# IEEESTEC

## 10th Student projects conference

Organized By:
EESTEC LC Niš
IEEE Student Branch Niš
Faculty of Electronic Engineering Niš

With the cooperation of:
IEEE Serbia and Montenegro section
IEEE Electron Devices/Solid-State Circuits Chapter
IEEE Microwave Theory and Techniques Chapter
Serbian Academy of Sciences and Arts (SASA), Branch Niš

ISBN: 978-86-6125-193-1

**Publisher:**
Faculty of Electronic Engineering, Niš
P.O.Box 73, 18000 Niš
http://www.elfak.ni.ac.rs

**Chairman:**
Danijel Danković

**Vice Chairman:**
Dušan Vučković
Miloš Marjanović

**Editors of Proceedings:**
Ninoslav Stojadinović
Vera Marković
Danijel Danković
Zlatica Marinković

**Secretary:**
Nikola Simić
Dragana Dimitrijević
Đorđe Veličković
Miloš Mitić
Nikola Krstić
Milan Stojanović
Nikola Mitrović
Nikola Radisavljević
Milan Ivković
Marija Živković
Filip Đorović
Stefan Jovanović
Miljan Jerotijević
Anđela Kocić
Jana Vračar

**Technical Editors:**
Miloš Marjanović
Jana Vračar
Danijel Danković

CIP - Каталогизација у публикацији - Народна библиотека Србије, Београд

621.3(082)
004(082)

IEEESTEC Student Project Conference (10 ; 2017 ; Niš)
IEEESTEC 10th Student Project Conference, [Niš, 2017] / organized by EESTEC LC, Niš ... [et al.] ; [editors Ninoslav Stojadinović ... et al.]. - Niš : Faculty of Electronic Engineering, 2017 (Niš : Unigraf). - 314 str. : ilustr. ; 30 cm

Kor. nasl. - Radovi na srp. i engl. jeziku. - Tekst štampan dvostubačno. - Tiraž 100. - Bibliografija uz svaki rad.

ISBN 978-86-6125-193-1

1. Stojadinović, Ninoslav, 1950- [уредник] 2. Institut inženjera elektrotehnike i elektronike. Studentski ogranak (Niš)
a) Електротехника - Зборници b) Рачунарство - Зборници
COBISS.SR-ID 249967884

**Printed by:** „Unigraf“, Niš, 2017.
**Circulation:** 100 copies

**Reviewers:**

1. Vladimir Simić
2. Goran Nikolić
3. Aleksandar Jocić
4. Nikola Stojanović
5. Ivica Marković
6. Goran Stančić
7. Milena Frtunić Gligorijević
8. Marija Veljanovski
9. Ljubomir Vračar
10. Milan Simić
11. Miona Andrejević Stošović
12. Vojkan Davidović
13. Aleksandar Milenković
14. Sanja Aleksić
15. Marija Milijić
16. Zlatica Marinković
17. Saša Nikolić
18. Emilija Živanović
19. Igor Jovanović
20. Vesna Paunović
21. Vladimir Mitić
22. Miloš Petković
23. Aleksandar Veljanovski
24. Aleksandar Dimitrijević
25. Miloš Bogdanović
26. Vladan Mihajlović
27. Vladimir Ćirić
28. Miloš Marjanović
29. Dejan Mirković
30. Nikola Simić
31. Ivan Anastasijević
32. Milan Radić
33. Milica Rašić
34. Vladica Đorđević
35. Filip Filipović
36. Nikola Danković
37. Dejan Milić
38. Milić Pejović
39. Miloš Đorđević
40. Oliver Vojinović
41. Sandra Đošić
42. Danijel Danković
43. Marko Dimitrijević
44. Aleksandra Đorić
45. Branislav Ranđelović
46. Stefan Ilić
47. Nenad Milošević
48. Mirjana Perić
49. Jana Vračar
50. Milan Ivković
51. Aleksandra Stojković
52. Tijana Dimitrijević
53. Jelena Anastasov
54. Milan Stojanović
55. Milan Dinčić
56. Miloš Radmanović
57. Nikola Radisavljević
58. Milivoje Mihajlović
59. Nikola Stojanović

**Table of contents:**


Deset godina Konferencije studentskih projekata IEEESTEC 1
Modifikacija algoritma za sortiranje na osnovu stabla koji koristi pristup baziran na adresi 5
Synthesizing Simulated Electrical Appliances Consumption for Typical Serbian Household 11
Merenje sobne temperature i vlažnosti vazduha pomoću Arduino Uno platforme 15
Aural Immersive Virtual Reality for Neuro-Developmental Disorder Therapy Based... 19
Tool for Modelling and Automatic Deployment of Data Intensive Applications 25
Senzor za merenje magnetne indukcije 31
Meteorološka stanica – Data Logger 35
M2M uređaj za merenje temperature i vlažnosti vazduha 39
Akustična sklopka 45
Mobilna android aplikacija za pomoć pri vožnji na autoputevima u Srbiji 51
Laboratorijska maketa za proučavanje kretanja tela po strmoj ravni zasnovana na Raspberry... 55
Analiza paketskog saobraćaja taktičke VVF radio mreže 59
Poređenje vremena izvršenja funkcije kloniranja objekata bez upotrebe i sa upotrebom refleksije... 63
Sistem za kontrolu garaže 67
Digitalni termometar realizovan na principu analogno-digitalne konverzije 73
Projektovanje solarnog punjača za baterije 77
Praktična realizacija bezbedonosnog sistema 81
Primena žiro senzora za upravljanje elektro motorima na primeru Lego platforme 85
Android aplikacija za muzički festival “Outhide” 89
Area Calculator - Softver za merenje površine objekata na slici 93
Uređaj za hvatanje grabljivica 97
Sistem za detekciju pokreta kroz nepomične objekte 99
Koncept Pametne kuće kroz nadgledanje ambijentalnih parametara i zaštitu kuće 101
Kolaborativni sistemi za preporučivanje sadržaja i primena NHSM metrike za određivanje sličnosti 105
Uticaj geometrije krstastog silicijumskog senzora magnetnog polja na vrednosti Holovog napona 111
Autentifikacioni sistem elektronske brave 115
Elektronski istorijski vodič kroz grad Niš 121
Fooling a neural network with common adversarial noise 125
Dance style recognition using SVM Classificators 129
Praktična realizacija robota za treniranje stonog tenisa 133
Optoelektronske komponente i njihova primena 139
Vizuelizacija podataka korišćenjem R jezika 143
Kolo za merenje kapacitivnosti diskretnih kondenzatora korišćenjem timer NE555 kola i Arduino... 147
Realizacija audio efekata za gitaru korišćenjem Raspberry Pi 151
Ultrazvučni senzor parkiranja 155
Integracija sofisticirane real time web aplikacije uz pomoć soketa i kompaktne custom dizajnirane ... 159
Optički predajnik audio signala sa impulsnom širinskom modulacijom 163
Optički prijemnik impulsno-širinski modulisanih audio signala 167
Design, Simulation and Fabrication of a Microstrip Low-Pass Filter with Open-Stubs 171
Obezbeđivanje konstantnosti napajanja u postrojenjima gde je moguć paralelan rad transformatora 177
Veštačka inteligencija na primeru LEGO robota 181
VR Vožnja 185
Numerička integracija na neograničenom intervalu primenom ortogonalnih polinoma 189
Automatizacija proizvodnje u staklenicima sa Arduino platformom 193
Elektronski "peščani" sat 195
Drajver kostantne optičke snage za lasersku diodu 197
Alarmni sistem za sprečavanje krađe na poljoprivrednim parcelama 201
Projektovanje i realizacija sistema za merenje napona 203
Primena Arduino platforme u sistemu za navodnjavanje 207
Sistem za testiranje vida i određivanje potrebne dioptrije korekcionog sredstva - OPTIS 211
Realizacija jednog ASK/PSK modulatora 217
Uređaj za inteligentno upravljanje radom električne ograde 221
Brojač realizovan korišćenjem senzora prolaza 223
Efekat primene visokog napona na asimetrični kondenzator 227
Solarna meteorološka stanica 231
Multifunkcionalni časovnik 235
Daljinska kontrola i nadgledanje “pametne” kuće 239
Ispitivanje uticaja debljine tunel oksida kod Flash EEPROM ćelije pomoću TCAD alata 243
DTW algoritam u formi glasovnog prepoznavanja 247

Razvoj Pygame aplikacije na Raspberry Pi-u 251
Sistem za daljinsko upravljanje 255
Život i studije u Lisabonu 259
Automatizirani sustav navodnjavanja upravljan mikrokontrolerom 263
Linearni modulator laserske diode 267
Duplo klatno 271
Sistem pametne kuće 275
Operacioni pojačavač u konfiguraciji diferencijatora i integratora 279
Vektorsko upravljanje mašinom sa stalnim magnetima 283
Real-Time Analysis of Sunflower Pollination Levels 287
Prošireno realna i virtuelno realna 10. IEEESTEC konferencija 291
All Sky Cloudiness Estimator 297
Sistem za udaljeni nadzor pacijenata pomoću pametnog sata 301
Erasmus+ student mobility between the University of Niš and the Technical University of Sofia 307
Multimedijalni sistem informisanja učenika škole 311

# Deset godina Konferencije studentskih projekata IEEESTEC

Miloš Marjanović, Danijel Danković

*Sadržaj* – U ovom radu opisan je razvoj IEEESTEC konferencije studentskih projekata od nastanka do danas. Predstavljeni su ciljevi studentske konferencije i načini motivacije studenata da se bave naučno-istraživačkim radom i pisanjem radova. Predstavljeni su relevantni podaci koji pokazuju razvoj i napredak konferencije.

## I. UVOD

IEEESTEC konferenciju studentskih projekata organizuju IEEE studentski ogranak Niš [1], EESTEC lokalni komitet Niš [2] i Elektronski fakultet, Univerziteta u Nišu [3] pod pokroviteljstvom društva IEEE Electron Devices (EDS)/ Solid-State Circuits (SSC), društva IEEE Micorwave Theory and Techniques (MTT), sekcije IEEE Srbija i Crna Gora [4] i Srpske akademije nauka i umetnosti (SANU) [5]. Konferencija je nastala kao rezultat saradnje IEEE studentskog ogranka Niš i EESTEC lokalnog komiteta Niš, kao što je jasno iz imena konferencije (IEEE + EESTEC = IEEESTEC) (slika 1).




Slika 1. Logoi IEEE, IEEESTEC i EESTEC-a.

IEEE (Institute of Electrical and Electronics Engineers) je najveće svetsko udruženje posvećeno unapređivanju tehnoloških inovacija vezanih za elektronske uređaje i tehnološkom razvoju. IEEE ima više od 423.000 članova u 160 zemalja sveta, od toga 117.000 studenata. Veliki broj članova učestvuje na preko 1300 IEEE konferencija u 80 zemalja širom sveta. Važno je napomenuti da je jedan od glavnih ciljeva IEEE saradnja sa sličnim udruženjima. Na nivou sekcije Srbija i Crna Gora pripremljen je memorandum o saradnji IEEE-a i EESTEC-a.

EESTEC je međunarodna studentska organizacija koja okuplja studente elektrotehnike i računarstva i ima vrlo slične ciljeve kao IEEE. Glavni cilj EESTEC-a je da podrži razvoj međunarodnih kontakata između studenata i stručnjaka. Razmena ideja i iskustava među studentima ostvaruje se kroz radionice, razmene, konferencije, seminare i druge aktivnosti. Trenutno, postoji oko 50 EESTEC lokalnih komiteta u više od 20 evropskih zemalja sa više od 3000 članova. Saradnja između IEEE studentskog ogranka Niš i EESTEC loklalnog komiteta Niš je vidljiva na primeru organizacije IEEESTEC konferencije.

