

UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

ZAKLJUČNA NALOGA
**SAMODEJNI SISTEM ZA KRMILJENJE
ZALIVALNO-NAMAKALNIH SISTEMOV**

TOMAŽ TOMAŽINČIČ

UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

Zaključna naloga

Samodejni sistem za krmiljenje zalivalno-namakalnih sistemov

(Automated system for controlling irrigation systems)

Ime in priimek: Tomaž Tomažinčič

Študijski program: Računalništvo in informatika

Mentor: doc. dr. Jernej Vičič

Koper, september 2015

Ključna dokumentacijska informacija

Ime in PRIIMEK: Tomaž TOMAŽINČIČ

Naslov zaključne naloge: Samodejni sistem za krmiljenje zalivalno-namakalnih sistemov

Kraj: Koper

Leto: 2015

Število listov: 45

število slik: 27

število tabel: 1

število referenc: 25

Mentor: doc. dr. Jernej Vičič

Ključne besede: krmilni sistem, vreme, Raspberry Pi, Arduino, odprto koden

Izvleček: Zaključna naloga opisuje prednosti in slabosti sistemov za krmiljenje zalivalnih in namakalnih sistemov na trgu, večji del naloge pa je namenjen opisu delovanja in pregledu posameznih faz razvoja odprtokodnega, nizko cenovnega samodejnega sistema za krmiljenje zalivalno-namakalnih sistemov, ki poleg ugodne, ponuja funkcijsko konkurenco ostalim sistemom za krmiljenje zalivalnih in namakalnih sistemov. Opisuje tudi možnosti nadaljnjega razvoja, optimizacije in poenostavitve upravljanja s krmilnim sistemom.

Key words documentation

Name and SURNAME: Tomaž TOMAŽINČIČ

Title of final project paper: Automated system for controlling irrigation systems

Place: Koper

Year: 2015

Number of pages: 45

Number of figures: 27

Number of tables: 1

Number of references: 25

Mentor: Assist. Prof. Jernej Vičič, PhD

Keywords: controlling system, weather, Raspberry Pi, Arduino, open source

Abstract: Water is a chemical compound, which covers almost 70% of the Earth's surface. All biological processes including life itself depend on it. For food production we consume enormous quantities of water, the source thereof is in most cases limited, which is a problem that many people face in agricultural production, especially in areas prone to drought. The existing solutions include a combination of plumbing fixtures and a variety of water networks, that make watering certain areas or plants possible. Recently, an increasing number of control systems for irrigation emerged, which aim to optimize water consumption in various ways. There are quite a few differences amongst before mentioned systems, both in price performance ratio and functionality. Despite the variety of different functionalities they all lack the ability to adapt to different weather conditions. In order to gain a better insight into the current state of the market, we dedicate a part of this thesis to a detailed review of the advantages and disadvantages of existing controllers for irrigation systems. The rest of the thesis focuses on outlining the development process of an open source, low-cost, automatic irrigation controller. Our own solution aims to build upon the existing features to include an adaptive algorithm that optimizes water usage based on weather conditions.

Zahvala

Zahvaljujem se mentorju, doc. dr. Jerneju Vičiču za strokovno pomoč in sodelovanje pri izdelavi zaključne naloge.

Posebna zahvala gre puncu, družini in prijateljem za podporo in pomoč tekom študija.

Kazalo vsebine

1	Uvod	1
1.1	Motivacija	2
2	Predstavitev domene	3
2.1	Krmilni sistem in zalivalno-namakalni sistem	3
2.1.1	Prednosti uporabe krmilnih sistemov	4
2.1.2	Enostavni krmilni sistemi	5
2.1.3	Pametni krmilni sistemi	5
2.2	Samodejni sistem za krmiljenje zalivalno-namakalnih sistemov	6
3	Metodologija	9
3.1	Arhitektura sistema	9
3.2	Uporabljena oprema, tehnologije in programski jeziki	10
3.2.1	Protokoli	13
3.2.2	Senzorji	19
3.3	Programska oprema	22
4	Rezultati	25
4.1	Končni izdelek	25
4.1.1	Uporabniški vmesnik in primeri delovanja	26
4.2	Nadaljnje faze razvoja in možnosti izboljšav	30
5	Dostopnost	32
5.1	Licenca izvirne kode	32
5.2	Dostopnost programa izvirne kode	33
6	Zaključek	34
7	Literatura in viri	35

Seznam tabel

1.1	Količina vode potrebna pridelavo posamezne vrste hrane. [3]	1
-----	---	---

Seznam slik

2.1	Primer namakalnega sistema	3
3.1	Arhitektura krmilnega sistema	10
3.2	Arduino UNO	11
3.3	Raspberry Pi 2	12
3.4	Radio frekvenčni modul NRF24L01P	12
3.5	Pretvarjanje digitalnega signala s tehniko digitalne frekvenčne modu- lacije. Vidimo lahko, kako se frekvenca nosilnega signala (v sredini) spremeni glede na modulacijski signal (na vrhu).	14
3.6	Format podatkovnega paketa sestoji iz preambule, naslova, paketnega kontrolnega polja, vsebine sporočila in CRC preverjanja.	15
3.7	Delovanje ESB protokola v primeru pisanja. Vidimo lahko, da spreje- mnik po sprejemu sporočila, samodejno preklopi v stranje za pošiljanje ter odgovori pošiljateljju s ACK paketom.	16
3.8	Arhitektura NRF24L01: iz slike je razvidna povezava med SPI, ESB in GFSK.	17
3.9	Shema povezave platforme Arduino UNO in Raspberry Pi 2 (B) z NRF24L01 modulom.	18
3.10	Struktura osnovnega sporočila	19
3.11	Struktura sporočila - akcija in podatki	19
3.12	Senzor za dež	19
3.13	Senzor zračnega pritiska BMP180	20
3.14	Senzor vlažnosti zraka in temperature - DHT22	20
3.15	Senzor vlažnosti zemlje	21
3.16	Fotoupor	21
3.17	Merilec vodnega pretoka YF-S201	22
4.1	Notranja enota krmilnega sistema	25
4.2	Zunanja enota krmilnega sistema s perifernimi napravami.	26
4.3	Samodejni krmilni sistem ob zagonu	27
4.4	Vrednosti senzorjev zunanje enote	27
4.5	Izpis vlažnosti zemlje po conah	28

4.6	Aktivacija ročnega namakanja	28
4.7	Preklic ročnega namakanja	28
4.8	Izpis vremenske napovedi	29
4.9	Aktivacija samodejnega namakanja	30

Seznam kratic

<i>PVC</i>	PolyVinyl Chloride
<i>EKS</i>	Enostavni Krmilni Sistemi
<i>PKS</i>	Pametni Krmilni Sistemi
<i>RF(modul)</i>	Radiofrekvenčni (modul)
<i>IDE</i>	Intgrated Development Enviornment
<i>USB</i>	Universal Serial Bus controller
<i>SSH</i>	Secure Shell
<i>TCP</i>	Transmission Control Protocol
<i>IP</i>	Internet Protocol
<i>API</i>	Application Programming Interface
<i>WiFi</i>	Wireless Fidelity
<i>TX</i>	Transmit / Transmitter
<i>RX</i>	Receive / Receiver
<i>ACK</i>	ACKnowledgement
<i>ART</i>	Auto Retransimission
<i>CRC</i>	Cyclic Redundancy Check
<i>Half – Duplex</i>	Dvosmerna komunikacija s preklapljanjem med prejemanjem in oddajanjem
<i>Full – Duplex</i>	Hkratna dvosmerna komunikacija
<i>NC(valve)</i>	Normaly Closed (valve)
<i>NO(valve)</i>	Normaly Open (valve)
<i>ESB</i>	Enhanced ShockBurst
<i>GFSK</i>	Gaussian frequency-shift keying
<i>DECT</i>	Digital Enhanced Cordless Telecommunications
<i>FIFO</i>	Fist In First Out
<i>LED</i>	Light-Emitting Diode
<i>BSD</i>	Berkley Software Distribution
<i>MIT</i>	Massachusetts Institute of Techonology
<i>GPL</i>	General Public License

1 Uvod

Voda, kemijska spojina, ki prekriva skoraj 70 % Zemljine površine. Od nje so odvisni vsi biološki procesi, ki omogočajo življenje na Zemlji, tudi naše. Za proizvodnjo hrane se porabi ogromne količine vode, vir le-te pa je v večini primerov omejen.

Tabela 1.1: Količina vode potrebna pridelavo posamezne vrste hrane. [3]

Izdelek	Količina izdelka	Količina vode (l)	% svetovne porabe vode
Pšenica	0,5 kg	500	12
Ječmen	0,5 kg	750	3
Sirek	0,5 kg	1400	2,7
Proso	0,5 kg	2500	2
Riž	0,5 kg	1700	21
Krompir	0,5 kg	450	0,09
Koruza	0,5 kg	409	8
Soja	0,5 kg	818	4,5
Bombaž	kratka majica	2700	/
Jabolko	1	18	/
Pomaranča	1	50	/

S tem problemom se velikokrat spopadajo osebe v agrikulturni proizvodnji, še posebej na območjih dovzetnih za sušo. Načinov za reševanje slednjega problema je več, večinoma pa se uporabljajo različni načini napeljave vodnega omrežja na zalivalnih površinah in načina zalivanja same rastline. V zadnjem obdobju se na trgu v vse večjem številu pojavljajo in uporabljajo sistemi za krmiljenje zalivalnih in namakalnih sistemov, ki na različne načine upravljajo z vodo. Med seboj se razlikujejo tako po funkcionalnosti kot tudi po ceni. Slabost večine sistemov za krmiljenje je, da se delovanje ne spreminja glede na vremenske vplive.

V prvem delu zaključne naloge bodo predstavljeni krmilni sistemi za zalivanje in namakanje (od sedaj naprej krmilni sistem), zalivalni in namakalni sistemi (od sedaj naprej namakalni sistemi), izpostavljene bodo prednosti in slabosti. S pregledom in

primerjavo različnih krmilnih sistemov si bomo ustvarili celovit vpogled v uporabnost tovrstnih rešitev.

Večji del zaključne naloge opisuje idejo in delovanje samodejnega krmilnega sistema. Podrobneje so predstavljene vse faze razvoja prototipa, težave ki na katere smo naleteli in njihove rešitve. V zaključku so predstavljeni rezultati in primeri delovanja, možnosti izboljšav in predlogi nadaljnjega razvoja.

1.1 Motivacija

Krmilne sisteme se za zalivanje in namakanje (od sedaj naprej namakanje) uporablja že dalj časa, tako na zasebnih kot tudi javnih površinah. Kljub pojavu pametnih krmilnih sistemov (PKS), med uporabniki še vedno prevladujejo enostavni-primitivni krmilni sistemi (EKS). Primarni razlog je gotovo nizka cena v primerjavi s PKS, eden od razlogov pa je zagotovo tudi ta, da so se slednji začeli pojavljati na trgu šele v zadnjih letih. Posledično se ne moremo izogniti situacijam, kot so namakanje dan pred napovedanim deževjem ali še huje namakanje medtem, ko dežuje.

Zaradi naštetih razlogov smo razvili samodejni sistem za krmiljenje, ki bo sposoben odločanja kdaj in koliko zalivati na podlagi vremenskih razmer in potrebe rastline po vodi. Poleg tega je krmilni sistem nizko cenoven in odprtokoden, kar pomeni, da je prosto dostopen.

