

Automatizirani sustav vodoopskrbe i navodnjavanja

Crljenica, Karlo

Undergraduate thesis / Završni rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Istrian University of applied sciences / Istarsko veleučilište - Università Istriana di scienze applicate**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:212:917365>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-09-27**



image not found or type unknown

Repository / Repozitorij:

[Digital repository of Istrian University of applied sciences](#)



image not found or type unknown



Istarsko veleučilište
Università Istriana
di scienze applicate

PULA

PREDDIPLOMSKI STRUČNI STUDIJ POLITEHNIKE

KARLO CRLJENICA

**AUTOMATIZIRANI SUSTAV
VODOOPSKRBE I NAVODNJAVANJA**

ZAVRŠNI RAD

PULA, kolovoz 2019.



Istarsko veleučilište
Università Istriana
di scienze applicate

PULA

PREDDIPLOMSKI STRUČNI STUDIJ POLITEHNIKE

AUTOMATIZIRANI SUSTAV VODOOPSKRBE I NAVODNJAVANJA

ZAVRŠNI RAD

Kolegij: Osnove automatskog upravljanja

Nositelj: mr.sc. Eduard Lorencin

Student: Karlo Crljenica

JMBAG:

PULA, kolovoz 2019.

ZAHVALA

Za početak, zahvaljujem se svome mentoru mr.sc. Eduardu Lorencinu koji me je uz pomoć svog znanja, vještina i kompetencija savjetovao i navodio tijekom izrade ovog završnog rada.

Zahvaljujem se i svojim prijateljicama, prijateljima, kolegama i svim dobrim ljudima koji su sve trenutke studiranja bili uz mene te mi pružali potporu.

Na kraju, najveću zahvalu prilaže svojoj obitelji koja me podržavala, kako u dobrim, tako i u teškim trenucima školovanja i bez koje ni ovog završnog djela mog visokog školovanja ne bi bilo.

IZJAVA

Kojom izjavljujem da sam diplomski rad s naslovom SUSTAV AUTOMATSKOG UPRAVLJANJA VODOOPSKRBE I NAVODNJAVANJA izradio samostalno pod voditeljstvom mr.sc. Eduarda Lorencin, pred. U radu sam primijenio znanje i vještine stečene tijekom studiranja na Politehnici Pula – Visoko tehničko-poslovnoj školi te koristio literaturu koja je navedena na kraju završnog rada. Tuđe spoznaje, stavove, zaključke, teorije i zakonitosti koje sam izravno ili parafrazirajući naveo u završnom radu na uobičajen, standardan način citirao sam i povezao s fusnotama s korištenim jedinicama literature.

Rad je pisan u duhu hrvatskog jezika.

Student: Karlo Crljenica

SAŽETAK

U ovom radu obrađena je tema korištenja podzemnih voda u svrhu vodoopskrbe gledano sa tehničkog aspekta i opisani svi dijelovi sustava vodoopskrbe podzemnom vodom. Navedene su prednosti i mane ovakvog sustava te parametri koji moraju biti zadovoljeni za realizaciju istoga. Uz rad, napravljen je i funkcionalni model automatiziranog sustava vodoopskrbe kućanstva pa su obrađeni i način izrade te funkcioniranja modela. Dalje u radu govori se o načinu izrade mobilne aplikacije kojom se upravlja modelom te o korištenju iste.

SUMMARY

This work discusses the topic of using groundwater for the purpose of water supply from the technical aspect and to describe all the parts of the underground water supply system. The advantages and disadvantages of such a system and the parameters that must be met for the realization of the same are discussed. Along with the work, a functional model of the automated water supply system has been developed, and the way of designing and functioning of the model is elaborated. The work talks about how to build a mobile application that manages the model and how to use that same application.

SADRŽAJ

SAŽETAK.....	1
SUMMARY	1
1. UVOD	3
1.1. Cilj rada	3
1.2. Metode rada	3
1.3. Struktura rada	3
2. PODZEMNE VODE	4
2.1. Sustav opskrbe vodom iz podzemnih izvora	5
2.2. Dijelovi sustava za vodoopskrbu iz podzemnih izvora	7
2.3. Financijski aspekt ugradnje automatiziranog sustava vodoopskrbe iz podzemnih izvora	8
3. MODEL SUSTAVA NAVODNJAVANJA I VODOOPSKRBE	9
3.1. Potrebne komponente	9
3.1.1. Arduino UNO mikro kontroler.....	9
3.1.2. Bluetooth HC 06 modul	10
3.1.3. Android uređaj.....	11
3.1.4. Elektronički elementi korišteni pri izradi modela.....	12
3.2. Logika automatskog upravljanja sustavom	15
3.2.1 Sustav automatskog upravljanja s povratnom vezom	18
3.3. Izrada programa za upravljanje Arduino uređajem	20
3.4. Izrada aplikacije za upravljanje pomoću Android uređaja	23
3.4.1. Izrada dizajna aplikacije.....	24
4. ZAKLJUČAK	35
LITERATURA.....	36
POPIS SLIKA	38
POPIS PRILOGA.....	40

1. UVOD

Podzemne vode su vode tekućice ili stajačice koje se nalaze unutar šupljina zemljine kore i kreću se isključivo pomoću gravitacije. Podzemne vode moguće je koristiti u svrhu navodnjavanja i vodoopskrbe kućanstava pomoću dubinskih vodenih pumpi za ispumpavanje na površinu. Posebno je povoljna upotreba podzemnih voda na područjima gdje nije dostupan izvor vode iz vodovodnog sustava. Automatizacijom sustava vodoopskrbe iz podzemnih voda moguće je postići udobnost i upotrebljivost jednake kao kod korištenja vode iz vodovoda.

1.1. Cilj rada

Cilj rada je objasniti funkcioniranje sustava vodoopskrbe iz podzemnih voda te temeljito objasniti i prikazati sve dijelove jednog takvog sustava sa naglaskom na centar automatizacije, tj. upravljačku jedinicu koja je u ovom radu realizirana uz pomoć Arduino uređaja.

1.2. Metode rada

U radu su korištene sljedeće metode:

- Pri definiranju elemenata uvodnog dijela te opisivanju dijelova sustava i izrade modela, korištena je opisna metoda.
- Kod izrade tablica i logičkih shema korištena je grafička metoda, a pri izračunu troškova materijala korištena je matematička metoda.

1.3. Struktura rada

Ovaj rad podijeljen je u četiri glavna poglavlja.

U uvodnom dijelu rada sadržani su kratak opis, cilj rada te metode korištene u radu.

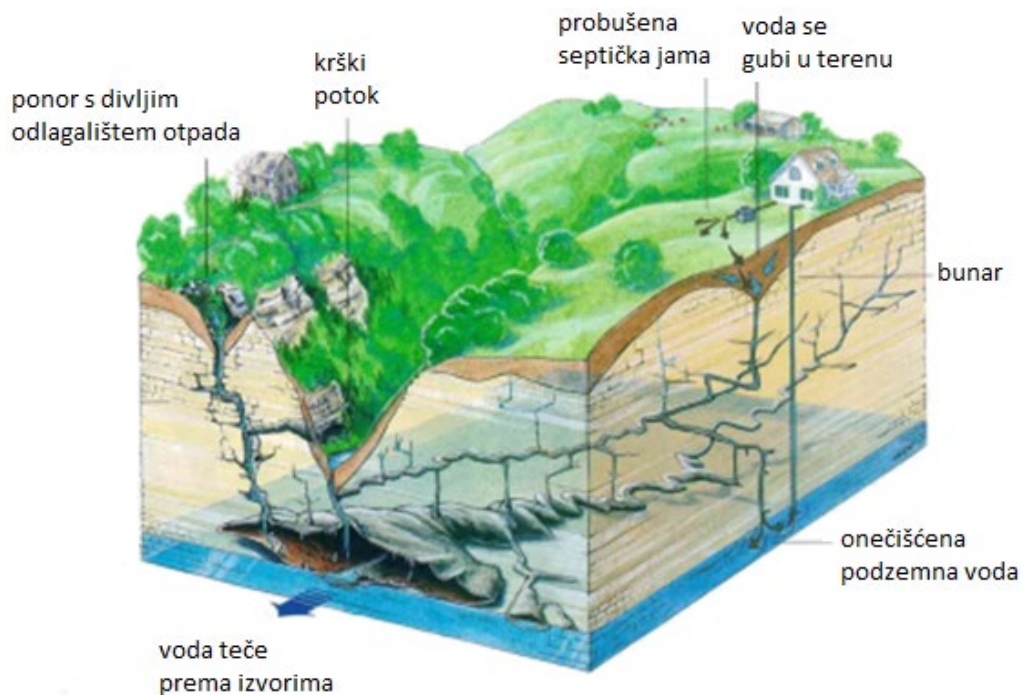
Drugo poglavlje sadrži opis sustava vodoopskrbe iz podzemnih voda u realnom okruženju. Ovdje su opisani svi dijelovi sustava i njihove karakteristike. Kod opisa dijelova sustava u obzir je uzeta situacija područja primjene na kojem ne postoji niti dovod pitke vode niti sustav opskrbe električnom energijom.

Treće poglavlje opisuje izradu modela sustava i u njemu se detaljno razrađuje programiranje Arduino uređaja te izradu mobilne aplikacije za upravljane sustavom.

U četvrtom poglavlju je zaključak, nakon kojega slijede popisi slika, tablica i priloga.

2. PODZEMNE VODE

Podzemne vode su vode koje se nalaze ispod zemljine površine u šupljinama. Znanost koja se bavi istraživanjem područja podzemnih voda je hidrogeologija. Podzemna voda kreće se isključivo putem gravitacije.¹ Podzemne vode na teritoriju Republike Hrvatske iznimno su važan, ali nedovoljno iskorišten vodni potencijal. U usporedbi s ostalim europskim zemljama Hrvatska je jedna od najbogatijih prema količini podzemnih voda i ukupnih obnovljivih vodnih resursa. Unatoč tomu, sustavi za navodnjavanje izgrađeni su na približno 0,5% površine ukupnoga poljoprivrednog zemljišta. Kako je u Hrvatskoj tijekom posljednjeg desetljeća sve češća pojava sušnih razdoblja, nedostatak navodnjavanja površina postaje jedno od ključnih pitanja hrvatske poljoprivrede. S druge strane, primjena nedovoljno učinkovitih tehnika pri navodnjavanju, nedostatak ili loše održavanje sustava navodnjavanja uz neadekvatno uključivanje struktura vlasti, može uzrokovati onečišćenje podzemnih voda.²



Slika 1.- Presjek zemljine kore sa podzemnim vodama te primjerima onečišćenja

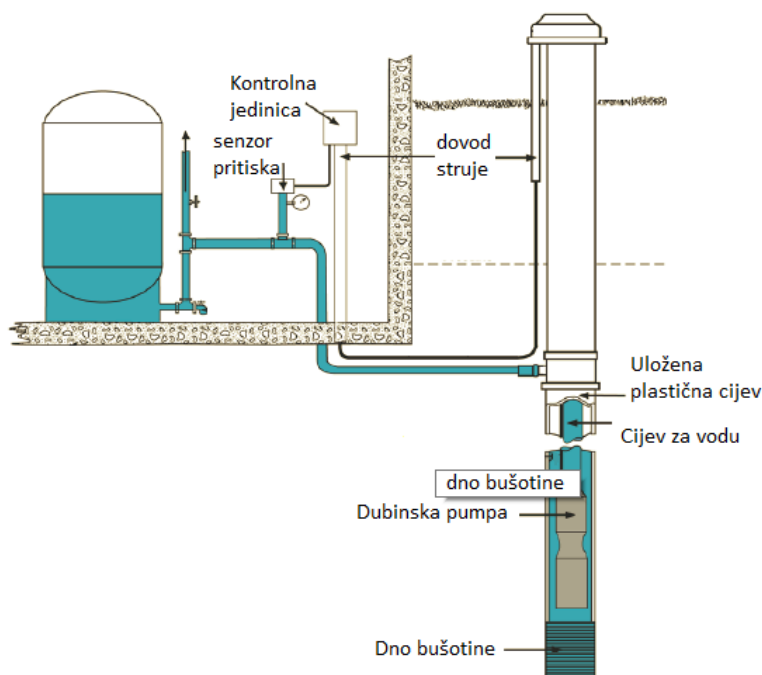
Izvor: <http://cistopodzemlje.info/node/10086> (29.12.2017.)

¹ https://hr.wikipedia.org/wiki/Podzemne_vode (12.12.2018.)

²Pejdo, A.;Šiljković, Ž.:MOGUĆNOSTI NAVODNJAVANJA PODZEMNIM VODAMA U HRVATSKOJ, pregledni članak, odjel za geografiju, Sveučilište u Zadru, listopad 2007., Zadar

2.1. Sustav opskrbe vodom iz podzemnih izvora

Pošto su podzemne vode sve potrebniji izvor vodoopskrbe, naročito u poljoprivredi, sve se više ljudi odlučuje na izgradnju infrastrukture za korištenje istih. Kako bi se voda dovela od dubine do slavine, potrebo je ispuniti više uvjeta. Prvi korak je pronalaženje podzemne vode, koje zahtijeva znanje i vještine kako bi se ispitalo tlo te utvrdilo ima li mogućnosti bušenja bušotine i na kojem mjestu točno. Prvi je korak jako bitan, jer se krivom procjenom mjesta bušenja mogu prouzrokovati veliki troškovi, iz razloga što je bušenje bušotine jedan od skupljih pothvata ovisno o dubini na kojoj je podzemna voda nađena. Nakon utvrđivanja mjesta bušenja, buši se bušotina specijalnim strojevima za tu namjenu i nerijetko se u izbušenu bušotinu ulaže plastična cijev kako bi se smanjio rizik urušavanja materijala plašta bušotine u bušotinu. Voda se iz bušotine na zemljinu površinu dovodi pomoću dubinskih pumpi dizajniranih posebno za tu namjenu. Kada je voda dovedena na zemljinu površinu, kako bi bila upotrebljiva, potrebno je postići pritisak iste da bi se mogla kretati cijevima sustava.

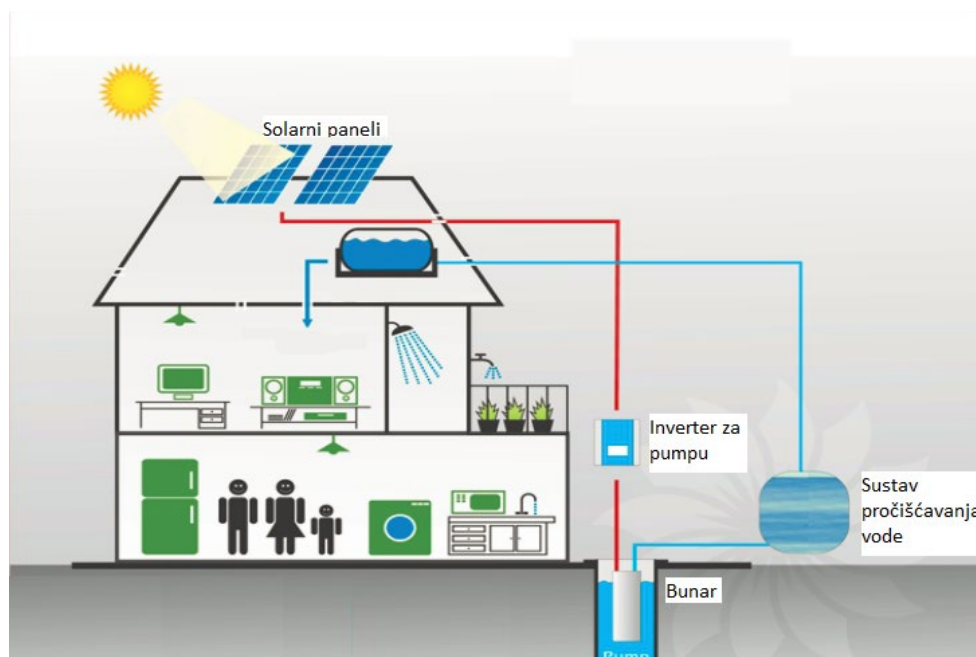


Slika 2. - Shema jednostavnog sustava vodoopskrbe iz bušotine

Izvor: <https://dluxeinternacional.com/automatic-water-supply-of-a-private-house/well-construction#image-1> (19.05.2019.)

