14.0 mA
10	6 dBm	20.0 mA
11	10 dBm	30.0 mA

Napon napajanja: 3.0V, Ta = 27°C, Impedancija antene= 400 Ω

Glavna komponenta modula je integrirani krug nRF905 tvrtke nordic semiconductor, koji koristi tehnologiju prijenosa ShockBurst™. ShockBurst™ tehnologija omogućuje brzi prijenos podataka sa korištenjem jeftinijih mikrokontrolera (poput korištenog ATmega 328P). Korištenjem ove tehnologije omogućava se komunikacija sa mikrokontrolerom putem SPI protokola, tu vrstu protokola danas podržava većina mikrokontrolera. Integrirani krug nudi 3 frekvencije rada: 433, 868 i 915 MHz.

U ShockBurst™ modu rada AM (Address Match) i DR (Dana ready) pinovi obavještavaju mikrokontroler o stanju modula i prisutnosti paketa. Na taj način mikrokontroler može biti u stand-by načinu rada, što znači da štedi energiju.

Sva konfiguracija i inicijalizacija nRF905 kontrolera se obavlja putem SPI sučelja. Sučelje se sastoji od ukupno pet registara, njihov raspored prikazan je na slici 3.23:



Slika 3.23: Raspored registara integriranog kruga nRF905,[3].

Registar je brza memorija u procesoru ili drugoj jedinici koja služi za privremeno pohranjivanje određenih podataka i obavljanje aritmetičkih i ostalih operacija; spremnik. Registri koje sadrži ovaj sklop su:

- **Statusni registar** – Sadrži statuse izlaznih pinova Data Ready (DR) i Address Match (AM)
- **RF – Konfiguracijski registar** – Registar koji sadrži informacije o postavkama kao što su frekvencija rada, izlazna snaga i slično…
- **TX-Adresa –** Registar koji sadrži informacije o adresi na koju će modul poslati podatak.
- **TX-Podatci** – U ovaj registar se sprema podatak koji će biti poslan u mrežu, na prethodno definiranu adresu.
- **RX-Podatci -** U ovaj registar se pohranjuju podatci koji dođu sa drugih modula.

Cijeli sklop, kao što je prije spomenuto, komunicira kroz SPI protokol. Protokolom se šalju naredbe na koje nRF905 odgovara. Lista naredbi i njihov opis za SPI protokol prikazana je u tablici 3.4. Kada je CSN pin SPI protokola postavljen u logičku „0“, nRF905 očekuje naredbu. Prije slanje nove naredbe potrebno je postaviti CSN signal u logičku „1“ pa ponovno u „0“.

Tablica 3.4: Lista naredbi za SPI komunikaciju, [3].

Popis instrukcija za SPI komunikaciju sa nRF905		
Naziv naredbe	**Format naredbe**	**Opis operacije**
W_CONFIG (WC)	0000 AAAA	Zapis u konfiguracijski registar. AAAA označava od kojeg bajta operacija zapisivanja počinje. Broj bajtova ovisi o početnoj adresi AAAA
R_CONFIG (RC)	0001 AAAA	Iščitavanje konfiguracijskog registra. AAAA označava od kojeg bajta operacija iščitavanja počinje. Broj bajtova ovisi o početnoj adresi AAAA
W_TX_PAYLOAD (WTP)	0010 0000	Zapis TX - podataka: 1-32 bajta. Operacija za zapis uvijek započinje na bajtu 0.
R_TX_PAYLOAD (RTP)	0010 0001	Iščitavanje TX - podataka: 1-32 bajta. Operacija za iščitavanje uvijek započinje na bajtu 0.
W_TX_ADDRESS (WTA)	0010 0010	Zapis TX - adrese: 1-4 bajta. Operacija za zapis uvijek počinje na bajtu 0.
R_TX_ADDRESS (RTA)	0010 0011	Iščitavanje TX - adrese: 1-4 bajta. Operacija za iščitavanje uvijek počinje na bajtu 0.
R_RX_PAYLOAD (RRP)	0010 0100	Iščitavanje RX - Podatka: 1-32 bajta. Operacija za iščitavanje uvijek počinje na bajtu 0.
CHANNEL_CONFIG (CC)	1000 pphc cccc cccc	Serijska naredba za brzo postavljanje mrežnog kanala, frekvencije rada i moda napajanja u konfiguracijskom registru. CH_NO = cccccccc, HFREQ_PLL = h PA_PWR = pp

3.3 Shema sustava

Shema sustava centralne jedinice skupa sa nacrtom tiskane pločice i popisom komponenata nalazi se u prilogu P9.3.

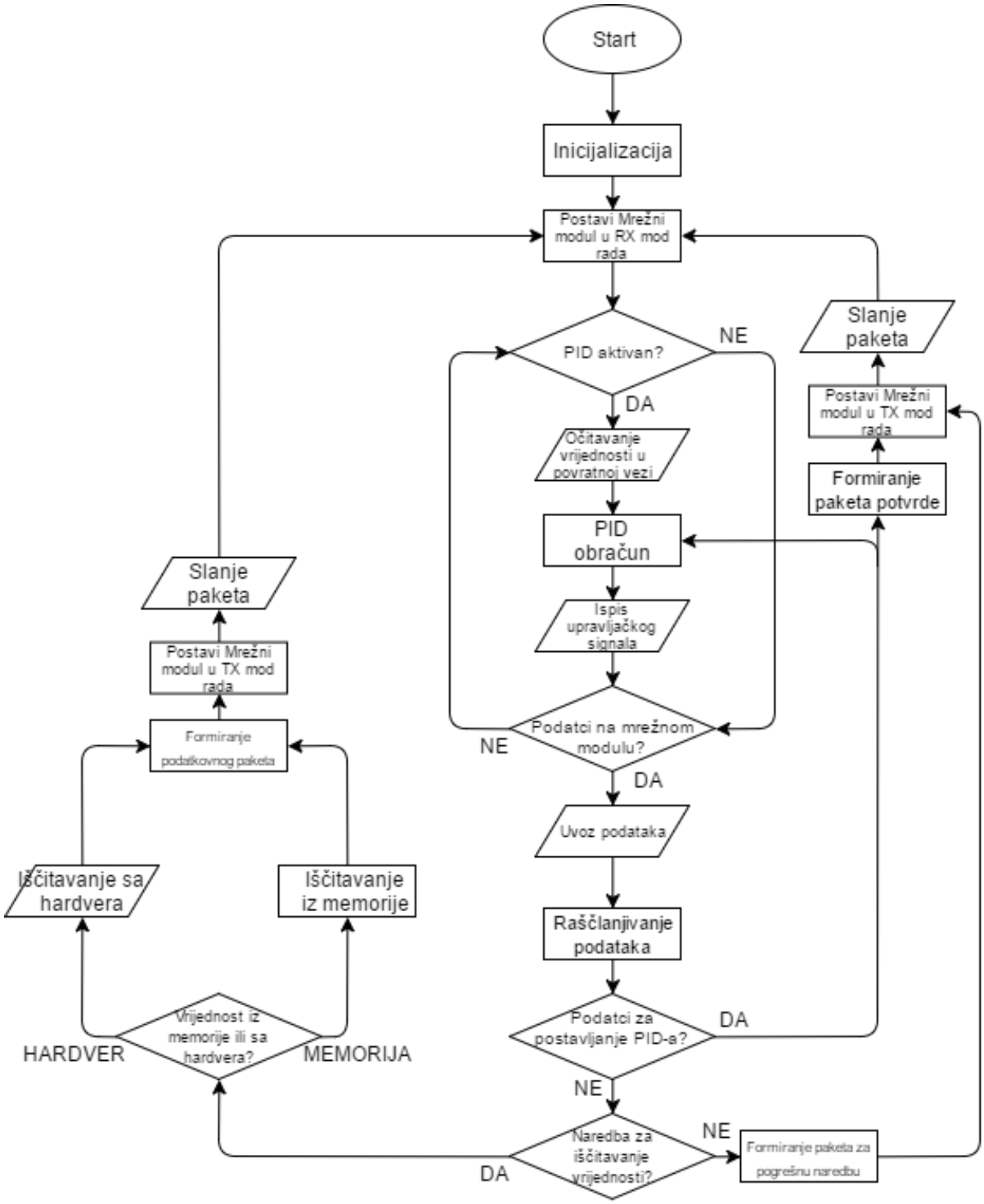
Shema sustava izvršne jedinice skupa sa nacrtom tiskane pločice i popisom komponenata nalazi se u prilogu P9.4.

Shema sustava nacrtana je u računalnom programu tvrtke Autodesk, EAGLE. Nacrtana je pregledno, u raznim komponentama. Kako bi sustav bilo lakše implementirati, vanjski senzori u shemi predstavljeni su konektorima (kao i na pločici).

3.4 Algoritam upravljanja

3.4.1 Upravljanje sustavom pumpi

Na slici 3.24 prikazan je blokovski dijagram toka programa za mikrokontroler u izvršnoj jedinici (prilog P9.2), ne razlikuje se puno od centralne jedince. Ima puno više komunikacije sa hardverskim djelom, iz tog razloga se i zove izvršna jedinica.



Slika 3.24: Prikaz blokovskog dijagrama toka programa za mikrokontroler u Izvršnoj jedinici.

Program započinje uključenjem mikrokontrolera, nakon toga pokreće se postupak inicijalizacije sa hardverom. Na početku mrežni modul se odmah postavlja u RX način rada, odnosno način rada za primanje podataka. Prilikom uključenja PID regulator je ugašen. Pod pojmom „ugašen“ se smatra da mu je referentna vrijednost 0.

Dok je PID regulator neaktivan program zapravo samo čeka pojavljivanje podatka na mrežnom uređaju. Kada se pojavi podatak, pomoću CM bajta u paketu se razlučuje koja je iduća zadaća mikrokontrolera. Ako mu je u paketu stigao podatak za postavljanje referentne vrijednosti PID-a, on se postavi a time se i pali PID regulator. Prilikom aktivacije regulatora, program šalje putem mrežnog modula podatak o aktivaciji regulatora. Podatak se prima u centralnu jedinicu i on označava uspješnu aktivaciju regulatora.

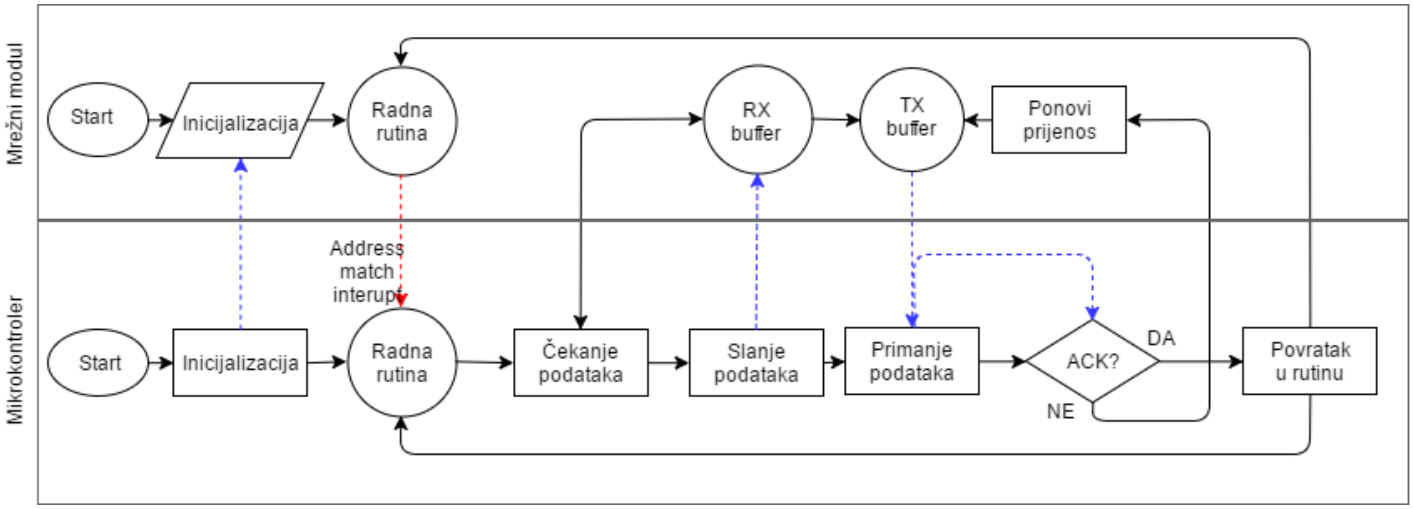
Nakon aktivacije PID-a, program se izvodi tako da regulator uzima podatke procesa, u ovom slučaju podatke sa senzora protoka, te na osnovu tih podataka izračunava potrebnu upravljačku vrijednost. Upravljačke vrijednost se u idućem koraku postavlja na PWM izlaz mikrokontrolera (vrijednosti od 0 do 255).