2 Predstavitev domene

2.1 Krmilni sistem in zalivalno-namakalni sistem

Zalivalno-namakalni sistem je sistem, ki skrbi za namerno (nenaravno) dodajanje vode v zemljo. Zgodovina namakalnih sistemov sega že v obdobje Mezopotamije in starega Egipta. Sistem je bil in je, ponekod tudi danes, sestavljen večinoma iz jarkov, nasipov in zapornic, ki usmerjajo določeno količino vode do in preko obdelovalnih površin. Najpogosteje se (predvsem za domačo uporabo) uporablja sistem cevi iz PVC materiala, ventilov in raznih dodatkov, kot so na primer: pršilniki, zalivalniki, kapljalniki, ipd. Tip dodatkov je odvisen tudi od zelenega načina namakanja. Na sliki 2.1 je primer sodobnega namakalnega sistema s pršilniki, ki je primeren za uporabo na večjih obdelovalnih površinah.



Slika 2.1: Primer namakalnega sistema. [6]

Krmilni sistem za namakanje je elektronski sistem sestavljen iz krmilnika in perifernih naprav. Krmilnik s pomočjo perifernih naprav pripravi namakalno shemo ter v nekaterih primerih nadzoruje tudi porabo vode. Krmilnik je običajno nameščen neposredno ob izvoru vode (vodovodne pipe, zbiralnika vode, ipd.), ta s pomočjo ventilov upravlja pretok vode. Za delovanje potrebuje vir električne energije, pri čemer je izvor energije neodvisen od tipa krmilnika.

Najpogosteje uporabljeni viri so neposreden priklop v električno omrežje, baterija oziroma akumulator in sončne celice, ki so običajno komplement bateriji oziroma akumulatorju.

2.1.1 Prednosti uporabe krmilnih sistemov

Prednost uporabe krmilnih sistemov je veliko, mednje štejemo praktičnost in odgovorno ravnanje z vodo.

Večina uporabnikov takih sistemov se za nakup ali izdelavo odloči, zaradi želje po avtomatizaciji. Uporabnik se tako izogne vsakodnevnih vprašanj, kot so:

- Ali je zalivanje opravil?
- Kdaj je zalivanje potrebno?
- Koliko vode potrebujejo rastline?

Z uporabo krmiljenega namakalnega sistema so ta vprašanja brez predmetna. Prav tako lahko sistem oceni primerno količino vode, ki jo rastlina potrebuje. Ob okvari komponente ali težavi z napajanjem krmilnega sistema moramo napake prepoznati, saj lahko v nasprotnem primeru pride do uničenja pridelka. Večina krmilnih sistemov ne podpira možnosti ugotavljanja primerne količine vode, ki je v določenem trenutku potrebna. Pri krmilnih sistemih, ki to funkcionalnost podpirajo, je natančnost vprašljiva. V večini primerov je krmilni sistem, ne glede na možnost okvare in točnost ugotavljanja potrebne količine vode za namakanje, učinkovitejši od človeka. Za uporabnika je torej največja prednost prihranek časa, na dolgi rok pa tudi denarni prihranek zaradi zmernejše in odgovornejše porabe vode. Slednje je sicer odvisno od tipa krmilnega sistema oziroma od uporabniških nastavitev. Krmilni sistemi lahko najdejo prostor tudi med manj zahtevnimi uporabniki, katerim sistem pomaga k bolj uspešnem pridelovanju.

Z razvojem PKS je možno natančneje ugotoviti čas in količino vode potrebne za namakanje, kar posredno vpliva na boljšo rast rastlin. S tako optimizacijo dodatno zmanjšamo količino porabljenе vode, kar posledično vpliva tudi na dolgoročni finančni prihranek. PKS imajo vsekakor velik potencial v prihodnosti. V razvitih državah postajajo ljudje vse bolj ozaveščeni glede porabe vode, na področjih, kjer je količina vode omejena pa so taki krmilni sistemi še posebej dobrodošli in še kako potrebni.

Slabosti je malo in so predvsem odvisne od uporabnika, uporabniških nastavitev in kakovosti samega krmilnega sistema.

2.1.2 Enostavni krmilni sistemi

Enostavni krmilni sistemi so v primerjavi s PKS (2.1.3) primitivnejši. Gre za precej trivialne sisteme, kjer lahko uporabnik določi interval zalivanja in v nekaterih primerih tudi količino vode. Slabost takih sistemov je, da so v veliki meri odvisni od uporabnika in uporabniških nastavitvev. Taki sistemi so cenovno ugodni in dostopni. Interakcija med uporabnikom in krmilnim sistemom je možna samo fizično na lokaciji, kjer se krmilni sistem nahaja.

2.1.3 Pametni krmilni sistemi

Pametni krmilni sistemi so vsi krmilni sistemi, ki so zmožni prilagajanja glede na vremenske razmere. PKS so v vseh pogledih boljša rešitev, vendar njihov razvoj krni kompleksnost in posledično cenovno dostopnost. V prihodnosti pričakujemo večje povpraševanje po PKS sistemih, kar bo pocenilo razvoj in gonilo nadaljnji razvoj. Pametne krmilne sisteme lahko v grobem razdelimo na tri skupine:

- **Krmilni sistemi s senzorji**

Nadgradnja EKS, ki vsebuje raznovrstne senzorje za spremljanje vremenskih razmer. Glavna funkcija EKS ostaja nastavitvev intervala namakanja, le da v tem primeru prilagaja krmiljenje glede na trenutne vremenske razmere. Tako se lahko izognemo nepotrebnemu namakanju med dežjem, saj bo senzor ob zaznavi dežja prekinil z namakanjem. Uporabniške nastavitve se kot pri EKS vnašajo neposredno na lokaciji.

- **Krmilni sistemi z brezžično povezavo na splet**

Krmilni sistem ima brezžični modul (najpogostje WiFi ali RF modul) preko katerega se povezuje na splet in s pomočjo različnih API-jev pridobiva podatke o vremenski napovedi. Na podlagi vremenskih podatkov PKS izdelava načrt namakanja. Prednost pred PKS s senzorji je ta, da ima tak sistem dinamičen načrt, ki se spreminja glede na vremensko napoved. V primeru netočne vremenske napovedi bo načrt namakanja neprimeren. Krmilniki z brezžično povezavo so večinoma razširjeni z mobilno in/ali spletno aplikacijo, ki omogoča spremljanje trenutnega statusa PKS, preverjanje načrta namakanja in vnašanje uporabniških nastavitvev. To seveda ne velja za vse PKS tega tipa, saj se lahko implementacija razlikuje od sistema do sistema.

- **Krmilni sistemi s senzorji in brezžično povezavo na splet**

Krmilni sistem s senzorji in brezžično povezavo združuje predhodna tipa PKS, kar pomeni, da ima tudi vse njune prednosti, hkrati pa izključuje omenjene slabosti.

Tak tip sistema je trenutno najbolj natančen. Načrt namakanja je poleg vremenske napovedi, odvisen tudi od trenutnega stanja, kar lahko v določenih primerih (npr. napačne vremenske napovedi, težava z brezžično povezavo) odigra veliko vlogo.

Tako kot pri PKS z brezžično povezavo, tudi tak tip PKS omogoča in v večini primerov nudi razširitev z mobilno in/ali spletno aplikacijo, preko katere je možno spremljanje in upravljanje sistema.

2.2 Samodejni sistem za krmiljenje zalivalno-namakalnih sistemov

V času nastanka ideje je bil trg PKS majhen, sama tehnologija pa še v povojih in poskusni fazi. Najbolj dostopni so sicer še vedno EKS z možnostmi priključitve različnih senzorjev, vendar v kombinaciji z vsemi potrebnimi senzorji, cena ni bila več zanesljiva in se je počasi približevala boljšim PKS (z brezžično povezavo ali kombinaciji slednje s senzorji). Interakcija takih sistemov z uporabnikom je še vedno fizična na lokaciji sistema, kar ni vedno zadoščalo našim potrebam. V primeru ustreznega PKS pa je v večini primerov problem predstavljala cena.

Želeli smo PKS, ki bo sposoben samodejnega delovanja brez, oziroma z minimalno človeško interakcijo. Da bi lahko to dosegli, mora biti način delovanja krmilnega sistema kar se da robusten; v primeru odpovedi enega ali več modulov, mora sistem še vedno pravilno delovati. Da lahko to dosežemo mora biti krmilni sistem opremljen z vsemi potrebnimi senzorji, s katerimi lahko pridobimo čim več informacij o trenutnih vremenskih razmerah in razmerah v prsti. Prav tako želimo izdelati načrt namakanja za naslednjih nekaj dni in imeti oddaljen dostop do krmilnega sistema. V kasnejši fazi projekta želimo izvajati tudi podatkovno rudarjenje nad vsemi pridobljenimi podatki (več o tem v poglavju 4.2), vendar potrebujemo primerno strojno opremo, ki bo slednje omogočala. Podatkovno rudarjenje je proces analize velike količine podatkov, z namenom pridobitve neke informacije [19]. Hkrati želimo, da je naš krmilnik robusten in zanesljiv, predvsem pa energijsko varčen.

Večina trenutnih PKS je namreč omejena na področje nekega objekta, ki zagotavlja električno energijo in WiFi dostop. Samodejni krmilni sistem bi radi izboljšali do te mere, da bi ga lahko upravljali v območju nekaj 100 metrov ali tudi več, kar pa je izven dosega standardnih WiFi modulov. Prav tako obstajajo situacije, ko električno omrežje ni na voljo in je potrebno napajanje preko baterije ali akumulatorja, zato mora biti krmilnik energijsko varčen.

Iz slednjih razlogov smo sistem implementirali na dveh različnih platformah, kjer je ena zadolžena za krmiljenje periferije (senzorji, ventili, itd.), druga pa obdeluje podatke na podlagi katerih izvaja potrebne operacije. Ker so običajni WiFi moduli omejeni na razmeroma majhna območja, smo se za povezavo med moduli odločili za radiofrekvenčne (RF) module, ki nam omogočajo prenos podatkov na bistveno večjih razdaljah. Domet je v veliki meri odvisen od moči oddajnika in antene RF modula, vendar lahko RF modul z nekaj manjšimi popravki brez težav nadgradimo in s tem povečamo domet. Podrobnosti o RF modulih in platformah si bomo ogledali v poglavju 3.1.

Med periferne naprave poleg RF modula, spadajo še razni senzori in elektromagnetni ventili. Elektromagnetni ventili so ventili z dvema stanjima (odprt ali zaprt), odpiranje in zapiranje ventila pa povzroči spremembo elektromagnetnega polja. Uporabljata se dva različna tipa ventilov in sicer NC ventil (normally closed - zaprt v privzetem stanju) in NO ventil (normally open - odprt v privzetem stanju). Pod napestjo se NC ventil odpre, medtem ko se NO ventil zapre. Za naše potrebe je uporabljen NC ventil, saj v primeru izgube električne energije, preprečuje nenadzorovano iztekanje vode. V zaključni nalogi se ventilom ne bomo podrobneje posvečali, saj je njihova uporaba odvisna od uporabnika.

Med perifernimi napravami so najpomembnejši senzori. Kot smo že v uvodnem poglavju 1 omenili, zaključna naloga zajema samo razvoj in delovanje prototipa samodejnega krmilnega sistema, saj obseg projekta presega okvirje zaključne naloge. Našemu krmilnemu sistemu, smo že na začetku dodali senzore za merjenje temperature, vlažnosti zraka in zračnega tlaka, čeprav jih ne bomo posebej obravnavali. Omenjeni senzori so predvideni za naslednje faze razvoja. Več o samem nadaljevanju projekta, možnosti nadgradnje in izboljšav je napisanega v poglavju 4.2. Uporabljeni senzori bodo predstavljeni in opisani v poglavju 3.2.2.