Pritisak se postiže na dva načina, prvi je način upotreba gravitacije, tj. izgradnja vodotornja, dok je drugi način stvaranje pritiska pomoću hidrofornih pumpi i spremnika. Prosječna temperatura podzemnih voda kreće se oko 4°C, a za navodnjavanje u poljoprivredi to je često preniska temperatura pa se nerijetko voda grije. Podizanje temperature vode najčešće se postiže korištenjem solarne energije, tj. puštanjem vode u spremniku ili vodotornju neko vrijeme prije početka navodnjavanja. Postoje razni sustavi za automatizaciju vodoopskrbe kućanstava, npr. na nekim mjestima gdje je podzemna voda dovoljno plitka, dubinska pumpa se koristi i kao tlačna pumpa, odnosno njome se direktno postiže pritisak u spremniku (Slika 2).

Postoje razne ideje za realizaciju automatskog sustava vodoopskrbe, a naročito kada se radi o kućanstvima koja nemaju velike zahtjeve za količinom vode pa je potrebna manja količina energije za opskrbu. Jedan takav sustav vidi se i na Slici 3 gdje je prikazan sustav vodoopskrbe iz bušotine uz korištenje solarne energije za pogon pumpe. Ovakav sustav je ugradiv kada se voda nalazi na maloj dubini pa je potrebna i manja snaga za istu ispumpavati na površinu.



Slika 3.- Integrirani sustav pametne kuće

Izvor: http://www.hyperion.rs/images/fancybox/Hyperion_Concept.jpg (29.08.2017.)

2.2. Dijelovi sustava za vodoopskrbu iz podzemnih izvora

- POTOPNA PUMPA

Jedan od najbitnijih dijelova sustava za vodoopskrbu iz podzemnih izvora jest potopna dubinska pumpa. Pumpe za ispušavanje vode iz bušotina su posebno dizajnirane za tu namjenu. Napravljene su tako da su uže i duže kako bi mogle biti spuštene u izvor vode kroz cijev bušotine. Elektromotori pumpi imaju često snage i nekoliko kilovata.

- SPREMNIK VODE

Kada bi se potopna pumpa koristila kao stalni izvor vode, za svaku, i najmanju količinu vode koja bi bila potrebna, pumpa bi se morala paliti. Takva situacija nije povoljna jer bi se stalnim paljenjem i gašenjem skratio vijek trajanja pumpe, a i sustav bi bilo teško izraditi da radi ujednačeno. Iz ovih razloga u većinu sustava ugrađuje se spremnik vode kao stalni izvor iz kojeg se napajaju sljedeći dijelovi sustava. Spremnik se često montira na visoki položaj kako bi se postigao efekt vodotornja, a u sustavima gdje je spremnik montiran nisko, iz njega ide dovod u hidroforu pumpu te zatim i u spremnik pod tlakom.

- HIDROFORNA PUMPA

Kod sustava gdje tlak nije postignut korištenjem sile gravitacije, a postoji potreba za vodom pod tlakom, ugrađuju se hidrofori sustavi koji tlače vodu na određeni tlak kako bi se koristila u sustavu. Hidrofor je uređaj koji omogućuje dopremu vode na više visine iz cisterne, bunara ili gradskog vodovoda. Svi hidrofori rade tlakom od najmanje 8 atmosfera. Tim tlakom se vodu može dopremiti na visine i do 80 metara. Hidrofor je najbolje smjestiti na najnižu razinu ili podrumsku prostoriju.³ Hidrofor se sastoji od pumpe za vodu te spremnika pod tlakom, tj. Hidrauličnog akumulatora. Pumpa služi kako bi se voda upumpavala u spremnik pod tlakom od kuda se isporučuje u daljnje sustave. Pošto je voda slabo stlačiva, a cilj je da imamo konstantni pritisak u mreži više vremena, spremnik u koji se upumpava voda se prethodno puni sa zrakom na određeni tlak iz razloga što je zrak puno više stlačiv.

³ <https://hr.wikipedia.org/wiki/Hidrofor> (12.12.2018.)

- UPRAVLJAČKA JEDINICA

Za učinkovitu automatizaciju sustava vodoopskrbe kućanstva iz podzemnih izvora potrebno je ugraditi upravljačku jedinicu koja će upravljati ulazima i izlazima. Kada se radi o jednostavnijim sustavima i kada vlasnik nema velike zahtjeve u pogledu automatizacije, ona se implementira u obliku nekoliko releja koji su spojeni na način da za određenu kombinaciju ulaznih vrijednosti postavljaju izlazne veličine pod napon ili ih ostavljaju bez napona.

2.3. Financijski aspekt ugradnje automatiziranog sustava vodoopskrbe iz podzemnih izvora

Kada se govori o financijskom aspektu ugradnje automatskog sustava vodoopskrbe iz podzemnih izvora, bitno je situaciju dobro proučiti i analizirati. Prije kretanja u realizaciju projekta, bitno je analizirati parametre prirodnog okruženja. Najbitnije je odrediti nalazi li se na ciljanom terenu podzemna voda te na kojoj dubini. To su veoma bitni podaci iz razloga što je samo bušenje jedna od najvećih investicija, ovisno o dubini bušenja. Cijena bušenja bušotine raste i s obzirom na sastav tla pa će tako bušenje u kamenitom tlu biti znatno skuplje i sporije od bušenja u zemljanom tlu. Poslije bušenja bušotine, cijenu uvelike diktira i količina potrebne vode i protoka. Ako je potreban veći protok i/ili pritisak vode, pumpe, cjevovodi i spremnik biti će skuplji od onih potrebnih za manje protoke. Što se tiče bušotine, srednja cijena po metru dubine je približno 550kn.⁴ Nadalje, pumpa za podizanje vode u spremnike u tom će primjeru koštati oko 2225 kn⁵. Navedeni iznosi su okvirnog karaktera budući da za iste dubine postoje pumpe različitih snaga, odnosno protoka. Ako smo, na primjer, ograničeni snagom izvora električne energije (vikendice sa benzinskim generatorima) tada će se nabaviti pumpa manje snage koja će na osnovu manje snage imati i manji protok vode u vremenu pa će biti potrebno duže vrijeme za punjenje spremnika. Jedan od najvažnijih dijelova sustava potreban da bi u cijevima u svakom trenutku bila dostupna voda pod određenim pritiskom je hidrofora jedinica. Kada govorimo o hidrofornoj jedinici izbor opet ovisi o specifičnoj situaciji jer sve ovisi o tome

⁴ https://www.emajstor.hr/cijene/dubisnko_busenje_busotina (12.12.2018.)

⁵ <https://www.ikoma.hr/hr/vodovodni-materijal/potopne-pumpe-za-vodu/pumpa-za-vodu-potopna-38-m-150-l-min-pedrollo-4blockm-6-6-7651/> (12.12.2018.)

kolika je potrošnja vode u vremenu te koliki protok vode je potreban pod određenim pritiskom u trenutku. Shodno tome, cijene mogu varirati od 900 kn⁶ za mali hidropak pa do preko 10.000 kn za velike hidroforne sustave.

Upravljačka jedinica je sljedeći trošak potreban za implementaciju automatski upravljivog sustava. Cijena jedinice ovisi o tome je li se programira u vlastitom aranžmanu ili se nabavlja već gotova jedinica za automatizaciju (PLC ili slično).

Kada se govori o financijskom aspektu bitan parametar je i utrošak električne energije budući da su pumpe veliki potrošači električne energije nazivnih snaga često preko 1 kW.

3. MODEL SUSTAVA NAVODNJAVANJA I VODOOPSKRBE

Model automatskog sustava navodnjavanja sastoji se od nekoliko dijelova. Prvi dio je razvodna električna kutija u kojoj su smještene sve komponente, drugi dio je Arduino upravljačka jedinica na koju su spojene bakreni vodiči koji vode do ulaznih i izlaznih elemenata. Treći dio su ulazni i izlazni elementi, prekidači kao ulazni, a LED lampice kao izlazni elementi. Bakreni vodiči su spojeni neposredno u ulaze na upravljačke jedinice, dok su LED diode spojene u seriju sa otpornikom pa nato na izlaze upravljačke jedinice. Namjena otpornika je usklađivanje naponske razine izlaza upravljačke jedinice i diode.

3.1. Potrebne komponente

3.1.1. Arduino UNO mikro kontroler

Arduino je ime za otvorenu računalnu i softversku platformu koja omogućava dizajnerima i konstruktorima stvaranje uređaja i naprava koje omogućuju spajanje računala s fizičkim svijetom tj. stvaranje interneta stvari. Arduino je stvorila talijanska tvrtka SmartProjects 2005. rabeći 8-bitne mikro kontrolere Atmel 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 sastajali kada su stvarali projekt.⁷

⁶ <https://www.ealati.hr/proizvod/rem-power-hidropak-wpem-340220-g/> (12.12.2018.)

⁷ <https://hr.wikipedia.org/wiki/Arduino> (12.12.2018.)

Arduino Uno je mikro kontrolerska ploča bazirana na ATmega328P mikročipu. Ima 14 digitalnih ulaznih odnosno izlaznih pinova, 6 analognih ulaza, 16 MHz kvarcni kristal, USB priključak, utičnicu za napajanje, ICSP zaglavlje i gumb za resetiranje. Sadrži sve što je potrebno za podršku mikro kontrolera. "Uno" znači jedan na talijanskom i odabran je za označavanje izdanja Arduino Software (IDE) 1.0. Uno ploča i verzija 1.0 Arduino Software (IDE) bile su referentne verzije Arduina, sada su evoluirale u novija izdanja. Uno ploča je prva u nizu USB Arduino ploča i referentni model za Arduino platformu.⁸



Slika 4. - Arduino uno mikrokontroler

Izvor: <https://shop.pimoroni.com/products/arduino-uno-rev3> (19.08.2019.)

Stvoritelji Arduina kažu da je on open-source platforma za kreiranje elektroničkih prototipova bazirana na sklopovlju i programskom paketu koji je fleksibilan i jednostavan za korištenje. Arduino je namijenjen umjetnicima, dizajnerima, hobbistima, elektroničarima i svima koji su zainteresirani za stvaranje interaktivnih objekata ili okruženja. Cijela Arduino platforma napravljena je i objavljena kao otvoreni sustav, što znači da su sve sheme i izvorni kod programa besplatno dostupni svima za preuzimanje sa službenih stranica proizvođača.⁹

3.1.2. Bluetooth HC 06 modul

HM-06 je Bluetooth modul dizajniran za uspostavljanje bežične podatkovne komunikacije kratkog dometa između dva mikro kontrolera ili sustava. Modul radi na Bluetooth 2.0 komunikacijskom protokolu i može djelovati samo kao podređeni uređaj. To je najjeftiniji način bežičnog prijenosa podataka i fleksibilniji u odnosu na druge metode i čak može prenositi datoteke brzinom do 2.1Mb / s.

⁸ <https://store.arduino.cc/arduino-uno-rev3> (12.12.2018.)

⁹Zenzerović, P. ARDUINO KROZ JEDNOSTAVNE PRIMJERE, Hrvatska zajednica tehničke kulture, Zagreb, 2006.

HC-06 koristi tehniku proširenog spektra skokova frekvencije (FHSS) kako bi se izbjegle smetnje s drugim uređajima i omogućio pun duplex prijenos. Uređaj radi na frekvencijskom području od 2.402 GHz do 2.480GHz.¹⁰

3.1.3. Android uređaj

Android Inc. su osnovali Andy Rubin, Rich Miner, Nick Sears i Chris White u listopadu 2003. godine kako bi razvijali programe za pametne mobilne uređaje koji bi uzimali u obzir korisničke postavke te njegovu lokaciju. Nakon dvije godine gotovo tajnog rada (jedino što je bilo poznato bilo je da se radi o softveru za mobitele), Google je odlučio kupiti Android te počinju spekulacije o ulasku Googlea na tržište pametnih telefona. Osnivači i ključni programeri, osnaženi Googleovim programerima, na tržište donose mobilnu platformu temeljenu na linuxovom kernelu koja bi trebala biti potpuno prilagodljiva zahtjevima korisnika.

Android platforma je prilagođena za uporabu na uređajima s većim zaslonima poput pametnih telefona koji rabe 2D grafičku knjižnicu ili 3D grafičku knjižnicu temeljenu na OpenGL ES 2.0 specifikacijama. Za pohranu podataka upotrebljava se SQLite relacijska baza podataka, napisana u C programskom jeziku. Karakteristike ovog softvera jesu njegova knjižnica koja u svega 275 kB implementira većinu SQL standarda. U odnosu na druge sustave za upravljanje bazama, SQLite nije zasebni proces već je sastavni dio aplikacije koja pristupa bazi podataka. [16] Za povezivanje s drugim uređajima primjenjuju se GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, WiFi (nije moguće povezivanje preko proxy poslužitelja i preko Ad hoc bežičnih mreža), LTE, NFC i WiMAX. Bluetooth podržava sučelje za upravljanje drugim uređajima (primjerice televizija, radio) te prijenos audio zapisa s jednog uređaja na drugi. Od Android verzije 3.0 postoji podrška za spajanje uređaja za upravljanje (tipkovnica, miš, igraće palice), dok na prijašnjim verzijama podrška je bila napravljena od samih proizvođača takvih uređaja. Za prijenos poruka upotrebljavaju se SMS i MMS servisi s podržanim prikazom razgovora. Android je do verzije 2.3 podržavao 26 različitih jezika, izlaskom Gingerbreada ta brojka se udvostručuje i danas je podržano čak 57 različitih jezika.[10] Internet preglednik je temeljen na WebKit-ovom engine-u uparenom s Googleovim Chrome V8 JavaScript engine. Uređaj s instaliranim Android Froyo sustavom kao i oni noviji imaju mogućnost tetheringa,

¹⁰ [https://components101.com/wireless/hc-06-bluetooth-module-pinout-datasheet\(12.12.2018.\)](https://components101.com/wireless/hc-06-bluetooth-module-pinout-datasheet(12.12.2018.))

odnosno uporaba uređaja kao žične (putem USB kabela) ili bežične pristupne točke za pristupanje internetu.¹¹

3.1.4. Elektronički elementi korišteni pri izradi modela

1. Prekidač

Pri izradi modela sustava navodnjavanja prekidači su korišteni u svojstu imitiranja senzora. Prekidač je jednostavan sklop koji propušta ili ne propušta struju ovisno o položaju polugice.



Slika 5.- Prekidač

Izvor: <https://elektronika.hr/proizvodi/prekidaci-ac-53> (10.07.2019.)