Ako se u paketu pojavi podatak za iščitavanje neke vrijednosti, razlučuje se radi li se o vrijednosti pojedinog senzora ili vrijednosti iz memorije. Vrijednosti koje smogu iščitati sa senzora su: temperatura zraka, vlažnost zraka, protok, temperatura medija, pritisak u spremniku. Vrijednosti koje se mogu dohvatiti iz memorije su: referentna vrijednost regulatora, trenutna izlazna vrijednost regulatora, otvorenost ventila, minimalna i maksimalna temperatura, minimalan i maksimalan protok, itd.… Nakon razlučivanja o kojem se podatku radi, podatak se dohvaća (ili sa hardvera ili iz softvera), formira se podatkovni paket sa iščitanom vrijednosti i šalje se putem mrežnog modula na adresu sa koje je stigao.

U slučaju da naredba ne spada niti u jedan od prethodna dva slučaja, ona se odbacuje. Potom se formira paket u kojem je specifičan odgovor za naredbe koje izvršna jedinica ne prepoznaje, taj paket se također šalje na adresu s koje je došao.

Svaki podatak koji dođe do izvršne jedinice se obradi i nakon obrade se uvijek šalje odgovor pošiljatelju. Bez obzira bio taj podatak koristan ili ne. Na taj način pošiljatelj (centralna jedinica) ima povratnu informaciju o stanju mreže.

3.4.2 Komunikacija i protokol



Slika 3.25: Blokovski prikaz komunikacije mikrokontrolera i mrežnog modula.

Prilikom uključenja mikrokontrolera, sa njegovog napajanja napaja se i mrežni modul (nRF905). Prvi korak je inicijalizacija mikrokontrolera, što znači da mikrokontroler inicijalizira svoje varijable i varijable mrežnog modula, te zna da je mrežni modul ispravan i spojen. Nakon toga mikrokontroler ulazi u radnu rutinu i obavlja neke druge zadatke dok čeka da mu dođe podatak sa mrežnog modula. Si koraci opisani u ovom poglavlju vidljivi su na slici 3.25.

Kada drugi mrežni modul šalje podatak na centralnu jedinicu, on uz podatak stavi adresu i odašilje ga. Kada se podudaraju adresa centralne jedinice i odašiljanog podatka, mrežna jedinica pošalje signal (Address Match) mikrokontroleru koji prestaje sa izvođenjem programa rutine i prelazi na prikupljanje podataka iz memorije mrežnog modula (plave isprekidane linije). Pritom mikrokontroler u memoriju spremi adresu programa na kojoj je stao kako bi poslije mogao nastaviti sa izvođenjem programa na tom mjestu. Prikupljanje podataka je dvosmjerna komunikacija zbog toga što nakon slanja svakog paketa podataka mikrokontroler šalje potvrdu da je primio paket. Ukoliko dođe do greške i mikrokontroler pošalje potvrdu da nije primio paket, mrežni modul će ponovno poslati paket.

Protokol koji se koristi u ovoj komunikaciji je SPI (*engl. Serial Peripheral Interface bus* – serijska periferna sabirnica) protokol. Protokol koji je razvijen za kratke komunikacijske veze, primarno u manjim sustavima. Razvila ga je firma Motorola 1980-tih godina, i od tada je praktički postao standard za primjenu u ovakvim uređajima. Temeljen je na master – slave arhitekturi komunikacije.

Nakon što se svi paketi pravilno prime i mrežni uređaj dobije potvrdu za to (ACK), mikrokontroler i mrežni modul se vraćaju u svoju rutinu. Mikrokontroler nastavlja izvoditi program redoslijedom gdje je stao.

Komunikacijski protokol između izvršne i centralne jedinice ne razlikuje se od protokola između računala i centralne jedinice. Podatci se također pakiraju u pakete, koji su malo izmijenjeni. Zbog toga što komunikacijski modul ima svoju mrežnu adresu, kako bi svaki modul bio jedinstven, i ona se mora dodati u podatkovni paket. Tako da sada paket poprima ovaj oblik:

[CA |CM|b1|b2|b3|b4]

CA je jedan bajt podatka koji označava adresu pojedinog mrežnog modula na izvršnoj jedinici. Svaki modul u mreži ima drugačiju adresu, kako bi podatak poslan sa centralne jedinice došao točno na ciljani komunikacijski modul, a ne na neki drugi. Ovaj protokol nije standardiziran niti jednim sustavom, dogovoren je samo za potrebe ovog projekta.

U slučaju neuspjele komunikacije između centralne jedinice i upravljačke jedinice, centralna jedinica računalu vraća paket u kojemu su sve vrijednosti postavljene u „0“. Na taj način računalo zna da je došlo do problema u komunikaciji i ponavlja postupak komunikacije.

Ako se pojavi zahtjev za dodavanjem nove upravljačke jedinice u mrežu (pritisak tipkala na centralnoj i upravljačkoj jedinici), izvršna jedinica šalje centralnoj svoju adresu i sve podatkovne bajtove postavlja u „01|02|03|04“. Centralna jedinica pohranjuje adresu upravljačke jedinice i prosljeđuje računalu informaciju da je u mrežu dodana nova upravljačka jedinica.

Svaki podatkovni paket koji se šalje između jedinica ili između jedinica i računala u sebi sadrži jedan bajt naredbe koju izvršna jedinica mora obraditi. Ta naredba u paketu označena je nazivom CM i nalazi se na drugom mjestu s lijeve stane paketa:

[CA|**CM**| b1|b2|b3|b4]

Ova vrsta paketa se koristi za komunikaciju centralne jedinice sa računalom, odnosno zadavanje zadatka od stane korisnika na računalu. Naredbe se razlikuju po brojevima. Postoji nekoliko vrsta naredbi, podijeljene su u sektore na koje se baziraju. Lista naredbi je prikazana u tablici 3.5:

Tablica 3.5: Lista naredbi za Izvršnu jedinicu.

Komanda	Veličina podatka	Opis	Adresa (DEC)	Adresa (HEX)	Zahtjev	Struktura odgovora	Raspon vrijednosti
Set address	8 bit	Promjena adrese Izvršne jedinice	1	1	ca01cd000000	ca/01/cd/00/00/00	0-255
Flow read current	16 bit	Trenutni protok	21	15	ca1500000000	ca/15/02/cd/dd/00/00	0-8,5 l/min
Flow read cumulative	32 bit	Ukupni protok od uključenja pumpe	22	16	ca1600000000	ca/16/04/	0-4,294,967,295 l
Flow read max	16 bit	Maksimalni zabilježeni protok	23	17	ca1700000000	ca/17/02/cd/dd/00/00	0-8,5 l/min
Flow read min	16 bit	Minimalni zabilježeni protok	24	18	ca1800000000	ca/18/02cd/dd/00/00	0-8,5 l/min
Flow set reference for PID	16 bit	Postavljanje referentne vrijednosti protoka za PID regulator	25	19	ca19cddd0000	ca/19/02/cd/dd/00/00	0-8,5 l/min
Temp read current	16 bit	Očitavanje trenutne temperature medija	31	1F	ca1f00000000	ca/1F/02/cd/dd/00/00	0-125°C
Temp min	16 bit	Minimalna zabilježena temperatura	32	20	ca2000000000	ca/20/02/cd/dd/00/00	0-125°C
Temp max	16 bit	Maksimalna zabilježena temperatura	33	21	ca2100000000	ca/21/02/cd/dd/00/00	0-125°C
Current water pressure	8 bit	Trenutni pritisak vode (stanje u spremniku vode)	41	29	ca2900000000	ca/29/01/cd/00/00/00	0-100%
Min pressure	8 bit	Minimalni zabilježeni pritisak vode	42	2A	ca2a00000000	ca/2A/01/cd/00/00/00	0-100%
Max pressure	8 bit	Maksimalni zabilježeni pritisak vode	43	2B	ca2b00000000	ca/2B/01/cd/00/00/00	0-100%
Valve set opening	1 bit	Upravljanje ventilom	51	33	ca33cd000000	ca/33/01/cd/00/00/00	0 ili 1
Valve read	1 bit	Iščitavanje stanja ventila	52	34	ca3400000000	ca/34/01/cd/00/00/00	0 ili 1
Pump set speed	8 bit	Postavljanje brzine pumpe	61	3D	ca3dcd000000	ca/3D/01/cd/00/00/00	0-100%
Pump read current speed	8 bit	Očitavanje trenutne brzine pumpe	62	3E	ca3e00000000	ca/3E/01/cd/00/00/00	0-100%
Air temperature	16 bit	Očitavanje temperature zraka	71	47	ca4700000000	ca/47/02/cd/dd/00/00	0-80°C
Air humidity	8 bit	Očitavanje vlažnosti zraka	72	48	ca4800000000	ca/48/01/cd/00/00/00	0-100%

Legenda tablice 3.5:

- Struktura paketa za slanje: ca+cm+b1+b2+b3+b4
- Struktura odgovora: ca+cm+bb+b1+b2+b3+b4

gdje su:

- **ca** => adresa
- **cm** => Komanda
- **bb** => Broj bajtova u odgovoru
- **b1** => Bajt 1
- **b2** => Bajt 2
- **b3** => Bajt 3
- **b4** => Bajt 4