Najpomembnejša senzorja sta po našem mnenju senzor za zaznavanje dežja in senzor za merjenje vlažnosti zemlje. Slednji nam podaja informacijo o dejanskem stanju in potrebi rastline po vodi (ob predpostavki, da poznamo rastlino in njeno zahtevo po vodi). Tako se lahko izognemo scenariju, ko rastlina prejme preveč ali premalo vode, kar posledično ovira njeno rast. Podobno vlogo ima v krmilnem sistemu tudi senzor za dež, saj lahko v primeru zaznanega dežja, preventivno zaustavimo proces namakanja. S tem preprečimo, da bi rastlina v kombinaciji namakanja in dežja, prejela preveliko količino vode, hkrati pa tudi bolj razumno in odgovorno ravnamo s porabo vode. V primeru manjših padavin se lahko izkaže, da vlažnost zemlje še vedno ni primerna, zato lahko nadaljujemo z namakanjem. Ker rastlino vedno namakamo samo toliko, kolikor je potrebno, privarčujemo tudi na vodi. Prav tako je pomemben tudi senzor vodnega pretoka s pomočjo katerega nadzorujemo in ocenimo porabo vode, pri posameznem

namakanju.

V idejni zasnovi smo določili, da bodo senzori za merjenje vlažnosti zemlje samostojni moduli, opremljeni z RF modulom, ki omogoča komunikacijo z zunanjo enoto krmilnega sistema 3.3. Prednost modularne in samostojne zasnove je predvsem v tem, da imamo lahko več sensorjev, ki ima za vsako cono (cone lahko razdelimo npr. po kulturi/vrsti rastline ali glede na potrebe po vodi) svoj senzor za merjenje vlažnosti zemlje. Vemo namreč, da ima vsaka vrsta rastline različne potrebe po vodi. S tako zasnovo lahko omogočimo, da vsako cono namakamo po potrebi in neodvisno od potreb ostalih con (podrobneje v poglavju 4.2). Za razvoj prototipnega primera smo sicer uporabili samo en senzor. Med senzori je tudi svetlobni senzor, ki sistemu sporoča svetlobno nasičenost, s pomočjo katere lahko razlikujemo med nočjo in dnevom. Glede na vremenske razmere lahko določimo, da sistem namaka samo ponoči ali v oblačnem vremenu. Takrat je namreč zaradi manjše temperature izhlapevanje vode manjše, kot sicer.

3 Metodologija

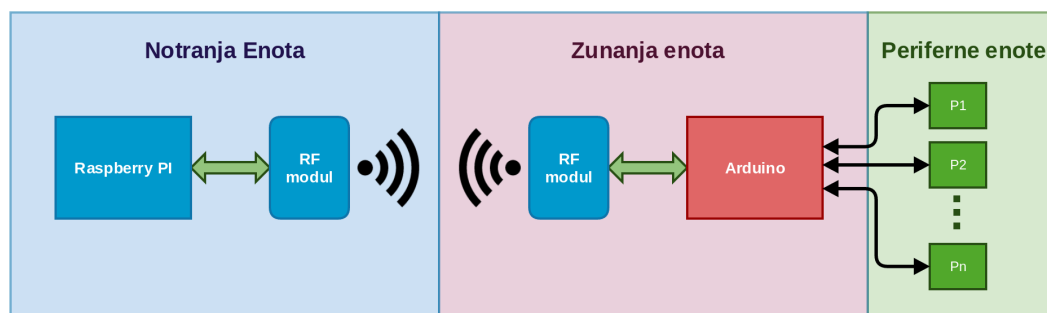
3.1 Arhitektura sistema

Načrtovanje arhitekture sistema je zelo pomembna faza razvoja, saj v primeru slabo zastavljene arhitekture lahko okrnimo nadaljnji razvoj sistema. To je še posebej pomembno v primeru odprtokodne rešitve, ki predvideva nadaljnji razvoj znotraj odprtokodne skupnosti. Poleg možnosti razširitev, ki dopolnjujejo sistem z novimi funkcionalnostmi, mora sistem omogočati dobro skalabilnost in robustnost. Robustnost sistema se običajno zagotovi z modularnim sistemom, pri katerem sistem razdrobimo na posamezne, med seboj neodvisne module. Komunikacijo med moduli običajno implementiramo s pomočjo odprtokodnih standardov in tehnologij. Na ta način omogočimo integracijo novih modulov in entitet sistema, če le te podpirajo zahtevane standarde. Skalabilnost lahko v najosnovnejši obliki opišemo kot sposobnost nepričakovane rasti sistema. Običajno pri načrtovanju in razvoju sistemov ne moremo natančno predvideti obsega. Ponovno je pri razvoju odprtokodnih rešitev to zelo pomembno, saj lahko skupnost sistem razširiti na način, ki si ga ob načrtovanju nismo predstavljali. Skalabilnost običajno dosežemo tako, da pri načrtovanju sistema ne omejujemo ničesar, če to ni nujno potrebno.

Arhitekturo sistema lahko razdelimo na tri krovne module in sicer, notranjo enoto, zunanjo enoto in periferne enote, kot je razvidno na sliki 3.1. Vsak od krovnih modulov je fizično in sistemsko ločen. Moduli med seboj komunicirajo s pomočjo standardnih komunikacijskih protokolov, ki so podrobneje opisani v poglavju 3.2. Med zunanjo in notranjo enoto poteka komunikacija s pomočjo radio frekvenčnih modulov [21], ki omogočajo stabilnost na želeni razdalji. Notranja enota je zadolžena za pridobivanje podatkov s strani uporabnika, spleta in zunanje enote. Zbrane podatke modul obdela in glede na vhodne podatke prilagodi program zalivanja. Zunanja enota je v večji meri odvisna od omejene električne energije zato je večino časa v varčevalnem načinu. V varčevalnem načinu modul porabi zelo malo električne energije tako, da samo vzdržuje aktivnost radio frekvenčnega modula. V primeru, da notranja enota pošlje signal zunanji enoti, se aktivira in preide iz varčevalnega načina v normalen način. Modul ostane v normalnem načinu, dokler je komunikacija z ostalimi moduli aktivna, nakar ponovno preide v varčevalni način. Primarna naloga zunanje enote je krmiljenje perifernih enot.

Iz perspektive sistemske arhitekture, število in tip perifernih enot ni pomembno.

Za realizacijo notranje enote smo uporabili popularno platformo za razvijalce Raspberry Pi, medtem ko smo za zunanjo enoto izbrali platformo Arduino, ki je primernejša za krmiljenje analognih perifernih enot [15]. Podrobnosti posamezne platforme so opisane v poglavju 3.2.



Slika 3.1: Prikaz arhitekture krmilnega sistema, kjer lahko vidimo delitev na notranjo, zunanjo in periferno enoto.

3.2 Uporabljena oprema, tehnologije in programski jeziki

Sistem smo realizirali s pomočjo cenovno ugodnih komponent in opreme. Poleg opreme smo uporabili vrsto standardov za komunikacijo med posameznimi moduli. Za razvoj programske opreme smo uporabili več programskih jezikov in razvojnih okolij.

Pri izbiri mikrokrmilnika je zelo pomembno, da podrobno pregledamo sistemske zahteve in predvsem podporo čipa, ki mikrokrmilnik poganja. Potrošniška elektronika je izjemno pocenila proizvodni proces elektronike, kar je posledično omogočilo hiter razvoj velike količine odprtokodnih platform za prototipiranje. Kljub visoki nasičenosti trga z mikrokrmilnimi platformami je daleč najpopularnejša platforma Arduino. Platforma Arduino je odprtokodni projekt, ki je nastal v Italiji in katerega glavni namen je bil postati pripomoček za učenje elektrotehničnih ved. Zaradi odprtokodnosti in dobre sistemske zasnove je postal zelo popularen tudi med drugimi proizvajalci elektrotehničnih komponent, ki danes že razvijajo komponente, kompatibilne z mikrokrmilniki Arduino. Mikrokrmilniki Arduino temeljijo na 8 bitnem Atmel AVR čipovju [14] medtem ko novejšje verzije že vsebujejo 32-bitne Atmel ARM procesorje. Poleg strojne rešitve Arduino ponuja tudi programsko okolje, ki je v primerjavi s konkurenti zelo uporabniku prijazno. Razvojno okolje (IDE) poleg prevajanja iz programskega jezika

v strojno kodo podpira tudi enostavno nalaganje strojne kode na mikrokrmilnik. To je implementacija USB protokola na mikrokrmilniku Arduino.

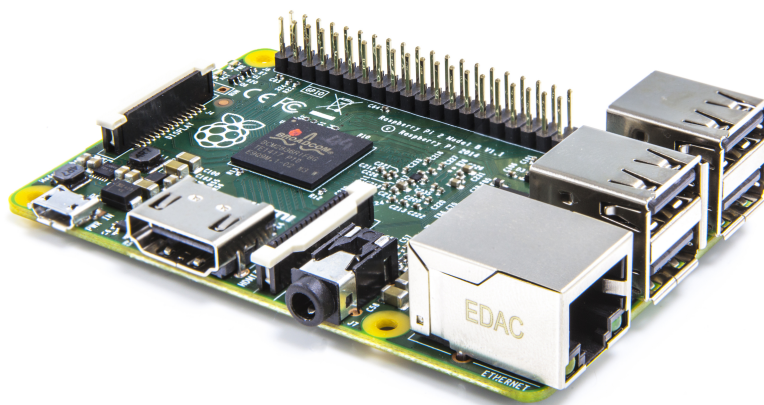


Slika 3.2: Arduino UNO: osnovna odprtokodna mikrokontrolna platforma, namenjena razvijalcem.

Poleg enostavnega razvoja nam Arduino ponuja odprtokodne knjižnice s pomočjo katerih enostavneje krmilimo periferne enote. Knjižnica (programska knjižnica) je skupek programov ali podprogramov, s katerimi si pomagamo pri razvoju programske opreme. Poleg tega je na voljo veliko zunanjih senzorjev in naprav, ki uradno podpirajo Arduino platformo. Arduino je izredno priročen za krmiljenje analognih naprav, nima pa računske zmogljivosti procesorja, da bi lahko celotno programsko opremo poganjali na Arduino platformi. V ta namen smo dodali platformo Raspberry Pi. [22]

Raspberry Pi je računalnik, ki ne presega velikosti običajne kreditne kartice, medtem ko v računski moči presega marsikateri mobilni telefon. Razvila ga je Angleška fundacija po imenu Raspberry Pi Foundation z namenom širitve in podpore pedagoškemu procesu računalništva v manj razvitih delih sveta. Že ob izidu leta 2012 je bilo na spletu na voljo več Linux distribucij, ki so bile posebej prevedene za platformo, kasneje pa so med podprte operacijske sisteme dodali tudi Windows 10. Raspberry Pi prav tako kot Arduino omogoča krmiljenje analognih naprav, vendar je poraba električne energije razmeroma večja. Za razvoj samodejnega krmilnega sistema smo sicer uporabili Raspberry Pi 2, ki je naslednik predhodno omenjenega, vendar je v našem primeru trenutno vseeno, katerega izberemo. Raspberry Pi uporabimo kot glavno računsko vozlišče, ki je zadolženo za zbiranje podatkov o stanju perifernih enot in optimizaciji porabe vode glede na vremenske razmere. Za operacijski sistem smo izbrali Linux distribucijo Raspbian, ki je posebna verzija popularne distribucije Debian prevedena posebej za Raspberry Pi platformo. Za komunikacijo smo na Raspberry Pi priklopili tudi USB Wifi modul, s pomočjo katerega lahko dostopamo do spleta ter radio frekvenčni modul,

ki omogoča komunikacijo z mikrokrmilnikom Arduino.



Slika 3.3: Raspberry Pi 2 (model B): nizkocenovni računalnik v velikosti kreditne kartice, namenjen predvsem v izobraževalne namene.