2. Svjetleća dioda

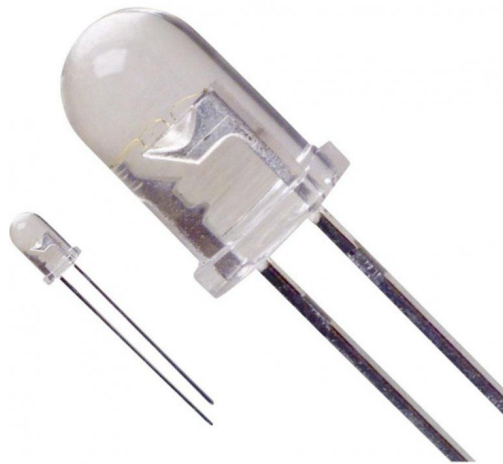
Općenito, dioda je nelinearni poluvodički elektronski element (ili, rjeđe, elektronska cijev) s dva priključka koji posjeduje ispravljačka osobine. Poluvodičke diode se izvedu se temelju pn-spoja ili, rjeđe, na temelju spoja metal-poluvodič. Dioda se mogu razvrstati po materijalu na kojemu su rađene (silicij, germanij, galij-arsenid, silicij-karbid) i po tipu (ispravljačke, svjetleće, foto-diode, Zener diode, Schottky diode, tunel-diode itd.).¹²

Svjetleća dioda (en. LED - Light Emiting Diode, često se naziva i LE dioda) je posebna vrsta poluvodičke diode koja emitira svjetlost kada je propusno polarizirana, tj. kada kroz nju teče električna struja. Prilikom direktne rekombinacije para elektron-šupljina, emitira se foton svjetla.

¹¹ [https://hr.wikipedia.org/wiki/Android_\(operacijski_sustav\)](https://hr.wikipedia.org/wiki/Android_(operacijski_sustav)) (12.12.2018.)

¹² <http://www.gradimo.hr/clanak/led-ndash-svjetlo-buducnosti/24040> (12.12.2018.)

Takvu osobinu imaju poluvodiči galijev-arsenid (GaAs), galijev fosfid (GaP) i silicijev karbid (SiC). Ta pojava se naziva elektroluminiscencija. Boja emitiranog svjetla ovisi o poluvodiču, kao i o primjesama u njemu i varira od infracrvenog preko vidljivog do ultraljubičastog dijela spektra



Slika 6. - Led dioda

Izvor: <https://elektronika.hr/proizvod/led-diode/1001006-led-dioda-5mm-plava-759>
(25.08.2019.)

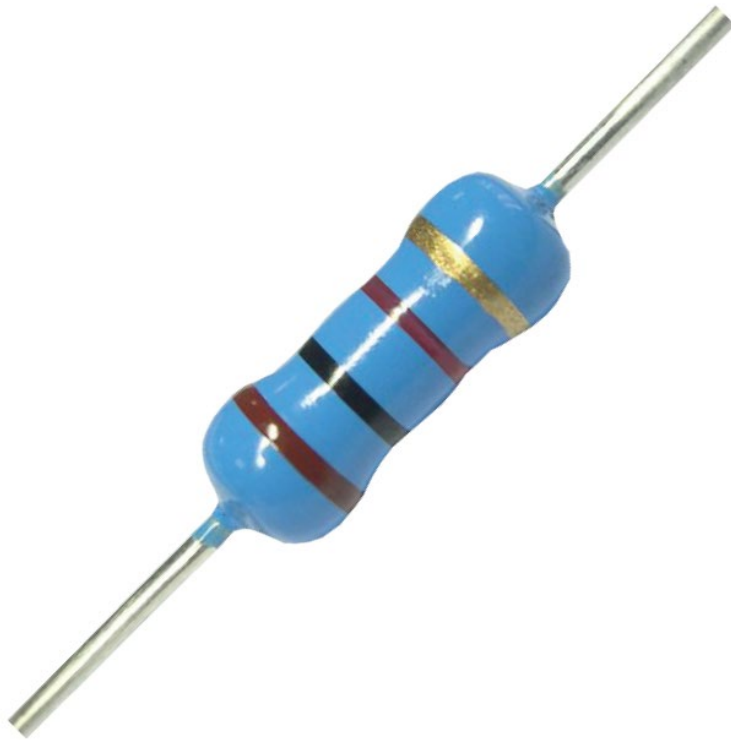
Na maketi sustava vodoopskrbe iz podzemnih izvora, LED dioda služiti će kao aktuator, tj. Led dioda će prikazivati stanja na izlazima arduino mikrokontrolera ovisno o vrijednostima na ulazima u mikrokontroler.

3. Otpornik

Otpornik je pasivna komponenta strujnog kruga, sa dva ulaza u kojemu je odnos jakosti struje i napona sukladan Ohmovom zakonu. Otpornik se koristi za ograničavanje struje. Ako spojimo otpornik u seriju prije neke druge komponente bilo to dioda ili zavojnica, struja kroz tu komponentu je smanjena na vrijednost koje je dozvoljena za tu komponentu. Možemo ga koristiti u ulozi prigušivača tako da spojimo mrežu dva ili više otpornika koji onda služe da bi smanjili napon signala. Najčešća primjena otpornika je za stvaranje strujno-naponskog odnosa u strujnom krugu. Naime ako znamo struju u strujnome krugu, tada otpornik koristimo da bi stvorili razliku potencijala koja je proporcionalna struji. Sa druge strane, ako je

poznata razlika potencijala između dvaju točaka u strujnom krugu, otpornik možemo upotrijebiti za stvaranje te poznate struje koja je proporcionalna razlici potencijala.

Na modelu sustava navodnjavanja otpornik ćemo koristiti kako bi smanjili razliku potencijala, odnosno, napon. Napon je potrebno smanjiti iz razloga što izlazni napon Arduino pločice iznosi 5V, a LED diode rade na naponu do 3V. Smanjenje razlike potencijala dobiti ćemo tako što će se otpornik spojiti serijski sa LED diodom te će na taj način na njemu nastajati pad napona od 2V koliko nam je potrebno da bi napon na diodi bio 3V.



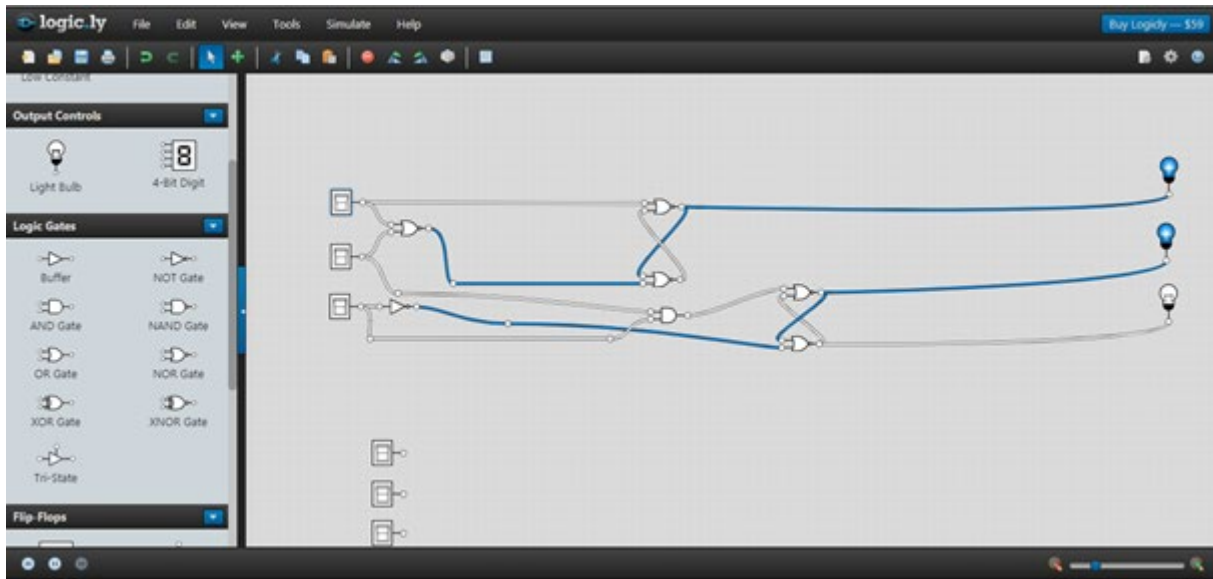
Slika 7. - Otpornik

Izvor: <https://www.elementa.rs/proizvod/28366/otpornik-m-film-1-4w-82r-1-rm5-08>

(11.07.2019.)

3.2. Logika automatskog upravljanja sustavom

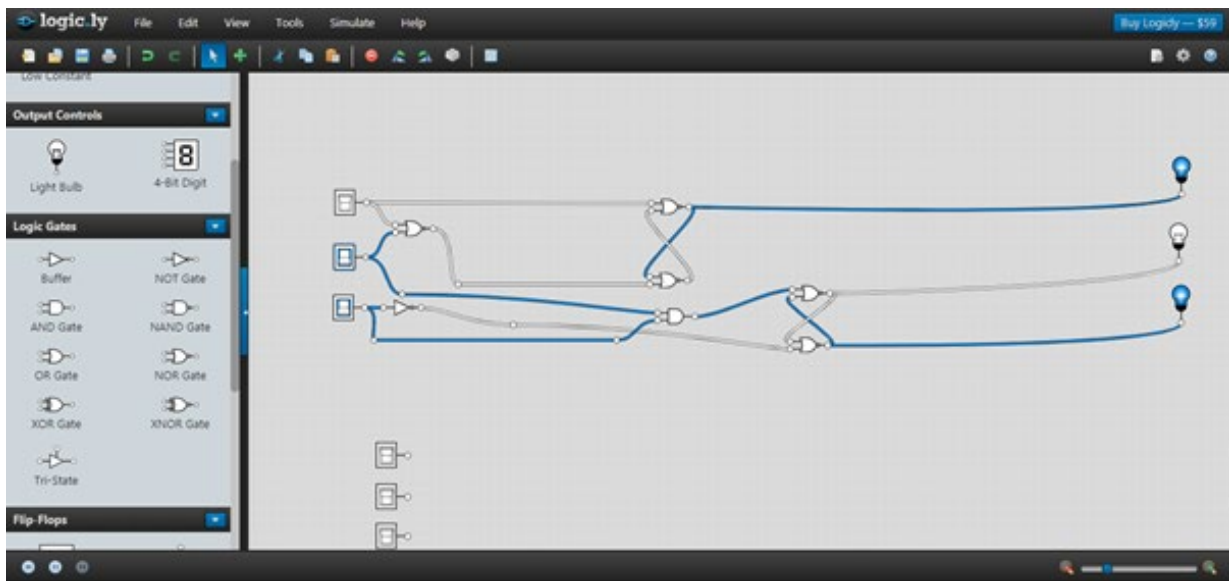
Za prikaz funkcionalnosti automatskog upravljanja tj. logiku upravljanja povratnom vezom korišten je besplatni online program „logicly“ kojim se pomoću jednostavnih logičkih sklopova kao što su „i“, „ili“, „ni“, „nili“ i ne, može jednostavno napraviti simulacija funkcioniranja sustava. Na sljedećim slikama prikazano je kako se ponašaju izlazi (LED diode) u ovisnosti o ulazima (prekidačima).



Slika 8. - Stanje logičkog upravljanja kod praznog spremnika

Izvor: Autor

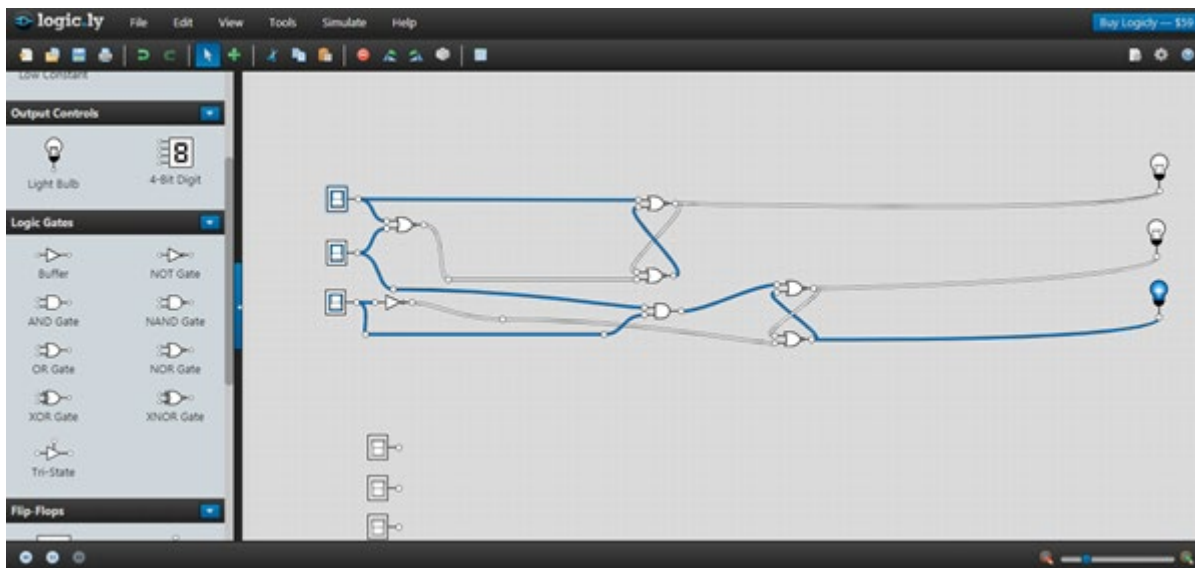
Slika 8 prikazuje situaciju kada u spremniku nema vode odnosno sva tri plovka (prekidača) su u položaju u kojem ne propuštaju struju te su tada aktivirani izlazi koji pale podzemnu pumpu te je otvoren ventil koji propušta napajanje vodom iz vodovoda.



Slika 9. - Stanje logičkog upravljanja kod spremnika napunjenog do drugog plovka

Izvor: autor

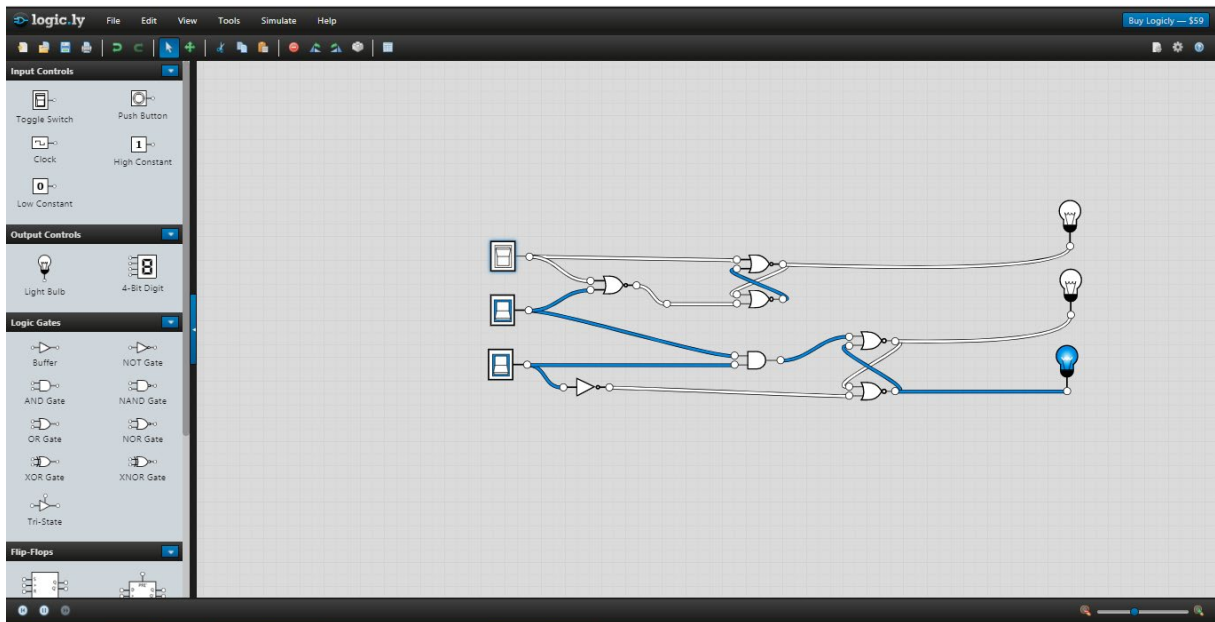
Na slici 9 je prikazana situacija kada voda u spremniku dostigne razinu veću od razine plovka na sredini spremnika te se tada napajanje vodom prebacuje s onog iz vodovoda na napajanje iz spremnika, odnosno hidroforne pumpe, a podzemna pumpa je i dalje upaljena dok se spremnik do kraja ne napuni. Kako bi se postigao drukčiji efekt kod punjenja i pražnjenja spremnika korišten je najjednostavniji memorijski sklop-bistabil.