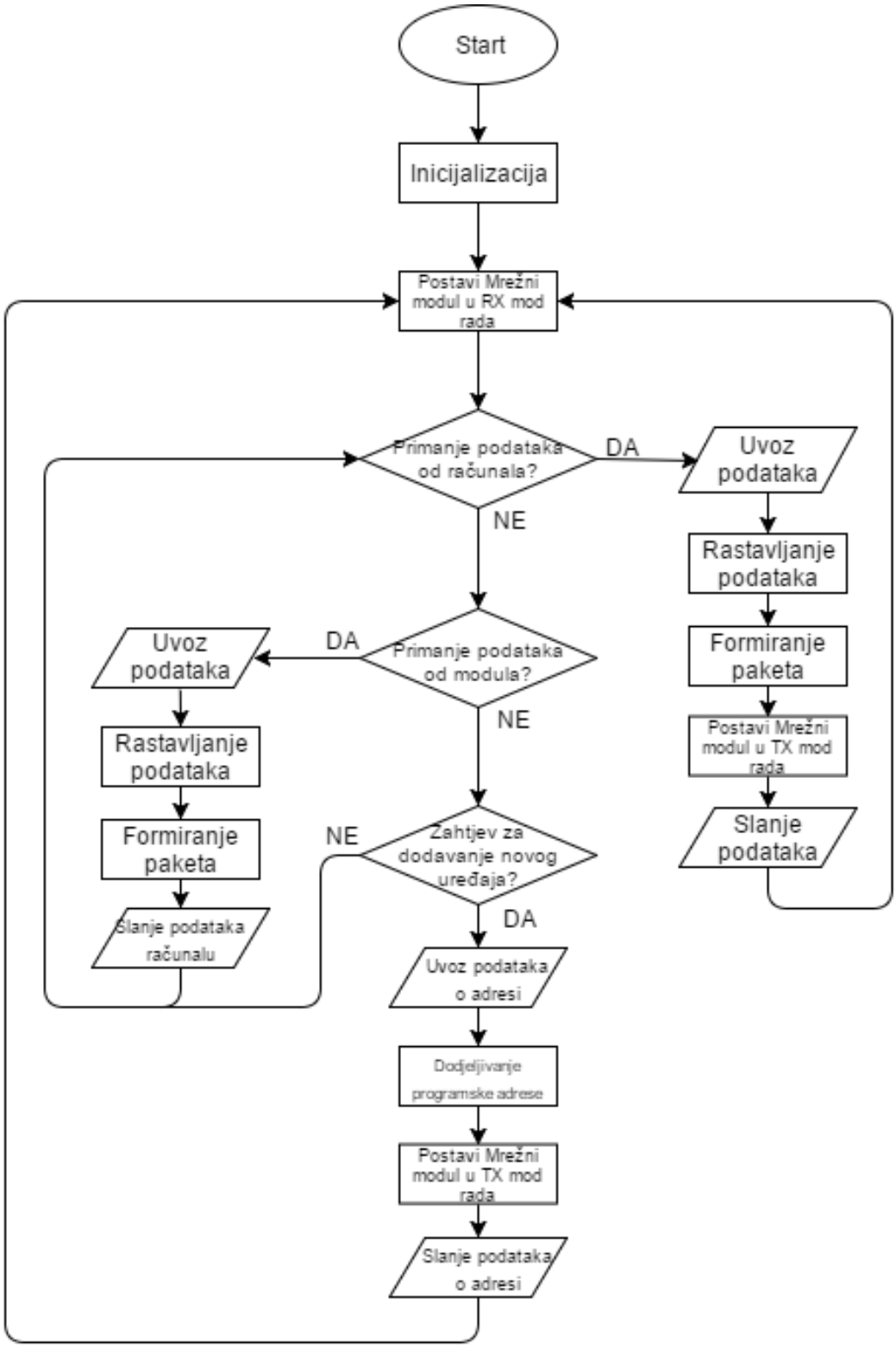
Kao što se može vidjeti u tablici, ima ukupno 18 naredbi. sve su numerirane brojevima od 1 do 72, ali ne redoslijedom. Brojevi se koriste iz razloga što ih je najlakše staviti u paket. Paket podržava brojeve od 0 do 255. Maksimalno je 255 zbog toga što se jedan bajt sastoji od 8 bitova, a sa 8 bitova može se ukupno napisati 256 kombinacija, $2^8 = 256$.

Naredbe se u paket stavljaju u brojevima koje je potrebno pretvoriti u heksadekadski brojevni sustav. Na taj način kada korisnik zadaje naredbu putem računala piše na primjer broj „72“ kao „48“.

Važno je da prilikom formiranja paketa obavezno na mjestu bajta CM budu dvije znamenke heksadekadskog broja. Odnosno, kod heksadekadskog broja 10 zapis u paket bi bio samo „A“. Ali zbog raščlanjivanja paketa na centralnoj jedinici obavezno je da budu dvije znamenke pa će biti potrebno broj 10 zapisati kao „0A“. Pošto prilikom preračunavanja iz heksadekadskog u dekadski sustav razlite težine (potencije) brojeva uzimaju s desne na lijevu stranu, nula ispred slova A ne remeti podatak pa i dalje vrijedi heksadekadski zapis broja 10.

3.4.3 Nadzor, upravljanje i prikaz procesnih parametara

Nadzor, upravljanje i prikaz procesnih parametara se provodi kroz centralnu jedinicu. Rutina rada mikrokontrolera centralne jedinice ustvari se sastoji od čekanja da se pojavi interupt za komunikaciju ili da se pojavi tekst na njegovom USB sučelju. Detaljniji grafički opis programa iz priloga P91 prikazan je na slici 3.26:



Slika 3.26: Prikaz blokovskog dijagrama toka programa za mikrokontroler u centralnoj jedinici.

Nakon početka programa, uređaj inicijalizira sve svoje varijable i komunikaciju sa vanjskom periferijom. U vanjsku periferiju centralne jedinice spadaju mrežni modul i USB sučelje sa računalom, te prije spomenuto tipkalo za dodavanje novog uređaja u mrežu.

Poslije inicijalizacije mikrokontroler šalje signal mrežnom modulu da se stavi u režim rada za primanje podataka, to znači da mrežni modul u tom režimu rada može samo primati poslane podatke od drugih jedinica, ne može ih sam slati. Ova mjera se provodi zbog hardvera samog mrežnog modula ali i kao neka vrsta sigurnosne mjere, odnosno kako ne bi došlo do greške u programu pa da sam program pošalje neke nepotrebne podatke.

Cijela radna rutina programa u mikrokontroleru, a tako i samog mikrokontrolera se provodi tako da se provjerava prisutnost signala. Odnosno, mikrokontroler provjerava koji su signali dostupni. Ako se pojavi podatak sa računala, mikrokontroler pregleda paket koji je došao sa računala, ustanovi o kojoj se izvršnoj jedinici radi, i razluči koju komandu računalo želi da izvršna jedinica obradi. Nakon toga u paket za slanje stavi informaciju o komandi i adresu izvršne jedinice koja treba obraditi komandu. Obavezno je potrebno prebaciti mrežni modul u TX (*engl. transmission* - odašiljanje) način rada, odnosno način rada u kojem mrežna jedinica šalje podatke. Nakon toga prebaci paket za slanje u memoriju mrežnog modula (nRF905) i da mu signal za slanje. Po završetku cijelog procesa potrebno je mrežni modul ponovno vratiti u RX (*engl. receive* - primanje) način rada, kako bi mogao primiti odgovor od izvršne jedinice. I mikrokontroler se vrača u rutinu. Ako kojim slučajem centralna jedinica ne primi odgovor od upravljačke jedinice u roku od pola sekunde, ona šalje računalu signal da veza nije uspostavljena.

Dok je u rutini, ako se pojavi signal na AM (*engl. Address Match* – podudaranje adrese) pinu mikrokontrolera, to signalizira prekid rutine i znači da je došao podatak na mrežni modul. Nakon nekoliko trenutaka mrežni modul daje signal da je primio paket putem pina DR (*engl. Data Ready* – podatci spremni). Idući korak mikrokontrolera je preuzimanje paketa, raščlanjivanje informacija, „prepakiranje“ u drugi okvir razumljiv računalu i na posljetku prosljeđivanje računalu.

Pritiskom tipkala za spajanje dodatne izvršne jedinice (na izvršnoj jedinici), rutina se prekida u trećem slučaju. Po prekidu rutine mikrokontroler uvozi podatke sa adresom iz mrežnog modula (koji je prethodno primio podatke od modula izvršne jedinice), dodjeljuje toj izvršnoj jedinici njezinu vlastitu adresu pomoću koje će komunicirati. Prosljeđuje tu adresu na mrežni modul koji se prebacuje u TX način rada i šalje adresu. Po završetku mikrokontroler vraća mrežni modul u RX način rada i vraća se u svoju rutinu.

3.5 Realizacija sustava

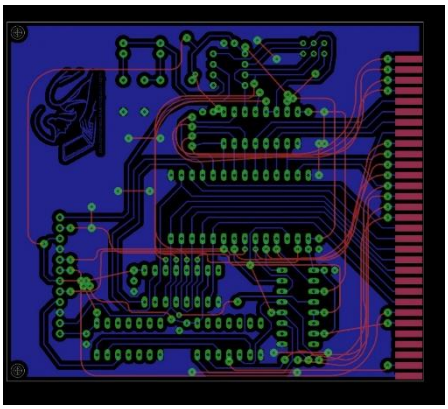
3.5.1 Izrada tiskane pločice

Tiskana pločica (*engl. PCB, printed circuit bord* – tiskana pločica) naziv je za sredstvo kojim se mehanički i električki povezuju elektroničke komponente. Sastoji se od podloge od izolatorskog materijala na kojoj se različitim postupcima oblikuje vodljiva struktura.

Podloga se tipično izrađuje od celuloznih vlakna impregniranih fenolnom smolom ("pertinaks") ili staklenih vlakana impregniranih epoksidnom smolom ("vitroplast"), međutim za više frekvencije koriste se i drugi materijali kao što su fluor polimeri i keramika.

Izrada tiskane pločice foto postupkom provodi se u nekoliko koraka:

1) Nacrt tiskane pločice – Nacrt tiskane pločice izrađuje se na osnovu sheme projekta na računalu ili ručno. Za ovaj sustav izrađena je tiskana pločica u softveru EAGLE (prilog P9.3 i P9.4).

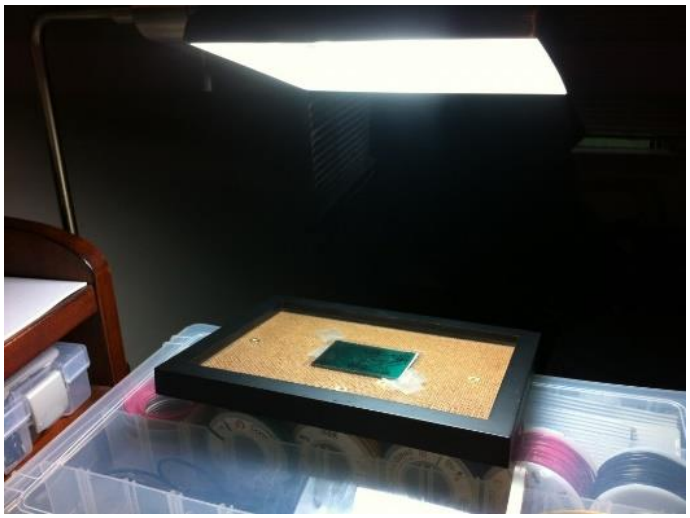


Slika 3.27: Primjer nacrtane tiskane pločice na računalu.

2) Ispis – Nakon što je pločica nacrtana na računalu, potrebno je isprintati nacrt vodova. Printa se na prozirnu foliju. Primjer je prikazan na slici 3.28:

Slika 3.28: Isprintan nacrt vodova na prozirnoj foliji.

3) Osvjetljavanje – Nakon printanja potrebno je prethodno kupljenu pločicu sa slojem UV foto osjetljivog laka izrezati i pripremiti za osvjetljavanje pod UV svjetlom. Priprema se svodi na skidanje zaštitnog sloja UV laka. Potrebno je isprintanu foliju staviti na pločicu, na stanu gdje je foto osjetljivi lak, i tako zajedno sa pločicom staviti pod izvor UV svjetla. Osvjetljavanje traje oko 10 minuta. Korak je prikazan na slici 3.29:



Slika 3.29: Osvjetljivanje pločice.

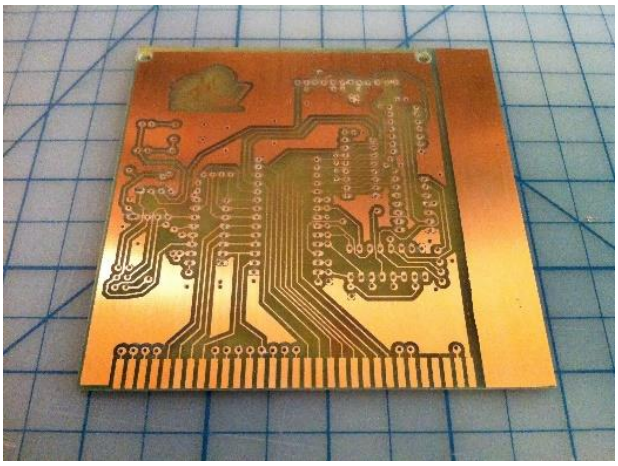
4) Razvijanje – U ovom koraku kemijski se odvaja osvjetljeni dio foto osjetljivog UV laka od ne osvjetljenog djela. Za skidanje laka koristi se NaOH. On reagira sa osvjetljenim lakom i na pločici ostaje samo dio laka do kojeg nije prodrla UV svjetlost. Prikaz pločice nakon 4. koraka je prikazan na slici 3.30:

Slika 3.30: Pločica nakon razvijanja.

5) Jetkanje – Jetkanje je također kemijski postupak, ali otklanjanja viška bakra sa pločice. U ovom postupku kiselina (mješavina hidrogena i solne kiseline) reagira sa bakrom koji na sebi nema foto osjetljivog sloja, odnosno uklanja sav nepotrebni bakar i ostavlja samo vodove. Ovaj korak prikazan je na slici 3.31:

Slika 3.31: Postupak jetkanja.