Da bi lahko platformi med sabo brezžično komunicirali, potrebujemo temu primerno opremo. Običajna brezžična omrežja nimajo dovolj velikega dometa, da bi bila v našem primeru uporabna. Iz slednjih razlogov smo se odločili, da bomo za brezžično komunikacijo uporabili radiofrekvenčni modul NRF24L01P (NRF24) [8]. NRF24 je razširjen med razvijalci zaradi ugodne cene, solidnega dometa in zanesljivosti pri prenosu podatkov. Modul deluje v frekvenčnem območju 2,4 GHz, največja teoretična razdalja pri kateri lahko RF moduli komunicirajo med seboj, je 1100 m (vidno polje). Prenos podatkov se izvršuje po paketih in sicer po protokolu Enhanced ShockBurst - ESB (poglavje 3.2.1), nastavimo lahko tri različne hitrosti prenosa, in sicer: 250kbps, 1Mbps in 2Mbps.



Slika 3.4: Radio frekvenčni modul NRF24L01P: glede na ceno, zanesljiv in razširjen RF modul med razvijalci.

3.2.1 Protokoli

Komunikacija med posameznimi moduli je izrednega pomena. V primeru izpada komunikacije je celoten sistem nefunkcionalen. Da bi zagotovili stabilnost in morebitno razširljivost sistema, smo komunikacijo implementirali s pomočjo standardnih protokolov. V našem sistemu uporabljamo tri tehnologije za komunikacijo in sicer, WiFi tehnologijo, USB protokol in radiofrekvenčni (RF) oddajnik in sprejemnik.

Namen USB protokola je komunikacija med sistemom in osebnim računalnikom ali zunanji napravami, ki niso del sistema. S pomočjo serijskega protokola na USB konektorju omogočamo kontrolo sistema, parametrov in razhroščevanje v primeru okvare.

WiFi tehnologija je prisotna na Raspberry Pi platformi v obliki USB modula, ki omogoča sistemu dostop do interneta ter hkrati varen in oddaljen dostop do sistema s pomočjo protokola SSH, ki uporablja standardni TCP/IP protokol.

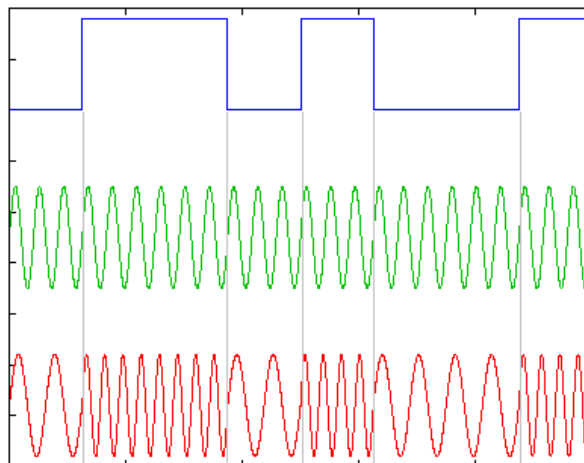
NRF24L01

Delovanje RF modula NRF24L01 je nekoliko bolj zapleteno in več nivojsko, zato bomo predstavili tri glavne ravni in sicer: pretvarjanje podatkov iz digitalne oblike v analogno in obratno, protokol za izmenjavo podatkov ter vmesnik za komunikacijo RF modula z zunanjo/notranjo enoto.

- **GFSK modulacija**

Modulacija je postopek spreminjanja lastnosti frekvenčnega signala (imenujemo ga nosilni signal) na podlagi modulacijskega signala. Modulacijski signal vsebuje podatke, ki jih želimo prenašati. Poznamo analogne, digitalne in pulzne modulacije. Naš modulacijski signal je digitalni, zato se bomo osredotočili na digitalno modulacijo. Digitalno modulacijo izvajamo z različnimi tehnikami. Najbolj znani sta digitalna frekvenčna modulacija (FSK) in digitalna amplitudna modulacija (AKS). Slednji se uporabljata tudi za modulacijo radijskih valov.

GFSK (Gaussian frequency-shift keying) modulacija, je vrsta digitalne frekvenčne modulacije, ki uporablja Gaussov filter za glajenje pozitivnih in negativnih frekvenčnih odklonov (tej predstavljajo binarno 0 in 1) [16]. GFSK se uporablja tudi v DECT (Digital Enhanced Cordless Telecommunications) in Bluetooth napravah [25]. Naprava, ki prejme prehodno moduliran signal, ustvari demodulacijo, torej obraten postopek, kot pri modulaciji. Tako iz moduliranega signala dobimo ponovno digitalen signal, ki ga nato naprava primerno obdela.



Slika 3.5: Pretvarjanje digitalnega signala s tehniko digitalne frekvenčne modulacije. Vidimo lahko, kako se frekvenca nosilnega signala (v sredini) spremeni glede na modulatorski signal (na vrhu).

- **Enhanced ShockBurst protokol**

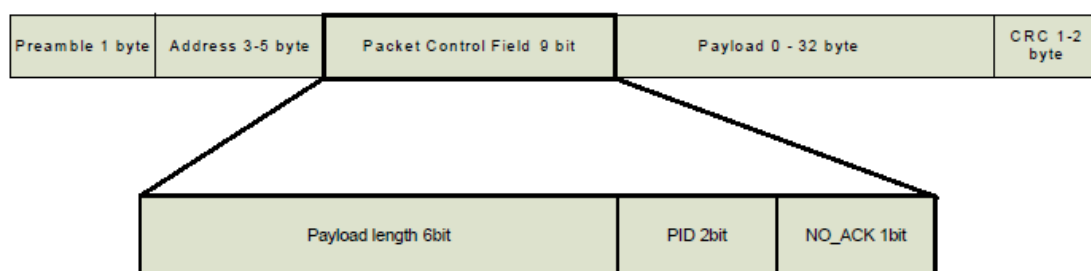
RF moduli imajo kar nekaj pomanjkljivosti, ena izmed teh je, da je RF tehnologija v splošnem podvržena motnjam in je izguba podatkov med komunikacijo relativno pogosta. Težava se pojavi tudi zaradi načina komunikacije, saj naš RF modul komunicira dvosmerno (branje, pisanje), če smo natančni komunikacija poteka v Half-Duplexa. Za razliko od Full-Duplexa, kjer se lahko napravi istočasno pogovarjata med seboj, Half-Duplex omogoča le branje ali pisanje v danem trenutku. V kolikor potrebujemo branje in pisanje, moramo programsko določiti, kdaj bomo brali in kdaj pisali. V ta namen je proizvajalec NRF24L01, Nordic Semiconductor, razvil poseben protokol ESB (Enhanced ShockBurst) [8], ki rešuje slednje probleme.

ESB je protokol, ki je paketno osnovan sloj podatkovne povezave, s samodejnim tvorjenjem podatkovnih paketov, sistemsko uro, samodejnim pošiljanjem potrditvenega paketa (ACK) in ponovnim pošiljanjem sporočila v primeru napake. Vse potrebne funkcionalnosti za pošiljanje in prejemanje podatkov se izvajajo neodvisno, brez vključevanja centralne mikrokrmilne enote.

Glavna funkcionalnost modula, ki jo ESB omogoča, je samodejno upravljanje paketnih transakcij, ki omogočajo dvosmerno komunikacijo. Paketna transakcija je izmenjava paketa med dvema RF moduloma, pri čemer se eden od modulov obnaša kot primarni sprejemnik, drugi pa kot primarni oddajnik. Primarni zato, saj modul kot primarni oddajnik, deluje za kratkem čas kot sprejemnik, obratno pa velja tudi za primarni sprejemnik. Razlog za to je pošiljanje potrditvenih paketov (ACK) med prenosom podatkovnih paketov (več v nadaljevanju). Za lažji

opis delovanja samega protokola bomo najprej opisali posamezne funkcionalnosti, ki se izvajajo med komunikacijo nato pa.

Paket sestoji iz različnih komponent, ki so določene po formatu (pravilu). Format je potreben za uspešno komunikacijo med moduloma, saj bi se lahko v nasprotnem primeru zgodilo, da modul ne bi znal prebrati paketa, posledično pa bi ga zavrgel. V našem primeru paket sestavljajo naslednje glavne komponente (najpomembnejši bit je na levi strani): preambula, naslov sprejemnika, polje s paketno kontrolo, vsebina sporočila in CRC komponenta.



Slika 3.6: Format podatkovnega paketa sestoji iz preambule, naslova, paketnega kontrolnega polja, vsebine sporočila in CRC preverjanja.

Preambula vsebuje sekvenco osmih bitov, ki se uporablja za sinhronizacijo demodulatorja na strani sprejemnika. ESB protokol poskrbi za vnos naslova sprejemnika, kateremu se pošilja paket. Naslov zagotavlja, da paket prejme točno določen sprejemnik in se s tem izognemo neželeni komunikaciji z drugimi sprejemniki. Paketno kontrolno polje sestoji iz treh komponent. Prva je šest bitna oznaka, ki predstavlja dolžino sporočila (payload) v bajtih. Kontrolno polje vsebuje še identifikacijsko številko paketa, ki nam pove ali gre za nov paket ali za ponovno transakcijo ter zastavico, ki določa ali se izvede pošiljanje potrditvenega paketa. V sporočilu jedra so shranjeni podatki, ki jih pošiljamo. Dolžina sporočila je lahko statična ali dinamična, vendar v obeh primerih ne večja kot 32 bajtov. ESB protokol zagotavlja tudi preverjanje napak pri prenosu podatkov. V ta namen je definirano zadnje polje v paketu, ki vsebuje CRC (Cyclic Redundancy Check) vrednost v dolžini 1-2 bajta. CRC vrednost tvori izračun naslova, paketne kontrole in sporočila.

Samodejno pošiljanje potrditvenega paketa je funkcionalnost, ki jo samodejno izvede sprejemnik s tem, da pošlje ACK paket oddajniku. S to funkcionalnostjo se zmanjša obremenitev mikrokontrolne enote in poraba energije. Ko

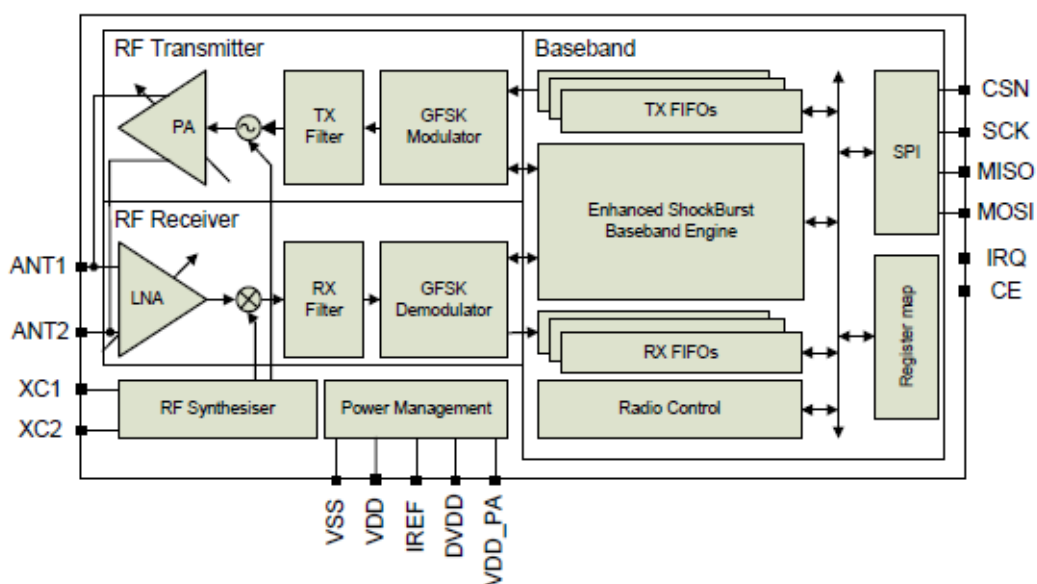
sprejemnik prejme paket, samodejno preklopi v stanje oddajanja in pošlje ACK paket oddajniku, nato pa preklopi nazaj v stanje oddajanja. Obratno velja v primeru oddajnika, ko pričakuje paket ACK 3.7.