Miloš Marjanović je asistent na Katedri za mikroelektroniku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: milos.marjanovic@elfak.ni.ac.rs

Dr Danijel Danković je docent na Katedri za mikroelektroniku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: danijel.dankovic@elfak.ni.ac.rs

## II. O IEEESTEC KONFERENCIJI

### *A. Ciljevi IEEESTEC konferencije*

Dobro je poznato da su glavni razlozi za pisanje rada i prisustvovanje konferenciji da se čuju prezentacije drugih radova i razmene mišljenja i ideje sa ostalim istraživačima, učesnicima konferencije. Međutim, ciljevi studentske konferencije su daleko složeniji. Osnovna ideja, na početku, je bila da studenti osnovnih akademskih studija imaju mogućnost da nauče kako se piše naučno-istraživački rad. Za najveći broj autora ovo je prvi rad, što je i odlična prilika za promociju IEEE šablona za pripremu rada. Drugi cilj je javna prezentacija projekata koje su radili učenici srednjih škola (elektrotehničkih smerova i gimnazije), kao i studenti tehnike. Naime, poznato je da veoma važni projekti često ostaju nezapaženi ili nikad ne napreduju dalje od početne faze istraživanja. Prezentacija realizovanih projekata na dan konferencije i objavljivanje članka u zborniku radova sa konferencije značajno doprinose promociji naučno-istraživačkih projekata studenata. Treći cilj je omogućiti direktnu komunikaciju između učesnika konferencije. Učesnici imaju mogućnost da razmene mišljenja sa učenicima, studentima i akademskim osobljem (asistentima i profesorima). Tom prilikom mogu da dobiju nove ideje za istraživanja i sugestije kako unaprediti trenutni projekat. Takođe, učesnici se mogu informisati o tome šta učenici i studenti iz drugih srednjih škola i fakulteta rade, ne samo vezano sa usku temu projekta, već i mnogo šire. Konačno, studenti počinju da grade veze sa drugim istraživačima u svojoj oblasti istraživanja. IEEESTEC konferencija promoviše timski rad. Studentski timovi predlažu, razvijaju i prezentuju projekte iz različitih oblasti istraživanja (elektronike, mikroelektronike, kompjuterskih nauka, telekomunikacija, upravljanja sistemima...). Ova konferencija je idealna mogućnost za promociju tehničkih nauka. Naime, učenici srednjih škola prisustvuju konferenciji i imaju mogućnost da vide sve projekte. Na taj način, može se povećati njihovo interesovanje za tehničke nauke i mogu se javiti ideje za njihova sopstvena istraživanja. Na kraju, IEEESTEC

konferencija je odlična prilika da predstavnicima malih kompanija u regionu da jasnu sliku o istraživanjima i studentskim projektima. Na taj način kompanije mogu sarađivati sa autorima projekata koji su interesantni za njihovu oblast rada. Ovo je najbrži način prenosa znanja sa univerziteta ka malim kompanijama. Veliki broj učesnika konferencije i predstavnika kompanija je osetio prednost ove mogućnosti.

*B. Motivacija studenata za učešće na IEEESTEC konferenciji*

Organizatorima je veoma teško reći: "Treba da napišete rad za IEEESTEC konferenciju". Veoma je važno da student nezavisno donese odluku da napiše rad. Ovo ukazuje da oni moraju biti veoma motivisani. Zavisno od interesovanja studenta, postoji širok opseg različitih motivacija za pisanje rada i učešće na IEEESTEC konferenciji. Prvo, prethodno navedeni ciljevi IEEESTEC konferencije su veoma važan razlog za studente da napišu radove.

Drugo, najbolji radovi se nagrađuju. Na osnovu ocena recenzenata nominuju se tri najbolja rada. Organizatori uz pomoć prijatelja konferencije obezbeđuju veliki broj nagrada za autore najboljih radova (detalji sa dodele nagrada na slici 2). Radovi koji su dobili najveće ocene recezenata, i kao nagrađeni radovi, prezentuju se usmeno u okviru svečanosti otvaranja konferencije (islustracija slika 3). Svi ostali radovi prezentuju se preko postera, dok studenti sa praktičnim projektima demonstriraju svoje radove (ilustracija slika 4). Organizacioni odbor konferencije bira i tri rada sa najboljom praktičnom realizacijom, koji takođe dobijaju nagrade.



Slika 2. Dodela nagrada nagrađenim autorima.

Učešće na konferenciji može doprineti poboljšanju biografije kandidata i predstavlja odličnu referencu za budući posao studenata. Ovo je posebno bitno, ukoliko student bude nagrađen na konferenciji. Takođe, jedna od nagrada na konferenciji je nominacija rada da predstavlja studentski ogranak na svetskom takmičenju IEEE Region 8 Student Paper Contest [6]. Važno je napomenuti da samo članovi-studenti IEEE mogu učestvovati na ovom takmičenju. Dodeljuje se i specijalna nagrada IEEE MTT autorima radova iz oblasti telekomunikacija. IEEE Women in Engineering (WIE) grupa iz IEEE Niš dodeljuje specijalnu nagradu ženskim autorima radova. Godine 2016. ustanovljena je specijalna nagrada za rad sa najvećim inovacionim potencijalom u saradnji sa NiCAT klasterom naprednih tehnologija.



Slika 3. Usmena prezentacija nagrađenih radova.



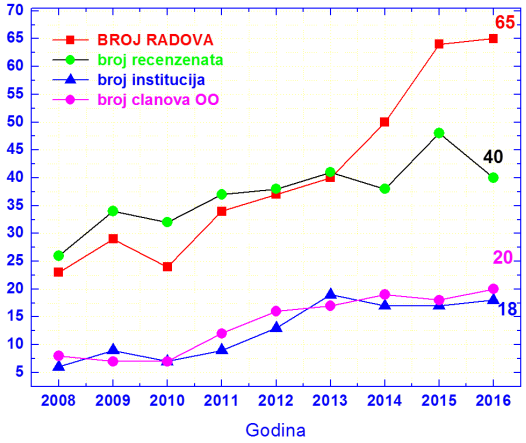
Slika 4. Demonstracija praktično realizovanih projekata.

*C. Napredak IEEESTEC konferencije*

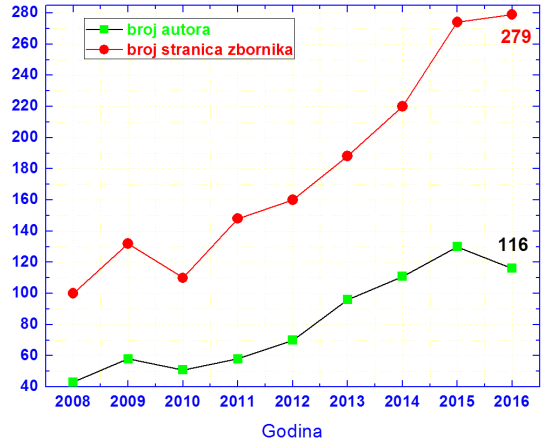
Značajnim angažovanjem organizatora konferencije ona se iz godine u godinu poboljšava i napreduje. Konferencija se organizuje u okviru obeležavanja dana Elektronskog fakulteta, tako da je veliki broj gostiju Elektronskog fakulteta prisutan na konferenciji, a i učesnici konferencije uveličavaju proslavu praznika fakulteta. Takođe, IEEESTEC konferencija je deo godišnjih aktivnosti IEEE sekcije Srbija i Crna Gora. Uz podršku Elektronskog fakulteta, IEEE sekcije Srbija i Crna Gora, i iskustvo članova organizacionog odbora, konferencija beleži trend napredovanja, što je jasno na osnovu parametara datih u Tabeli 1. Kao što se može videti na slikama 5 i 6, trend napretka je jasno vidljiv. Svi analizirani parametri pokazuju značajan porast, od 53 % (broj recenzenata) do 200 % (broj institucija odakle dolaze učesnici) u periodu od 2008. do 2017. godine. Glavni cilj oranizatora konferencije je da se ovaj trend napretka nastavi i održi.

TABELA I
IEEESTEC KONFERENCIJA U BROJEVIMA

| Godina | Radovi | Autori | Recenzenti | Stranice zbornika | Insistucije | Organizatori |
|---|---|---|---|---|---|---|
| 2008. | 23 | 43 | 26 | 100 | 6 | 8 |
| 2009. | 29 | 58 | 34 | 132 | 9 | 7 |
| 2010. | 24 | 51 | 32 | 110 | 7 | 7 |
| 2011. | 34 | 58 | 37 | 148 | 9 | 12 |
| 2012. | 37 | 70 | 38 | 160 | 13 | 16 |
| 2013. | 40 | 96 | 41 | 196 | 19 | 17 |
| 2014. | 50 | 111 | 38 | 218 | 17 | 19 |
| 2015. | 64 | 130 | 48 | 282 | 17 | 18 |
| 2016. | 65 | 116 | 40 | 279 | 18 | 20 |
| 2017. | 75 | 151 | 59 | 314 | 22 | 20 |




Slika 5. Broj radova, recenzenata, institucija i članova organizacionog odbora IEEESTEC konferencije tokom godina.




Slika 6. Broj autora i stranica zbornika IEEESTEC konferencije tokom godina.

*D. Druge bitne informacije o IEEESTEC konferenciji*

Osnovna ideja je da konferencija iz godine u godinu bude interesantnija. Tako da su organizatori ustanovili više nagrada. Autori radova biraju najbolji rad korišćenjem aplikacije koju je za potrebe konferencije izradio VTŠ Apps tim [7]. Autor rada koji sakupi najveći broj glasova tradicionalno dobija knjigu „101 heroj svetske istorije“. Takođe, nekoliko godina dodeljivana je nagrada za autore radova koji su dobili najviše glasova u online glasanju. Godine 2012. više od 24.000 posetlica posetilo je zvanični sajt Konferencije i glasalo za najbolji rad.

Vrlo je važno istaći da profesori i asistenti Elektronskog fakulteta u Nišu daju ogroman doprinos konferenciji kroz mentorstva i recenzije radova. Preporučljivo je da svaki autor ima mentora. Naime, najveći broj autora piše svoj prvi rad, pa je potrebno da imaju mentora da im pomogne u rešavanju problema i da smernice u istraživanju. Za svaki rad obezbeđeno je minimum troje recenzenata.

Sve konferencije bile su posvećene nekom događaju. Prva konferencija je bila posvećena saradnji između IEEE studentskog ogranka Niš i EESTEC lokalnog komiteta Niš. Druga je bila posvećena 125. godišnjici IEEE, treća – 50 godina Elektronskog fakulteta Niš, četvrta je bila posvećena 40. godišnjici IEEE sekcije Srbija i Crna Gora. Peta je bila godišnjica same konferencije, pet godina IEEESTEC konferencije. Cilj organizatora je da daju svoj mali doprinos obeležavanju ovih važnih godišnjica i praznika. Članci o svim događajima vezanim za konferenciju objavljeni su u časopisima i žurnalima: IEEE Solid-State Circuits Magazine, IEEE Electron Devices Newsletters i u IEEE Region8News.

## III. ZAKLJUČAK

Svi navedeni ciljevi IEEESTEC konferencije studentskih projekata se iz godine u godinu ostvaruju i dolazi do napretka, što jasno pokazuju statistički podaci predstavljeni u radu. Organizacioni odbor će i dalje raditi i uvoditi novine kako bi konferencija bila interesantnija i učenicima srednjih škola i studentima. Iako je konferencija popularna što ilustruje slika 7, radiće se i dalje na promociji i popularizaciji naučno-istraživačkog rada i pisanja radova.




Slika 7. Učesnici IEEESTEC konferencije.

## ZAHVALNICA

Ovaj tekst je nastao kao deo regularnih aktivnosti IEEE studentskog ogranka Univerziteta u Nišu.