6) Bušenje – Posljednji korak izrade tiskane pločice je bušenje rupa kroz koje prolaze nožice komponenata i leme se na vodove. Nakon ovog koraka pločica je spremna za lemljenje komponenata. Ovaj korak prikazan je na slici 3.32:



Slika 3.32: Probušena tiskana pločica.

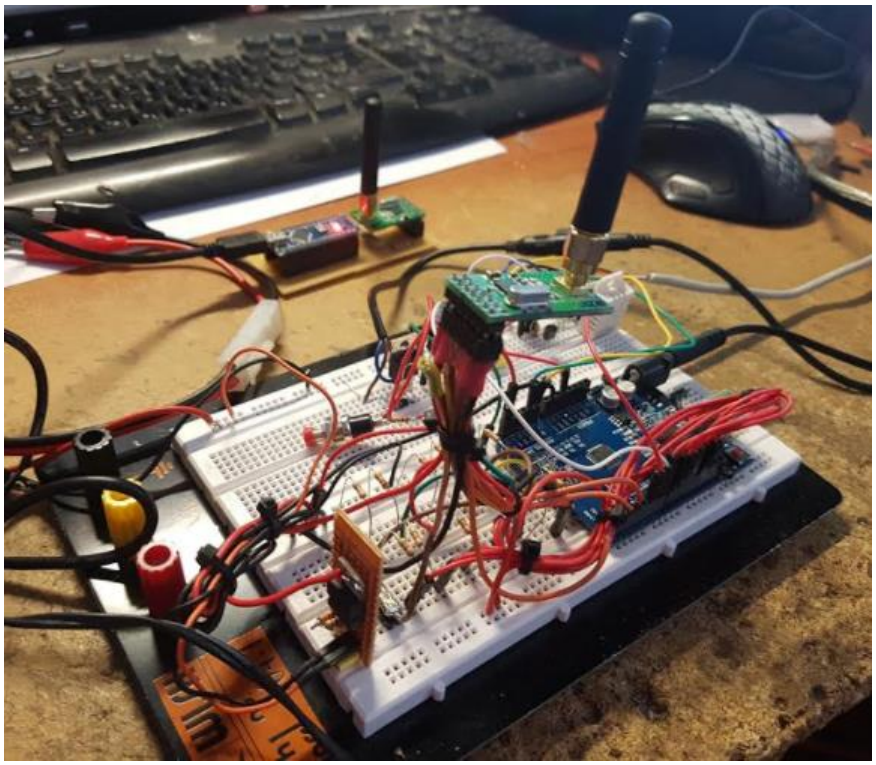
Važno je napomenuti da tokom cijelog procesa izrade zbog rada sa agresivnim kiselinama i opasnim ručnim alatima važno je upotrebljavati zaštite rukavice, naočale i zaštitnu masku kako bi se izbjegle trajne tjelesne ozljede.

3.5.2 Sastavljanje sustava

Polazište pri sastavljanju sustava je neka vizija sustava i dogovor sa mentorom u vezi funkcioniranja sustava. Njegove zadaće, njegova robusnost, mjesto montaže i sl. Neki od osnovnih zahtjeva sustava su da ima RF komunikaciju na 433MHz kako bi se izbjegla zagađenost etera na 2.4GHz zbog mogućih smetnji.

Idući korak bio je pronaći potrebne komponente koje bi odgovarale zahtjevima. Dogovoreno je da se za mrežnu komunikaciju koristi prethodno spomenuti mrežni modul nRF905. Za njegovo upravljanje odabrana je razvojna platforma arduino uno sa ATmega 328P mikrokontrolerom. Uz to, dogovoreno je da će sustav upravljati pumpom za vodu i elektromagnetnim ventilom, za čije je upravljanje potreban mosfet. Izabrani mosfet je IRF540. To su neki osnovne komponente i zahtjevi sustava, idući korak je razrada.

Sastavljanje sustava kreće od skiciranje prve sheme na papiru i sastavljanja sustava na eksperimentalnoj pločici. Kao što joj samo ime kaže, na njoj se eksperimentiraju različiti spojevi na mnogim projektima, tako i na ovom. Pored osnovnog mrežnog modula, arduina i mosfeta pokazala se potreba za dodatnim tipkalom za slanje adrese, 2 regulatora napona, i raznim senzorima, dodatnim kondenzatorima i sl. Izgled završenog sustava na eksperimentalnoj pločici može se vidjeti na slici 3.33.



Slika 3.33: Automatizirani sustav upravljanja pumpama za navodnjavanje sastavljen na eksperimentalnoj pločici.

Idući korak, nakon što je cijeli sustav isproban na eksperimentalnoj pločici i provjereno funkcionira, je dorada prethodno skicirane sheme, crtanje sheme i tiskane pločice na računalu. Za ovaj korak korišten je računalni softver tvrtke Autodesk, EAGLE. Zbog potrebe da sustav bude što manjih dimenzija korištene su SMD (*engl. surface mount device* – površinski montirane komponente) komponente. Ove komponente su posebne po tome što su manjih dimenzija i leme se na bakar na tiskanoj pločici. Za razliku od običnih komponenata, SMD komponente ne zahtijevaju da se pločica buši kako bi se zalemila komponenta. Korištene su SMD komponente otpornika i kondenzatora u 1206 i 0805 kućištu.

Nakon što se pločica nacrta, dobavljač izrađuje pločicu prema prethodno dostavljenoj dokumentaciji. Izgled pločice i sheme centralne i izvršne jedinice prikazan je u prilogu P9.3 i P9.4. Na slici 3.34 može se vidjeti izgled sastavljenog sustava izvršne jedinice.

Slika 3.34: Izgled izvršne jedinice nakon sastavljanja na tiskanoj pločici.

Izrazito je vidljiva transformacija u urednosti i dimenzijama između završne verzije (slika 3.34) i početne verzije na eksperimentalnoj pločici (slika 3.33).

3.5.3 Podešavanje sustava

Centralna jedinica sustava je vrlo jednostavna i ne zahtjeva neka dodatna podešavanja i ugađanja. Dok izvršna jedinica u svom mikrokontroleru ima iš programiran PID regulator koji je potrebno podesiti. Odnosno, odrediti njegova 3 parametra: proporcionalni, integralni i derivacijski član.

Postoje razne metode za određivanje parametara PID regulatora. Jedna od njih je i relejna metoda. Ona funkcionira tako da sustav dovodi na rub stabilnosti, i mjeri vrijeme između stanja. Na osnovu tog vremena izračunavaju se Kp, Ki i Kd parametri.

No, ovaj sustav ima transportno kašnjenje očitavanja trenutne vrijednosti protoka. Zbog toga ova metoda (barem u obliku u kojem je netko napisao arduino PID datoteku) ne daje dobre rezultate. Transportno kašnjenje očitanja protoka javlja se zbog toga što se unutar jedne sekunde broje impulsi sa senzora protoka. Tek nakon jedne sekunde poznat je protok sustava.

Ostaje metoda ručnog ugađanja regulatora koja je malo dugotrajnija ali daje zadovoljavajuće rezultate. Ona se provodi tako da korisnik sam mijenja parametre sustava i promatra njegovo ponašanje. Tako su eksperimentalnom metodom dobiveni sljedeći parametri:

$$Kp = 2$$
$$Ki = 5$$
$$Kd = 1$$

Ovi parametri važeći su samo za pumpu navedenu u poglavlju 3.2.3 i za napon napajanja pumpe od 24V! Promjenom napona pumpa slabi, a time se mijenja radna točka sustava i potrebno je promijeniti parametre PID regulatora.

3.5.4 Testiranje funkcionalnosti

Testiranje funkcionalnosti je posljednji korak u izradi projektnog zadatka. Ovaj korak je ključan kako bi se utvrdila pouzdanost, točnost, funkcionalnost sustava, a na posljetku i cijena sustava.

Testiranje funkcionalnosti automatiziranog sustava može se provesti na način da se provjeravaju sljedeći zahtjevi:

- Funkcionira li senzori temperature (DS1820 i DHT 22)
- Funkcionira li senzor mjerenja pritiska
- Funkcionira li senzor mjerenja protoka
- Funkcionira li PID regulacija

Provjera funkcionalnosti senzora temperature, vlažnosti, pritiska i protoka se vrši bez upotrebe nekih vanjskih alata ili pomagala. Provjerava se slanjem zahtjeva od strane centralne jedinice i provjerom odgovora koji se primi. Ukoliko je odgovor neočekivano visoka ili niska vrijednost može se zaključiti da je senzor neispravan. Neispravni senzor temperature će prikazati temperaturu zraka u kojoj se korisnik nalazi na primjer 85°C, što je poprilično ne realno jer bi na toj temperaturi došlo do ozljeđivanja korisnika. Ista metodologija može se primijeniti na senzor temperature medija i senzor vlažnosti zraka.

Ispravnost PID regulatora se provodi na način da se izvršnoj jedinici zada određeni protok i provjerava se ponašanje regulatora. Pravilno podešen regulator mora bez problema kroz određeno vrijeme postaviti traženi protok na izlazu sustava. Dok kod pogrešno podešenog regulatora ili ne odgovarajućeg napona napajanja sustava regulator ne može pravilno podesiti traženi protok ili se ne kontrolirano pali i gasi, mlaz vode je ne ujednačen itd…

4. TESTIRANJE I REZULTATI

4.1 Metodologija testiranja

U ovom poglavlju pobliže ću pojasniti metodu testiranja komunikacijskog dometa između centralne jedinice i izvršne jedinice. Ovo testiranje s provodi kako bi se mogla utvrditi najveća dopuštena udaljenost između centralne i izvršne jedinice. Na osnovu toga korisnik može predvidjeti implementaciju sustava.

Za provedbu ovog ispitivanja potreban je geodetski metar, njegov izgled prikazan je na slici 4.1. Mjerenje se provodi na način da jedna od jedinica (izvršna ili centralna) stoji fiksirana u prostoru, a druga jedinica se udaljava od nje. Tokom udaljavanja geodetskim metrom se mjeri udaljenost između dvije jedinice i bilježi se udaljenost te vrijeme potrebno da fiksna jedinica primi odgovor od pomične jedinice. Fiksna jedinica šalje zahtjev pod određenim brojem pomičnoj jedinici, te čeka njezin odgovor. Mjeri se vrijeme potrebno da se dobije taj odgovor.

Slika 4.1: Geodetski metar.

Testiranje se provodi unutar grada i izvan grada, na otvorenom prostoru (na polju). Očekuje se da će komunikacija izvan gradskih naselja ostvariti veći domet nego ova unutar grada, zbog prethodno spomenute zagađenosti etera raznim bežičnim komunikacijama.

4.2 Rezultati testiranja

Tablica 4.1: Domet između jedinica izvan grada.

Izvan grada	
Udaljenost	Vrijeme odgovora
0 m	14 ms
10 m	14 ma
20 m	14 ms
30 m	14 ms
40 m	14 ms
50 m	14 ms
60 m	15 ms
70 m	15 ms
80 m	15 ms
90 m	15 ms
100 m	15 ms
110 m	16 ms
120 m	16 ms
130 m	-

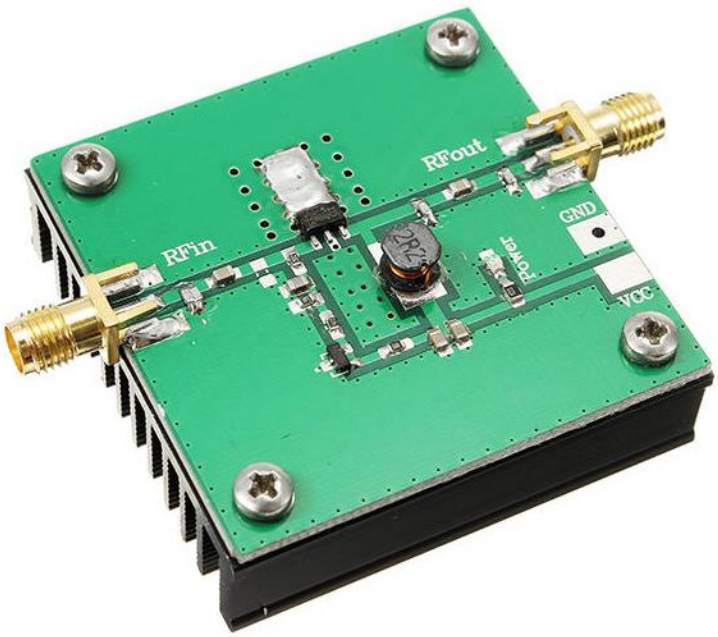
Tablica 4.2: Domet između jedinica unutar grada.