Slika 3.7: Delovanje ESB protokola v primeru pisanja. Vidimo lahko, da sprejemnik po sprejemu sporočila, samodejno preklopi v stanje za pošiljanje ter odgovori pošiljatelju z ACK paketom. Prav tako, oddajnik takoj po poslanem paketu, preklopi v stanje poslušanja in čaka na ACK paket.

Samodejno prepošiljanje se izvede v primeru, ko imamo omogočeno pošiljanje ACK paketa 3.6, ampak tega ne prejmemo po poslanem podatkovnem paketu. Število ponovitev in časovni interval pošiljanja je možno programsko nastaviti.

Pošiljanje in sprejemanje paketa poteka po vnaprej določenih ceveh (mehanizem za komunikacijo - angleško *pipe*). Med oddajanjem ESB protokol tvori podatkovne pakete na podlagi sistemske ure. NF24L01 modul ima za sporočila v pošiljanju in prejemanju implementirano FIFO vrsto (TX in RX vrsto), zato se v primeru večje količine sporočil, le-ta shranijo v temu primerno vrsto. Po izdelanem paketu se signal pretvori z GFSK modulacijo in pošlje primarnemu sprejemniku. Primarni oddajnik nato začne s poslušanjem in čaka na ACK paket. Primarni sprejemnik, ki je v stanju sprejemanja, konstanto išče signal z ustreznim naslovom v demodulacijskem signalu. Ko ESB najde pravilen naslov, obdela preostanek paketa, zamenja stanje v pisanje in pošlje ACK paket primarnemu oddajniku. Med obdelavo paketa primarni sprejemnik validira paket s CRC preverjanjem. Če je paket brez napak, se zapiše v RX vrsto. V primeru, da je med prenosom prišlo do napake in primarni oddajnik ni prejel ACK paketa, pošilja paket, dokler ga ne prejme ali se izteče število poizkusov (določeni programsko s strani uporabnika).



Slika 3.8: Arhitektura NRF24L01: iz slike je razvidna povezava med SPI, ESB in GFSK moduli.

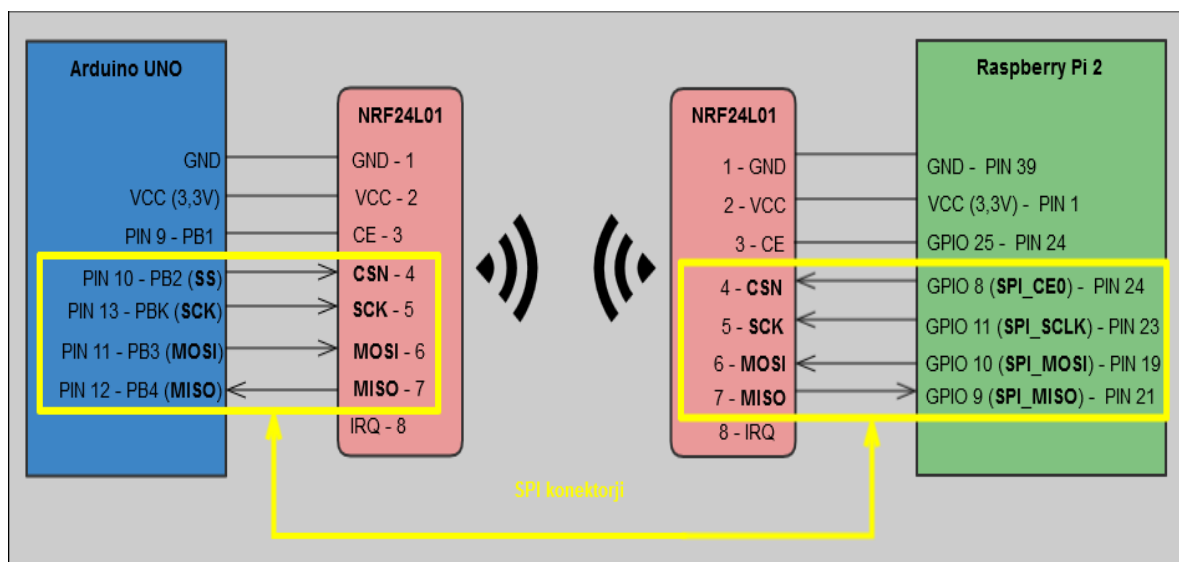
• SPI vmesnik

Serial Peripheral Interface bus (SPI) ali serijski vmesnik za povezavo naprav, je sinhroniziran serijski podatkovni protokol [17]. Uporablja se za komunikacijo med napravo (mikrokrmilnik, analogno-digitalni pretvorniki,...) in perifernimi napravami na kratkih razdaljah. SPI povezavo vedno tvori primarna naprava (master device) in ena ali več perifernih naprav, povezava pa poteka v full duplex načinu. SPI vodilo ima 4 logične signale:

- SCLK (Serial Clock): je izhodi signal systemske ure v primarni napravi
- MOSI (Master Output, Slave Input): povezava po kateri primarna naprava pošilja podatke periferni napravi
- MISO (Master Input, Slave Output): povezava po kateri primarna naprava prejema podatke s strani periferne naprave.
- SS (Slave Select): povezava na vsaki napravi, preko katere lahko primarna naprava aktivira ali onemogoči periferno napravo

V našem primeru se SPI vmesnik uporablja za komunikacijo, med platformama Arduino in Raspberry Pi z modulom NRF24L01. Platforma Arduino ima SPI vmesnik že privzeto aktiviran. Za njegovo uporabo moramo v razvojno okolje Arduino IDE, uvoziti SPI knjižnico, s katero lahko upravljamo SPI povezave [1]. Na starejših verzija platforme Raspberry Pi, je potrebno v konfiguracijskem meniju, omogočiti SPI modul [7].

Na sliki 3.9 lahko vidimo shemo povezave platforme Arduino UNO in Raspberry Pi 2 z modulom NRF24L01. Prikazana je celotna povezava vključno z napajanjem. IRQ konektor (vhod za prekinitve) v našem primeru nismo potrebovali, zato ga nismo povezali s platformama.

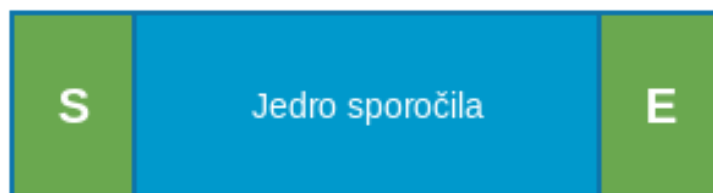


Slika 3.9: Shema povezave platforme Arduino UNO in Raspberry Pi 2 (B) z NRF24L01 modulom. V rumenem okvirju so prikazani SPI konektorji.

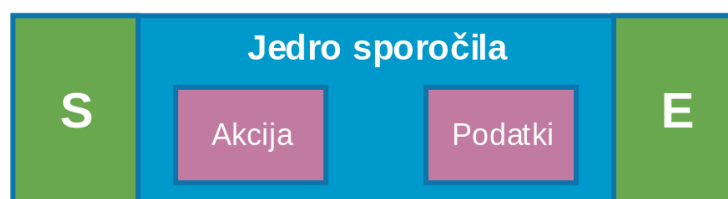
Kljub temu, da NRF24L01 modul, že samodejno preverja, ali je bil paket uspešno poslan, potrebujemo v našem primeru tudi višje nivojski protokol. Ob izvedbi določenih akcij namreč pričakujemo povratno sporočilo, ki ga moramo znati obdelati. Prav tako se lahko zgodi, da ravno med prenosom pride do težav z zunanjo ali notranjo enoto (na primer izguba napajanja), zato smo zaradi strmenja k robustnosti sistema izdelali lasten format za pošiljanje sporočil na visokem nivoju.

Sporočilo je definirano z dvema zastavicama, ki signalizirata začetek in konec sporočila. Vsi podatki, ki so prejeti ali poslani med dvema zastavicama imenujemo jedro sporočila kot je razvidno iz 3.10 slike

Oba modula imata enak nabor akcij, ki jih podpirata. Jedro sporočila je sestavljeno iz dveh delov, ki sta ločena po velikosti. Prvih 16 bitov identificira akcijo, medtem ko je ostali del namenjen podatkom. Modul, ki akcijo prejme, jo izvede in pošiljatelju odgovori z enako akcijo. V primeru, da akcija od prejemnika ne zahteva odgovora v obliki podatkov jo prejemnik vseeno pošlje in namesto podatkov pošlje zastavice, ki pošiljatelju signalizirajo ali se je akcija izvedla uspešno ali neuspešno.



Slika 3.10: Groba struktura sporočila, kjer 'S' predstavlja začetek, 'E' pa konec sporočila. Jedro sporočila je sestavljen iz oznake akcije in zahtevanih podatkov.



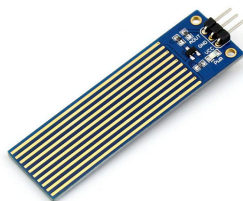
Slika 3.11: Jedro sporočila sestoji iz 16 bitne oznake akcije in niza podatkov, če jih akcija zahteva.

3.2.2 Senzorji

Da bi lahko sistem napovedal kdaj in koliko je treba zalivati, potrebuje podatke o trenutnem stanju na lokaciji. V ta namen sistem dovoljuje priključitev perifernih enot v obliki senzorjev s pomočjo katerih lahko izmerimo stanje na lokaciji.

1. Senzor za dež YL-83

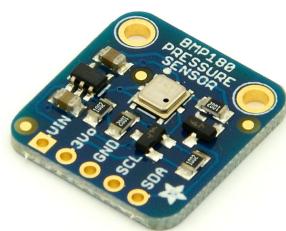
Senzor za dež je tehnično razmeroma enostaven, kljub temu zelo pomemben senzor. Upornost vezja je odvisna od količina dežja na vezju. Večja, kot je količina dežja, manjša je upornost. Senzor vrača vrednosti v analogni obliki, kar pomeni, da moramo podatke sami interpretirati [9].



Slika 3.12: Senzor za dež: vezje predstavlja variabilni upor. Upornost vezja se spreminja glede na količino tekočine na vezju.

2. Senzor zračnega pritiska BMP180

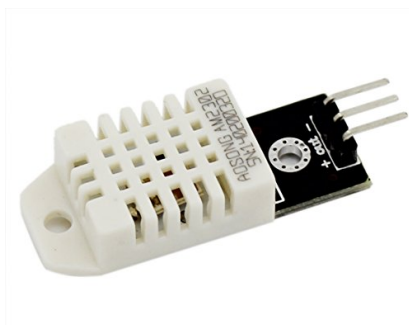
Senzor Bosch Sensortec BMP180, je modul za merjenje zračnega pritiska, višine in temperature, ki je združljiv s platformo Arduino. Višine in temperature sicer ne odčitavamo. Višine v našem konkretnem primeru (vsaj trenutno) ne potrebujemo, poleg tega pa nam senzor vrne absolutno višino glede na zračni pritisk. Točno višino bi v primeru potreb lahko s primernim API-jem (Google Maps Elevation API [2]) na podlagi vnesene lokacije krmilnega sistema. Temperaturo merimo s senzorjem za temperaturo in vlažnostjo zraka DHT22 4, saj so temperaturni odčitki natančnejši ($\pm 0.5^{\circ}\text{C}$ odstopanja od realne vrednosti) v primerjavi z BMP180 ($\pm 1^{\circ}\text{C}$ odstopanja od realne vrednosti). Za modul smo se odločili, ker je cenovno zelo ugoden, predvsem na račun masovne proizvodnje zaradi vgradnje v mobilne naprave. Delovno območje senzorja je med 300 - 1100 hPa z odstopanjem $\pm 0,12$ hPa pri 25°C , temperaturno območje pa je med -40 - 80°C [11].



Slika 3.13: Senzor BMP180 za merjenje zračnega pritiska, temperature in absolutne višine. V našem ga uporabljamo samo za merjenje zračnega tlaka.