## LITERATURA

[1] IEEE studentski ogranak Niš. Dostupno na: http://ieee.elfak.ni.ac.rs/
[2] EESTEC lokalni komitet Niš. Dostupno na: http://www.eestec.rs/
[3] Elektronski fakultet Niš. Dostupno na: http://www.elfak.ni.ac.rs/cir/
[4] IEEE sekcija Srbija i Crna Gora. Dostupno na: http://www.ieee.uns.ac.rs/
[5] Srpska akademija nauka i umetnosti, ogranak u Nišu. Dostupno na: https://www.sanu.ac.rs/Clanstvo/Nis.aspx
[6] Student Paper Contest. Dostupno na: http://www.ieeer8.org/category/student-activities/awards-and-contests/student-paper-contest/
[7] VTŠ Apps Team. Dostupno na: http://vtsappsteam.edu.rs/

# Modifikacija algoritma za sortiranje na osnovu stabla koji koristi pristup baziran na adresi

Jovan Milojković

*Sadržaj* – U ovom radu može se pogledati modifikacija algoritma za sortiranje brojeva koji koristi: n-arna stabla kao strukturu za smeštanje podataka, sortiranje bazirano na adresi, i N-arnog K-koda za pakovanje brojeva. Modifikacija bi trebalo dati poboljšanja performansi sekvencijalne i paralelne implementacije nemodifikovanog algoritma.

## I. DEFINICIJE

Na početku, biće date definicije pojmova koji su korišćeni u radu.

Sortiranje se može definisati na sledeći način [1]. Neka je ulaz uvek skup $X = \{x_1, x_2, x_3 \dots x_n\}$ ključeva koji pripadaju linearno uređenom skupu proizvoljne veličine i dozvoljene operacije su operacije poređenja dva elementa kao i osnovne operacije *fetch* i *store*. Problem sortiranja predstavlja problem pronalaženja permutacije $\pi = \{1 \dots n\}$ takve da važi $x_{\pi(1)} \leq x_{\pi(2)} \leq x_{\pi(3)} \leq \cdots x_{\pi(n)}$, ili ispisati sortiranu sekvencu u n izlaznih registara, gde je $x_{\pi(k)}$ k-ti najmanji element polaznog skupa X ili element iz X ranka k. Problem selekcije je sličan problemu sortiranja s tim što kao ulaz pored skupa X ima i broj k, gde je zadatak pronaći $\pi(k)$ ili $x_{\pi(k)}$.

Graf G je par (V,E), gde je V skup čvorova, a E je skup veza između čvorova. Put u grafu predstavlja podgraf tog grafa koji je put. Ciklus u grafu je podgraf koji je ciklus [2].

Stablo je povezan graf G koji je acikličan, to jest koji ne sadrži cikluse [2],[3],[4],[5]. Binarno stablo je stablo za koje važi da je svaki čvor stabla povezan sa najviše dva druga čvora. N-arno stablo je stablo za koje važi da je svaki čvor stabla povezan sa najviše n drugih čvorova. Dalje može se i definisati N-arno stablo gde je N ne konstanta, nego vektor čime dobijamo stablo koje u zavisnosti od nivoa ima različite faktore granjanja čvora.

U literaturi se može naći pod nazivom kod koji koristi K manje bitnih bitova broja [6], [7] a na dalje u radu će se zvati K-kod. Kod koji posmatra K manje bitnih bitova broja se sastoji u tome, kako mu i samo ime kaže, da k manje bitnih bitova se predstavi kao broj koji ima $2^k$ bita. Na primeru će biti objašnjena upotreba K-koda. Posmatrajmo četvorobitne brojeve i četvorobitni K-kod. Neka su nam dati npr. broj 7 i broj 13. Kako je rečeno broj bitova koji nam treba za pamćenje brojeva u 4 K-kodu, broj bitova je 16. Broj 7 se predstavlja kao 0000000010000000, a broj 13 kao 0010000000000000. Ova dva broja možemo pamtiti u jednoj memorijskoj jedinici kao 0000000010000000 ∨ 0010000000000000 = 0010000010000000. Može se izvući generalna definicija, da se broj pamti kao postojanje jedinice na i-tom mestu u kodu i da to postojanje označave broj i. Odavde se vidi i da je nedostatak korišćenja ovakvog koda taj da u jednoj memorijskoj jedinici kada pamtimo broj 7 možemo ga jednom zapamtiti. A šta ako ga treba zapamtiti više puta, tj. zapamtiti da se broj sedam više puta javio? Dobra stvar kod K-koda je ta što se korišćenjem soritrajučih mreža [3],[8] može dobiti K-kod od broja i obrnuto za kratko vreme. Kombinatorna kola [6] se mogu iskoristiti za implementaciju sortirajučih mreža, gde se lako može i unapred odrediti granica broja k [9], a u [6] kažu da bi broj k trebao biti između 2 i 4 zbog složenosti samog kombinatornog kola.

U slučaju javljanja jedno te istog broja više puta može se koristiti N-arni K-kod [6]. Kod ovog koda potreban broj cifara za pamćenje k-tobitnih brojeva je $n * 2^k$. Običan K-kod je u stvari N-arni K-kod gde je n=1. Ovaj kod se može koristiti ako je potrebno pamćenje svih ponavljanja nekog broja, ali broj ponavljanja mora biti manji od $2^n$.Vratimo se na prethodni primer. Pamtimo četvorobitne brojeve 7 i 13, koristeći N-arni K-kod gde je n=2, odavde proizilazi da će nam trebati 32 bita za pamćenje četvorobitnih brojeva. Za broj 7 kod koji se dobija je 00-00-00-00-00-00-00-00-01-00-00-00-00-00-00-00, a broj 13 bi bio 00-00-01-00-00-00-00-00-00 -00-00-00-00-00-00-00, dok oba upakovana 00-00-01-00-00-00-00-00-01-00-00-00-00-00-00-00.

## II. SORTIRANJE BAZIRANO NA ADRESI I K-KODU

Prvi pristup je taj da nam sam broj označava adresu na kojoj treba izvršiti nešto. Na početku, celu memoriju popuniti nulama. Kako koji broj dođe tako na toj adresi inkrementirati datu vrednost. Ovaj način sam po sebi jeste brz dok se struktura memorije popunjava, ali ima mnoge nedostatke. Vađenje elemenata iz ove strukture zahteva pretraživanje cele memorije što je sa pogleda performansi sporo. Dalje, mnogo mesta ostaće prazno. Onda veličina memorije koja je potrebna za ovakav način rada je velika. Radi ilustracije daćemo primer koliko je ovakav način sortiranja loš po pogledu memorije. Neka sortiramo brojeve veličine 32 bita i koristimo memorijske lokacije od 8 bita,

Jovan Milojković je diplomirani inžinjer računarstva i informatike Elektronskog fakulteta, Univerzitet u Nišu, Aleksandra Medvedeva 14, 1800 Niš, Srbija, E-mail: jovanmilojkovic94@gmail.com

jednog bajta. Tada nam je potrebno 4GB memorije da bismo sortirali niz 32-bitnih brojeva bilo koje dužine. Isto se javlja problem šta ako se neki broj toliko puta javi u nizu, a mi nismo predvideli da će se toliko puta javiti. Tada nam možda mesto koje smo ostavili za inkrementiranje broja može postati malo.

Može se uočiti i velika prednost ovog sistema, a to je brzina smeštanja podataka u strukturu. Jer mi, čim nam dođe novi element, odmah znamo koje mesto u memoriji inkrementirati. Ovakav način sortiranja može se iskoristiti jedino ako je veličina niza koji se sortira velika. Upotrebom verovatnoće lako se može pokazati da ako imamo Gausovu raspodelu brojeva u nizu, sortiramo niz čiji su brojevi p-bitni, broj brojeva u nizu, n, je poželjno da bude $n \geq 2^p$ da bi zauzeće memorije bilo što efikasnije, a samim tim bismo imali i manje "rupa" prilikom čitanja iz te memorije.

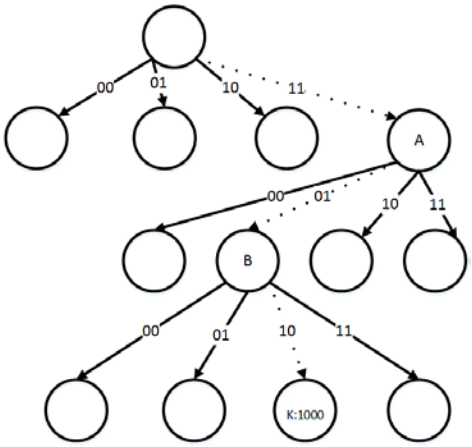
Drugi pristup je korišćenjem K-koda. Neka su brojevi koje sortiramo M-bitni. M-K viših bita broja nam označava lokaciju na kojoj treba upisati K-kod od preostalih k bita datog broja. Ovaj pristup je bolji jer povećavamo lokalnost. Dobre i loše strane ovog pristupa iste su kao kod prethodnog. Realne karakteristike prilikom sortiranja koristeći ovakav pristup mogu se videti u radu [6]. Takođe treba obratiti pažnju na veličinu broja k kod K-koda, deo koji je naveden ranije u radu.

Na kraju može se koristiti i N-arni K-kod, da bi pamtili i brojeve i njihova ponavljanja.

## III. SORTIRANJE KORISTEĆI N-ARNA STABLA BAZIRANO NA ADRESI I K-KODU (NEMODIFIKOVAN ALGORITAM)

Isto kao i u prethodnom poglavlju, ideja je iz rada [6] a proistekla iz rada [10]. U ovom pristupu takođe koristimo K-kod za smeštanje brojeva. Može se koristiti i N-arni K-kod. Svi elementi se pamte u listovima stabla. Sam put od korena stabla pa do lista predstavlja više bitove broja dok se K nižih bita pamti u obliku K-koda u listu. Ova metoda je poznata kao i *tree-walk-table* pristup rešavanju problema.

U uvodu, ovog dela, smo dali neke najkraće karakteristike, a sada ćemo detaljnije opisati sam algoritam. Neka su brojevi koje sortiramo M-bitni. Unapred odredimo broj K za K-kod koji ćemo koristiti. Ostalo je M-K bitova. Oni se koriste za definisanje N-arnog stabla. To N-arno stablo na kraju krajeva ne mora biti N-arno stablo gde je N konstantno, nego N može biti vektor, čime se omogućuje na različitim nivoima različito granjanje. Faktor granjanja stabla se može odrediti odmah na početku, a može se i dinamički menjati. Mi ćemo ga za sada odrediti odmah na početku, a posle ćemo reći nekoliko rečenica o tome kako se može upotrebiti različit faktor granjanja čvorova.




Slika 1. Sortiranje bazirano na adresi, izgled stabla.

Pogledajmo N-arna stabla sa konstantnim faktorom granjanja N. Dodatno N treba biti stepen dvojke jer radimo sa binarnim brojevima. Neka je $G = \log_2(N)$. Tada je broj segmenata (dubina stabla) koji dobijamo ovakvim pristupom jednak $(M-K)/G$, pa prvi veći ceo broj ako se dobije razlomljeni. U slučaju razlomljenog broja može se zapaziti da će se stablo u jednom od nivoa granati različito, nego u ostalim nivoima. Čvorovi iz tog nivoa će imati manje dece. Taj nivo može biti bilo koji. Više bitove adrese delimo na $(M-K)/G$ segmenata gde svaki segment, sekvencijalno, služi za delimično izračunavanje adrese sledećeg potomka. Daćemo kratak primer da bi bilo jasnije. Neka je $M = 8, K = 2, N = 4, G = 2$. Odavde se vidi da stablo ima stepen granjanja 4 i da će do kraja taj stepen granjanja biti odžan. Uzmimo bilo koji 8-bitni broj npr. 11011000. Razdelimo taj broj 11 - 01 - 10 - 00. Zadnja dva bita služe za pravljenje K-koda od broja, dok sekvencijalno 11 - 01 - 10 služe za adresiranje sledećeg potomka u stablu. Na stablu sa slike se može uočiti kako je zamišljeno funkcionisanje sortiranja baziranog na adresi korišćenjem N-arnog stabla. Slika 1 ilustruje prethodni primer. Na Slici 1 linije koje su tačkasto označene, predstavljaju put od korena do lista gde se treba zapamtiti vrednost koja je ušla u sistem. Odavde se vidi i činjenica da se svi brojevi pamte u listovima. To znači da uvek prilikom stizanja novog broja mi moramo doći do lista i tu upisati njegov K-kod.