Unutar grada	
Udaljenost	Vrijeme odgovora
0 m	15 ms
10 m	15 ms
20 m	15 ms
30 m	15 ms
40 m	15 ms
50 m	15 ms
60 m	15 ms
70 m	15 ms
80 m	16 ms
90 m	-
100 m	-

Iz priloženih rezultata u tablicama 4.1 i 4.2 da se zaključiti da se rezultati izvan i unutar grada dosta razlikuju. Očekivani domet ovog sustava je bio oko 1000 metara. Postoje razni faktori koji su uzročnici ovako slabog dometa izvan grada, jedan od njih je napon napajanja mrežnog modula centralne. Trenutni napon mrežnog modula iznosi 3.2V, dok je optimalni napon rada 3.3 – 3.6V.

Domet unutar grada očekivano je umanjen u odnosu na domet izvan grada. Razlog tomu je zagađenost etera, građevinski objekti (zgrade, kuće, drveća, itd..).

Za korekciju rezultata, odnosno pojačavanje signala može se koristiti RF pojačalo koje pojačava frekvenciju od 433 MHz. Ono se instalira umjesto postojeće antene. Primjer takvoga pojačala prikazan je na slici 4.2:



Slika 4.2: Antensko pojačalo 433 MHz

5. ZAKLJUČAK

Svakodnevna potreba za zalijevanjem biljaka na raznim poljoprivrednim površinama dovela je do razvoja sustava za automatizirano upravljanje pumpama za navodnjavanje. Ideja da se radnja upravljanja pumpama automatizira i samim time poveća iskoristivost samog zalijevanja iskorištena je u ovom radu za projektiranje i realizaciju sustava za automatsko upravljanje pumpama za navodnjavanje pomoću neke mikrokontrolerske platforme ili računala. Prednost ovog sustava je što se sastoji od određenog broja izvršnih jedinica koje su bežično povezane sa centralnom jedinicom. To znači da korisnik može upravljati pumpama i navodnjavanjem udaljen kilometrima od samog poljoprivrednog zemljišta.

Odabirom arduino UNO razvojne platforme, koja svojom cijenom, specifikacijama i mogućnostima zadovoljava zadatke ovog završnog rada. Dobiva se ju se dvije komponente sustava, to su centralna jedinica i izvršna jedinica. Centralna jedinica zadužena je za komunikaciju računala i izvršnih jedinica. Dok izvršne jedinice su zadužene da ispunjavaju zadatke koje dobiju od računala. Izvršna jedinica sastoji se od senzora vlage, senzora temperature zraka, senzora temperature medija, senzora pritiska vode u spremniku, pumpe i elektromagnetnog ventila.

Za potrebe završnog rada izrađen je prototip opisanog sustava koji se sastoji od centralne i izvršne jedinice. Za svaku jedinicu ručno su izrađene tiskane pločice sa svim komponentama i smještene unutar zaštitnog kućišta. Svaka jedinica testirana je najprije zasebno, a nakon toga i u sklopu prototipa.

Osim fizičkog prototipa razrađene su i metode obrade signala korištenih za potrebe komunikacije i upotrebe raznih senzora. Osmišljen je komunikacijsko zaglavlje pomoću kojeg centralna jedinica, računalo i izvršna jedinica razmjenjuju podatke i zadaće. Premda isti daju dobre i pouzdane rezultate, postoji prostor za daljnje usavršavanje sklopa. Primjerice, u sljedećim inačicama sklopa mogao bi se poboljšati domet mrežnih modula ili realizirati solarno napajanje cijelog sustava. Realizacijom solarnog napajanja sustav postaje još ekonomičniji.

Određeni problemi mogu nastati prilikom testiranja prototipa u netipičnim atmosferskim uvjetima, ubog nepredvidljivih smetnji. Također, ne može se sa sigurnošću predvidjeti ponašanje sustava u punoj konfiguraciji koja podrazumijeva veliki broj izvršnih jedinica spojenih na centralnu jedinicu. Iako su unaprijed osmišljeni mehanizmi zaštite od kolizija, moguće je da pored određene veličine sustava zakažu. U tom slučaju problem će se riješiti uvođenjem nove centralne jedinice ili nekog drugog komunikacijskog protokola.

LITERATURA

[1]. Arduino Products, https://www.arduino.cc/en/Main/Products, pristup: 18.6.2017

[2]. Communications protocol, https://en.wikipedia.org/wiki/Communications_protocol, pristup: 18.6.2017

[3]. nRF905 product specification, http://infocenter.nordicsemi.com/pdf/nRF905_PS_v1.5.pdf, pristup: 18.6.2017

[4]. Diskretni PID regulator, http://laris.fesb.hr/digitalno_vodjenje/text_5-5.htm, pristup: 18.6.2017

[5]. Power MOSFET, https://www.vishay.com/docs/91019/91019.pdf, pristup: 18.6.2017

[6]. Temperature and humidity module AM2302 Product Manual, http://akizukidenshi.com/download/ds/aosong/AM2302.pdf, pristup: 18.6.2017

[7]. ATmega328P, http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf, pristup: 18.6.2017

[8]. Pulse-width modulation, https://en.wikipedia.org/wiki/Pulse-width_modulation, pristup: 18.6.2017

[9]. Arduino PWM, https://www.arduino.cc/en/Tutorial/PWM, pristup: 18.6.2017

[10]. YF-S201 Hall Effect Water Flow Meter / Sensor, http://www.hobbytronics.co.uk/yf-s201-water-flow-meter, pristup: 18.6.2017

[11]. Solenoid Actuator, http://www.about-air-compressors.com/solenoid-actuator.html, pristup: 18.6.2017

[12]. K. Trednjak, Realizacija sustava za automatsko zalijevanje biljke, završni rad, 2015

[13]. V. Ravlić, Automatika za elektrotehničke škole, 2004

[14]. USB, https://hr.wikipedia.org/wiki/usb, pristup: 3.9.2017

[15]. PID controller, https://en.wikipedia.org/wiki/PID_controller, pristup: 3.9.2017

[16]. Programmable Resolution 1-Wire Digital Thermometer, https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf, pristup: 3.9.2017

[17]. JT-600 Water pump, https://goo.gl/jhBi8a, pristup 3.9.2017

[18]. Tranzistory MOSFET, http://www.et-pocitacovesystemy.wz.cz/, pristup: 3.9.2017

SAŽETAK

Uspješan uzgoj poljoprivrednih kultura ključan je uvjet za razvoj prehrambene industrije u smislu potrebe za općom ljudskom prehranom. U tom procesu uzgoja poljoprivrednih kultura sustavi za navodnjavanje imaju značajnu ulogu. Cilj automatiziranog sustava upravljanja pumpama za navodnjavanje je dodatno automatizirati sustav navodnjavanja. To se postiže sustavom za upravljanje pumpama koji se sastoji od dva ključna dijela: centralne i izvršne jedinice sustava. Sve izvršne jedinice prikupljaju podatke sa poljoprivredne površine i putem centralne jedinice se podatci prenose do računala odnosno do korisnika. Korisnik odlučuje o načinu navodnjavanja i povratnom informacijom „odgovara“ izvršnoj jedinici koliku količinu vode da pusti u sustav navodnjavanja. Sustav je projektiran da mu potrošnja električne energije bude minimalna, pa je u budućnosti kroz razne inačice sustava moguće je dodati i solarno napajanje.

Ključne riječi: Automatika, PID regulacija, automatizirano zalijevanje, bežična komunikacija, arduino

ABSTRACT

Summary

Successful cultivation of agricultural crops is a key condition for the development of the food industry in terms of the need for general human nutrition. Irrigation systems have a significant role to play in the stated process of growing crops. The goal of an automated pump management system for irrigation is to further automate the irrigation system. This is achieved by a pump management system that consists of two key parts: central and executive system units. All the executive units collect data from the agricultural area and transfer the data via the central unit to the computer, respectively to the user. The user decides on the irrigation mode and "responds" with feedback information to the executive unit the exact amount of water that is allowed to release to the irrigation system. The system is designed to have minimal power consumption, so in the future, solar power can be added in various versions of the system. The system is designed to have minimal power consumption, so in the future solar power could be added in various versions of the system.

Keywords: Automation, PID controller, irrigation system, wireless communication, arduino

ŽIVOTOPIS

Vedran Kukrika rođen je 12. veljače 1995. godine u Osijeku. Pohađao je osnovnu školu Franje Krežme od 2003. do 2010. godine i završio ju sa vrlo dobrim uspjehom. Po završetku osnovne škole, 2010. godine upisuje Elektrotehničku i prometnu školu Osijek, smjer tehničar za mehatroniku. U trećem razredu odlazi na državno natjecanje iz mehatronike u Zagreb i završava na 2. mjestu, no već iduće godine (u četvrtom razredu) na istome natjecanju završava na 1. mjestu. Tokom cijelog srednjoškolskog obrazovanja redovno je sudjelovao na smotrama radova Osječko-baranjske županije. Srednjoškolsko obrazovanje završava 2014. godine sa odličnim uspjehom. Po završetku školovanja od Grada Osijeka dobiva jednokratnu stipendiju za darovite učenike zajedno s pohvalnicom. Iste godine polaže državnu maturu i upisuje Sveučilište Josipa Jurja Strossmayera u Osijeku na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek, stručni studij smjer Automatika, na kojem stječe radne navike i znanja iz područja automatike.

Potpis studenta:

_____

PRILOZI

P9.1 Softver za mikrokontroler centralne jedinice

```
#include <nRF905.h>
#include <SPI.h>
#include "nRF905_types.h"
#include "nRF905_config.h"

#define ADRESA_CENTRALNE_JEDINICE 0
#define RXADDR {ADRESA_CENTRALNE_JEDINICE, 0x6F, 0x2E, 0x10}
#define TXADDR {0xFE, 0x4C, 0xA6, 0xE5}
typedef struct {
  byte br1;
  byte br2;
  byte br3;
  byte br4;
  byte br5;
  byte br6;
} paket_slanje;
char serial_input [12];
boolean Flag = 0;
byte adresa = 0;
unsigned int Vrijeme_cekanja = 0;

void setup() {

  nRF905_init();

  nRF905_setTransmitPower(NRF905_PWR_10);
  nRF905_setFrequency(NRF905_BAND_433, 422.8);
  nRF905_setChannel(NRF905_BAND_433, 0);

  byte addr[] = RXADDR;
  nRF905_setRXAddress(addr);

  nRF905_receive();

  Serial.begin(9600);

  Serial.println(F("Spreman"));
}

void loop() {
  paket_slanje paket;


  if (Serial.available() > 0) {
    for (int i = 0; i < 13; i++) {
      serial_input[i] = Serial.read();
      if (serial_input[i] == '\n') {
        serial_input[i] = 0;
        i = 12;
      }
      delay (10);
    }
    Serial.end();
    Serial.begin(9600);
    paket.br1 = get_byte(serial_input[1], serial_input[0]);
    paket.br2 = get_byte(serial_input[3], serial_input[2]);
    paket.br3 = get_byte(serial_input[5], serial_input[4]);
    paket.br4 = get_byte(serial_input[7], serial_input[6]);
    paket.br5 = get_byte(serial_input[9], serial_input[8]);
```

```
  paket.br6 = get_byte(serial_input[11], serial_input[10]);
  adresa = paket.br1;
  Serial.print("Adresa: ");
  Serial.println(paket.br1);
  Serial.print("Komanda: ");
  Serial.println(paket.br2);
  Serial.println("4 bajta: ");

  Serial.println(paket.br3);
  Serial.println(paket.br4);
  Serial.println(paket.br5);
  Serial.println(paket.br6);

  Slanje_Paketa(&paket);
  Flag = 1;
  Vrijeme_cekanja = millis();
  nRF905_receive();
}


nRF905_receive();

while (1)
{
  if ((Flag == 1) && ((unsigned int)(millis() - Vrijeme_cekanja) > 300)) {
    Serial.println(F("Odgovor nije primljen"));
    Flag = 0;
  }
  if (Primanje_paketa(&paket))
  {
    if ( (Flag == 1) && (adresa == paket.br1)) {
      Serial.println(F("Odgovor primljen"));
      Flag = 0;

    }
    else if ((Flag == 0) || ((Flag == 1) && (adresa != paket.br1))) {
      Serial.println(F("Zahtjev primljen"));
      nRF905_receive();
    }
    Serial.println(paket.br1);
    Serial.println(paket.br2);
    Serial.println(paket.br3);
    Serial.println(paket.br4);
    Serial.println(paket.br5);
    Serial.println(paket.br6);

    byte nb = 0;
    if (paket.br3 > 0) nb++;
    if (paket.br4 > 0) nb++;
    if (paket.br5 > 0) nb++;
    if (paket.br6 > 0) nb++;
    Serial.println("----------------");
    if (paket.br1 < 16) Serial.print ('0');
    Serial.print(paket.br1, HEX);
    Serial.print("/");
    if (paket.br2 < 16) Serial.print ('0');
    Serial.print(paket.br2, HEX);
    Serial.print("/");
    Serial.print ('0');
    Serial.print(nb, HEX);
    Serial.print("/(");
    if (paket.br3 < 16) Serial.print ('0');
```

```
      Serial.print(paket.br3, HEX);
      Serial.print("/");
      if (paket.br4 < 16) Serial.print ('0');
      Serial.print(paket.br4, HEX);
      Serial.print("/");
      if (paket.br5 < 16) Serial.print ('0');
      Serial.print(paket.br5, HEX);
      Serial.print("/");
      if (paket.br6 < 16) Serial.print ('0');
      Serial.print(paket.br6, HEX);
      Serial.print(")");
      Serial.println();
    }
    else if (Serial.available())
      break;
  }
}

//-------------------------------------------------------------
// Send a packet

static void Slanje_Paketa(void* _paket)
{
  // Void pointer to packet_s pointer hack
  // Arduino puts all the function defs at the top of the file before packet_s being declared :/
  paket_slanje* paket = (paket_slanje*)_paket;
  byte tmpBuff[6];
  tmpBuff[0] = paket->br1; //adresa
  tmpBuff[1] = paket->br2; //komanda
  tmpBuff[2] = paket->br3; //byte 1
  tmpBuff[3] = paket->br4; //byte 2
  tmpBuff[4] = paket->br5; //byte 3
  tmpBuff[5] = paket->br6; //byte 4

  // Set payload data
  nRF905_setData(tmpBuff, 6);
  delay (30);
  byte addr[] = {paket->br1, 0x4C, 0xA6, 0xE5};
  nRF905_setTXAddress(addr);

  // Send payload (send fails if other transmissions are going on, keep trying until success)
  while (!nRF905_send());
}

static bool Primanje_paketa(void* _paket)
{
  // Void pointer to packet_s pointer hack
  // Arduino puts all the function defs at the top of the file before packet_s being declared :/
  paket_slanje* paket = (paket_slanje*)_paket;

  byte buffer[NRF905_MAX_PAYLOAD];

  // See if any data available
  if (!nRF905_getData(buffer, sizeof(buffer)))
    return false;


  paket->br1 = buffer[0];
  paket->br2 = buffer[1];
  paket->br3 = buffer[2];
  paket->br4 = buffer[3];
  paket->br5 = buffer[4];
```

```
  paket->br6 = buffer[5];

  return true;
}

unsigned int get_byte (unsigned int lsb, unsigned int msb) {

  if ( (lsb >= 48) && (lsb <= 57)) {
    lsb = lsb - 48;
  }
  if ( (lsb >= 97) && (lsb <= 102))
  {
    lsb = lsb - 87;
  }
  if ( (msb >= 48) && (msb <= 57)) {
    msb = msb - 48;
  }
  if ( (msb >= 97) && (msb <= 201))
  {
    msb = msb - 87;
  }

  int rez = (lsb * 1) + (msb * 16);
  return rez;
}
```

P9.2 Softver za mikrokontroler izvršne jedinice

```
#include <PID_v1.h>
#include <SimpleTimer.h>
#include <DHT.h>
#include <nRF905.h>
#include <SPI.h>
#include <EEPROM.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include "nRF905_types.h"
#include "nRF905_config.h"

#define ADRESA_CENTRALNE_JEDINICE 0
#define DHTPIN A0
#define DHTTYPE DHT22
#define ONE_WIRE_BUS A1
#define IZLAZ_PID 6

double trenutno_stanje = 0;
double staro_stanje;
int broj_impulsa;
double aggKp = 2, aggKi = 5, aggKd = 2;
double consKp = 2, consKi = 0.5, consKd = 2;
float Ukupan_prokotk = 0, temp = 0;
float cijeli, decimalni;
float protokmin = 255;
float protokmax = 0;
float tempmin = 255;
float tempmax = 0;
int pressure = 0;
int pressmin = 255;
int pressmax = 0;

DHT dht(DHTPIN, DHTTYPE);
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
double Setpoint, Input, Output;
PID myPID(&Input, &Output, &Setpoint, 2, 5, 1, DIRECT);

typedef struct {
  byte br1;
  byte br2;
  byte br3;
  byte br4;
  byte br5;
  byte br6;
} paket_slanje;
SimpleTimer timer;

void setup() {
  pinMode(6, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(A5, INPUT);
  pinMode(A2, INPUT);
  myPID.SetMode(AUTOMATIC);
  nRF905_init();

  nRF905_setTransmitPower(NRF905_PWR_10);

  nRF905_setFrequency(NRF905_BAND_433, 422.8);
  nRF905_setChannel(NRF905_BAND_433, 0);
```

```
  nRF905_setCRC (NRF905_CRC_16);
  byte addr[] = {EEPROM.read (0), 0x4C, 0xA6, 0xE5};
  nRF905_setRXAddress(addr);

  nRF905_receive();
  myPID.SetMode(AUTOMATIC);
  Serial.begin(9600);

  TCCR1A = 0;
  TCCR1B = bit (CS10) | bit (CS11) | bit (CS12);
  timer.setInterval(1000, Protok);

  Serial.println("Spreman");
  Serial.println(EEPROM.read (0));
}



void loop() {
  timer.run();
  paket_slanje paket;

  protokmin = min(Input, protokmin);
  protokmax = max(Input, protokmax);
  pressmin = min (pressure, pressmin);
  pressmax = max (pressure, pressmax);
  pressure = analogRead (A2) / 4;


  if (digitalRead (A5) == HIGH) {
    delay (250);
    paket.br1 = EEPROM.read (0);
    paket.br2 = 0;
    paket.br3 = 1;
    paket.br4 = 2;
    paket.br5 = 3;
    paket.br6 = 4;

    Slanje_Paketa(&paket);

    Serial.println ("ZAHTIJEV");

  }


  if (Primanje_paketa(&paket)) {

    Serial.print ("Adresa: ");
    Serial.println (paket.br1);
    Serial.print ("Komanda: ");
    Serial.println (paket.br2);
    Serial.print ("byte br1: ");
    Serial.println (paket.br3);
    Serial.print ("byte br2: ");
    Serial.println (paket.br4);
    Serial.print ("byte br3: ");
    Serial.println (paket.br5);
    Serial.print ("byte br4: ");
    Serial.println (paket.br6);

    if (paket.br2 >= 0) {
      switch (paket.br2) {
```

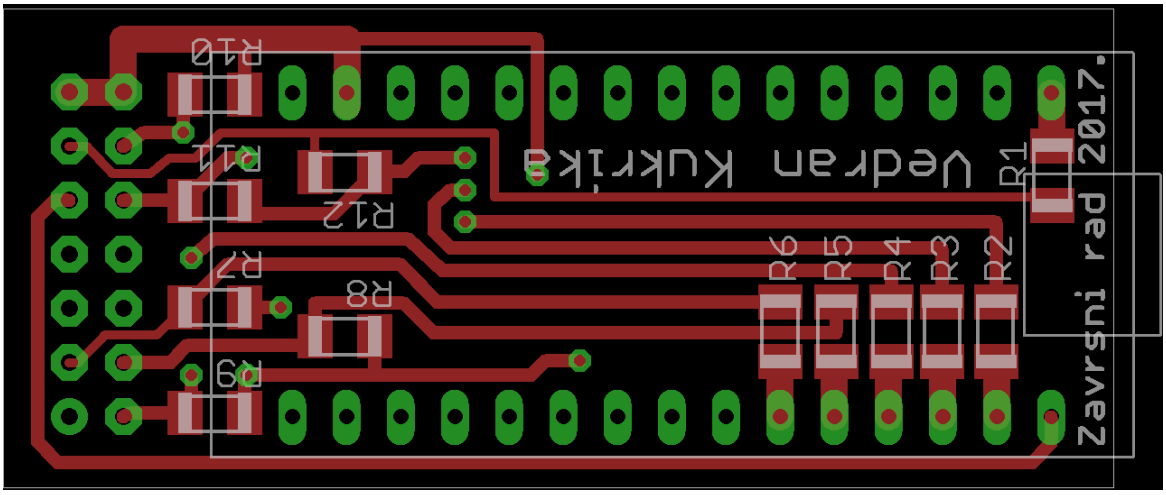
```
case 1: {
    EEPROM.write (0, paket.br3);
    byte addr[] = {EEPROM.read (0), 0x4C, 0xA6, 0xE5};
    nRF905_setRXAddress(addr);
    nRF905_receive();
  } break;
case 21:                    //Flow read current
  paket.br3 = Input;
  paket.br4 = (Input * 100) - ((byte)Input * 100);;
  paket.br5 = 0;
  paket.br6 = 0;
  break;
case 22:                    //Flow read cumulative
  paket.br3 =  Ukupan_prokotk;
  paket.br4 =  (Ukupan_prokotk * 100) - ((byte)Ukupan_prokotk * 100);
  paket.br5 = 0;
  paket.br6 = 0;
  break;
case 23:                    //Flow read max
  paket.br3 =  protokmax;
  paket.br4 =  (protokmax * 100) - ((byte)protokmax * 100);
  paket.br5 = 0;
  paket.br6 = 0;
  break;
case 24:                    //Flow read min
  paket.br3 =  protokmin;
  paket.br4 =  (protokmin * 100) - ((byte)protokmin * 100);
  paket.br5 = 0;
  paket.br6 = 0;
  break;
case 25:                    //Flow set reference for PID regulation
  cijeli = paket.br3;
  decimalni = paket.br4;
  paket.br5 = 0;
  paket.br6 = 0;
  Setpoint = (decimalni / 100) + cijeli;
  break;
case 31:                    //Water temperature current
  sensors.requestTemperatures();
  temp = sensors.getTempCByIndex(0);
  tempmax = max(temp, tempmax);
  tempmin = min(temp, tempmin);
  paket.br3 = temp;
  paket.br4 = (temp * 100) - ((byte)temp * 100);
  paket.br5 = 0;
  paket.br6 = 0;
  break;
case 32:                    //Water temperature min
  sensors.requestTemperatures();
  temp = sensors.getTempCByIndex(0);
  tempmin = min(temp, tempmin);
  paket.br3 = tempmin;
  paket.br4 =  (tempmin * 100) - ((byte)tempmin * 100);
  paket.br5 = 0;
  paket.br6 = 0;
  break;
case 33:                    //Water temperature max
  sensors.requestTemperatures();
  temp = sensors.getTempCByIndex(0);
  tempmax = max(temp, tempmax);
  paket.br3 = tempmax;
  paket.br4 =  (tempmax * 100) - ((byte)tempmax * 100);
```

```
      paket.br5 = 0;
      paket.br6 = 0;
      break;
    case 41:                    //Pressure read current
      paket.br3 = pressure;
      paket.br4 = 0;
      paket.br5 = 0;
      paket.br6 = 0;
      break;
    case 42:                    //Pressure read min
      paket.br3 = pressmin;
      paket.br4 = 0;
      paket.br5 = 0;
      paket.br6 = 0;
      break;
    case 43:                    //Pressure read max
      paket.br3 = pressmax;
      paket.br4 = 0;
      paket.br5 = 0;
      paket.br6 = 0;
      break;
    case 51:                    //Valve set openning
      if (paket.br3 == 0) {
        Setpoint = 0;
      }
      else {
        digitalWrite(4, HIGH);
      }
      paket.br4 = 0;
      paket.br5 = 0;
      paket.br6 = 0;
      break;
    case 52:                    //Valve read
      paket.br3 = digitalRead(4);
      paket.br4 = 0;
      paket.br5 = 0;
      paket.br6 = 0;
      break;
    case 71:                    //DHT temp read
      paket.br3 = dht.readTemperature();
      paket.br4 = 0;
      paket.br5 = 0;
      paket.br6 = 0;
      break;
    case 72:                    //DHT humidity read
      paket.br3 = dht.readHumidity();
      paket.br4 = 0;
      paket.br5 = 0;
      paket.br6 = 0;
      break;
    default:
      paket.br1 = 1;
      paket.br2 = 1;
      paket.br3 = 1;
      paket.br4 = 1;
      paket.br5 = 1;
      paket.br6 = 1;

      break;
  }
}
Slanje_Paketa(&paket);
```