3. Senzor za merjenje vlažnosti zraka in temperature DHT22

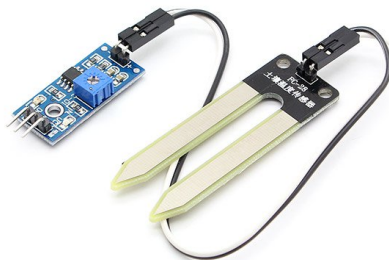
DHT22 je cenovno ugoden digitalni in relativno zanesljiv senzor za merjenje zračne vlažnosti in temperature. Deluje v območjih s temperaturo od -40 in 80° z odstopanjem $\pm 0,5^{\circ}\text{C}$ in vlažnostjo zraka med 0 in 100% z odstopanjem $\pm 2\%$ RH. Dobljeni podatki so v digitalni obliki [12].



Slika 3.14: Senzor DHT22 uporabljamo za merjenje vlažnosti zraka in temperature.

4. Senzor vlažnosti zemlje FC-28

Senzor za vlažnost zemlje je trivialen senzor, ki meri napetost med polaritetama. Višina napetosti je odvisna od stopnje vlažnosti zemlje. Večja, kot je vlaga zemlje, večja je napetost. Senzor nam vrača analogen podatek, katerega nato sami interpretiramo [10].



Slika 3.15: Senzor vlažnosti zemlje s pripadajočim vmesnikom. Vmesnik pretvori podatke iz senzorja, v berljivo obliko za platformo Arduino.

5. Svetlobni senzor - fotoupor

Svetlobni senzor - fotoupor je variabilni upor, katerega upornost se spreminja v obratnem sorazmerju glede na svetlobo. Torej je v temnem prostoru, upornost fotoupora največja (imenujemo jo dark resistance - upornost v temnem prostoru), z večanjem intenzivnosti svetlobe pa upornost pada. Kot že omenjeno v poglavju 2.2, uporabljamo fotoupor predvsem za predvidevanje, kdaj se mrači/dani in oblačnosti tekom dneva. Upornost fotoupora v temnem prostoru je $1\text{M}\Omega$, deluje pa v temperaturnem območju med -20 in $80\text{ }^{\circ}\text{C}$. Dobljeni podatki so v analogni obliki, le-te pa sami interpretiramo [13].



Slika 3.16: Fotoupor uporabljamo za merjenje svetilnosti.

6. Merilec vodnega pretoka YF-S201

Merjenje vodnega pretoka je nekoliko bolj zapleteno od ostalih senzorjev. Poenostavljeno, senzor izmeri pretok tekočine s pomočjo vgrajene vetrnice, ki pošlje elektromagnetni pulz za vsako rotacijo. Število zabeleženih pulzov je premo sorazmerno s količino pretočene tekočine. Signal je analogen, zato moramo z določeno formulo na podlagi opisanih vrednosti izračunati pretok vode v standardnih mer-skih enotah [4].



Slika 3.17: Merilec vodnega pretoka YF-S201, bomo uporabili za oceno porabljene vode za namakanje.

3.3 Programska oprema

Programsko opremo lahko razdelimo na dva ločena dela, in sicer, programsko opremo za zunanjo enoto in notranjo enoto. Poleg različnih vlog je, različna tudi njuna arhitektura 3.1. Raspberry Pi je tako strojno, kot tudi programsko pravi računalnik. Za delovanje je potreben operacijski sistem (v našem primeru je to Raspbian), z njim pa upravljamo, tako kot z drugimi računalniki. Temu primerna je tudi računrska (procesorska) moč. To nam služi kot hitre izračune namakalnih planov. Tej sicer v tej fazi niso tako zahtevni, da jih ne bi mogli izvajati na podobnih platformah, vendar bo v kasnejši fazi to postalo zelo pomembno 4.2. Arduino Uno je mikrokrmilnik, ki vsebuje nabor funkcij, ki jih programsko upravljamo in je njegovo delovanje v primerjavi z Raspberry Pi platformo zelo enostavno. Razlike bomo opazili tudi v različnih programskih jezikih in nekaj dodatnih funkcionalnostih na zunanji enoti, zaradi kontrole perifernih enot in senzorjev.

Programska oprema za zunanjo enoto je bila razvita v programskem jeziku Python [23] medtem ko je na zunanji enoti razvita v programskem jeziku C. Notranja enota podpira uporabniški vmesnik, s pomočjo katerega lahko uporabnik nastavlja in krmili

delovanje sistema. Notranja enota z uporabo protokola, opisanega v poglavju 3.2.1 sporoči akcijo, ki se mora izvesti na zunanji enoti.

Funkcije, ki jih sistem podpira so:

1. **Ping external module**

S pomočjo funkcije ping lahko preverimo, če je povezava med zunanjim in notranjo enoto vzpostavljena in stabilna. Funkcija ping pošlje sporočilo zunanji enoti, ki sporočilo prepozna in odgovori. Če prejmemo odgovor, vemo, da je povezava vzpostavljena in zanesljiva.

2. **Waterflow data**

Ta funkcija je namenjena pregledu količine iztečene vode od trenutka začetka namakanja, če to poteka. V kolikor poteka namakanje v trenutku akcije, izpišemo dobljeni podatek v konzolo.

3. **Weather data**

Funkcija sproži prebiranje senzorskih podatkov na zunanji modulu, ki jih vrne notranjemu, ter izpiše v konzoli. Pridobimo naslednje podatke: temperatura, vlažnost zraka, stopnja dežja, zračni pritisk in stopnja svetlobe. Najpomembnejši podatek je podatek o nivoju dežja, saj je od tega odvisno namakanje v vsakem trenutku delovanja sistema.

4. **Soil moisture**

Funkcija sproži akcijo na zunanji enoti, ki iz sensorja odčita vlažnost zemlje. Ko vrednost odčitamo, jo inkapsuliramo v pravilno obliko in notranji enoti odgovorimo s podatki. Podatek se nato izpiše v konzoli.

5. **Start auto watering**

S pomočjo funkcije sprožimo samodejno namakanje. Pred začetkom samodejnega namakanja uporabnik za vsako cono vnese minimalno in maksimalno vrednost, ki predstavlja območje vlažnosti za posamezno cono. Če je dejanska vlažnost pod minimalno vrednostjo, se sistem na podlagi vremenske napovedi odloči ali bo namakal. Če vlažnost zemlje preseže maksimalno vrednost in je namakanje v teku, sistem namakanje prekine. V samodejnem načinu notranja enota vsakih 30 minut preverja stanje senzorjev na zunanji enoti. V kolikor je potrebno namakanje, preveri vremensko napoved. Če je možnost in količina padavin dovolj velika, sistem ne prične z namakanjem, temveč nadaljuje s preverjanjem senzorjev na zunanji enoti ter vremenske napovedi. V kolikor se podatki na senzorjih ali vremenska napoved spremenijo, notranja enota temu primerno ukrepa.

6. **Stop auto watering**

Funkcija, ki prekliče samodejno namakanje. Pred izvedbo funkcije mora uporabnik vnesti številko cone, za katero želi preklicati samodejno namakanje.

7. **Start manual watering**

Ročno namakanje je funkcija, ki uporabniku omogoča klasično nastavitev namakanja in sicer časovni interval namakanja. Uporabnik pred zagonom vnese začetni in končni čas namakanja za cono, ki jo želi namakati. V tem primeru sistem ne upošteva senzorjev in vremenske napovedi in prepušča nadzor uporabniku. Sistem vseeno preverja podatke iz senzorja in vremenske napovedi ter v primeru, ko sistem zazna dežne kaplje in vrednost senzorja za vlago zemlje preseže predhodno vneseno maksimalno vrednost ali vremenska napoved napoveduje dež, ugotovitve sporoči uporabniku.

8. **Stop manual watering**

Funkcija predčasno preneha z namakanjem cone, ki jo uporabnik vnese ob izvedbi akcije.

9. **Enter location coordinate**

S pomočjo te funkcije sporočimo sistemu, kje se nahaja. Lokacijo nastavimo tako, da kot parameter funkciji pošljemo podatka o geografski širini in dolžini, kjer se sistem nahaja. Lokacija je pomembna, saj je od nje odvisna vremenska napoved.

10. **Get weather forecast**

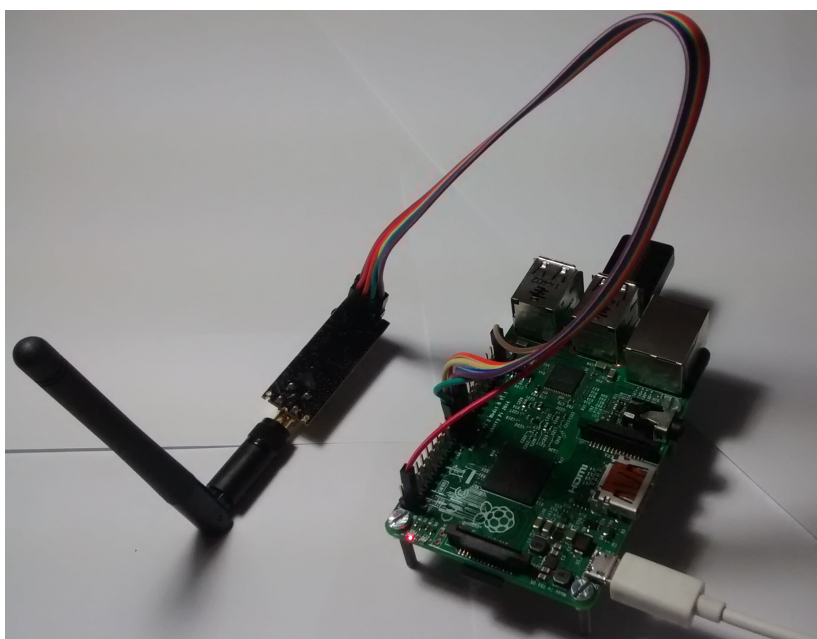
Funkcija pridobi vremensko napoved, ki jo potrebujemo za planiranje namakanja. Pridobimo le podatke za trenutno stanje vremena, za naslednjih 12 ur ter naslednje 3 dni. Od pridobljenih podatkov nas najbolj zanima možnost in količina padavin. Vremensko stanje in notranja enota se redno posodablja in hkrati pomeni napoved, ki jo uporabimo v primeru izpada internetne povezave oziroma nedostopnosti storitve, ki nam omogoča dostop do podatkov.

11. **Quit** S to funkcijo prekinemo izvajanje programa.