Slika 10. - Stanje logičkog upravljanja kod punog spremnika

Izvor: autor

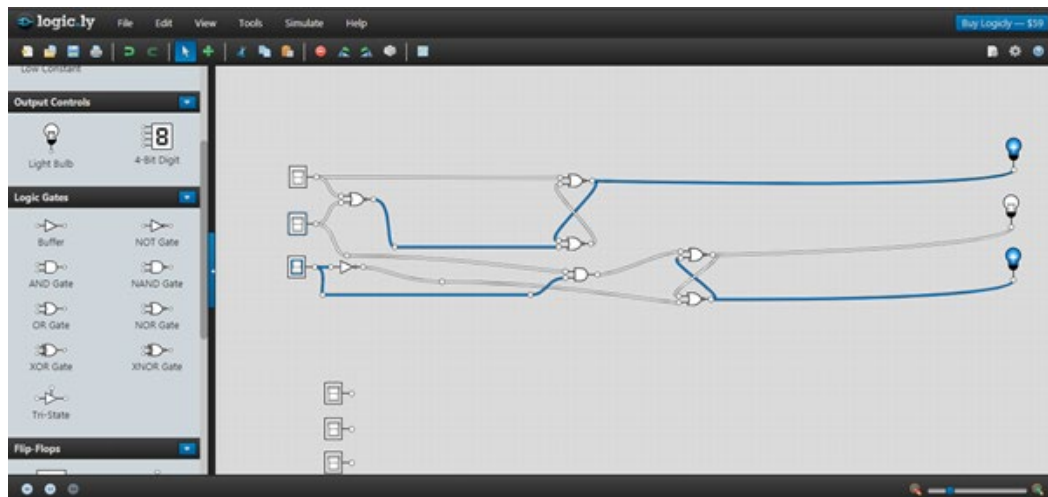
Slijedeća situacija na slici 10 je ona kada je spremnik pun te se tada gasi napajanje podzemne pumpe.



Slika 11. - Stanje logičkog upravljanja kod malo ispražnjenog spremnika

Izvor: autor

Na slici 11 se vidi efekt korištenja bistabilnog sklopa, odnosno kada se spremnik malo isprazni i opet se najviši plovak vrati u negativno stanje, pumpa se neće upaliti te kako bi izbjegli stalno paljenje i gašenje oko jedne točke ona će se upaliti tek kada voda dostigne razinu srednjeg plovka.



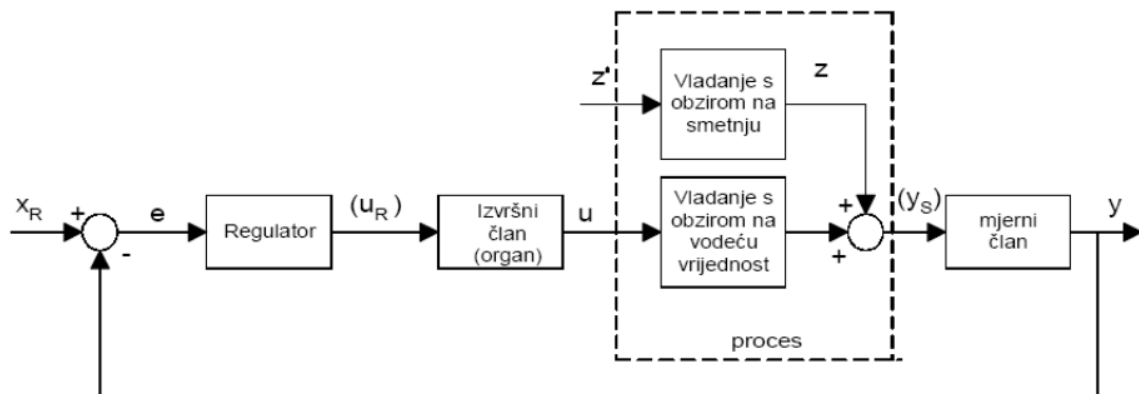
Slika 12. - Stanje logičkog upravljanja kod spremnika ispražnjenog do drugog plovka

Izvor: autor

Na slici 12 je vidljiva situacija u kojoj voda dostiže razinu vode srednjeg plovka te se tada pali pumpa i puni spremnik do kraja, u slučaju da u izvoru ne bi bilo vode te pumpa ne bi punila spremnik, situacija bi se vratila u stanje kao na slici 1.

3.2.1 Sustav automatskog upravljanja s povratnom vezom

Kako se postizanje željene vrijednosti regulirane veličine ostvaruje u zatvorenoj petlji, na način da se izmjerena vrijednost regulirane veličine uspoređuje sa referentnom vrijednošću, a signal njihove razlike (regulacijsko odstupanje) ulazi u regulacijski član koji nato djeluje na izvršne članove, govorimo o sustavu automatskog upravljanja. Na temelju sustava povratne veze djeluju brojni procesi u prirodi i društvu: evolucija, biološki sustavi, društveno – politički sustavi. Osnovna ideja povratne veze jest usporedba aktualne vrijednosti sa željenom vrijednošću veličine koju reguliramo te djelovanje na temelju njihove razlike. Ljudi su u svojim izumima oduvijek oponašali prirodu, pa su i u tehničke sustave uveli povratnu vezu. Veliki je broj primjera, od vrlo jednostavnih do iznimno složenih poput robota, transportnih sredstava, električnih ili internetskih mreža. Štoviše, rijetki su moderni tehnički sustavi koji ne sadrže barem neki podsustav automatske regulacije.¹³



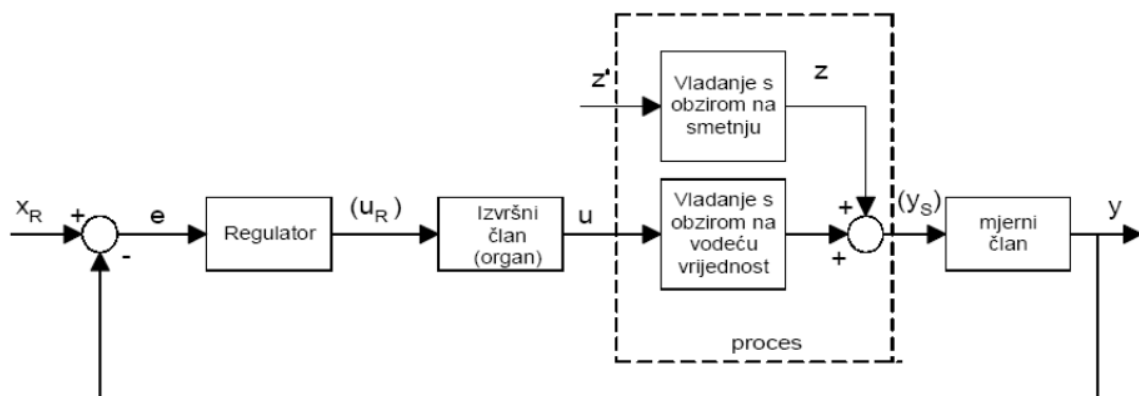
Slika 13. - Blok shema automatskog upravljanja razine vode u spremniku

Izvor: Autor

Model automatskog upravljanja vodoopskrbom podijeljen je na dva regulacijska kruga.

¹³https://zoranpericsplit.weebly.com/uploads/1/2/4/9/12491619/09._poglavlje_regulacija_g.p-osnove.pdf(12.12.2018.)

Prvi regulacijski krug opisuje upravljanje sustavom vodoopskrbe. Ovaj krug opisuje kako izrađeni sustav na temelju razine vode u spremniku regulira izvore vodoopskrbe, odnosno elektro ventile i pumpu vode. Varijabla X_r označava referentnu veličinu, a u ovom slučaju je to razina vode u spremniku, ova veličina ulazi u točku zbrajanja u koju ulazi i negativna povratna veza koja se spaja na izlaz sustava gdje se nalazi regulirana veličina y što je u ovom slučaju razina vode u spremniku. Kada se od željene tj., referentne vrijednosti oduzme regulirana veličina ako postoji razlika ona se naziva regulacijsko odstupanje i ona se označava sa e . Regulacijsko odstupanje ulazi u regulator, što je u ovom slučaju Arduino kontroler koji tu veličinu obradi te izlazni signal pretvori u fizikalnu veličinu kojom će se moći pokretati izvršni član. U slučaju izvršni član je elektro ventil koji otvara dotok vode s obzirom na regulacijsko odstupanje. Mjerni član predstavlja dio sustava koji pomoću senzora mjeri reguliranu veličinu. U ovom slučaju mjerni član predstavljaju plovci koji mjere razinu vode u spremniku, odnosno reguliranu veličinu. U procesu se neprestano odvija mjerenje i reguliranje regulirane veličine. Varijabla z' predstavlja nepredvidive vanjske smetnje u procesu, npr. Nastanak rupe na spremniku iz koje ističe voda, neki mehanički utjecaj na plovak ili slično.



Slika 14. - Blok shema automatskog upravljanja navodnjavanjem u odnosu na vlažnost tla

Izvor: Autor

Drugi regulacijski krug opisuje jednostavno upravljanje navodnjavanjem pomoću istog Arduino kontrolera. Ovaj krug opisuje kako sustav u ovisnosti od vlažnosti tla upravlja navodnjavanjem neke kulture. U drugom krugu Regulirana veličina je vlažnost tla. Referentna

vrijednost X_r je optimalna vlažnost tla koja se definira na temelju usjeva, odnosno potreba biljaka koje ćemo navodnjavati. Izvršni član je elektro ventil koji propušta vodu na špricaljke za navodnjavanje, a regulator je Arduino kontroler koji na temelju ulaznih veličina izlazne veličine postavlja u potrebna stanja. Mjerni član je senzor vlažnosti tla koji daje određene veličine napona u odnosu na vlažnost tla.

3.3. Izrada programa za upravljanje Arduino uređajem

Kod izrade programskog koda za Arduino upravljačku jedinicu koristi se jedinstveni programski jezik i to u Arduino programu iz kojeg se kod direktno upisuje u upravljačku jedinicu.

Na slici 14 je prikazan prvi dio programa u kojemu se najprije definiraju varijable koje će se koristiti u programskom kodu. Prilikom definiranja varijabli istima se dodjeljuje naziv i početna vrijednost.

```
int automatski = 2;           //definiramo varijablu sklopke za odabir auomatskog načina na port 2
int plovak1 = 3;             //definiramo varijablu plovak1 na port 3
int plovak2 = 4;             //definiramo varijablu plovak2 na port 4
int plovak3 = 5;             //definiramo varijablu plovak3 na port 5
int senzorr1 = 6;           //definiramo varijablu senzor vlažnosti na port 6
int LEDpumpa = 7;           //definiramo varijablu LED pumpa na port 7
int LEDev1 = 8;             //definiramo varijablu LEDev1 na port 8
int LEDev2 = 9;             //definiramo varijablu LEDev2 na port 9
int LEDnav1 =10;            //definiramo varijablu LEDnav1 na port10
int LEDnav2 = 11;           //definiramo varijablu LEDnav2 na port 11
char data = 0;               //definiramo varijablu data koja će poprimati vrijednosti iz serijske komunikacije
int autobt = 0;              //definiramo varijablu autobt na koju ćemo spremati vrijedosti automatskog načina rada sustava
```

Slika 15. - Definiranje varijabli u programu

Izvor: Autor

Drugi dio programa počinje sa *void setup* i to je onaj u kojemu prije definiranim varijablama dodjeljujemo dodatne parametre, odnosno varijablama koje ćemo koristiti kao ulazne ili izlazne pinove putem funkcije *pinMode* dodjeljujemo parametre *input* (ulaz) ili *output* (izlaz). Naredbom *digitalWrite* izlazne pinove postavlja se u visoko ili nisko stanje, a naredbom *digitalRead* čitaju se vrijednosti digitalnih ulaza.

```

void setup() {

pinMode(automatski, INPUT);           //varijalu automatski definiramo kao ulazni pin
pinMode(plovak1, INPUT);             //varijablu plovak1 definiramo kao ulazni pin
pinMode(plovak2, INPUT);             //varijablu plovak2 definiramo kao ulazni pin
pinMode(plovak3, INPUT);             //varijablu plovak3 definiramo kao ulazni pin
pinMode(senzorv1, INPUT);            //varijablu senzorv1 definiramo kao ulazni pin
pinMode(LEDpumpa, OUTPUT);           //varijablu LEDpumpu definiramo kao izlazni pin
pinMode(LEDev1, OUTPUT);             //varijablu LEDev1 definiramo kao izlazni pin
pinMode(LEDev2, OUTPUT);            //varijablu LEDev2 definiramo kao izlazni pin
pinMode(LEDnav1, OUTPUT);            //varijablu LEDnav1 definiramo kao izlazni pin
pinMode(LEDnav2, OUTPUT);           //varijablu LEDnav2 definiramo kao izlazni pin
digitalWrite(LEDpumpa, LOW);         //izlaz LEDpumpa postavljamo u nisku vrijednost
digitalWrite(LEDev1, LOW);           //izlaz LEDev1 postavljamo u nisku vrijednost
digitalWrite(LEDev2, LOW);           //izlaz LEDev2 postavljamo u nisku vrijednost
digitalWrite(LEDnav1, LOW);          //izlaz LEDnav1 postavljamo u nisku vrijednost
digitalWrite(LEDnav2, LOW);          //izlaz LEDpumpa postavljamo u nisku vrijednost
Serial.begin(9600);                  //ovom naredbom započinjemo serijsku komunikaciju
}