Veoma je bitna i memorijska prezentacija ovakvog stabla. Pretpostavimo da je memorija na početku popunjena nulama. Memorijska prezentacija lista je veoma jednostavna, jer list sadrži samo K-kod broja. Može se koristiti i N-arni K-kod. Za koren se bira neka početna adresa, neka je to adresa 0x0000. Adresa potomka koji se nalazi na nekoj grani, neka je to grana i, se dobija tako što se na početnu adresu čvora doda i. Odavde se može videti da je od početne adrese čvora potrebno zauzeti još N adresa za pamćenje adresa potomaka. Posle tih N adresa mogu se nalaziti dodatni podaci vezani za taj čvor.

Ovo je dinamičko stablo, tj. ovo stablo je zamišljeno da se dinamički gradi, da bi se uštedelo na vremenu tokom čitanja podataka iz memorije (ušteda se ogleda u tome što se neće celo stablo obići, nego samo delovi u kojima se nalaze podaci). Zbog toga dobro je, u ovakvoj arhitekturi, imati još jedan podsistem koji će vršiti samo dodelu slobodne memorije procesima koji je potraže. Lako je odrediti koliko je maksimalno moguće imati procesa u sistemu i taj broj predstavlja dubinu stabla, a dubina stabla je u suštini broj segmanata na koje delimo dobijeni binarni broj. Po našem mišljenju, celokupnu memoriju bi na startu trebalo podeliti na h delova, gde je h visina stabla, da ne bi dolazilo do konflikta prilikom dodele memorije kada je više procesa aktivno. Na Slici 2 je prikazana memorijaska prezentcija stabla, iz prethodnog primera, koje je na početku bilo prazno a onda je došao, kao i u prethodnom primeru broj 11011000.

| Adresa | Podatak | |
|---|---|---|
| 0x0000 | - | Koren |
| 0x0001 | - | |
| 0x0002 | - | |
| 0x0003 | 0x0004 | |
| 0x0004 | - | Čvor A |
| 0x0005 | 0x0008 | |
| 0x0006 | - | |
| 0x0007 | - | |
| 0x0008 | - | Čvor B |
| 0x0009 | - | |
| 0x000A | 0x000C | |
| 0x000B | - | |
| 0x000C | 1000 | List |
| 0x000D | - | |
| 0x000E | - | |
| 0x000F | - | |
| 0x0010 | - | |
| 0x0011 | - | |

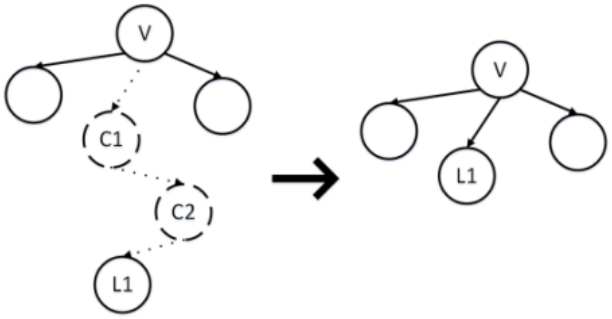
Slika 2. Memorijska prezentacija dinamičkog stabla.

## IV. SORTIRANJE KORISTEĆI N-ARNA STABLA BAZIRANO NA ADRESI I K-KODU (MODIFIKOVAN ALGORITAM)

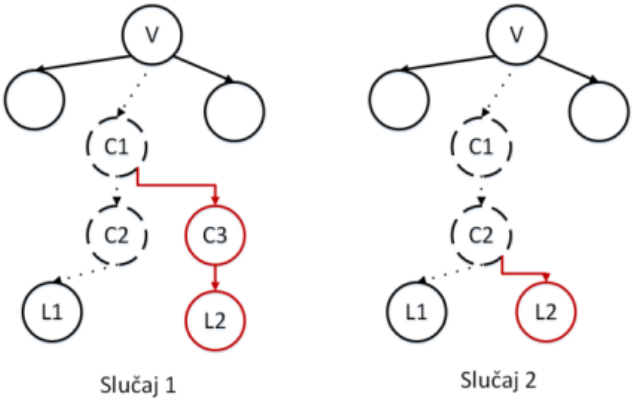
Modifikacija u odnosu na prethodni algoritam proistekla je iz pitanja zašto uvek pamtiti ceo put do lista. Dodatnim zauzećem memorije možemo ovo uraditi.

Mi namerno želimo da dozvolimo da se, dok ne bude bilo potrebe, deo podstabla koji predstavlja put od korena do lista ne pamti. Neka je dat čvor V. Nailazi novi broj. Broj ide putem od korena do čvora V. Neka čvor V ima neke od svojih potomaka, ne sve. Neka novi broj koji nailazi ima takve bitove da teži da pravi nove čvorove od čvora V do lista u kome se treba smestiti. Pitanje je zašto praviti ceo put, kada direktno možemo rezervisati memoriju u listu, reći da se tu nalazi broj, zapamtiti sam broj u tom listu i nastaviti dalj. Na Slici 3 je došao podatak koji se treba smestiti u list L1, i napraviti put preko čvorova C1 i C2.

Problem se javlja kada naiđe podatak koji se ne treba smestii preko čvora C2 ili preko čvora C1 pa na nekom novom mestu, kao na Slici 4. Ovde dolazi u pomoć broj koji smo smestili u L1 pored same vrednosti K-koda. Obavljanjem XOR operacije broja koji je smešten u L1 i broja koji se treba smestiti u L2 dobijamo nešto nalik sledećem 000...0 |V ??? |V KKK . Kako nas k zadnjih bita ne zanima, njih odbacujemo. Bitovi do |V označavaju bitove koji smo prešli dok smo došli do čvora V. Bitovi počevši od oznake |V do druge te oznake označavaju bitove na kojima se mogu razlikovati broj smešten u L1 i novi broj. Nas zanima mesto prvog bita (počevši od najviših) na kome se javlja razlika da bi znali čvor iz koje dubine trebamo napraviti.




Slika 3. Razlika između modifikovanog i nemodifikovanog stabla.




Slika 4. Problem koji se javlja kod dodavanja čvora u modifikovanom stablu.

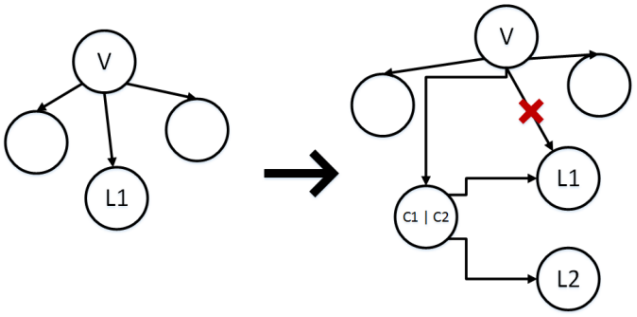
Čvor C3 ne bi trebalo praviti. Posmatrajmo Slučaj 1 gde se javlja problem pravljenja čvora C3. Stavimo se u situaciju da čvor C1 postoji. Tada ne bismo pravili čvor C3 već bi direktno lepili list na čvor C1. Odavde sledi da i u Slučaju 1 i u Slučaju 2 rešenje je napraviti jedan čvor, gde ćemo taj čvor nadovezati na čvor V tj. u čvor V napisati adresu njegovog novog potomka, prepisati na odgovarajućem mestu adresu lista L1, zauzeti memoriju za list L2 i napisati njegovu adresu na odgovarajućem mestu u novom čvoru.

Dalje, da li je bitno da li je to čvor C1 ili je to čvor C2? Naš odgovor je da, jer moramo znati na kojoj dubini se

nalazi čvor, plus, granjanje mora biti zadovoljavajuće, tj. da se ne bi novi čvor i stari preklopili. Za Slučaj 1, treba napraviti čvor C1, dok za Slučaj 2, čvor C2. Ako bismo pored čvora sada pamtili i bilo koji broj iz njegovih listova (uočava se da je dovoljan jedan broj), ponovo proverom XOR operacije novodolazećeg broja i broja koji je upisan u čvoru utvrđujemo da li postoji slaganje nad određenim segmentom bitova. Ako postoji, nastavlja se dalje, a ako ne postoji, mora se izvršiti prethodno opisana procedura. Ovo objašnjenje se dodatno može posmatrati uz Sliku 5. Isto tako očigledno je da Slučaj 1 i Slučaj 2 pokrivaju sve ostale slučajeve, a predloženo rešenje rešava problem konflikta, čak i za slučajeve kada bi se pojavilo još međučvorova.

Dalje, može se desiti da se vrši granjanje i pre nego što se dođe do lista. Lako se može uočiti da je taj slučaj isti kao i slučaj sa listom.

Izmene u strukturi koje je ovaj način razmišljanja uvele su sledeće. Za čvor moramo pamtiti da je čvor u pitanju, a za list da je list, i na kojoj se dubini nalazi. U svakom čvoru i svakom listu pored toga šta je moramo pamtiti i bilo koji njegov broj, tj. broj iz njegovog podstabla. Za slučaj čvora, jedan broj iz bilo kog njegovog lista, a za slučaj lista, jedan bilo koji njegov broj. Dalje memorijska struktura predložena u nemodifikovanom algoritmu može ostati ista, gde samo treba dodati napomenute atribute. Nažalost, zbog toga da li je bitno da li se radi o čvoru ili o listu moguć je dodatni *overhead* prilikom samog prolaska kroz čvorove.




Slika 5. Rešenje problema dodavanja novog čvora u modifikovanom stablu.

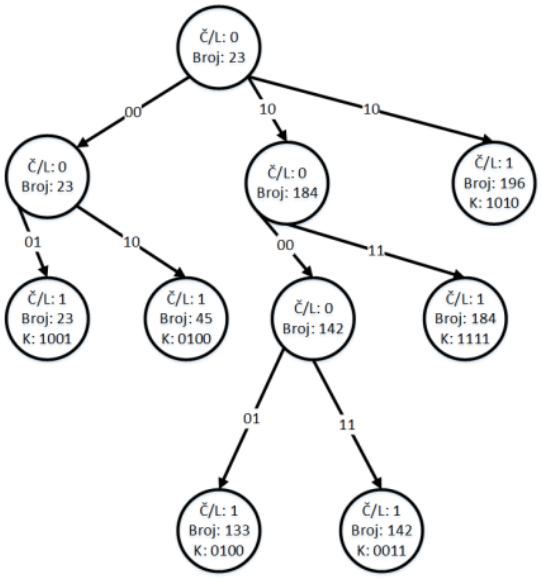
Dodatne izmene koje su uvedene, koje je autor uveo, jesu da ako se nalazimo trenutno na nekom čvoru, moramo pamtiti i adresu njegovog posrednog roditelja u slučaju nastanka konflikta. Algoritamski će se pokazati da ako gledamo čvor unapred, ne moramo pamtiti adresu neposrednog pretka datog čvora. Što se paralelnog izvršenja tiče u trenutku nastanka konflikta javlja se problem, jer bi trebalo izvršiti nekoliko upisa u memorijsku strukturu. To su oni upisi koje nismo želeli da izvršimo. Paralelizaciju algoritma možemo vršiti tako što ćemo na svakom čvoru uvesti da se pamti i referenca na semafor koji služi za zaključavanje datog čvora, ili svaka nit može praviti svoje stablo, a onda sva stabla spojiti u jedno.