4 Rezultati

4.1 Končni izdelek

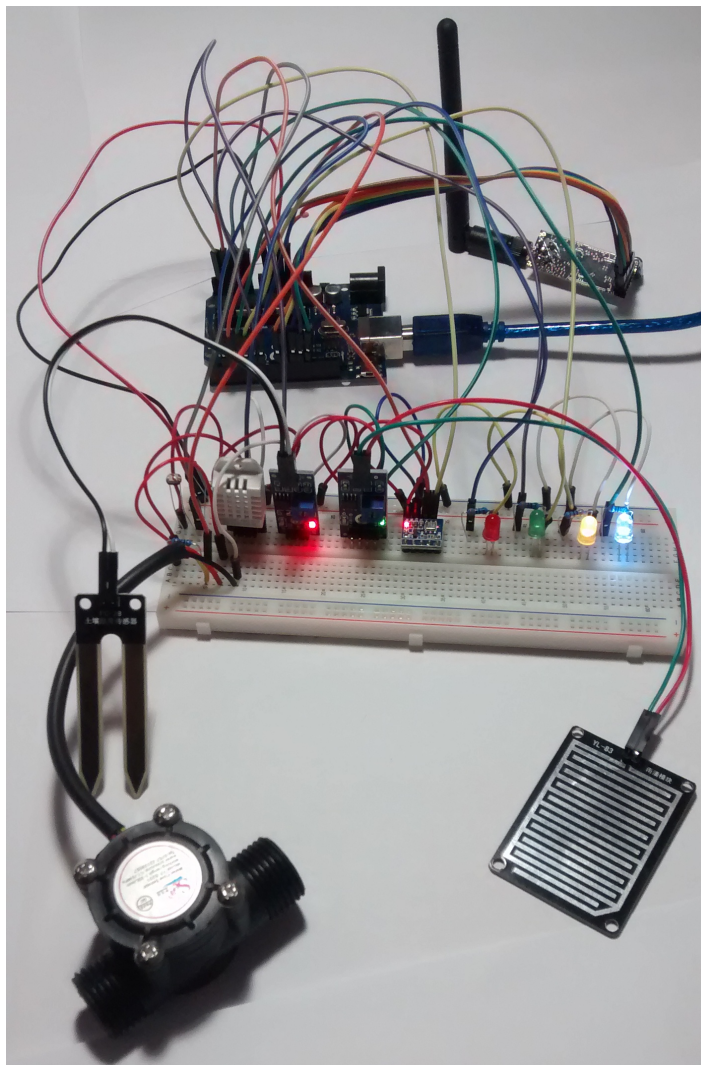
Končni izdelek te zaključne naloge je prototipni samodejni kontrolni sistem. Vse periferne naprave so s platformo Arduino, povezane z razvojnimi pripomočki, kot so didaktična (preizkusna) ploščica, žične spojke in vtični mostički. RF module smo prav tako povezali s platformami s pomočjo žičnih spojk. Poleg tega smo uporabili tudi nekaj elektronskih pasivnih elementov, kot so upori (za delilnike napetosti) in kondenzatorji (za stabilizacijo napetosti). Obe platformi napajamo preko USB povezave, ki nam omogoča 5 V izhodne napetosti.



Slika 4.1: Notranja enota krmilnega sistema sestavljena iz RF modula in Raspberry Pi 2 (B model). Za potrebe dostopa do notranje enote in pridobivanja vremenske napovedi s spleta je nanjo nameščen USB WiFi modul.

Na sliki 4.2 je vidna zunanja enota s perifernimi napravami in RF modulom. Na didaktični ploščici (na sredini slike, pod krmilnikom Arduino) so nameščeni vsi uporabljeni senzorji (od leve proti desni): fotoupor, DHT22 za merjenje vlažnosti zraka in temperature, senzor za merjenje vlažnosti zemlje z vmesnikom, senzor za merjenje

dežnih kapelj z vmesnikom, BMP180 za merjenje zračnega pritiska in senzor za pretok tekočine (spodaj levo). Elektromagnetnih ventilov nismo uporabljali v fazi razvoja, saj je brez pretoka vode, težko ugotavljati pravilno delovanje le-teh. Za simulacijo elektromagnetnih ventilov smo uporabili LED diode, v našem primeru 4, kar predstavlja 4 cone. V kolikor LED dioda žari, pomeni, da je elektromagnetni ventil odprt, v nasprotnem primeru pa zaprt. Na sliki lahko vidimo, da sta v dveh conah ventila odprta, na drugih dveh pa zaprta.



Slika 4.2: Zunanja enota krmilnega sistema s perifernimi napravami in RF modulom.

4.1.1 Uporabniški vmesnik in primeri delovanja

V trenutni fazi razvoja poteka interakcija med uporabnikom in sistemom preko oddaljenega dostopa. Uporabnik se s pomočjo protokola SSH poveže na notranji modul Raspberry Pi 2. Za to je potrebna namestitev programa, ki deluje kot SSH odjema-

lec. SSH protokol omogoča komunikacijo med poljubnimi računalniškimi sistemi, ki protokol podpirajo. V našem primeru je bil uporabljen program MobaXterm [5]. Po vzpostavitvi seje, se z osnovnimi Unix [24] ukazi pomaknemo v direktorij, kjer se nahaja izvorna koda in zaženemo program. Takoj ob zagonu program izpiše status RF modula, ki je povezan na notranjo enoto. Program se sedaj že izvaja. V kolikor smo v program že vnesli potrebne podatke (lokacijske koordinate) in omogočili ročno ali samodejno namakanje, se program že izvaja. Če želimo izvesti akcijo, pritisnemo tipko "Enter" (ang. Return), nakar se nam prikažejo vse razpoložljive akcije.

```

pi@raspberrypi ~/zalivalni_sistem $ sudo python csvraspberry.py
/home/pi/zalivalni_sistem/nrf24.py:373: RuntimeWarning: This channel is already in use, continuing anyway. Use GPIO.setwarning
s(False) to disable warnings.
  GPIO.setup(self.ce_pin, GPIO.OUT)
STATUS   = 0x0e RX_DR=0 TX_DS=0 MAX_RT=0 RX_P_NO=7 TX_FULL=0
RX_ADDR_P0-1 = 0xf0f0f0f0e1 0xf0f0f0f0d2
RX_ADDR_P2-5 = 0xc3 0xc4 0xc5 0xc6
TX_ADDR    = 0xf0f0f0e1
RX_PW_P0-6 = 0x20 0x20 0x00 0x00 0x00 0x00
EN_AA     = 0x3f
EN_RXADDR = 0x03
RF_CH     = 0x4c
RF_SETUP  = 0x0f
CONFIG    = 0x0f
DYNPD/FEATUR = 0x03 0x06
Data Rate = 2Mbps
Model     = nRF24l01+
CRC Length = 16 bits
PA Power  = PA_HIGH

*****Automated system for controlling irrigation systems*****

Press enter/return to start action.
Available actions:
> 1 - ping external module
> 2 - waterflow data
> 3 - weather data
> 4 - soil moisture
> 5 - start auto watering
> 6 - stop auto watering
> 7 - start manual watering
> 8 - stop manual watering
> 9 - enter location coordinate
> 10 - get weather forecast
> 0 - QUIT
Enter action: █

```

Slika 4.3: Samodejni krmilni sistem ob zagonu. Takoj po zagonu se nam izpiše status RF modula, po pritisku tipke "Enter" pa seznam vsej akcij.

Če želimo preveriti kakšno, je trenutno stanje senzorjev na zunanji enoti, vnesemo primerno številko akcije in izpišejo se nam naslednji podatki:

```

Enter action: 3
Photo cell: 2
Humidity: 46 %
Rain level: 0
Pressure: 1005 mb
Temperature: 23 *C
█

```

Slika 4.4: Izpis podatkov iz senzorjev na zunanji enoti.

```
Enter action: 4
Zone 1: 0
Zone 2: 45
Zone 3: 56
Zone 4: 12
```

Slika 4.5: Izpis vlažnosti zemlje po conah.

Ročno namakanje lahko omogočimo s številko akcije 7. Ob vnosu izberemo cono, kjer želimo začeti z namakanjem. Nato vnesemo čas začetka in konca namakanja za izbrano cono.

```
Enter action: 7
Start man watering in zone 1? (Y/N): y
Insert start time for watering (hh:mm DD:MM:YYYY): 12:30 03:10:2015
Insert end time for watering (hh:mm DD:MM:YYYY): 16:00 03:10:2015
Stop auto watering in zone 2? (Y/N): n
Stop auto watering in zone 3? (Y/N): y
Insert start time for watering (hh:mm DD:MM:YYYY): 17:00 05:10:2015
Insert end time for watering (hh:mm DD:MM:YYYY): 21:00 05:10:2015
Stop auto watering in zone 4? (Y/N): n
Man watering activated in:
                                zone 1 : 12:30 03:10:2015 - 16:00 03:10:2015
                                zone 3 : 17:00 05:10:2015 - 21:00 05:10:2015
```

Slika 4.6: Aktivacija ročnega namakanja po conah z vnosom začetka in konca namakanja.

Ob akciji preklič ročno namakanje, se nam prikažejo tiste cone, kjer so aktivirana ročna namakanja. Sistem nas za vsako tako namakanje vpraša, ali ga želimo preklicati.

```
Enter action: 8
Stop man watering in zone 1? (Y/N): y
Stop man watering in zone 3? (Y/N): y
Man watering stopped in zone: 1, 3!
```

Slika 4.7: Preklic ročnega namakanja po posameznih conah.

Po vnosu koordinat, ki določajo našo lokacijo, lahko preverimo tudi vremensko napoved. Izpisujemo trenutno stanje, za naslednjih 12 ur in naslednje 3 dni. Izpisani podatki so upoštevani tudi v primeru ko je omogočeno samodejno namakanje.

```
Enter action: 10
Powered by Forecast
::: Reloading cache...
### Current weather ###
- 2015-09-24 04:23:55
- Ozone: 285 DU
- Cloud cover: 57%
- Conditions: Rain
- Precip intensity: 2 mm/m2
- Precip probability: 63%
- Apparent temp: 10.4 *C
- Humidity: 64.0%
- Pressure: 1012 mb
- Wind: 17 km/h ENE
- Temperature: 10.4 *C
- Dew Point: 4.0 *C
- High: 18.3 *C
- Low: 10.4 *C
### Hourly weather for next 12 hours ###
04:00 Rain 10.4 *C 2 mm/m2 65%
05:00 Rain 10.5 *C 1 mm/m2 62%
06:00 Rain 10.6 *C 1 mm/m2 59%
07:00 Light Rain 10.9 *C 1 mm/m2 56%
08:00 Light Rain 11.3 *C 1 mm/m2 55%
09:00 Light Rain 11.9 *C 0 mm/m2 53%
10:00 Light Rain 12.7 *C 0 mm/m2 50%
11:00 Light Rain 13.7 *C 0 mm/m2 49%
12:00 Light Rain 14.9 *C 0 mm/m2 46%
13:00 Light Rain 16.2 *C 0 mm/m2 44%
14:00 Light Rain 17.2 *C 0 mm/m2 42%
15:00 Drizzle 17.9 *C 0 mm/m2 39%
16:00 Drizzle 18.3 *C 0 mm/m2 24%
17:00 Mostly Cloudy 18.0 *C 0 mm/m2 14%
18:00 Mostly Cloudy 17.1 *C 0 mm/m2 8%
19:00 Mostly Cloudy 15.9 *C 0 mm/m2 5%
20:00 Mostly Cloudy 14.9 *C 0 mm/m2 3%
21:00 Mostly Cloudy 14.2 *C 0 mm/m2 1%
22:00 Mostly Cloudy 13.5 *C 0 mm/m2 0%
23:00 Mostly Cloudy 12.8 *C 0 mm/m2 0%
00:00 Mostly Cloudy 12.2 *C 0 mm/m2 3%
01:00 Mostly Cloudy 11.6 *C 0 mm/m2 9%
02:00 Drizzle 11.1 *C 0 mm/m2 23%
03:00 Light Rain 10.8 *C 0 mm/m2 42%
### Daily weather for next 3 days ###
2015-09-25 Light rain until evening. 10.8 *C 21.1 *C 0 mm/m2 0%
2015-09-26 Mostly cloudy until evening. 11.7 *C 22.0 *C 0 mm/m2 0%
2015-09-27 Partly cloudy throughout the day. 12.7 *C 22.5 *C 0 mm/m2 0%
```

Slika 4.8: Izpis podatkov za trenutno stanje vremena, napoved za naslednjih 12 ur ter naslednje 3 dni.

Najpomembnejša akcija je zagotovo začetek samodejnega namakanja. Pred začetkom izberemo cono/e, katero/e želimo namakati. Za izbrano cono vnesemo tudi minimalno in maksimalno vlažnost zemlje. Na podlagi teh podatkov se bo sistem odločal kdaj in koliko namakati. Na sliki 4.9 je prikazan postopek aktivacije samodejnega namakanja za coni 2 in 3.