```

Slika 16. - Definiranje ulaznih i izlaznih pinova u programu

Izvor: Autor

Treći dio programa, prikazan na Slici 15, počinje sa *void loop* funkcijom, što znači da ovaj dio programa predstavlja petlju u kojoj se proces ciklično ponavlja sve dok se regulirana veličina ne izjednači sa referentnom vrijednosti. U ovom dijelu programa uz pomoć osnovnih logičkih funkcija kao što su *or*, *and* te *if* sustav obrađuje podatke prikupljene na mjernom članu te ih neprestano uspoređuje sa referentnom vrijednosti. Ovisno o regulacijskom odstupanju, sustav na izvršne članove šalje određene signale kojima regulira veličinu koja se mjeri. Kada se regulirana veličina izjednači sa referentnom, sustav više ne šalje napon na izvršne članove.

```

void loop() {
if(digitalRead(senzorvl) == HIGH){ //ovaj vlok naredbi služi kako bi se izlazi za navodnjavanje postavili u nisko stanje kada je senzor vlage u visokom
digitalWrite(LEDnav1,LOW);
digitalWrite(LEDnav2,LOW);
}
}
if((digitalRead(plovak1)==HIGH)and(digitalRead(plovak2)==HIGH)and(digitalRead(plovak3)==HIGH)){
digitalWrite(LEDpumpa,LOW); //ovaj blok naredbi služi da se u slučaju kada su sva 3 plovka podignuta pumpa stavlja u nisko stanje
delay(5);
}
if(digitalRead(senzorvl == LOW){
if(data == '2'){digitalWrite(LEDnav1,HIGH);} // ovaj blok služi da se u izlazi za navodnjavanje postavljaju u visoko ili nisko stanje
if(data == '3'){digitalWrite(LEDnav2,HIGH);} // s obzirom na vrijednost koja dolazi u serijskoj komunikaciji
if(data == '9'){digitalWrite(LEDnav1,LOW);}
if(data == '1'){digitalWrite(LEDnav2,LOW);}
}
}
if(digitalRead(automatski)==LOW){ //kada se sklopka za automatsko upravljanje stavlja u nisko stanje taka varijabla autobt poprima vrijednost nula
autobt = 0;
}
if(digitalRead(automatski)==HIGH){ //kada se sklopka za automatsko upravljanje stavlja u visoko stanje taka varijabla autobt poprima vrijednost jedan
autobt = 1;
}
}
}

```

Slika 17. - Upravljački kod za kontrolu vodoopskrbe

Izvor: Autor

```

if(autobt == 0){
if(data == '6'){digitalWrite(LEDpumpa,HIGH);} //kada varijabla autobt ima vrijednost nula tada se izlazi upravljaju ulazima na serijsku komunikaciju
}
if(data == '8'){digitalWrite(LEDpumpa,LOW);}
}
if(data == '4'){
digitalWrite(LEDdev1,HIGH);
digitalWrite(LEDdev2,LOW);
}
}
if(data == '5'){
digitalWrite(LEDdev2,HIGH);
digitalWrite(LEDdev1,LOW);
}
}
}
if(autobt == 1){ //kada varijabla autobt ima vrijednost jedan tada se izlazi upravljaju automatski s obzirom na stanja očitana na ulasima
}
if(digitalRead(plovak1)==LOW){
digitalWrite(LEDdev1,HIGH);
digitalWrite(LEDdev2,LOW);
digitalWrite(LEDpumpa,HIGH);
}
}
if((digitalRead(plovak1)==HIGH)and(digitalRead(plovak2)==LOW)){
digitalWrite(LEDpumpa,HIGH);
}
}
if((digitalRead(plovak1)==HIGH)and(digitalRead(plovak2)==HIGH)){
digitalWrite(LEDdev2,HIGH);
digitalWrite(LEDdev1,LOW);
}
}
}
delay(5);
}
}

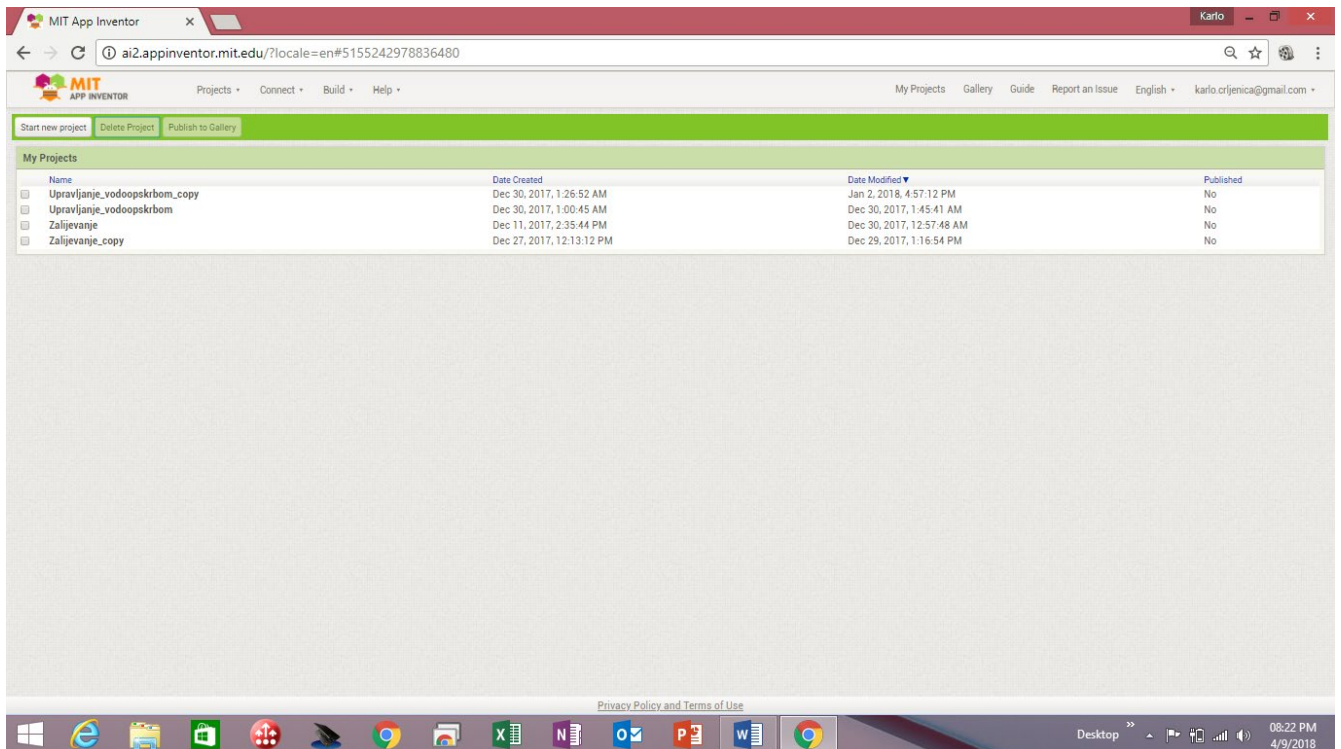
```

Slika 18 - Upravljački kod za upravljanje navodnjavanjem

Izvor: Autor

3.4. Izrada aplikacije za upravljanje pomoću Android uređaja

Sustav automatskog upravljanja uporabom Arduino uređaja nadograđen je aplikacijom za Android uređaje. Za izradu aplikacije korišten je MIT app inventor online program. Ovaj je program besplatan, a razvijen na američkom sveučilištu MIT. Slika 18 prikazuje razvojno okruženje programa MIT app inventor. MIT app inventor je razvojni program koji kao podlogu koristi JAVA programski jezik, ali je za razliku od staromodnog programiranja mnogo jednostavniji za korištenje budući da se programira u grafičkom načinu uporabom blokova. Veza između Arduino upravljačke jedinice i Android uređaja uspostavljena je pomoću bluetooth komunikacije. Odabirom željenih opcija putem aplikacije na Android sustavu šalju se određene vrijednosti na bluetooth modul spojen na Arduino kontroler.

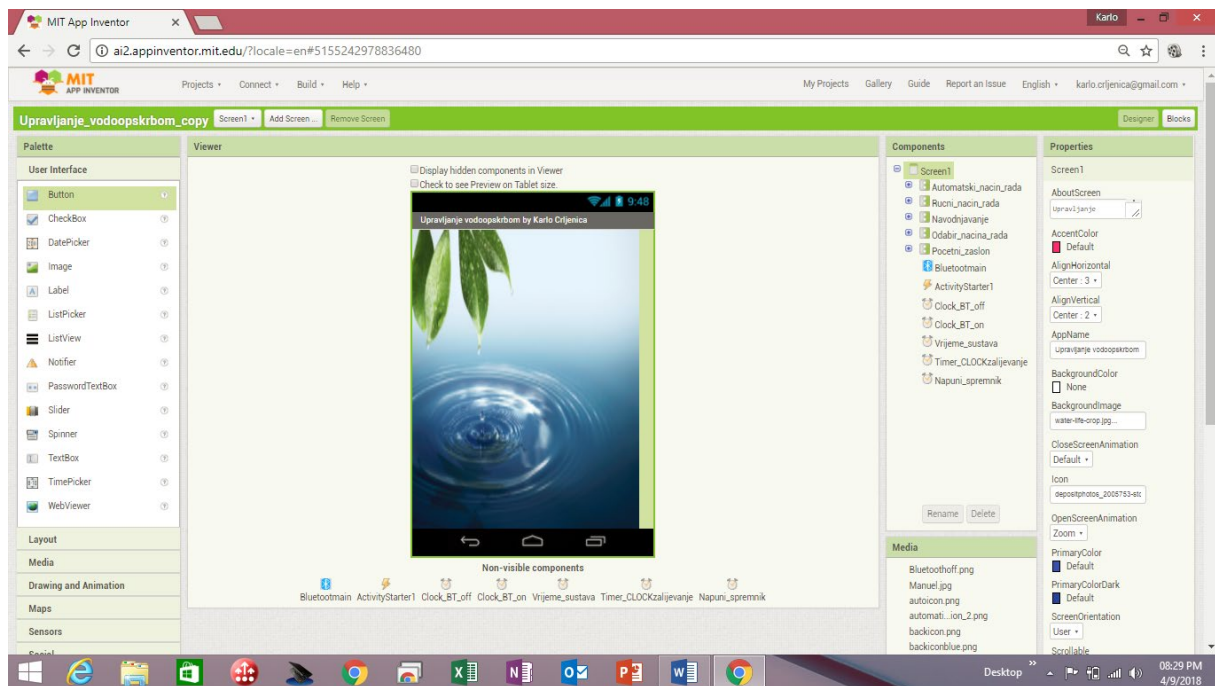


Slika 19. - MIT App Inventor razvojno okruženje

Izvor: <http://ai2.appinventor.mit.edu/?locale=en#5155242978836480> (31.03.2018.)

3.4.1. Izrada dizajna aplikacije

Slika 19 prikazuje razvojno okruženje za dizajniranje aplikacije. Aplikacija može sadržavati više slojeva koji se nazivaju „Screen“ pri čemu svaki sloj djeluje kao zasebna pod aplikacija. U ovom slučaju cijela aplikacija izrađena je u jednom sloju budući da se kroz sve dijelove aplikacije protežu parametri postavljeni na početku rada sa aplikacijom (npr. parametri uspostave Bluetooth veze). Na samom početku rada odabire se izgled pozadine. Pozadina može biti jednostavna jednobojna, ali u ovom slučaju za pozadinu je postavljena fotografija sa motivom vode i biljke, što upućuje korisnika na svrhu rada aplikacije. Sa lijeve strane nalazi se izbornik sa svim vidljivim i nevidljivim elementima koje možemo dodavati na Screen, a sa desne strane prikazana je lista dodijeljenih elemenata.



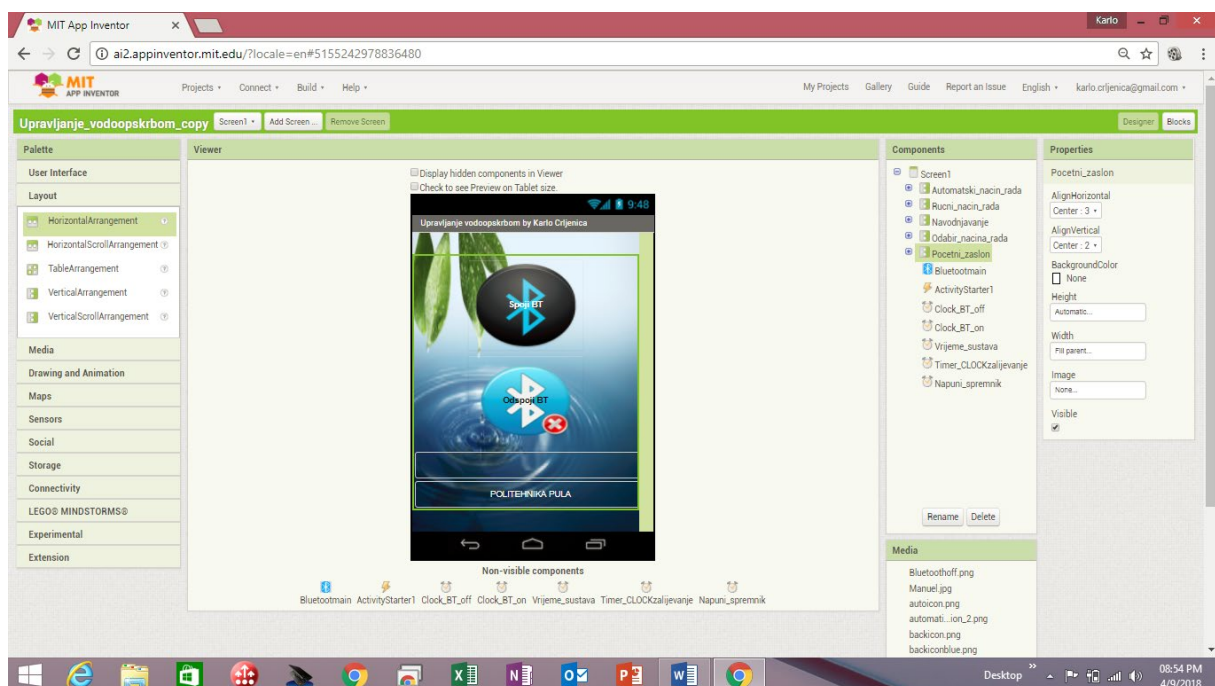
Slika 20. - Dizajniranje početnog zaslona aplikacije

Izvor: Autor

Nakon što je odabran izgled pozadine dodaju se elementi. Prije dodavanja gumbova i ikona, potrebno je odabrati odgovarajući *layout*, odnosno raspored elemenata na ekranu.

Na slici 20 je vidljiv prvi raspored koji sadrži funkcijske ikone za uspostavljanje bluetooth veze ili prekidanje iste. Prvi raspored također sadrži i prostore za tekst, od kojih je na jednom ispisan naziv visoke škole, a drugi je prazan te će se na njemu ispisivati određeni tekst s obzirom na stanje bluetooth veze. Osim ikona i prostora za tekstove, u ovom je koraku u

aplikaciju ugrađeno i više nevidljivih elemenata. Nevidljivi elementi nemaju nikakav utjecaj na dizajn nego služe isključivo za funkcionalnost aplikacije. Ugrađene su tri vrste nevidljivih elemenata. Prva vrsta je *Activity starter* koji služi za pozivanje nekog događaja u aplikaciji, a u ovom konkretnom slučaju služi kako bi pri samom pokretanju aplikacije potražio android uređaj za uključivanje i korištenje bluetooth modula. Druga vrsta korištenog nevidljivog elementa je *Bluetooth.main* koji služi za upravljanje bluetooth vezom Android uređaja. Treća i posljednja vrsta upotrebljenog nevidljivog elementa je *timer* koji ima ulogu brojača. Brojači služe kao podloga za mnoge funkcije u aplikaciji, pa će se za različite namjene postaviti različiti parametri brojača.