Ovakvu implementaciju sortiranja dobro je izvršiti ako sortiramo brojeve sa velikim brojem bitova, gde je stablo duboko. Time što ne pamtimo ceo put do lista od trenutka stizanja novog broja smo učinili da stablo na pojedinim mestima ima različitu dubinu. Ono svoju maksimalnu dubinu dostiže samo u delovima u kojima su se desili "pogoci" koji su naterali stablo da se spusti do svoje maksimalne dubine. Još jedna dobra strana ovog pristupa je što se ne moramo spustiti do određene dubine da bi pokupili podatak iz lista, već ako se nijedan pogodak nije desio u okolini tog lista, sam list će biti blizu čvora iz nekog gornjeg nivoa.

Radi ilustracije uštede daćemo primer. Neka je $M = 8, K = 2, N = 4, G = 2$. Brojevi koji će biti sortirani dati su u tabeli *Tabela I*. Nebitno je po kom redosledu su došli brojevi, uvek će stablo strukturno isto izgledati, samo će se pomoćne vrednosti u čvorovima možda razlikovati. Na Slici 6 može se videti izgled dobijenog stabla kada se sortiraju brojevi iz tabele, redosled brojeva u kojem su došli neće promeniti strukturu stabla.

TABELA I
BROJEVI KOJI SE SORTIRAJU SA NJIHOVOM BINARNOM PREZENTACIJOM I NJIHOVIM K-KODOM

| ***Broj*** | ***Binarna vrednost*** | ***K-Kod*** |
|---|---|---|
| 20 | 00 - 01 - 01 - 00 | 1000 |
| 23 | 00 - 01 - 01 - 11 | 0001 |
| 45 | 00 - 10 - 11 - 01 | 0100 |
| 133 | 10 - 00 - 01 - 01 | 0100 |
| 142 | 10 - 00 - 11 - 10 | 0010 |
| 143 | 10 - 00 - 11 - 11 | 0001 |
| 184 | 10 - 11 - 10 - 00 | 1000 |
| 185 | 10 - 11 - 10 - 01 | 0100 |
| 186 | 10 - 11 - 10 - 10 | 0010 |
| 187 | 10 - 11 - 10 - 11 | 0001 |
| 196 | 11 - 00 - 01 - 00 | 1000 |
| 198 | 11 - 00 - 01 - 10 | 0010 |




Slika 6. Izgled stabla dobijenog kada se sortiraju brojevi iz tabele 1.

## V. PARALELNA IMPLEMENTACIJA ALGORITAMA

Ovaj odeljak, u kratkim crtama sadrži, kako bi se mogla isprojektovati paralelna implementacija oba algoritma. Biće opisana dva načina na kojima bi se mogla isprojektovati paralelna implemntacija.

Kod oba načina, vrši se podela početnog niza po nitima, svejedno kako, bitno je samo da svaka nit dobije određenu količinu posla.

Prvi način se sastoji u tome da se sve niti rade nad istim stablom. Tu se javlja problem istovremenog pristupa istom čvoru/listu. To se može rešiti tako što će se u čvoru/listu pored svih atributa pamtiti i referenca na semafor. Semafor služi da se zauzme čvor/list i da ne dozvoli pristup više niti istom podatku. Semafor je nad celim čvorom, tako da nijedna nit neće moći imati pristup tom čvoru, dok ga nit koja vrši obradu nad njim ne oslobodi. Sve niti rade nad istim stablom i dodaju elemente nad tim stablom. Na kraju samo treba proći kroz stablo i pročitati brojeve koji su sortirani.

Druga implementacija podrazumeva da svaka nit stvara svoje stablo. Ovde nema međusobnog čekanja, nego sve niti rade non stop. Ušteda na vremenu je kompenzovana dodatnim zauzećem memorije. Na kraju, treba proći kroz sva stabla, i to može biti implementirano sekvencijalno, tako što će se čuvati pokazivači na svaki trenutno aktivni čvor iz svakog stabla. Kada svi ti pokazivači budu na listu ili budu NULL (NULL bi bili ako taj put kojim se išlo u tom stablu ne postoji), tada se vrednost K-koda iz svakog lista OR-uje sa K-kod drugih listova. Na kraju se samo štampaju vrednosti iz tog lista, i ide se dalje.

Poređenje oba pristupa se svodi na to da je vreme čekanja koje se javlja u prvom pristupu kompenzovano većim memorijskim zauzećem i težim krajnjim prolaskom kroz stablo u drugom pristupu.

## V. DOBIJENI REZULTATI I DISKUSIJA

Merenja su vršena na personalnom računaru. Sortirani su 16to bitni brojevi. Parametri vezani za stabla su G = K = 2. Generisani su nizovi sa $2^i, i = 1 \ldots duzina_broja_u_bitovima - 1$ brojeva. Za svaku dužinu niza generisano je $2^{duzina_broja_u_bitovima-1}$ niza. Merenje memorijskog zauzeća nije odrađeno. U algoritmu, da bi se i memorijsko zauzeće izmerilo, se mora izračunati broj čvorova i broj listova koji se generišu, i taj broj pomnožiti sa veličinom čvora/lista.

Može se zapaziti da je prilikom random izabranog niza 16to bitnih brojeva gde je G=K=2, dobijeno poboljšanje. Sa urađenog ispitivanja može se uočiti da dok je dužina niza manja od 64 broja, skoro je svejedno koji se algoritam koristi. Ako je dužina ulaznog niza između 64 ($2^6$) i 8192 ($2^{13}$) bolje je koristiti modifikovani algoritam. U ostalim slučajevima bolji je nemodifikovan algoritam jer odmah zauzima memoriju za ceo put do lista bez dodatnih provera, a kada ima više brojeva, veća je verovatnoća da će većina puteva u stablu postojati.

Dodatno treba zapaziti da performanse agloritma zavise od samih brojeva u ulaznom nizu. Dodatno istraživanje, radi otkrića matematičke formule, se treba izvršiti. Dobijena matematička formula bi opisivala kakav raspored svih *velicina_broja* brojeva treba biti u ulaznom nizu da bi se utvrdilo koji algoritam je bolji.

Rezultati merenja, same implementacije algoritama, teorijska poređenja algoritama, kao i jos nekoliko zanimljivih algoritama se može videti u radu [11].

## ZAHVALNICA

U ovom radu, voleo bih se zahvaliti prof. dr Ivanu Milentijeviću i dipl. inž. Aleksi Ravnihar koji su mi pružili enormnu podršku prilikom rada na ovoj temi.

## LITERATURA

[1] Rüdiger Reischuk - Probabilistic parallel algorithms for sorting and selection 1985
[2] David Guichard - An Introduction to Combinatorics and Graph Theory
[3] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stain - Introduction to Algorithms, 2nd Edition, MIT press, 2002
[4] K.H Rosen, J.G. Michaels, J.L. Gross, D.S. Shier - Handbook of Discrete and Combinatorical Mathematics, CRC Press, Boca Raton, 2000
[5] F.M. Carrano - Data Abstraction and Problem Solving with C++, Walls and Mirrors, Addison Wesley, Boston, 2005
[6] V.Sklyarov, I. Skliarova, D. Mihhailov, A. Sudnitson - Implementation in FPGA of address-based data sorting, in Proc. 21st Int. Conf. on Field Programmable Logic and Applications - FPL'2011, Creete, Greece, 2011

[7] Wim Vanderbauwhede, Khaled Benkrid - High-Performance Computing Using FPGAs, Springer Science+Business Media, LLC 2013
[8] Donald E. Knuth - The art of Computer Programming, vol 3, Addison-Wesley, 1973
[9] R. Muller, J. Teubner, G. Alonso - Data processing on FPGAs, in Proc. VLDB Endwment, vol. 2(I), Lyon, France, 2009
[10] J.D. Davis, Z.Tan, F. Yu, L. Zhang - A practical reconSlikable hardware accelerator for Boolean satisfability solvers, in Proc. 45th ACM/IEEE Design Automation Conference, DAC'2008, Anaheim, California, USA, 2008
[11] J. Milojković - Algoritmi za sortiranje podataka zasnovani na stablima, 2017, Elektronski fakultet, Univerzitet u Nišu (https://drive.google.com/file/d/0Bxu4BjRGe6Wtbk0wMDJFbTJjbEU/view?usp=sharing

# Synthesizing Simulated Electrical Appliances Consumption for Typical Serbian Household

Filip Filipović, Bojan Banković

*Abstract* – This paper focuses on the creation of synthetic data for the typical Serbian household using Simulink models for its consumption. Individual diagrams are generated for twelve types of appliances in each of three consumer categories (concerned about consumption, less concerned and not concerned). After that, the data is fed into another algorithm, the user can select a number of households to be generated, and various probabilities in order to make different consumers. In the end, the generated data is saved in .csv files along other generated .csv files that serve the purpose of easier extraction automation.

## I. INTRODUCTION

Forecasting power consumption by electrical energy distributers is usually done for one group of consumers (usually within the same substation) not for each user individually. Distinguishable power consumption profiles do not have to model precisely each user. Empirical formulas based on load factor and simultaneous load coefficient, among many other factors are used for substation sizing and energy demand forecast. The opening of Serbian electricity market to other suppliers and opportunity for buyers to select their own supplier led to a more competitive market. In order to make as much profit as possible, suppliers will have to develop more complex algorithms for better consumption prediction. Power distribution companies will have to improve their efficiency by reducing both technical and non-technical losses. All of that implies that models for power loads will have to be developed for smaller groups.

Computational power can no longer be seen as an issue for more detailed algorithms. The main challenge, besides the creation of the algorithm itself, has now shifted to search for the relevant data for algorithm verification. Although many open data sets can be found online containing electrical consumption data for single household there can be several problems. Data belongs to a household that is located in a different region than desired (UK, USA, Canada, Finland or Norway for example). Thus, different habits of dwellers, different appliances and different time of their usage are expected.

As mentioned in [1], stochastically superimposed series of power perks present in residential building consumption profile cannot be distinguished when larger groups are observed. Thus, algorithms that address issues related to individual consumer cannot be fully tested on power consumption data records available from substations. Collecting electrical consumption data from each individual household is often followed by cost, privacy and practicality issues. For that reason, usage of synthetically generated data, at least for the initial version of algorithm tests, can be a good substitute.

In order to overcome this obstacle, MATLAB's Simulink model, described in [2], was used for simulation of various consumers. Once the user defines ambient temperature by hour, it can define the number of windows in each room and size of the rooms. These data including temperature setpoint in household is used in thermal model of the house. Operation of some appliances is directly related to room temperature, like air conditioner or heater. Some appliances are indirectly related, like refrigerator or water heater, and some are independent like lighting.

Considering the fact that Serbia has approximately 3.8 million electricity household meters and this group utilizes around 14000 GWh of electrical energy per year, the average consumption can be calculated around 300 kWh per month [3]. In [4] consumers are micro grouped based on their different consumption behavior and using real data from smart meters. Synthetic household data generated by simulation cannot exhibit those patterns, but daily power consumption diagrams shown in [4] and in [1] can be used as a starting point for daily consumption diagram. Further modification is done using geographically and economically closer consumption data [5].

The purpose of this paper is to present an alternative way of generating data for initial algorithm evaluation. Although MATLAB's Simulink is used for initial diagram generation, Python is used for data post processing and formatting in a way suitable for later use.