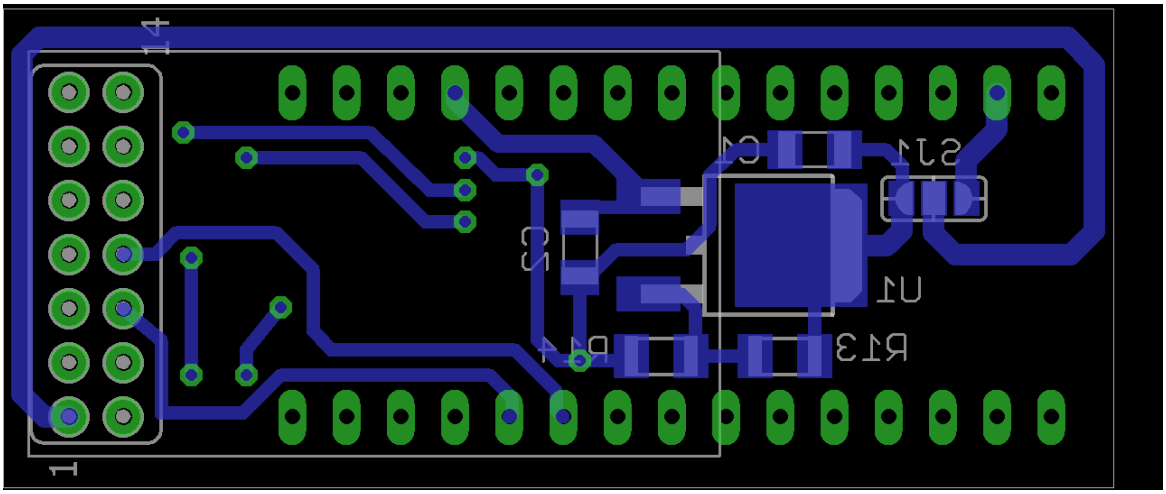
```
  }



  if (Setpoint > 0) {
    digitalWrite(4, HIGH);
    Input = broj_impulsa / 7.5 ;
    //   Serial.println (Input);
    myPID.Compute();
    analogWrite(6, Output);
    if (millis() % 60000 == 0) {
      Ukupan_prokotk = Ukupan_prokotk + Input;
      Serial.println (Ukupan_prokotk);
      delay (1);
    }
  }
  else {
    Input = 0;
    digitalWrite(4, LOW);
    analogWrite(6, LOW);

  }
}



//--------------------------------------------------------
static void Slanje_Paketa(void* _paket)
{
  // Void pointer to packet_s pointer hack
  // Arduino puts all the function defs at the top of the file before packet_s being declared :/
  paket_slanje* paket = (paket_slanje*)_paket;
  byte tmpBuff[6];
  tmpBuff[0] = paket->br1;
  tmpBuff[1] = paket->br2;
  tmpBuff[2] = paket->br3;
  tmpBuff[3] = paket->br4;
  tmpBuff[4] = paket->br5;
  tmpBuff[5] = paket->br6;

  nRF905_setData(tmpBuff, 6);
  delay (10);
  byte addr[] = {ADRESA_CENTRALNE_JEDINICE, 0x6F, 0x2E, 0x10};
  nRF905_setTXAddress(addr);
  while (!nRF905_send());
  nRF905_receive();
}

static bool Primanje_paketa(void* _paket)
{

  paket_slanje* paket = (paket_slanje*)_paket;
  byte buffer[NRF905_MAX_PAYLOAD];
  if (!nRF905_getData(buffer, sizeof(buffer)))
    return false;

  paket->br1 = buffer[0];
  paket->br2 = buffer[1];
  paket->br3 = buffer[2];
  paket->br4 = buffer[3];
  paket->br5 = buffer[4];
  paket->br6 = buffer[5];
```

```
  return true;
}

//-------------------------------------------------------
void Protok() {
  trenutno_stanje = TCNT1;
  broj_impulsa = (trenutno_stanje - staro_stanje);
  staro_stanje = trenutno_stanje;
}
```

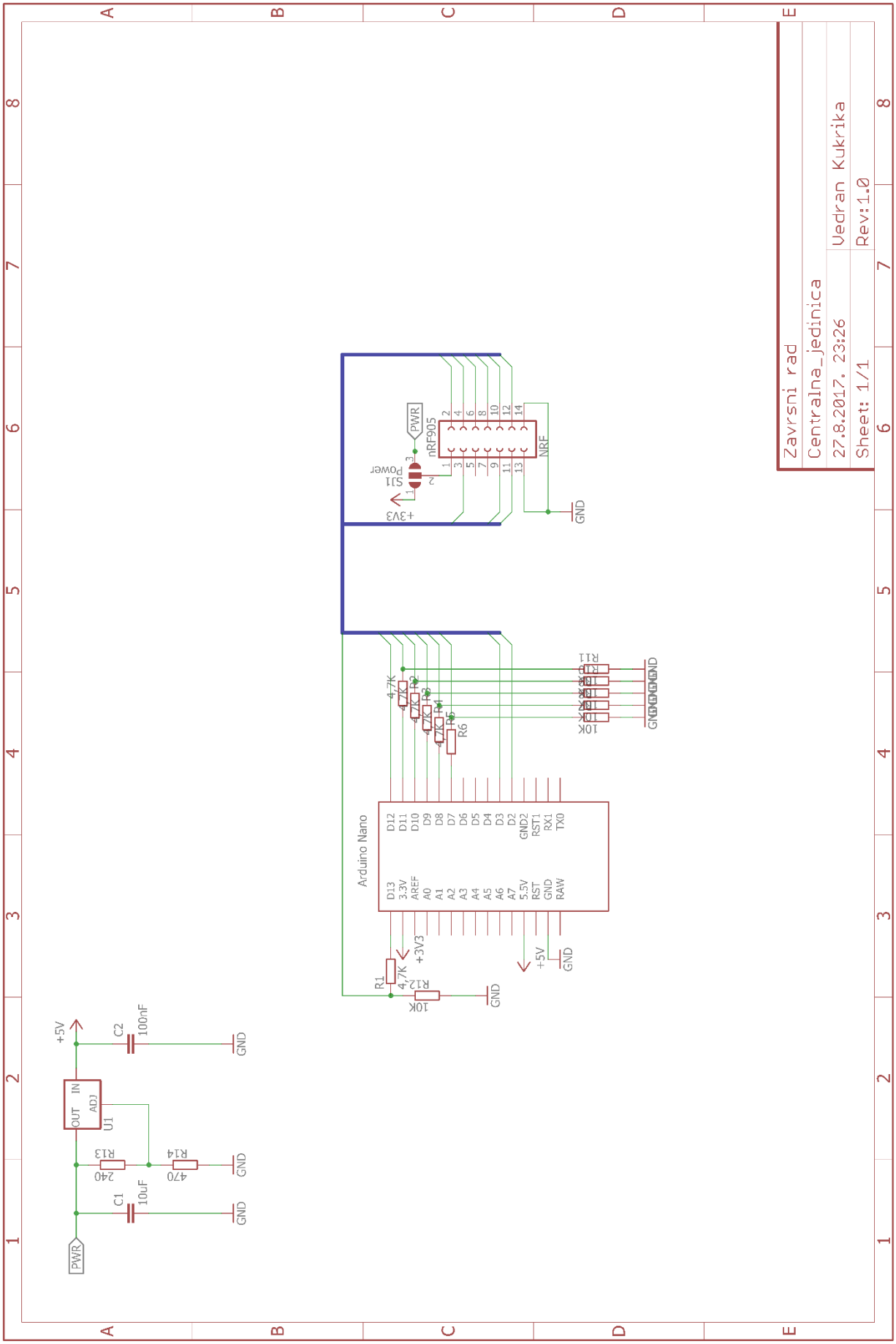
P9.3 Shema, nacrt i popis komponenata tiskane pločice za centralnu jedinicu



Slika 9.1: Gornji sloj vodova



Slika 9.2: Donji sloj vodova

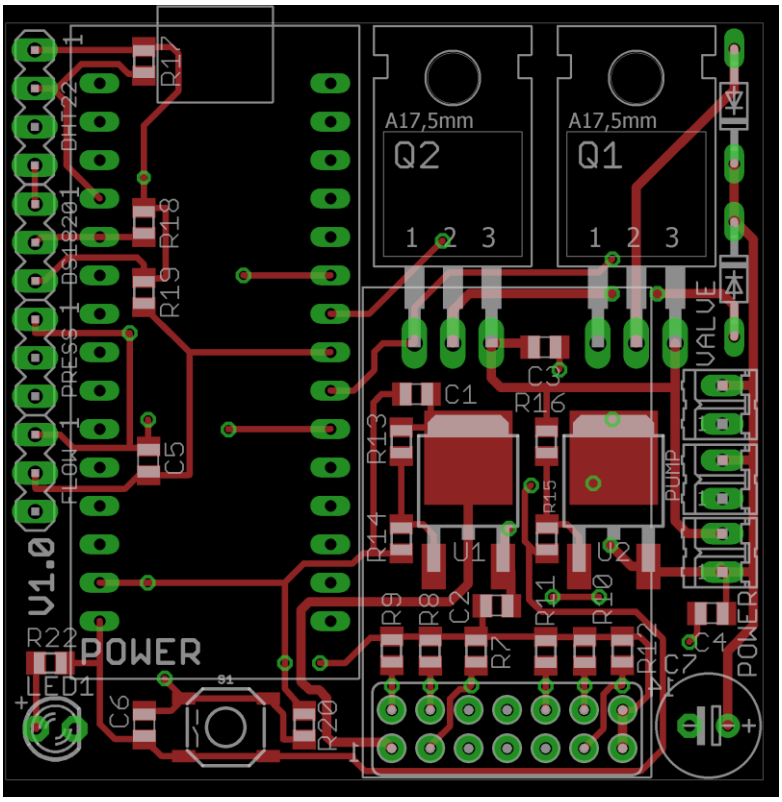


Shema 9.1: Shema centralne jedinice

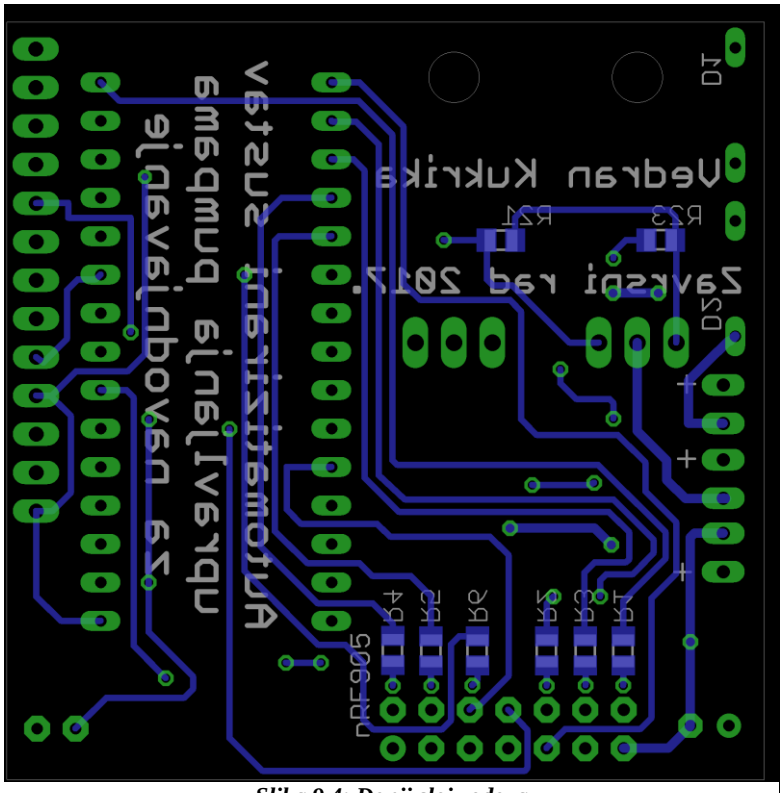
Tablica 9.1: Popis komponenata centralne jedinice