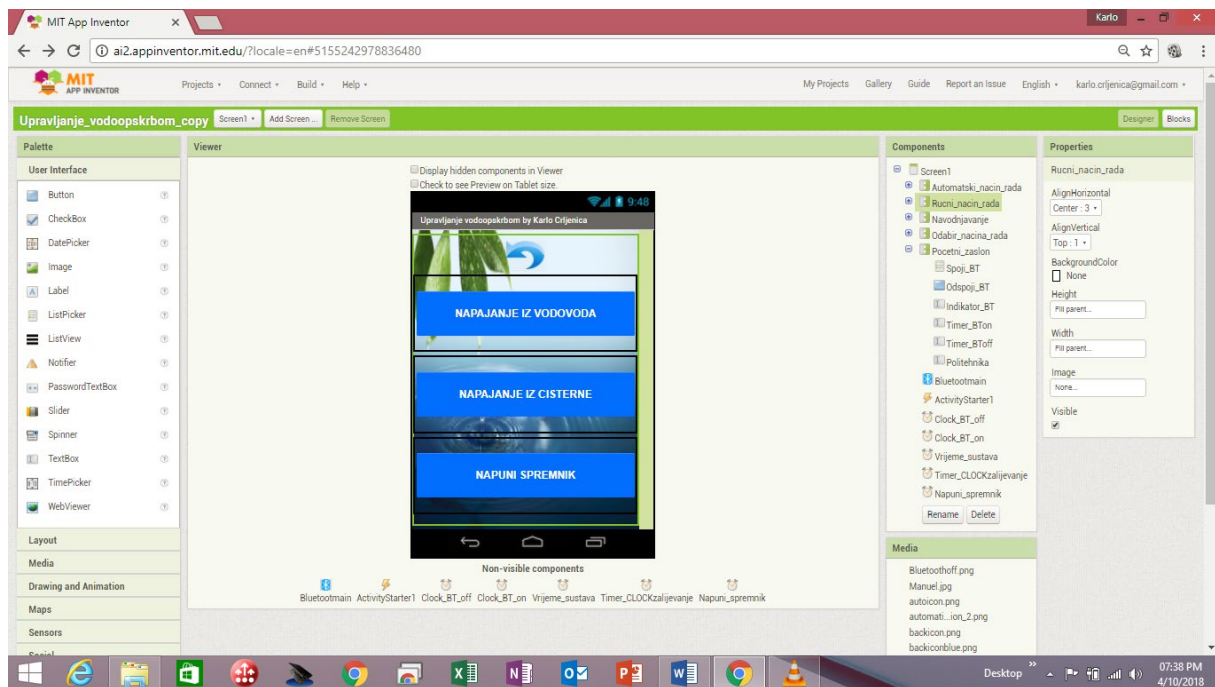


Slika 21. - Dizajniranje početnog zaslona aplikacije

Izvor: Autor

Na slici 21 prikazan je raspored ikona, koji služi za odabir načina rada navodnjavanja. Ovaj zaslon javlja se pola sekunde nakon što je uspješno uspostavljena bluetooth veza između Android uređaja i Arduino upravljačke jedinice. Pola sekunde vremena za izmjenu zaslona ostavljeno je kako bi korisnik mogao pročitati tekst statusa veze nakon što je uspostavljena i to u praznom polju za tekst. Funkcija čekanja pola sekunde dobivena je pomoću elementa *timer* kojeg smo dodali kao nevidljivi dio aplikacije u ovom koraku, zajedno sa bluetooth elementom, a način postavljanja parametara ovih dijelova biti će objašnjen kasnije u poglavlju logičkog dijela aplikacije. U izborniku odabir načina rada imamo 4 gumba kojima su dodane ikone zbog lakšeg korištenja i boljeg korisničkog iskustva. Prva ikona služi za izbor automatskog načina

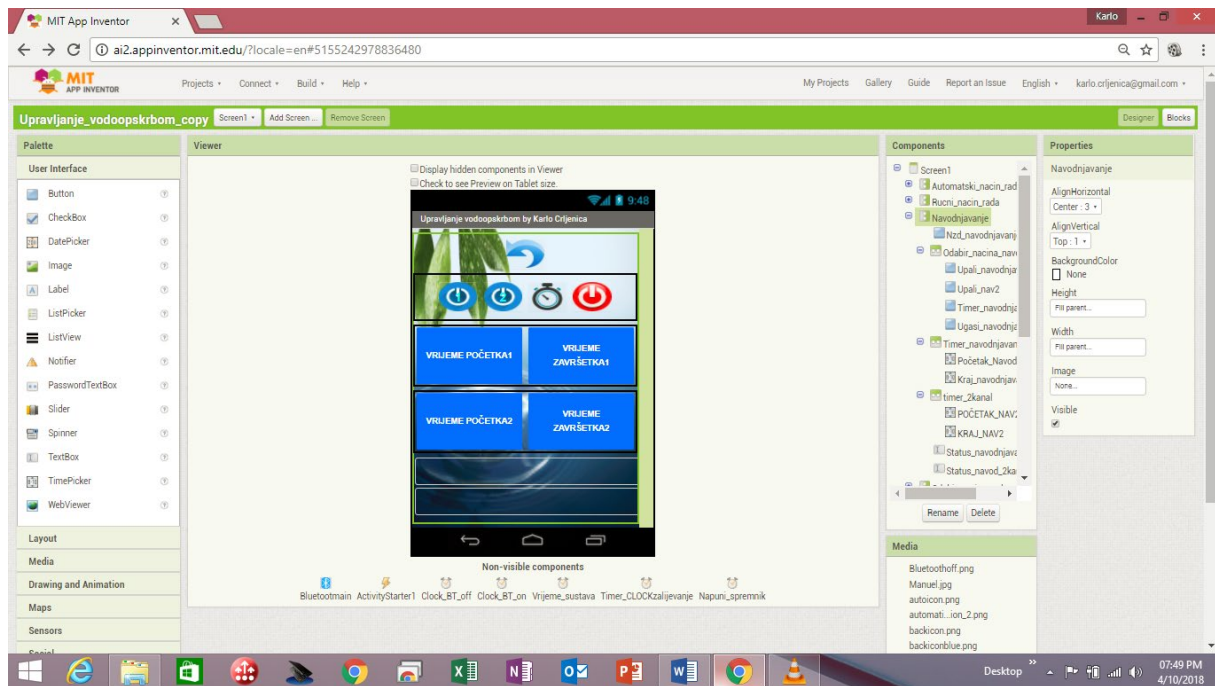
navodnjavanja, čime se na kontroler šalje vrijednost koja će njega postaviti u stanje da bira izvor vodoopskrbe sukladno sa parametrima senzora, dok će se pritiskom na ikonu ručni način rada otvoriti izbornik (Slika 21) na kojem se može odabrati napajanje sustava iz vodovoda, napajanje sustava iz cisterne te opcija „napuni spremnik“ kojom će se cisterna napuniti neovisno o parametrima senzora, koje na modelu imitiraju prekidači. Na vrhu izbornika nalazi se i ikona u obliku strelice unazad koja služi za povratak na prethodni izbornik. Takva ikona sa istom funkcijom nalazi se i na svim ostalim izbornicima.



Slika 22. - Dizajniranje početnog zaslona aplikacije

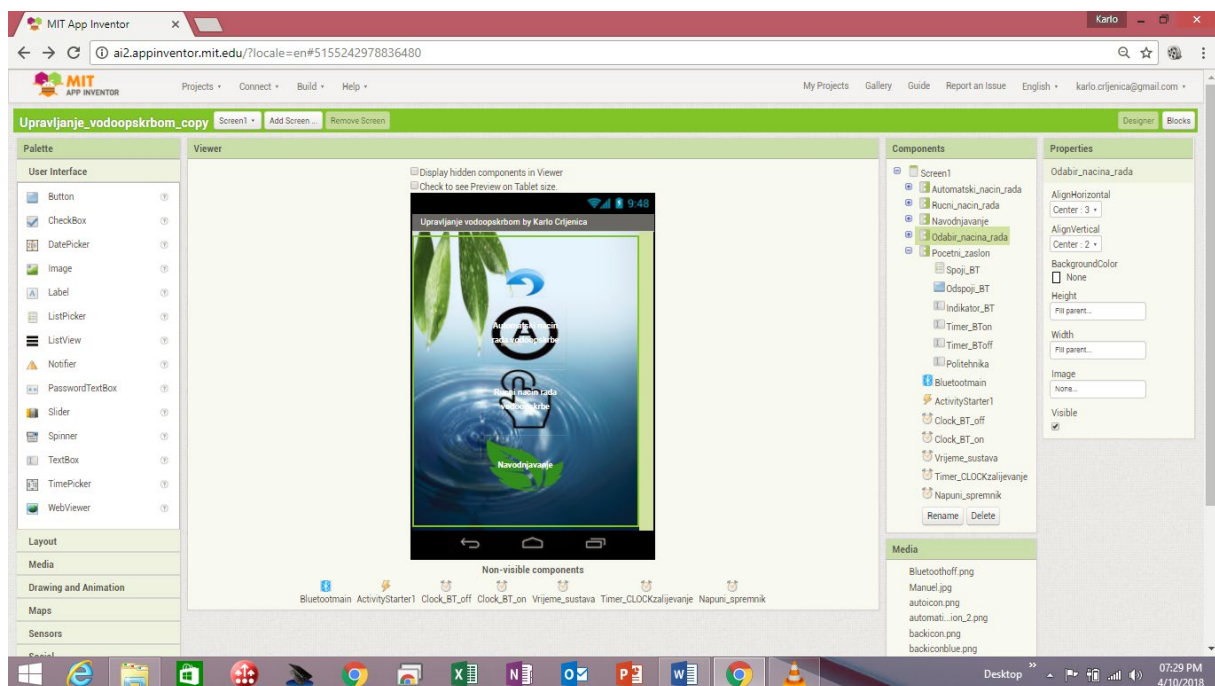
Izvor: Autor

Treća opcija u izborniku „odabir načina rada“ je opcija „Navodnjavanje“, odabirom koje se otvara izbornik u kojemu ima više rasporeda ubačenih u jedan veliki raspored. Ovaj izbornik ujedno sadrži i izbornik za ručno paljenje navodnjavanja te 2 timera kojima možemo odabrati vrijeme početka i završetka navodnjavanja za 2 kanala (cijeli je sustav izrađen na način da se neovisno upravlja sa 2 kanala navodnjavanja). Na podnožju izbornika nalaze se 2 prazna polja za tekst u kojima se automatski ispisuje status oba kanala navodnjavanja u realnom vremenu, a koje aplikacija preuzima iz aplikacije na Android sustavu.



Slika 23. - Dizajniranje početnog zaslona aplikacije

Izvor: Autor



Slika 24. - Dizajniranje izbornika aplikacije

Izvor: Autor

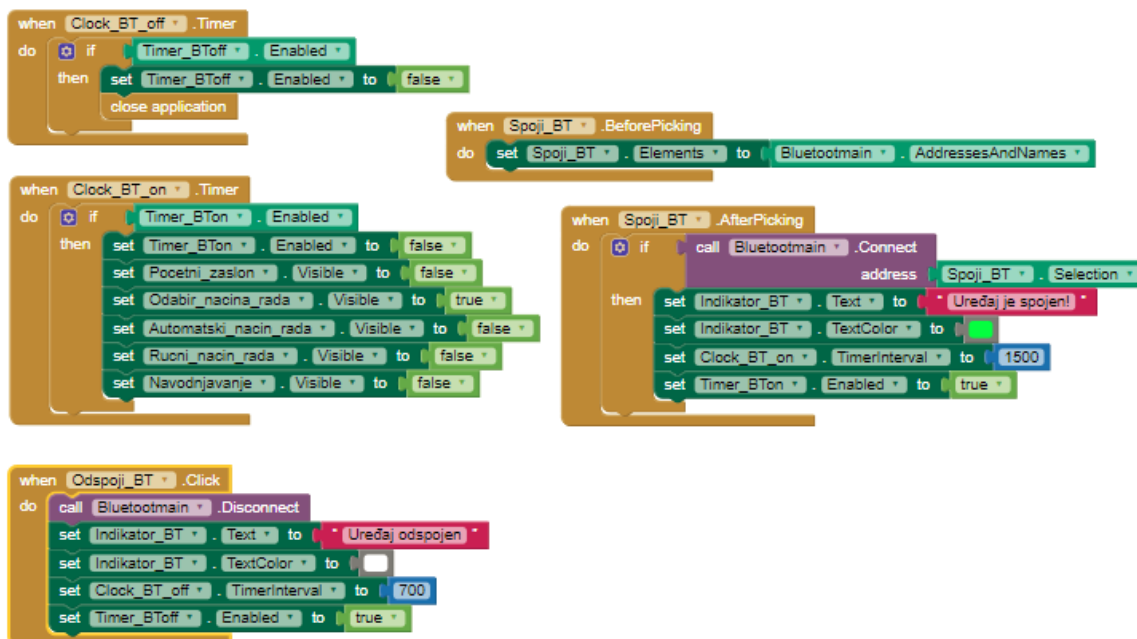
3.4.2. Izrada logičkog dijela aplikacije

Slika 25 prikazuje prvi programski blok. Blok započinje petljom „*when screen1 initialize do*“ što znači da će se u ovoj petlji izvršavati naredbe odmah pri potretanju same aplikacije. U prvoj petlji *Activity starter-u* je dodjeljen zadatak da zatraži uključivanje bluetooth modula android uređaja te je on odmah i pokrenut. Pošto pri pokretanju aplikacije uređaj nije spojen sa bluetooth modulom Arduino kontrolera, u prostor za tekst na početnom zaslonu dana je naredba da se ispiše tekst „Nema spojenih uređaja“ i to u crvenoj boji. Svi izbornici osim početnog zaslona koji je postavljen u stanje *visible true* postavljeni su u stanje *visible false* što znači da nisu vidljivi, što je praktično iz razloga da se na ekranu prikazuje samo jedan izbornik odjednom, zbog bolje preglednosti. U svim ostalim koracima programiranja u kojim se zahtjeva promjena izbornika, na isti ovaj način će se svi izbornici postaviti u nevidljivo stanje osim onog koji je potreban u datom trenutku. Osim postavljanja vidljivosti izbornika, u prvom koraku su aktivirani i neki brojači čiji parametri će biti potrebni u drugim koracima programa.



Slika 25. - Prvi programski blok

Izvor: Autor



Slika 26. - Drugi programski blok

Izvor: Autor

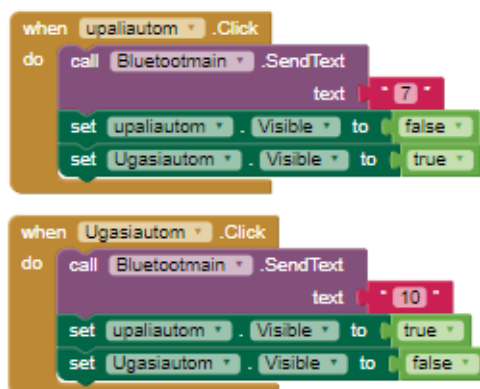
Na slici 25 je prikazan logički dio aplikacije koji upravlja bluetooth vezom i pokretanjem dijelova aplikacije. Elemente „clock“ iskoristilo se kao brojače, njih se koristilo iz praktičnih razloga, odnosno da bi se „uštedjelo“ nekoliko gumbova u aplikaciji čime se dobiva bolja ergonomija i praktičnost. Prilikom odabira bluetooth uređaja, aplikacija na dnu zaslona ispiše „Uređaj je spojen“ te se pokrene brojač koji ima interval 1.5 sekundu kako bi korisnik mogao pročitati tekst. Nakon tog intervala aplikacija automatski prelazi na sljedeći izbornik. Na sličan način je realiziran i završetak rada sa aplikacijom, kada korisnik želi prekinuti bluetooth komunikaciju i izaći iz aplikacije, klikom na ikonu odspoji BT uređaj, odspaja se, a na dnu zaslona ispisuje se tekst „uređaj je odspojen“. Nato se pokreće se brojač koji odbrojava 0.7 sekundi, nakon čega se aplikacija automatski zatvara.