V primeru preklica samodejnega namakanja je postopek enak kot pri preklicu

```
Enter action: 5
Zone 1? (Y/N): n
Zone 2? (Y/N): y
Min moisture for zone 2: 35
Max moisture for zone 2: 40
Zone 3? (Y/N): y
Min moisture for zone 3: 38
Max moisture for zone 3: 52
Zone 4? (Y/N): n
Auto watering started in zone: 2, 3
```

Slika 4.9: Aktivacija samodejnega namakanja, z izbiro zelenih con in minimalnih ter maksimalnih vrednosti za vlažnost zemlje.

ročnega namakanja. Uporabniku se prikažejo cone, ki so trenutno aktivirane, uporabnik nato za vsako cono določi, ali bo namakanje preklical.

4.2 Nadaljnje faze razvoja in možnosti izboljšav

Samodejni sistem trenutno deluje tako, da v primeru prenizke vlažnosti zemlje, preverja vremensko napoved in senzor dežja. Torej gre za sprotno preverjanje glede na potrebe. Delovanje sistema želimo razširiti tako, da bi sistem napovedoval tudi nekaj naslednjih namakanj, za vsako tako namakanje pa tudi oceno količine vode potrebne za namakanje. Želimo si tudi, da bi sistem znal oceniti točnost napovedi.

Da bi to dosegli, moramo sistem nadgraditi tako strojno, kot tudi programsko. Naslednja namakanja lahko napovemo tako, da z različnimi metodami ugotovimo, optimalne intervale namakanja, glede na trenutne razmere, vremensko napoved za naslednje dni, hitrost izparevanja vode iz zemlje in vrste rastlin. Za prototipni sistem smo uporabili le en senzor za merjenje vlažnosti zemlje, kar pa se v praksi ne izkaže za dobro rešitev. V prihodnosti bodo senzorji za vlažnost zemlje modularni, torej ločena enota, ki bo imela svoje baterijsko napajanje. Polnila se bo s pomočjo sončnih celic, komunikacija pa bo potekala, preko RF modula 3.2.2. Ker nam uporabljeni RF moduli omogočajo vzpostavitev zančnega omrežja, ima lahko vsaka cona svojo enoto, ki je neodvisna od druge. Vse enote lahko pošiljajo stanje senzorjev zunanjemu modulu, ta pa jih posreduje notranjemu (glavnemu) modulu.

Za izračun izparevanja vlage potrebujemo naslednje vrednosti: sončno radiacijo, temperaturo, hitrost vetra in vlažnost zraka. V poglavju 3.2.2 smo opisali senzor, s katerim merimo vlažnost zraka in temperaturo. Periferijo je torej potrebno razširiti še s senzorjem za sončno radiacijo in senzorjem za veter. Notranji modul bi programsko

nadgradili s podatkovno bazo [18] v katero bi shranjevali vse podatke, pridobljene s senzorji in vremenskimi napovedmi. Z algoritmi prilagojenimi za naše potrebe, bi tako lahko izvajali podatkovno rudarjenje nad dobljenimi podatki. S tem bi lahko naš sistem učili in sčasoma na podlagi preteklih odločitev izboljšali napovedi.

Kljub vsemu smo v veliki meri odvisni od točnosti vremenske napovedi. To bi poskusili rešiti tako, da bi podatke pridobivali iz več različnih virov. Tako bi lahko s pomočjo algoritmov in podatkovnega rudarjenja ocenili točnost posameznega vira. Poleg tega bi poskusili napake uravnovežiti s povprečjem podskupine izbranih virov.

Ključnega pomena v nadaljnji fazi je izdelava aplikacije za oddaljen dostop preko mobilne naprave ali spletnega brskalnika. Tako uporabniku omogočimo, da lahko kadarkoli in kjerkoli dostopa do svojega krmilnega sistema, preverja njegovo stanje in ga upravlja.

5 Dostopnost

V tem poglavju bomo predstavili uporabljene knjižnice in njihove licence. Opisali bomo tudi, kako lahko dostopamo do izvorne kode samodejnega krmilnega sistema

5.1 Licenca izvorne kode

Licenca odprtokodnih programov je zelo pomembna, saj kljub temu da so uporabljene knjižnice brezplačne, obstajajo določena pravila. Ta določajo predvsem, za katere namene lahko uporabljamo kodo programov in njihovo distribucijo.

Med najpogostejše licence spadajo BSD (Berkeley Software Distribution), MIT (Massachusetts Institute of Technology) Licence in GPL (General Public License). BSD in MIT Licenci spadata med najmanj zahtevne, saj dovoljujejo dodajanje izvorne kode, ki ne sodi pod isto licenco, hkrati pa dovoljuje spreminjanja licence izvorne kode. GPL licenca ne dovoljuje spreminjanja licence izvorne kode in dodajanja izvorne kode pod drugo licenco.

V našem projektu je bilo uporabljenih več knjižnic, tako za notranji kot tudi za zunanji modul. Pri predstavitvi se bomo osredotočili samo na tiste knjižnice, ki niso že vgrajene oziroma niso del neke platforme.

Na zunanjem modulu smo potrebovali knjižnice za upravljanje z naslednjimi napravami:

- **RF modul (RF oddajnik in sprejemnik) - RF24**
Knjižnica vsebuje metode za enostavno upravljanje RF modulov s pomočjo Arduina. Zaščitena je z GPL licenco.
- **Senzor DHT22 za merjenje temperature in vlažnosti zraka - DHT22**
Knjižnica za Arduino platforme vsebuje metode za uporabo temperaturnega senzorja in senzorja za vlažnost zraka. Knjižnica je zaščitena z MIT Licence.
- **Senzor BMP180 uporabljen za merjenje zračnega pritiska - BMP180**
Knjižnica za Arduino platforme vsebuje metode za uporabo temperaturnega senzorja, senzorja za merjenje višine in nadmorskega pritiska. V našem primeru smo

uporabili samo del knjižnice za temperaturni senzor. Knjižnica je izdana pod GPL licenco.

Na notranjem modulu smo uporabili naslednji knjižnici:

- **RF modul (RF oddajnik in sprejemnik) - RF24**

Knjižnica za upravljanje RF modulov s pomočjo Raspberry Pi platforme. Zaščiten je z GPL licenco.

- **Forecast.io**

Knjižnica za upravljanje s Forecast.io API [2]. S pomočjo knjižnice pridobivamo podatke o vremenski napovedi. Zaščiten je z BSD licenco.

5.2 Dostopnost programa izvirne kode

Izvorna koda programa je javno dostopna na spletnem portalu Github preko spletne povezave <https://github.com/ttomaz/AutomatedIrrigationSystem>.

Spletni portal Github smo izbrali, ker je odprt za javnost in brezplačen za odprtokodne projekte. Zasnovan je na osnovi Gita (sitem za razvijanje in verzioniranje projektov) [20], kar pomeni, da lahko vsakdo izdela svojo verzijo projekta, pri čemer ostane izvorna koda nedotaknjena. Vsa navodila, za izdelavo nove verzije projekta in upravljanje z njim so dostopna na spletni povezavi <http://git-scm.com/docs/gittutorial>.

6 Zaključek

Kljub temu, da trg postaja nasičen s primerljivimi napravami kot je predstavljen prototip, je poleg dodatnih funkcionalnosti, doprinos tudi v sami arhitekturi sistema. Trg se na tem področju še razvija in zaradi porodnih krčev ni dozorel za hiter razvoj. Trenutna ponudba je zelo kaotična in polna med seboj nekompatibilnih rešitev. Za uspešen razvoj trga je potrebna konsolidacija in standardizacija protokolov, ki bo omogočila priključitev poljubne periferne enote na poljuben krmilni sistem. V prihodnje pričakujemo, da se bo ponudba in povpraševanje tovrstnih sistemov povečala tako v obsegu kot v funkcionalnosti. Prav tako je v prihodnosti veliko prostora za razvoj pametnih sistemov, ki bodo bili bolj samostojni in bolje optimizirali celoten proces.

7 Literatura in viri

- [1] Arduino spi knjižnica. <https://www.arduino.cc/en/Reference/SPI>. (*Citirano na strani 17.*)
- [2] Google maps elevation api. <https://developers.google.com/maps/documentation/elevation/intro>. (*Citirano na straneh 20 in 33.*)
- [3] Kalkulator za izračun potrebne vode za izdelavo različnih izdelkov. <http://environment.nationalgeographic.com/environment/freshwater/embedded-water/>. (*Citirano na straneh VI in 1.*)
- [4] Merilec pretoka tekočine. https://www.olimex.com/Products/Components/Sensors/SNS-FLOW201/resources/Water_flow_sensor_datasheet.pdf. (*Citirano na strani 22.*)
- [5] MobaXterm - ssh client. <http://mobaxterm.mobatek.net/>. (*Citirano na strani 27.*)
- [6] Namakalni sistem. <http://www.access-irrigation.co.uk/design-services/irrigation-agriculture/field-irrigation>. (*Citirano na strani 3.*)
- [7] Navodila za konfiguracijo spi vmesnika na platformi raspberry pi. <https://www.raspberrypi.org/documentation/configuration/raspi-config.md>. (*Citirano na strani 17.*)
- [8] Nrf2401. www.nordicsemi.com/eng/content/download/2726/34069/file/nRF24L01P_Product_Specification_1_0.pdf. (*Citirano na straneh 12 in 14.*)
- [9] Senzor dežnih kapelj yl-83. <http://www.dx.com/p/produino-yl-83-single-pcb-raindrop-sensor-module-black-silver-292123#.VgbyXpeJ3AY>. (*Citirano na strani 19.*)
- [10] Senzor vlažnosti zemlje fc-28. <http://www.dx.com/p/fc-28-b-soil-humidity-detection-sensor-module-184821#.VgbyqZeJ3AY>. (*Citirano na strani 21.*)

-
- [11] Tehnični podatki bmp180. https://www.bosch-sensortec.com/en/homepage/products_3/environmental_sensors_1/bmp180_1/bmp180. (*Citirano na strani 20.*)
- [12] Tehnični podatki dht22. <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>. (*Citirano na strani 20.*)
- [13] Tehnični podatki fotoupor. http://www.ic-elect.si/pub/files/product_files/199002815500.pdf. (*Citirano na strani 21.*)
- [14] S. F. Barrett and D. J. Pack. Atmel AVR Microcontroller Primer: Programming and Interfacing, Second Edition, 2012. (*Citirano na strani 10.*)
- [15] Brandon. Arduino Uno vs Raspberry Pi vs BeagleBone Black, 2013. (*Citirano na strani 10.*)
- [16] D. C. Chang and T. H. Shin. Digital GFSK carrier synchronization. In *IEEE Asia-Pacific Conference on Circuits and Systems, Proceedings, APCCAS*, pages 1523–1526, 2006. (*Citirano na strani 13.*)
- [17] M. R. Choudhury S., Singh G.K. Design and Verification Serial Peripheral Interface (SPI) Protocol for Low Power Applications. 2014. (*Citirano na strani 17.*)
- [18] R. Elmasri and S. B. Navathe. *Fundamentals of Database Systems*, volume 28. 2003. (*Citirano na strani 31.*)
- [19] D. J. Hand. Principles of data mining. In *Drug Safety*, volume 30, pages 621–622, 2007. (*Citirano na strani 6.*)
- [20] M. McCullough. Getting Started with Git. *RefCardz*, pages 1–6, 2010. (*Citirano na strani 33.*)
- [21] D. K. Misra. *Radio-Frequency and Microwave Communication Circuits*, volume 8. 2001. (*Citirano na strani 9.*)
- [22] G. Mitchell. The Raspberry Pi single-board computer will revolutionise computer science teaching, 2012. (*Citirano na strani 11.*)
- [23] J. Payne. *Beginning Python: Using Python 2.6 and Python 3.1*. 2010. (*Citirano na strani 22.*)
- [24] K. Shell. *Unix in a Nutshell*, volume 280. 2013. (*Citirano na strani 27.*)
- [25] D. Sweeney. An introduction to bluetooth a standard for short range wireless networking. 2013. (*Citirano na strani 13.*)