The paper is structured as follows: section 2 gives details and methods for algorithm creation; section 3 gives overview of results and section 4 discusses conclusions and future work.

## II. MATERIALS AND METHODS

Precise modeling of household power consumption diagram must also include some habits of residents. These habits are manifested through routines like waking up time, usage of stove, television and many others. If we are only

Filip Filipović and Bojan Banković are with the Department of Electrical Power Engineering, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: f.filipovic@elfak.ni.ac.rs and b.bankovic@elfak.ni.ac.rs

concerned about energy consumption, most of these habits can be neglected.

Twelve appliances used for simulation were modeled for three classes of consumers (concerned about consumption, less concerned and not concerned). Three classes correspond to three slightly different pattern of appliance usage. Peak power load of an appliance is also correlated with the consumer class the appliance is in. Smart Residential Load Simulator (SRLS) tool from University of Waterloo was used for simulation of daily load diagram of individual appliance. Process of synthetic households' generation can be seen in Fig. 1.

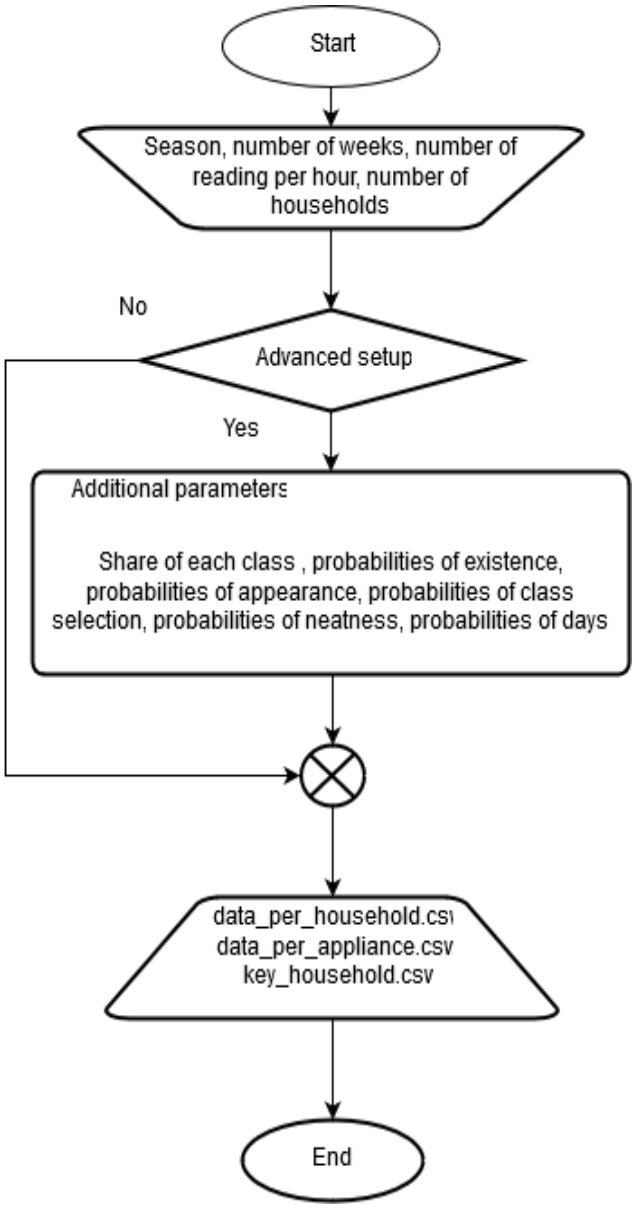



Fig. 1. Flowchart for synthetic household creation.

For basic synthetic data generation, user is only asked to enter season (summer or winter), number of readings per hour and number of households. Advanced setup offers full control over probabilities for diagram generation.

In the following sections, further explanation of each probability function will be explained. Also, for reasons of convenience, classes will be marked as shown in Table I.

TABLE I
DEFINITION OF CLASSES

| Old class names | New class names |
|---|---|
| Concerned about consumption | CC |
| Less concerned | LC |
| Not concerned | NC |

List of appliances and shorter names are present in Table II.

TABLE II
APPLIANCE LABELS

| Appliance | Label |
|---|---|
| Larger water heater | LWH |
| Smaller water heater | SWH |
| Stove | ST |
| Electronics | EL |
| Light | LI |
| Washing machine | WM |
| Fridge | FR |
| Air condition | AC |
| Heater | HE |
| Dishwasher | DW |
| Clothes dryer | CD |
| Watering | WT |

*A. Share of each class*

This represents array of probabilities that certain household belongs to certain class. Default settings are shown in Table III.

TABLE III
SHARE OF EACH CLASS IN TOTAL NUMBER OF HOUSEHOLDS

| Class | CC | LC | NC |
|---|---|---|---|
| Probability | 0.15 | 0.75 | 0.1 |

*B. Probability of existence*

This represents multidimensional array. It is used to model the probability for the existence of one or several specific appliances during workdays and weekends independently. Probabilities differ depending on the appliance and the class that the household belongs to. The output of the function that uses this probability is how many appliances of each category are present in the household.

*C. Probability of appearance*

This represents multidimensional array. It is used to model the probability of appliance appearance several times during week (or weekend). Probabilities differ depending on the appliance and the class that household belongs to. The output of the function that uses this probability is how many days during a week (or at weekends) the appliance will be used.

*D. Probability of days*

This represents a multidimensional array. It is used to model the probability of days of week that each appliance will be used. Probabilities differ according to the appliance class household belongs to and ambient temperature. An air conditioner or heater will be used more often on days when the ambient temperature has a greater deviation than desired. The output of the function that uses this probability is on what specific days what appliance will be used, considering probability of appearance.

*E. Probability of class selection*

This represents a multidimensional array. It contains probabilities for each individual appliance to take the power consumption diagram belonging to the class of household or some other class. The output of the function that uses this probability is what appliance will have power consumption diagram from what class.

*F. Probability of neatness*

This represents a multidimensional array. It contains probabilities for each individual appliance that will be activated next week on the same day, day before, day after, or if the activation will be skipped. The output of this function is an array of numbers that represent days when each appliance will be activated.

*G. Other variations*

There is a probability that activation hours of appliances such as light and stove are different at the weekend compared to weekdays. Each appliance can have up to half an hour earlier or later activation each day, compared to base configuration. The median value of generated consumption is around average Serbian household consumption (300 kWh per month).

## III. Main Results

The steps in the flowchart presented in the previous section led to generation of .csv files that can be used for further research. Algorithm for data generation does not have the graphical user interface, but all probabilities used here with other variables intended of being changed are stored in separate python file. The idea behind program files organization is that user can expand existing models of devices or consumers easily. Adding a new device requires from user to insert its name in list, provide a relative path to typical power consumption diagram data and to define device's earlier discussed probabilities.

Simulated power consumption diagrams of three typical appliances (air conditioner, refrigerator and small water heater) are shown in Fig. 2.

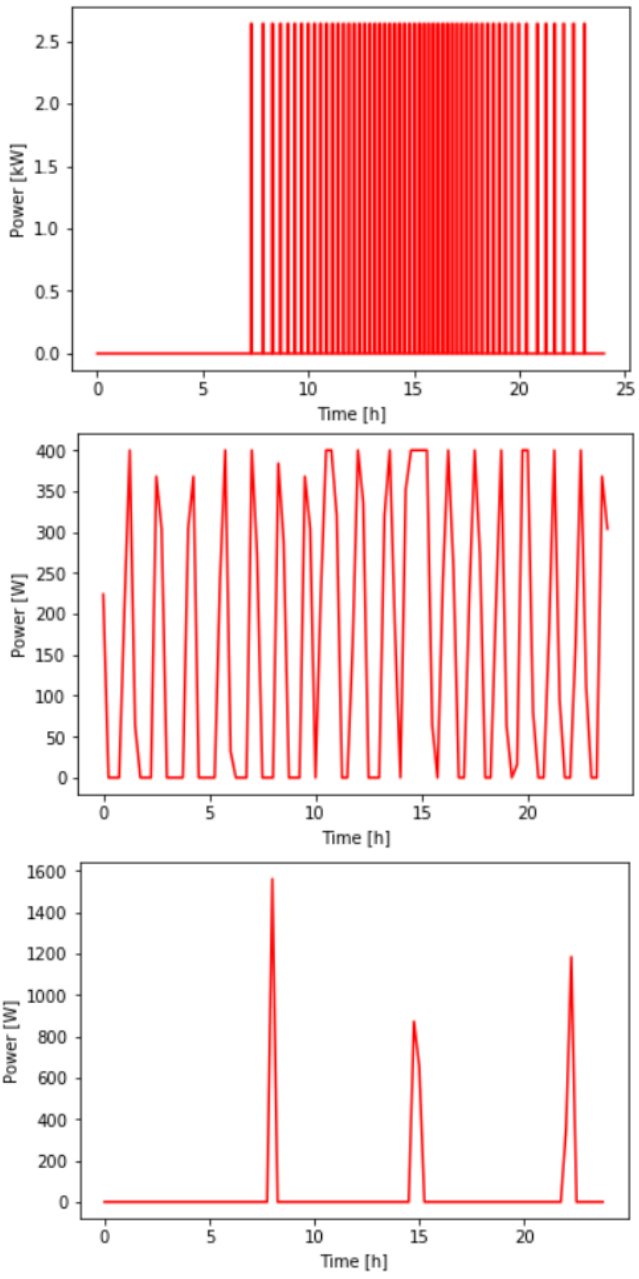



Fig. 2. Simulated power consumption of air conditioner (upper), refrigerator (middle) and small water heater (bottom).

Situation of a warm summer day with an air conditioner programmed to keep constant temperature is

shown in Fig 2. It is turned on more frequently during hours of maximum outdoor temperature. Also, a higher turn on frequency can be seen in late afternoon hours. This can be due to house door opening and stove usage.

A typical power consumption diagram of refrigerator shows slightly longer turn on times after breakfast, lunch and dinner because of the refrigerator door opening.

A simulated power load diagram for a small substation of twenty consumers is shown in Fig. 3.

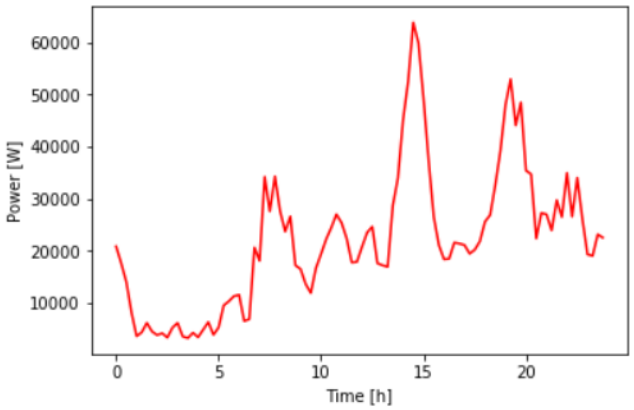



Fig. 3. Simulated power consumption for a small electrical substation that powers twenty household consumers.

In all previous Figures only the daily power consumption is shown for the reason of a more detailed insight. Fig. 4. shows 4 weeks of simulated power consumption diagram for one household.

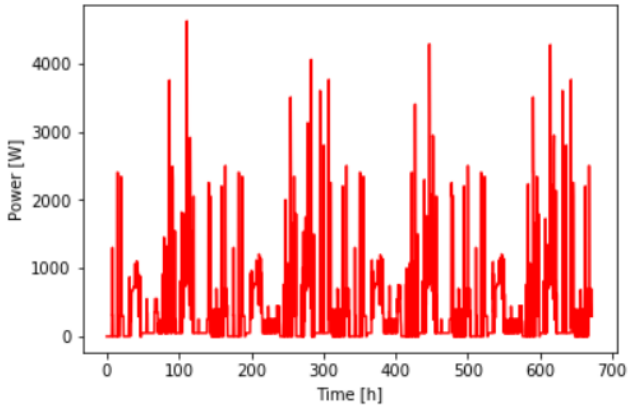



Fig. 4. Simulated 4 weeks consumption of household.