Slika 27. - Treći programski blok

Izvor: Autor

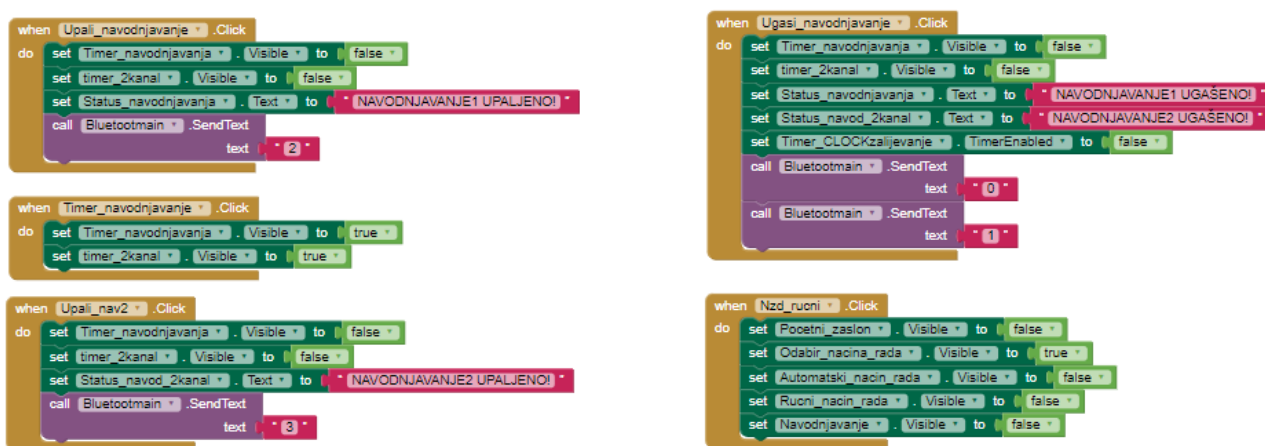
Na slici 27 prikazani se blokovi funkcija koji služe za kontrolu elemenata koji će se prikazivati na zaslonu. Sukladno tome što je potrebno, na ekranu se prikazuju samo određeni dijelovi. Ovaj dio realiziran je na način da se klikom na određene gumbове, određeni dijelovi stavljaju u stanje *visible true* ili *visible false*, odnosno u stanje vidljivo ili nevidljivo.



Slika 28. - Četvrti programski blok

Izvor: Autor

Kada je u izborniku načina rada odabrano „upali automatsko upravljanje“ aplikacija putem BT modula šalje Arduino kontroleru naredbu (vrijednost 7) da pokrene automatsko upravljanje vodoopskrbom. Isto tako, kada se pokrene automatsko upravljanje vodoopskrbom, ikona u aplikaciji zamjeni se crvenom ikonicom koja služi za gašenje automatskog načina rada. Dakle, jedan gumb koristimo za obje aplikacije te iz njega možemo i očitati stanje funkcije. Ovakav sustav uvelike pridonosi kompaktnosti i jednostavnosti korištenja aplikacije.



Slika 29. - Peti programski blok

Izvor: Autor

Na slici 29 prikazani su logički blokovi koji služe za upravljanje funkcijom navodnjavanja. Ovi blokovi su pozadina funkcija koje se u aplikaciji vizualno pojavljuju kao „upali navodnjavanje“, „ugasi navodnjavanje“ i to u dva kanala, te funkcija „timer“ koja otvara prozor sa dva brojača vremena na kojima se postavlja vrijeme paljenja ili gašenja navodnjavanja.



Slika 30. - Šesti programski blok

Izvor: Autor

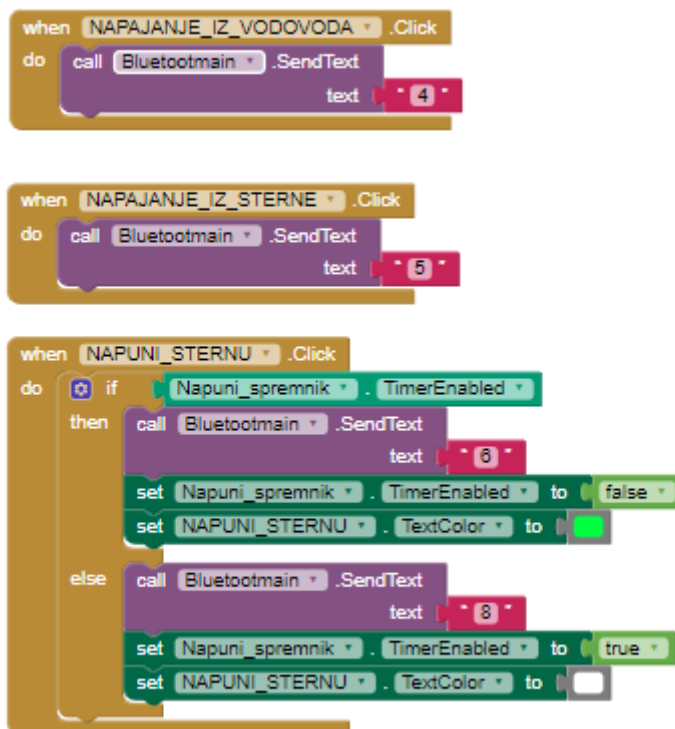
Slika 30 prikazuje dio aplikacije namijenjen vremenskom upravljanju navodnjavanjem. Početak svega je definiranje globalnih varijabli koje će poprimati vrijednosti trenutnog vremena koje se preuzima sa android uređaja te varijabli koje će poprimati vrijednosti vremena početaka i završetaka navodnjavanja u 2 kanala. Nakon što su definirane varijable ispod njih vide se blokovi koji obavljaju funkcije brojača vremena, a u narančastim blokovima vidljivo je kako se postavljene vrijednosti u brojačima dodjeljuju globalnim varijablama, a te vrijednosti se koriste bluetooth upravljanju koje je prikazano na slici 31.



Slika 31. - Sedmi programski blok

Izvor: Autor

Slika 31 prikazuje dio aplikacije koji, na temelju vrijednosti varijabli koji su ranije postavljani, upravlja sustavom navodnjavanja na način da kada se vrijeme sustava podudara sa nekom od postavljenih vrijednosti, šalje preko BT modula na Arduino uređaj određene vrijednosti.



Slika 32. - Osmi programski blok

Izvor: Autor

Slika 32 prikazuje posljednju grupu blokova aplikacije koji su namijenjeni ručnom načinu rada vodoopskrbe. Kada je početnom zaslonu odabrana funkcija „Ručni način rada vodoopskrbe“ tada se u aplikaciji otvara izbornik koji nudi tri opcije: napajanje iz cisterne, napajanje iz vodovoda te funkcija koja puni spremnik vode. Budući da se radi o ručnom načinu rada, ovi blokovi funkcioniraju na način da uz odabranu funkciju preko BT modula na Arduino šalju određene vrijednosti kako bi se izvršile željene naredbe.

4. ZAKLJUČAK

Voda je jedan od najvažnijih elemenata na svijetu i bez nje ne bi bilo ni života. Kada govorimo o vodi, ona je oduvijek bila čovjeku neophodna pa su se od davnina koristili razni izvori vode te razni načini akumulacije i skladištenje vode. Korištenje vode iz podzemnih izvora puno pomaže u svakodnevnom životu ljudi, a naročito onih koji nisu u mogućnosti koristiti vodu iz javne infrastrukture. Dobrobit korištenja automatiziranog sustava za upravljanje vodoopskrbom manifestira se tako što korisnik može uživati u punom komoditetu vodovodnog sustava, a da pritom koriste samo vodu iz podzemnog izvora.

Mana ovakvog sustava je ta što su pumpe često veliki potrošači električne energije, a one čine većinu ovog sustava. Uz potrošnju električne energije, mana ovog sustava je i potreba za izgradnjom spremnika te određena razina buke koje pumpe proizvode.

Dodatne prednosti i poboljšanja mogla bi se postići korištenjem solarne energije za napajanje pumpi čime bi se smanjila potrošnja električne energije.

Uz korištenje inženjerskih znanja, vještina i kompetencija, moguće je mnoge primitivne radnje učiniti jednostavnim te primjerenim suvremenom načinu života, a u isto vrijeme koristiti izvore vode i energenata bliže prirodi.

LITERATURA

Knjige:

1. Zenzerović, P.; **ARDUINO KROZ JEDNOSTAVNE PRIMJERE**, Hrvatska zajednica tehničke kulture, Zagreb, 2006.

Stručni radovi:

2. Pejdo, A.; Šiljković, Ž.: **MOGUĆNOSTI NAVODNJAVANJA PODZEMNIM VODAMA U HRVATSKOJ**, pregledni članak, odjel za geografiju, Sveučilište u Zadru, listopad 2007., Zadar

Materijali sa predavanja:

3. Lorencin, E.; **Osnove automatskog upravljanja**, materijali sa predavanja

Internet izvori:

4. https://www.emajstor.hr/cijene/dubisnko_busenje_busotina (12.12.2018.)
5. <https://www.ikoma.hr/hr/vodovodni-materijal/potopne-pumpe-za-vodu/pumpa-za-vodu-potopna-38-m-150-l-min-pedrollo-4blockm-6-6-7651/> (12.12.2018.)
6. <https://www.ealati.hr/proizvod/rem-power-hidropak-wpem-340220-g/> (12.12.2018.)
7. <https://hr.wikipedia.org/wiki/Arduino> (12.12.2018.)
8. <https://hr.wikipedia.org/wiki/Hidrofor> (12.12.2018.)
9. https://bs.wikipedia.org/wiki/Sistem_ku%C4%87ne_automatike (16.08.2017.)
10. <http://www.eko.zagreb.hr/default.aspx?id=109> (16.08.2017.)
11. <http://www.gradimo.hr/clanak/led-ndash-svjetlo-buducnosti/24040> (12.12.2018.)
12. https://zoranpericsplit.weebly.com/uploads/1/2/4/9/12491619/09_poglavlje_regulacija_g.p-osnove.pdf (12.12.2018.)
13. <http://cistopodzemlje.info/node/10086> (29.12.2017.)
14. <https://dluxeinternacional.com/automatic-water-supply-of-a-private-house/well-construction#image-1> (19.05.2019.)
15. http://www.hyperion.rs/images/fancybox/HyperION_Concept.jpg (29.08.2017.)
16. <https://shop.pimoroni.com/products/arduino-uno-rev3> (19.08.2019.)
17. : <https://elektronika.hr/proizvodi/prekidaci-ac-53> (19.08.2019.)

18. <https://elektronika.hr/proizvod/led-diode/1001006-led-dioda-5mm-plava-759>(25.08.2019.)
19. <https://www.elementa.rs/proizvod/28366/otpornik-m-film-1-4w-82r-1-rm5-08>
(11.07.2019.)
20. <http://ai2.appinventor.mit.edu/?locale=en#5155242978836480> (31.03.2018.)

POPIS SLIKA

Slika 1.- Presjek zemljine kore sa podzemnim vodama te primjerima onečišćenja	4
Slika 2. - Shema jednostavnog sustava vodoopskrbe iz bušotine	5
Slika 3.- Integrirani sustav pametne kuće	6
Slika 4. - Arduino uno mikrokontroler.....	10
Slika 5.- Prekidač	12
Slika 6. - Led dioda	13
Slika 7. - Otpornik.....	14
Slika 8. - Stanje logičkog upravljanja kod praznog spremnika.....	15
Slika 9. - Stanje logičkog upravljanja kod spremnika napunjenog do drugog plovka.....	16
Slika 10. - Stanje logičkog upravljanja kod punog spremnika.....	16
Slika 11. - Stanje logičkog upravljanja kod malo ispražnjenog spremnika	17
Slika 12. - Stanje logičkog upravljanja kod spremnika ispražnjenog do drugog plovka.....	17
Slika 13. - Blok shema automatskog upravljanja razine vode u spremniku	18
Slika 14. - Blok shema automatskog upravljanja navodnjavanjem u odnosu na vlažnost tla..	19
Slika 15. - Definiranje varijabli u programu	20
Slika 16. - Definiranje ulaznih i izlaznih pinova u programu	21
Slika 17. - Upravljački kod za kontrolu vodoopskrbe.....	22
Slika 18 - Upravljački kod za upravljanje navodnjavanjem	22
Slika 19. - MIT App Inventor razvojno okruženje.....	23
Slika 20. - Dizajniranje početnog zaslona aplikacije	24
Slika 21. - Dizajniranje početnog zaslona aplikacije	25
Slika 22. - Dizajniranje početnog zaslona aplikacije	26
Slika 23. - Dizajniranje početnog zaslona aplikacije	27
Slika 24. - Dizajniranje izbornika aplikacije.....	27
Slika 25. - Prvi programski blok	28
Slika 26. - Drugi programski blok.....	29
Slika 27. - Treći programski blok.....	30
Slika 28. - Četvrti programski blok.....	30
Slika 29. - Peti programski blok.....	31
Slika 30. - Šesti programski blok	32
Slika 31. - Sedmi programski blok.....	33

Slika 32. - Osmi programski blok 34

POPIS PRILOGA

Prilog 1. – programski kod za arduino mikrokontroler

```
int automatski = 2;           //definiramo varijablu sklopke za odabir auomatskog načina na
port 2

int plovak1 = 3;             //definiramo varijablu plovak1 na port 3

int plovak2 = 4;             //definiramo varijablu plovak2 na port 4

int plovak3 = 5;             //definiramo varijablu plovak3 na port 5

int senzorvl = 6;           //definiramo varijablu senzor vlažnosti na port 6

int LEDpumpa = 7;           //definiramo varijablu LED pumpa na port 7

int LEDev1 = 8;              //definiramo varijablu LEDev1 na port 8

int LEDev2 = 9;              //definiramo varijablu LEDev2 na port 9

int LEDnav1 =10;             //definiramo varijablu LEDnav1 na port10

int LEDnav2 = 11;           //definiramo varijablu LEDnav2 na port 11

char data = 0;               //definiramo varijablu data koja će poprimati vrijednosti iz serijske
komunikacije

int autobt = 0;              //definiramo varijablu autobt na koju ćemo spremati vrijedosti
automatskog načina rada sustava

void setup() {

pinMode(automatski,INPUT);    //varijalu automatski definiramo kao ulazni pin

pinMode(plovak1,INPUT);      //varijablu plovak1 definiramo kao ulazni pin

pinMode(plovak2,INPUT);      //varijablu plovak2 definiramo kao ulazni pin

pinMode(plovak3,INPUT);      //varijablu plovak3 definiramo kao ulazni pin

pinMode(senzorvl,INPUT);     //varijablu senzorvl definiramo kao ulazni pin
```

```

pinMode(LEDpumpa,OUTPUT);           //varijablu LEDpumpu definiramo kao izlazni pin
pinMode(LEDev1,OUTPUT);             //varijablu LEDev1 definiramo kao izlazni pin
pinMode(LEDev2,OUTPUT);             //varijablu LEDev2 definiramo kao izlazni pin
pinMode(LEDnav1,OUTPUT);            //varijablu LEDnav1 definiramo kao izlazni pin
pinMode(LEDnav2,OUTPUT);            //varijablu LEDnav2 definiramo kao izlazni pin
digitalWrite(LEDpumpa,LOW);          //izlaz LEDpumpa postavljamo u nisku vrijednost
digitalWrite(LEDev1,LOW);            //izlaz LEDev1 postavljamo u nisku vrijednost
digitalWrite(LEDev2,LOW);            //izlaz LEDev2 postavljamo u nisku vrijednost
digitalWrite(LEDnav1,LOW);           //izlaz LEDnav1 postavljamo u nisku vrijednost
digitalWrite(LEDnav2,LOW);           //izlaz LEDpumpa postavljamo u nisku vrijednost
Serial.begin(9600);                  //ovom naredbom započinjemo serijsku komunikaciju
}