Kol.	*Vrijednost*	*Uređaj*	*Pakovanje*	*Naziv*
6	4,7K	Otpornik	1206	R1, R2, R3, R4, R5, R6
6	10K	Otpornik	1206	R7, R8, R9, R10, R11, R12
1	240	Otpornik	1206	R13
1	470	Otpornik	1206	R14
1	10uF	Kondenzator	1206	C1
1	100nF	Kondenzator	1206	C2
1	LM317	Reg. napona	TO-263	U1
1	Arduino Nano	Mikrokontroler	NANO	Arduino Nano
1	nRF905	Mrežni modul	7x2 Pin	NRF

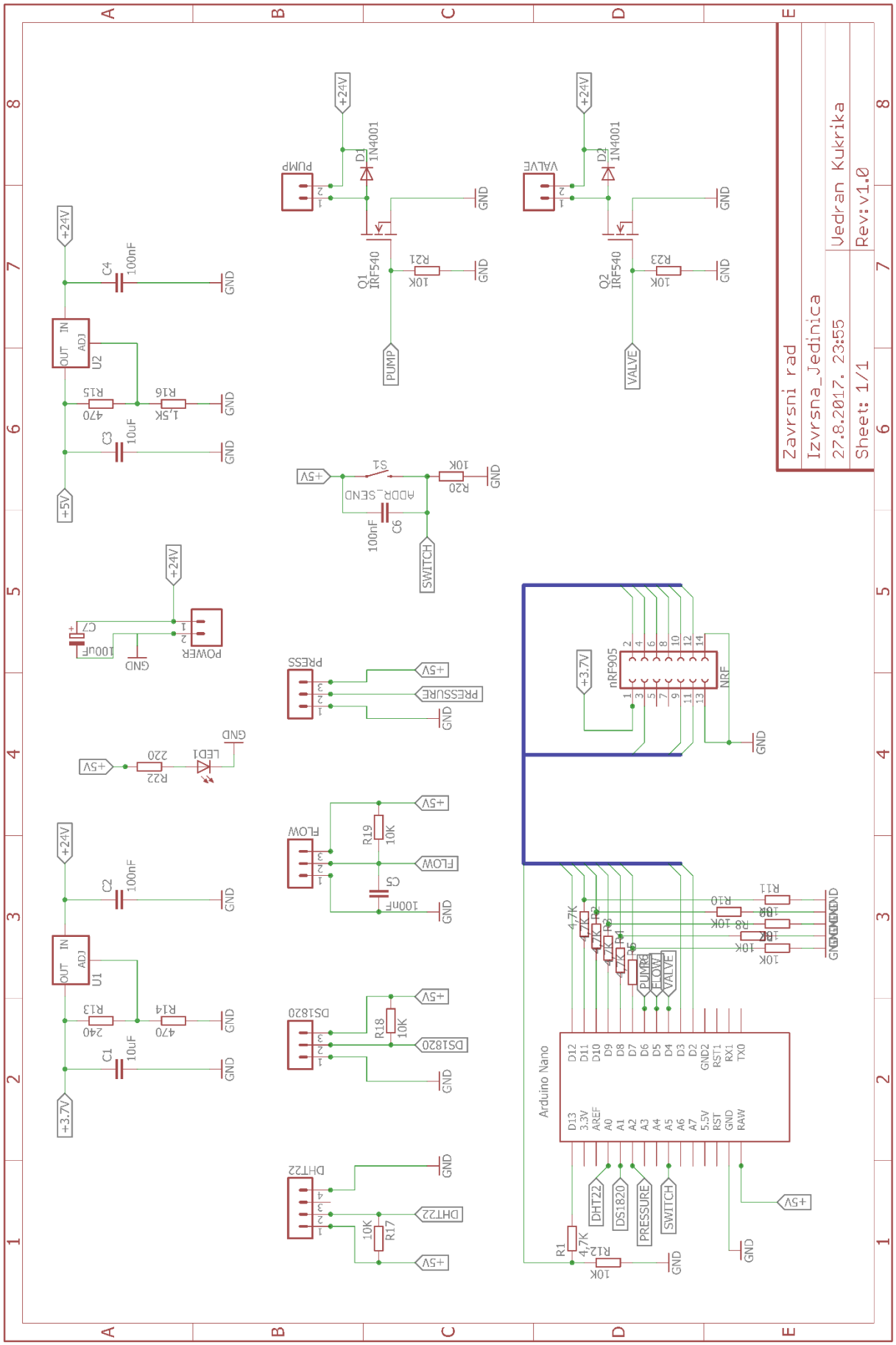
P9.4 Shema, nacrt i popis komponenata tiskane pločice za izvršnu jedinicu



Slika 9.3: Gornji sloj vodova



Slika 9.4: Donji sloj vodova

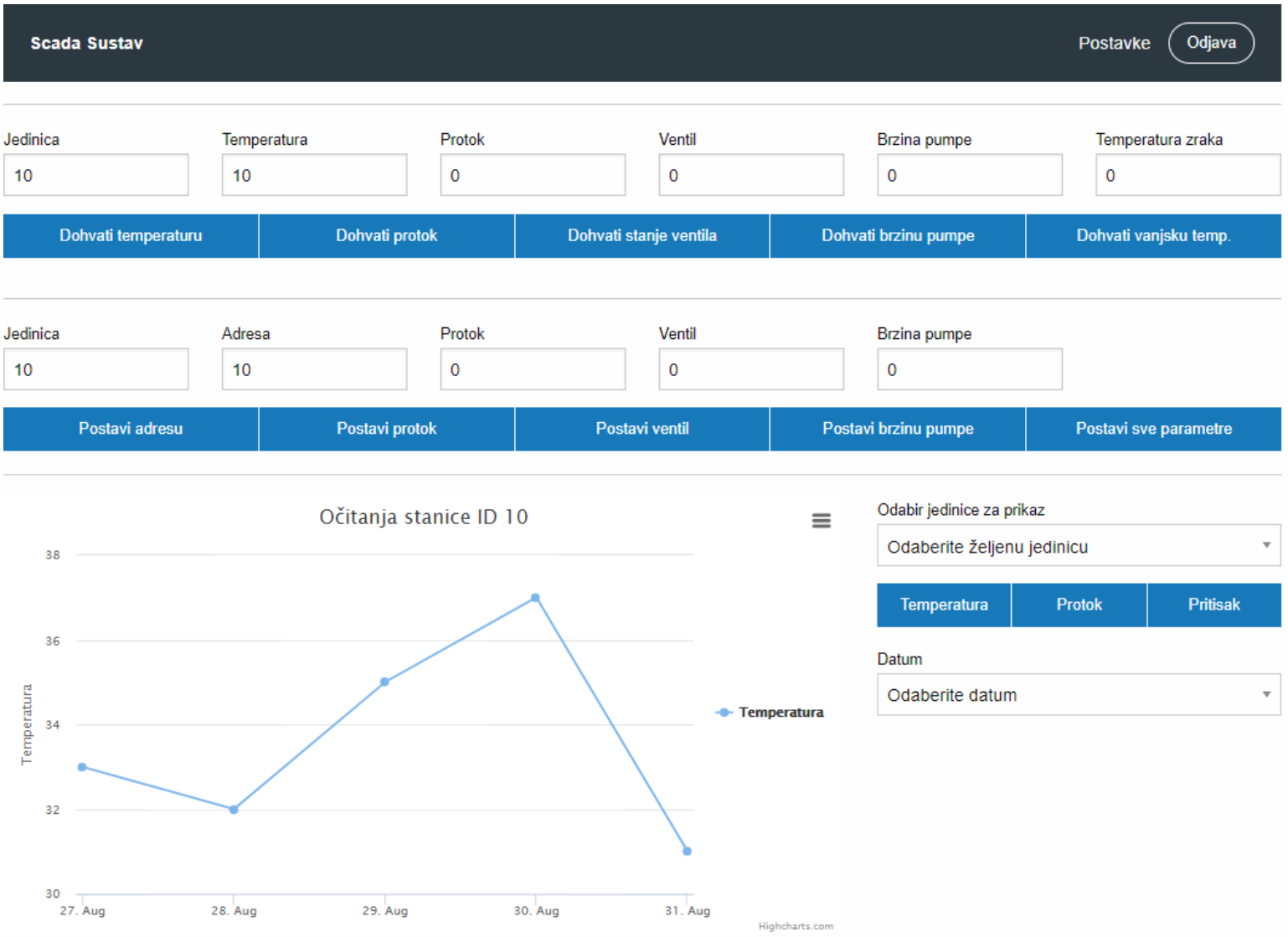


Shema 9.2: Shema izvršne jedinice

Tablica 9.2: Popis komponenata izvršne jedinice

Kol.	*Vrijednost*	*Uređaj*	*Pakovanje*	*Naziv*
1	1,5K	Otpornik	0805	R16
6	4,7K	Otpornik	0805	R1, R2, R3, R4, R5, R6
12	10K	Otpornik	0805	R7, R8, R9, R10, R11, R12, R17, R18, R19, R20, R21, R23
1	220	Otpornik	0805	R22
1	240	Otpornik	0805	R13
2	470	Otpornik	0805	R14, R15
1	100uF	Kondenzator	E2,5-7	C7
2	10uF	Kondenzator	0805	C1, C3
4	100nF	Kondenzator	0805	C2, C4, C5, C6
2	LM317	Reg. napona	TO-263	U1,U2
2	1N4001	Dioda	DO34-7	D1,D2
1	LED	LED	LED 3mm	LED1
1	Arduino Nano	Mikrokontroler	NANO	Arduino Nano
1	nRF905	Mrežni modul	7x2 Pin	NRF
2	IRF540	N-MOSFET	TO220	Q1, Q2
1	Tipkalo	Tipkalo	SMD 5.2mm	S1

P9.5 Snimka zaslona sustava automatiziranog upravljanja pumpama za navodnjavanje



ID	Vrijeme	Protok	Max. protok	Min. protok	Temp. medija	Min. temperatura	Max. temperatura	Pritisak vode	Min. pritisak	Max. pritisak	Stanje ventila	Brzina pumpe	Temp. zraka	Vlažnost zraka
10	12:23	6,3	7,3	2,4	42	23	42	72	43	80	66	72	32	46
10	12:45	6,3	7,3	3,5	38	23	42	66	40	72	70	68	32	46
10	13:04	4,3	5,7	3,7	35	22	40	52	37	66	60	50	32	46