Although Fig. 4. lacks detailed look it is shown in order to demonstrate probability of neatness and probability of days. Power consumption during next week is not a simple reproduction of last week's diagram. General power consumption profile is kept but variations are induced. Even when an appliance is switched on the same day in two consecutive weeks, its time of activation generally varies for up to an hour. Residents can show different behavior at weekends compared to weekdays, as it can be seen in Fig. 4.

## IV. Conclusion

Synthetic data generated by human, and based on a set of rules can hardly completely substitute real data. The main reason for that is human bias itself. We can only make synthetic data as good as the models we design. This means that a design flaw induced in a model might produce data that is completely useless.

Artificial intelligence can solve many of problems present, but it cannot be expected to solve all of them. First, at least for now, it has to be designed and tuned by a human. This implies that biases induced by a designer are contained in it, and in generated data. Secondly, training of artificial intelligence requires lots of data. The main reason for synthetic data generation is compensation for the lack of real data.

## References

[1] Jenkins, D. P., S. Patidar, and S. A. Simpson. "Synthesising electrical demand profiles for UK dwellings." Energy and Buildings 76 (2014): pp. 605-614.

[2] Pouresmaeil, E., et al. "Development of a smart residential load simulator for energy management in smart grids." IEEE Transactions on Power Systems (2013): pp. 1-8.

[3] Eps.rs. (2017). EPS - Potrošnja. [online] Available at: http://www.eps.rs/Lat/Article.aspx?lista=Sitemap&id=100 [Accessed 11 Sep. 2017].

[4] Anderson, Ben, et al. "Electricity consumption and household characteristics: Implications for census-taking in a smart metered future." Computers, Environment and Urban Systems 63 (2017): pp. 58-67.

[5] Anon, (2017). [online] Available at: https://www.researchgate.net/profile/Zdravko_Milovanovic/publication/271701211_POTROSNJA_ENERGIJE_DIJAGRAMI_POTROSNJE_KARAKTERISTIKE_POTROSACA_I_RACIONALNO_KORISCENJE_ENERGIJE/links/54cf77c40cf298d6566378f4/POTROSNJA-ENERGIJE-DIJAGRAMI-POTROSNJE-KARAKTERISTIKE-POTROSACA-I-RACIONALNO-KORISCENJE-ENERGIJE.pdf [Accessed 11 Sep. 2017].

# Merenje sobne temperature i vlažnosti vazduha pomoću Arduino Uno platforme

Dragan Kostić

*Sadržaj* – U ovom radu biće opisan jedan od načina za merenje sobne temperature i vlažnosti vazduha pomoću Arduino Uno platforme. Komponente koje su korišćenje u ovom radu jesu: senzor za merenje temperature i vlažnosti vazduha DHT11 [1], mrežni senzor WiFi modul ESP8266 [2], protoploča, otpornici otpornosti 1x220Ω, 2x4.7kΩ i Arduino Uno platforma koja omogućava da ovaj sistem funkcioniše kao jedna celina. Programiranje celokupnog sistema se odvija u softverskom paketu Arduino IDE [3].

## I. UVOD

Brojne studije su pokazale da temperatura i relativna vlažnost vazduha na radnom mestu znatno utiču na komfort i produktivnost radnika. Prvi korak ka optimizaciji ova dva parametra ambijenta je njihovo pouzdano merenje [4]. Ceo ovaj sistem ima za cilj da nam u bilo kom trenutku bude dostupna informacija o temperaturi i vlažnosti vazduha u okruženju u kome radimo. Takođe, prednost ovog sistema je kontrola ovih parametara iako nismo u tom okruženju. To nam omogućava *ThingSpeak channel* na kome šaljemo vrednosti ovih parametara.

## II. PRINCIP RADA

Za precizno merenje vlažnosti i temperature u zatvorenom okruženju koristili smo jedan od najdostupnijih senzora na tržištu. Vrednosti parametara sa ovog senzora smo slali na Arduino Uno platformu koja je kontrolisala celokupnu funkcionalnost ovog sistema. Ona je obrađivala podatke sa senzora DHT11 i prosleđivala Wifi modulu. WiFi modul ESP8266 smo iskoristili za slanje već obrađenih podataka na mrežu odnosno na *ThingSpeak channel*. *ThingSpeak* omogućava kreiranje aplikacija za snimanje senzora, aplikacija za praćenje lokacije, itd.

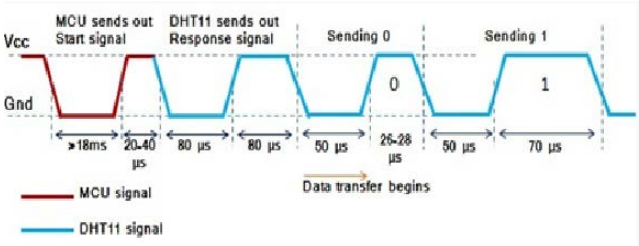
### A. *Opis komponenti korišćenih u radu*

DHT11 je trenutno najjeftiniji senzor dostupan na tržištu koji pruža kalibrisane digitalne izlaze za temperaturu i relativnu vlažnost vazduha.

Karakteristike senzora:

Relative humidity
Resolution: 16Bit
Repeatability: ±1% RH
Accuracy: At 25 ℃ ±5% RH
Hysteresis: <± 0.3% RH
Temperature Resolution: 16Bit
Repeatability: ±0.2 ℃
Range: At 25 ℃ ±2℃
Response time: 1 / e (63%) 10S
Electrical Characteristics
Power supply: DC 3.5～5.5V
Sampling period: more than 2 seconds
Pin Description
1.the VDD power supply 3.5～5.5V DC
2.DATA serial data, a single bus
3.NC empty pin
4.GND

Senzor se sastoji od otporno vlažnih osetljivih elemenata i NTC temperaturnih mernih uređaja, koji su povezani sa 8-bitnim mikrokontrolerom. Vremenski dijagram koji sledi na slici 1 prikazuje protokol kojim senzor komunicira sa mikrokontrolerom.



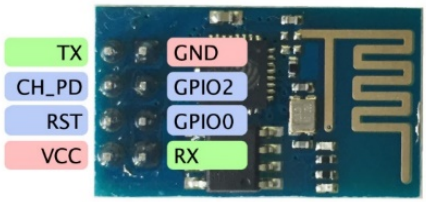

Slika 1. Vremenski dijagram.

Crvenom bojom prikazan je signal sa mikrokontrolera, a plavom signal sa senzora. Mikrokontroler inicira prenos podataka slanjem signala *"start"* i ukoliko je ostvarena komunikacija, dobija odzivni signal. Nakon slanja signala *"start"*, pin mikrokontrolera mora biti rekonfigurisan kao ulazni pin. Kada jednom dobije odziv sa senzora mikrokontroler postane spreman za prijem signala. Senzor tada šalje 5 bajtova podataka kontinualno (kod ovih bajtova, prvi bit je najznačajniji). Ovih pet bajtova podataka se sastoje od:
Podaci = Celi broj RV + Decimalni broj RV + Celi broj Temp. + Decimalni broj Temp. + bajt provere

WiFi modul - ESP8266 je samostalan SOC sa integrisanim TCP/IP protokolom koji omogućava pristup bilo kom mikrokontroleru na WiFi mrežu. Svaki ESP8266 modul dolazi pre-programiran sa AT skupom firmware

Dragan Kostić je student četvrte godine na smeru Upravljanje sistemima, Elektronski Fakultet Niš, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: dragan.kostic@elfak.rs

komandi. Ovaj modul ima odlične mogućnosti obrade i skladištenja podataka na ploči, koje mu dozvoljavaju da se integriše sa senzorima i drugim specifičnim uređajima preko njegovih GPIO pinova. Visok stepen integracije na čipu sa minimalnim dodatnim kolima, uključujući *front-end* module, omogućava ovom modulu da zauzme minimalno PCB područje. ESP8266 podržava APSD za VoIP aplikacije i *Bluetooth* interfejse, sadrži samopodešavajući RF dozvoljavajući mu da radi pod svim radnim uslovima. Izgled WiFi modula je prikazan na slici 2.




Slika 2. Wifi modul ESP8266.

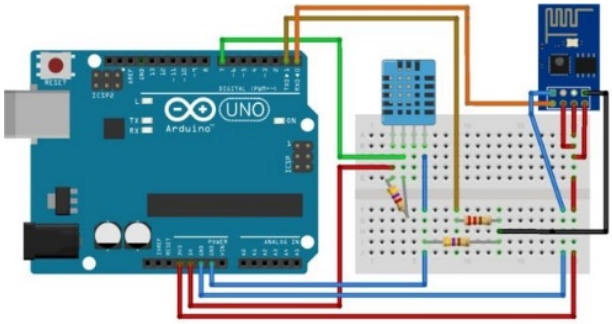
Arduino Uno platformu čine 8-bitni Atmel AVR mikrokontroler sa pripadajućim komponentama koje omogućavaju programiranje i povezivanje sa drugom elektronikom. Bitan aspekt Arduino projekta je standardizovan raspored konektora koji omogućava lako povezivanje sa dodatnim modulima. Arduino ploče uglavnom koriste megaAvr seriju čipova, konkretno ATmega8, ATmega168, ATmega328, ATmega1280 i ATmega2560. Većina ploča poseduje 5V linearni naponski regulator i 16MHz kristalni oscilator ili keramički rezonator. Arduino mikrokontroleri se isporučuju sa programiranim bootloaderom koji pojednostavljuje postupak prebacivanja prevedenog koda u fleš memoriju na čipu. Na slici 3 je prikazan izgled Arduino Uno r3 koja je korišćena u samoj realizaciji ovog projekta.




Slika 3. Arduino Uno r3 platforma.

### *B. Opis sistema*

Sa izlaznih pinova Arduino Uno ploče, uzimamo napon od 5V sa kojim napajamo senzor DHT11 preko pina 1. Naponom od 3.3 V napajamo WiFi modul ESP8266 takođe preko Arduino Uno ploče. Oni su priključeni na određenim pinovima protoploče. Sa Arduino platforme uzimamo izlazne priključke za masu, koji su povezani sa senzorom DHT11 preko pina 4 i WiFi modulom. Pin 2 senzora DHT11 povezan je sa 7 digitalnim pinom I/O Arduino ploče. Pin TXD WiFi modula je povezan na pinu RXD Arduino ploče, i obrnuto. Povezivanje komponenata je odradjeno prema slici 4.




Slika 4. Šema povezivanja.

Posle povezivanja šeme kao sa slike 4 sledi programiranje celokupnog sistema. Kod je napisan u softveru Arduino IDE. Arduino platforma se USB kablom povezuje sa računarom i uploaduje se kod na Arduino Uno razvojnu ploču. Pri završetku učitavanja koda, sistem kreće sa nesmetanim radom. Slike 5, 6, 7, 8 i 9 prikazuju važnije linije u glavnom kodu.

```
// Očitavanje temperature i vlažnosti traje oko 250 ms
// Ova petlja se neprekidno izvršava
void loop(){
  // Očitavanje vrednosti vlašnosti
  float h = dht.readHumidity();
  // Očitavanje vrednosti temperature
  float f = dht.readTemperature(true);
```

Slika 5. Opis početnog dela koda.

```
// Osvežavanje ThingSpeak channel-a sa novim vrednostima
updated = updateTemp(String(f), String(h));

// Ako je osvežavanje uspelo zasvetleće zelena LED
// u slučaju da nije uspelo zasvetleće crvena LED
if(updated){
  LightGreen();
}else{
  LightRed();
}
```

Slika 6. Slanje podataka na *ThingSpeak channel.*

```
// Korišćenje AT instrukcije za slanje podataka
Serial.print("AT+CIPSEND=");
Serial.println(cmd.length());
if(Serial.find(">")){
  // Izvršavanje komande za ažuriranje vrednosti parametara
  Serial.print(cmd);
}else{
  Serial.println("AT+CIPCLOSE");
}
```

Slika 7. Korišćenje AT instrukcije.