```

```

void loop() {

```

```

if(digitalRead(senzorvl) == HIGH){ //ovaj blok naredbi služi kako bi se izlazi za
navodnjavanje postavili u nisko stanje kada je senzor vlage u visokom

```

```

    digitalWrite(LEDnav1,LOW);

```

```

    digitalWrite(LEDnav2,LOW);

```

```

    delay(500);

```

```

}

```

```

if((digitalRead(plovak1)==HIGH)and(digitalRead(plovak2)==HIGH)and(digitalRead(plovak
3)==HIGH)){

```

```

    digitalWrite(LEDpumpa,LOW); //ovaj blok naredbi služi da se u slučaju kada su sva 3
plovka podignuta pumpa stavlja u nisko stanje

```

```

    delay(5);

```

```

}

```

```

if(digitalRead(senzorvl == LOW)){

```

```

    if(data == '2'){digitalWrite(LEDnav1,HIGH);} // ovaj blok služi da se u izlazi za
navodnjavanje postavljaju u visoko ili nisko stanje

```

```
    if(data == '3'){digitalWrite(LEDnav2,HIGH);} // s obzirom na vrijednost koja dolazi u
serijskoj komunikaciji
```

```
    if(data == '9'){digitalWrite(LEDnav1,LOW);}

```

```
    if(data == '1'){digitalWrite(LEDnav2,LOW);}

```

```
    delay(500);

```

```
    }

```

```
if(digitalRead(automatski)==LOW){ //kada se sklopka za automatsko upravljanje stavlja u
nisko stanje taka varijabla autobt poprima vrijdnost nula
```

```
autobt = 0;

```

```
delay(500);

```

```
    }

```

```
if(digitalRead(automatski)==HIGH){ //kada se sklopka za automatsko upravljanje stavlja u
visoko stanje taka varijabla autobt poprima vrijdnost jedan
```

```
autobt = 1;

```

```
delay(500);

```

```
    }

```

```
if(autobt == 0){

```

```
    if(data == '6'){digitalWrite(LEDpumpa,HIGH);} //kada varijabla autobt ima vrijednost nula
tada se izlazi upravljaju ulazima na serijsku komunikaciju

```

```
    if(data == '8'){digitalWrite(LEDpumpa,LOW);}

```

```
    if(data == '4'){

```

```
        digitalWrite(LEDev1,HIGH);

```

```
        digitalWrite(LEDev2,LOW);

```

```
    }

```

```
    if(data == '5'){

```

```
        digitalWrite(LEDev2,HIGH);

```

```
        digitalWrite(LEDev1,LOW);

```

```

    }
    delay(500);
}

if(autobt == 1){ //kada varijabla autobt ima vrijednost jedan tada se izlazi upravljaju automatski
s obzirom na stanja očitana na ulazima

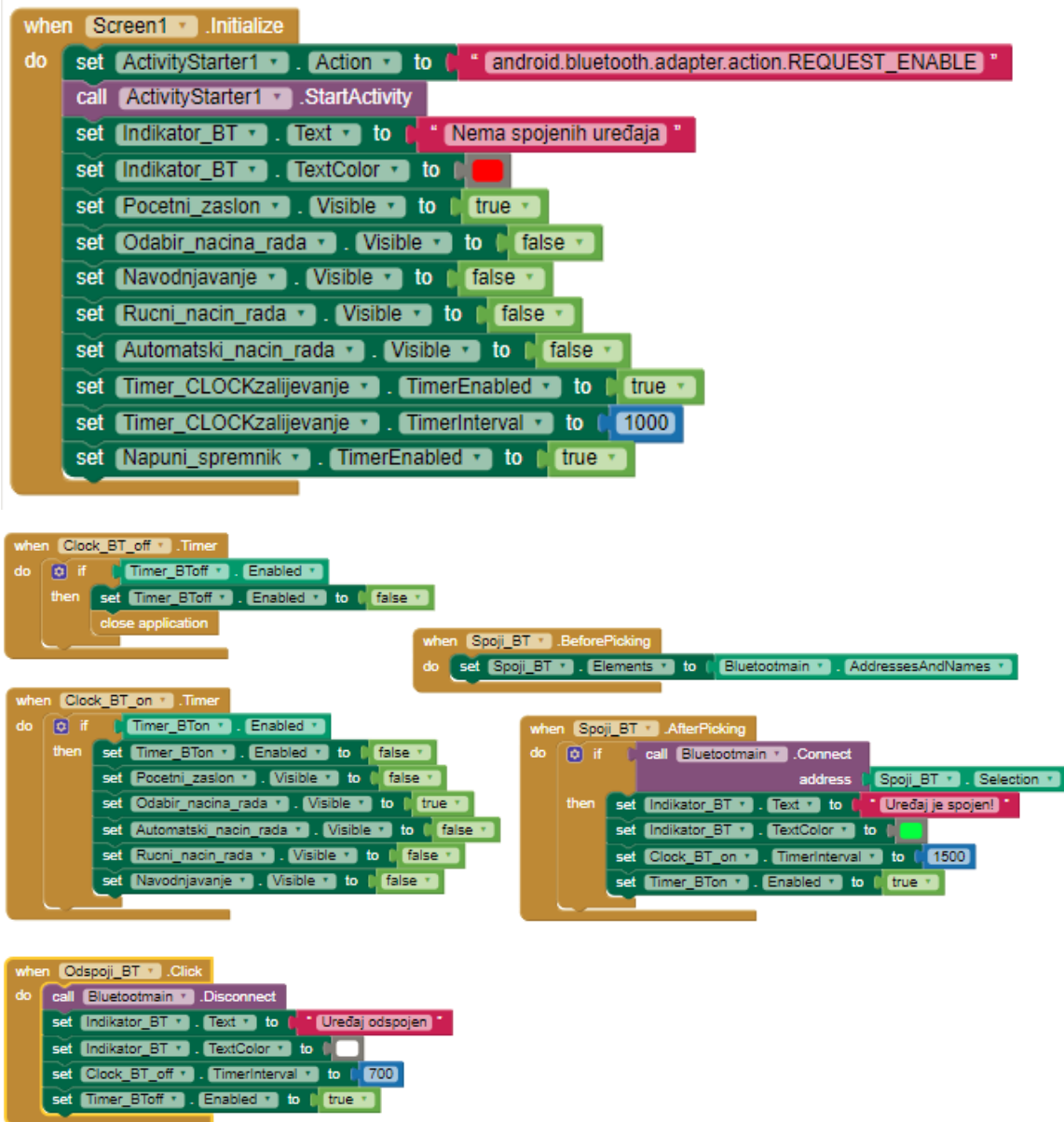
    if(digitalRead(plovak1)==LOW){
        digitalWrite(LEDv1,HIGH);
        digitalWrite(LEDv2,LOW);
        digitalWrite(LEDpumpa,HIGH);
    }

    if((digitalRead(plovak1)==HIGH)and(digitalRead(plovak2)==LOW)){
        digitalWrite(LEDpumpa,HIGH);
    }

    if((digitalRead(plovak1)==HIGH)and(digitalRead(plovak2)==HIGH)){
        digitalWrite(LEDv2,HIGH);
        digitalWrite(LEDv1,LOW);
    }
}
delay(500);
}

```


Prilog 2. – Blok dijagram programa aplikacije



```

when Automatski_nacin .Click
do
set Pocetni_zaslon .Visible to false
set Odabir_nacina_rada .Visible to false
set Automatski_nacin_rada .Visible to true
set Ručni_nacin_rada .Visible to false
set Navodnjavanje .Visible to false
set upaliautom .Visible to true
set Ugasiautom .Visible to false

```

```

when Navodnjavanje_butun .Click
do
set Pocetni_zaslon .Visible to false
set Odabir_nacina_rada .Visible to false
set Automatski_nacin_rada .Visible to false
set Ručni_nacin_rada .Visible to false
set Navodnjavanje .Visible to true
set Odabir_nacina_navodnjavanja .Visible to true
set Timer_navodnjavanja .Visible to false
set timer_2kanal .Visible to false

```

```

when Nzd_Pocetni_zaslon .Click
do
set Pocetni_zaslon .Visible to true
set Odabir_nacina_rada .Visible to false
set Automatski_nacin_rada .Visible to false
set Ručni_nacin_rada .Visible to false
set Navodnjavanje .Visible to false

```

```

when Nzd_Automatski .Click
do
set Pocetni_zaslon .Visible to false
set Odabir_nacina_rada .Visible to true
set Automatski_nacin_rada .Visible to false
set Ručni_nacin_rada .Visible to false
set Navodnjavanje .Visible to false

```

```

when Ručni_nacin .Click
do
set Pocetni_zaslon .Visible to false
set Odabir_nacina_rada .Visible to false
set Automatski_nacin_rada .Visible to false
set Ručni_nacin_rada .Visible to true
set Navodnjavanje .Visible to false

```

```

when Nzd_navodnjavanje .Click
do
set Pocetni_zaslon .Visible to false
set Odabir_nacina_rada .Visible to true
set Automatski_nacin_rada .Visible to false
set Ručni_nacin_rada .Visible to false
set Navodnjavanje .Visible to false

```

```

when upaliautom .Click
do
call Bluetoothmain .SendText
text 7
set upaliautom .Visible to false
set Ugasiautom .Visible to true

```

```

when Ugasiautom .Click
do
call Bluetoothmain .SendText
text 10
set upaliautom .Visible to true
set Ugasiautom .Visible to false

```

```

when Upali_navodnjavanje .Click
do
set Timer_navodnjavanja .Visible to false
set timer_2kanal .Visible to false
set Status_navodnjavanja .Text to NAVODNJAVANJE1 UPALJENO
call Bluetoothmain .SendText
text 2

```

```

when Ugasi_navodnjavanje .Click
do
set Timer_navodnjavanja .Visible to false
set timer_2kanal .Visible to false
set Status_navodnjavanja .Text to NAVODNJAVANJE1 UGAŠENO
set Status_navod_2kanal .Text to NAVODNJAVANJE2 UGAŠENO
set Timer_CLOCKzaljivanje .TimerEnabled to false
call Bluetoothmain .SendText
text 0
call Bluetoothmain .SendText
text 1

```

```

when Timer_navodnjavanje .Click
do
set Timer_navodnjavanja .Visible to true
set timer_2kanal .Visible to true

```

```

when Upali_nav2 .Click
do
set Timer_navodnjavanja .Visible to false
set timer_2kanal .Visible to false
set Status_navod_2kanal .Text to NAVODNJAVANJE2 UPALJENO
call Bluetoothmain .SendText
text 3

```

```

when Nzd_ručni .Click
do
set Pocetni_zaslon .Visible to false
set Odabir_nacina_rada .Visible to true
set Automatski_nacin_rada .Visible to false
set Ručni_nacin_rada .Visible to false
set Navodnjavanje .Visible to false

```

```

when Timer_CLOCKzajevanje .Timer
do
  set global vrijeme_sys to call (Vrijeme_sustava) .Now
  if
    get (global minute_pocetka) = call (Vrijeme_sustava) .Minute
    and
    get (global sati_pocetka) = call (Vrijeme_sustava) .Hour
    and
    get (global vrijeme_sys)
  then
    set (Status_navodjavanja) .Text to "NAVODJAVANJE1 UPALJENOI"
    call (Bluetoothmain) .SendText
    text "2"
  else if
    get (global minute_zavrseka) = call (Vrijeme_sustava) .Minute
    and
    get (global sati_zavrseka) = call (Vrijeme_sustava) .Hour
    and
    get (global vrijeme_sys)
  then
    set (Status_navodjavanja) .Text to "NAVODJAVANJE1 UGASENOI"
    call (Bluetoothmain) .SendText
    text "0"
  else if
    get (global minute_pocetka2) = call (Vrijeme_sustava) .Minute
    and
    get (global sati_pocetka2) = call (Vrijeme_sustava) .Hour
    and
    get (global vrijeme_sys)
  then
    set (Status_navod_ikana) .Text to "NAVODJAVANJE2 UPALJENOI"
    call (Bluetoothmain) .SendText
    text "3"
  else if
    get (global minute_zavrseka2) = call (Vrijeme_sustava) .Minute
    and
    get (global sati_zavrseka2) = call (Vrijeme_sustava) .Hour
    and
    get (global vrijeme_sys)
  then
    set (Status_navod_ikana) .Text to "NAVODJAVANJE2 UGASENOI"
    call (Bluetoothmain) .SendText
    text "1"

```

```

when NAPAJANJE_IZ_VODOVODA .Click
do
  call (Bluetoothmain) .SendText
  text "4"

```

```

when NAPAJANJE_IZ_STERNE .Click
do
  call (Bluetoothmain) .SendText
  text "5"

```

```

when NAPUNI_STERNU .Click
do
  if (Napuni_spremnik) .TimerEnabled
  then
    call (Bluetoothmain) .SendText
    text "6"
    set (Napuni_spremnik) .TimerEnabled to false
    set (NAPUNI_STERNU) .TextColor to green
  else
    call (Bluetoothmain) .SendText
    text "8"
    set (Napuni_spremnik) .TimerEnabled to true
    set (NAPUNI_STERNU) .TextColor to grey

```

```
initialize global minute_pocetka to 0
initialize global sati_pocetka2 to 0 pocetka to 0
initialize global minute_pocetka2 to 0
initialize global sati_zavrsetka to 0
initialize global minute_zavrsetka to 0
initialize global sati_zavrsetka2 to 0
initialize global vrijeme_sys to 0
initialize global minute_zavrsetka2 to 0
```

```
when Početak_Navodnjavanja .AfterTimeSet
do set global minute_pocetka to Početak_Navodnjavanja . Minute
set global sati_pocetka to Početak_Navodnjavanja . Hour
set Timer_CLOCKzalijevanje . TimerEnabled to true
```

```
when POČETAK_NAV2 .AfterTimeSet
do set Timer_CLOCKzalijevanje . TimerEnabled to true
set global minute_pocetka2 to POČETAK_NAV2 . Minute
set global sati_pocetka2 to POČETAK_NAV2 . Hour
```

```
when KRAJ_NAV2 .AfterTimeSet
do set Timer_CLOCKzalijevanje . TimerEnabled to true
set global minute_zavrsetka2 to KRAJ_NAV2 . Minute
set global sati_zavrsetka2 to KRAJ_NAV2 . Hour
```

```
when Kraj_navodnjavanja .AfterTimeSet
do set Timer_CLOCKzalijevanje . TimerEnabled to true
set global minute_zavrsetka to Kraj_navodnjavanja . Minute
set global sati_zavrsetka to Kraj_navodnjavanja . Hour
```