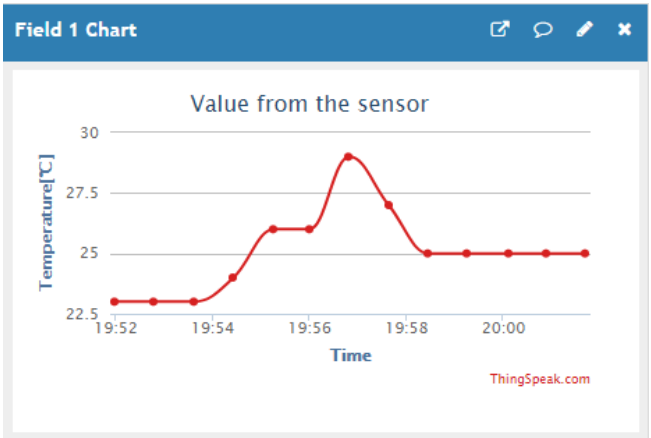
```
// Ako se u nekom slučaju dogodi greška, javlja se alarm
// u vidu svetljenja zelene i crvene LED u roku od 5s
void Error(){
  while(true){
    LightRed();
    delay(5000);
    LightGreen();
    delay(5000);
  }
}
```

Slika 8. Opis koda ako se javi greška.

```
DHT dht(DHTPIN, DHTTYPE);
//  U ovoj liniji koda se inicijalizuje DHT senzor.
// Starije verzije biblioteka su opciono uzimale treći parametar
// kako bi optimizovale vreme za brži rad cpu-ova.
// Ovaj parametar više nije potreban zato što trenutni DHT senzor
// koristi algoritam za samopodešavanje kako bi ubrzao rad cpu-a.
```

Slika 9. Inicijalizacija senzora.

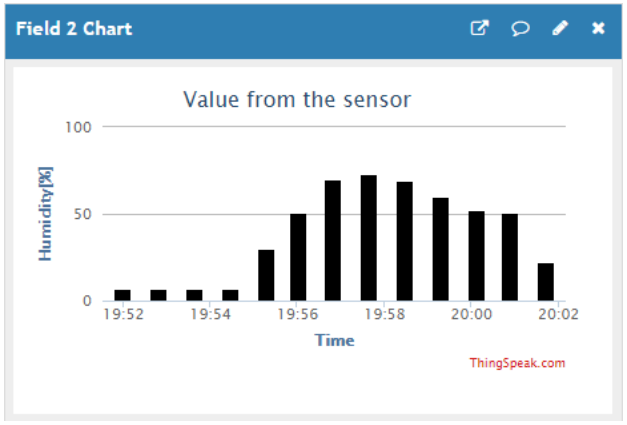
Na slici 10 i 11 su prikazani grafici koji su dobijeni slanjem vrednosti temperature i vlažnosti vazduha na *ThingSpeak channel* pomoću *API key-a*. Kod programiranja ključ za aplikaciju (*API key*) je kod koji se prenosi pomoću računarskih programa koji pozivaju programski interfejs aplikacija (API) da identifikuje pozivni program, njegovog programera ili njegovog korisnika na web lokaciju.



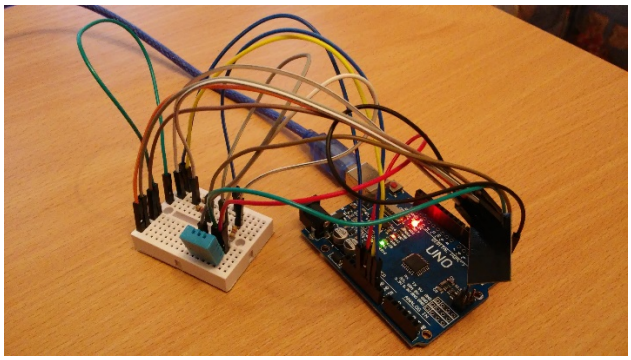

Slika 10. Grafik temperature vazduha.

## III. Zaključak

Ovaj rad se u praksi pokazao veoma uspešnim i funkcionalnim. S obzirom na to da senzor DHT11 spada u nižu klasu, sa *ThingSpeak channel-a* smo očitavali vrednosti na svakih minut vremena kako bismo imali što tačnija očitavanja što je sasvim zadovoljavajuće. Svrha ovog rada je da se pokaže kako pomoću Arduino platforme je moguće na lak i jednostavan način dobiti sistem koji će vršiti određenu funkciju kao i primena usvojenog znanja tokom studiranja.




Slika 11. Grafik vlažnosti vazduha.



Slika 12. Finalni izgled uređaja.

## Zahvalnica

Autor rada se posebno zahvaljuje dugogodišnjem kolegi Milanu Ivkoviću radi pružanja materijalnih sredstava.

## Literatura

[1] https://akizukidenshi.com/download/ds/aosong/DHT11.pdf
[2] https://cdn-learn.adafruit.com/downloads/pdf/dht.pdf
[3] https://www.sparkfun.com/products/13678
[4] https://en.wikipedia.org/wiki/Arduino
[5] https://www.automatika.rs/projekti/merenje-temperature-i-relativne-vlaznosti-vazduha-upotrebom-senzora-dht11.html

# Aural Immersive Virtual Reality for Neuro-Developmental Disorder Therapy Based on Storytelling Approach

Nenad Petrović

*Abstract* – In this paper, an implementation of virtual reality application supporting the therapy of pre-school children with neuro-developmental disorder, based on interactive storytelling is presented. The goal is to provide an affordable solution to help them improve language and communication skill by utilizing state-of-the-art technology. The application is running on Android smartphones and implemented using the combination of Unity 3D framework and Google Cardboard for the graphical part, while the speech processing is done by utilizing the capabilities of IBM Watson.

## I. Introduction

Wearable immersive virtual reality has shown an enormous promise as learning tool, especially when it comes to therapy of pre-school children with verbal Neuro-Developmental Disorder (NDD), which affects their social communication, learning, understanding, concentration and language skill [1].

Traditional storytelling is one of the common methods in such therapy, but we wanted to make it innovative and more interesting for the main target group – the children, so we consider utilizing state-of-the-art technology which is commonly used in modern video game development. Children are wearing VR glasses with mobile phone inside, which gives them perception of presence inside the virtual, video-game alike environment.

The user interaction approach is virtual reality-supported, interactive storytelling [2]. Patient listens to the story, and at certain points she/he has to answer the questions related to the content heard in order to proceed. The answers could be derived from the content heard or objects observed in the environment. Also, the environment is adaptive and it takes into account individual characteristics of the patient, letting them a certain degree of freedom, such as bringing decisions in certain situations.

There are similar virtual reality projects in this area, starting from late 90s, and most of them have been focused on topics related to simulating real-world situations [3] such as bus stops, classroom, caffé, cinema etc. Most of them are for PC platform and require specialized VR glasses and GPUs, which could cause additional expenses and are not easily portable from one place to another. However, smartphones are now powerful enough to handle more intensive graphical and multimedia content, so we decided to use this approach which is more cost effective and portable compared to the existing solutions, while giving us physical portability. This is also important for therapists and therapy institutions, as the therapy is not strictly related to the location where it is performed. Also, existing storytelling-oriented solutions in most cases contain stories with linear flow, which is the same for each patient, while in this project we want to provide adaptivity based on user characteristics, providing customized stories.

Nenad Petrović, M.Sc. in Computer Science and Engineering, Politecnico di Milano, Piazza Leonardo da Vinci, 32, 20133 Milano, Italy, E-mail: penenadpi@gmail.com

## II. Working Principle

Taking into account the type of interaction and the environment (aural immersive virtual reality), user input actions are limited to moving head in virtual environment (in order to observe the surrounding and objects) and providing spoken answers. On the other side, user actions change the state of the environment, which are, practically, visual changes and aural response.

As Figure 1 displays, the input of the engine are user speech and head movement, while the outputs are spoken response and VR environment changes (objects appear/disappear, change their position etc.).

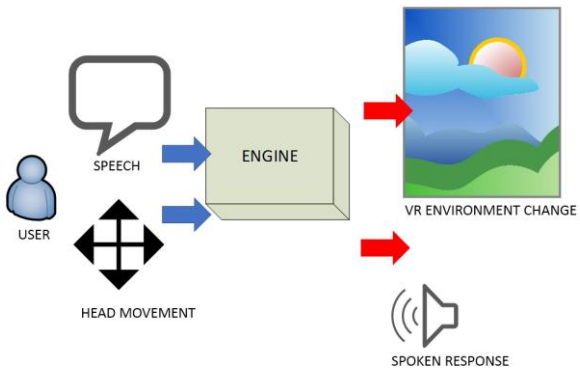



Fig. 1. Working principle overview.

We can distinguish between two separate processing paths: speech and head movements. Both of them hold meaningful information which can be extracted during the processing by system components and have separate paths.

In what follows, the flow of the operations necessary to be performed in order to produce the output from the input is presented, by following these paths.

*Speech processing path:* The speech is obtained from the patient and recorded as a .wav file. Then, the speech analysis component extracts meaningful information from the audio file. The most important thing that is extracted in this step is the text (speech-to-text transformation). Furthermore, advanced speech analysis techniques can be used in order to capture more parameters related to level of attention, concentration, emotion, such as speech breaks, pitch strees being emphasized on certain words etc. Furthermore, the textual response is forwarded to the conversational logic, which is answering according to the current context. Figure 2 presents the visualization of speech processing path.

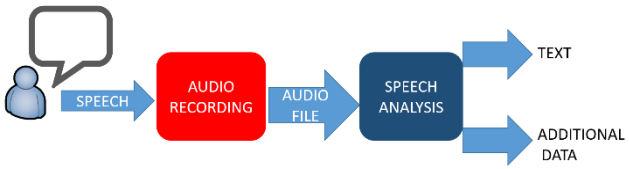



Fig. 2. Speech processing path.

*Movement processing path:* During the therapy session, the patient's head movement is analyzed to obtain useful information. For example, the level of attention can be evaluated according to the point where patient is currently looking. In some scenarios, the goal is to find or follow a certain object. Therefore, this information is useful to determine the patient's progress of this scenario. Also, the head movement speed can be useful information in order to determine that some specific event occured. It can be used to detect a certain type of patient reaction during the usage of the system, based on head movement. For example, very sharp and unexpected turn could mean that the patient is scared. Figure 3 presents the visualization of head movement processing path.

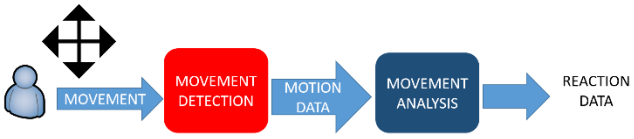



Fig. 3. Head movement processing path.

Once the logic engine gets all the mentioned results (obtained after speech and head movement processing), it is ready to create the output that consists of virtual reality image and spoken response. Depending on the context, the objects are inserted to the environment, destroyed or their state is changed (animations). For example, the patient decides to attack the goblin (in a fairy tale-alike story), by speaking the word "attack". After that, the goblin is destroyed and disappears from the virtual reality scene.

## III. System Architecture

According to the interaction type and requirements in terms of input and output during the interaction and previously mentioned challenges, we conclude that the solution architecture should contain the following components (Figure 4 displays the relations between them):

- *Speech processing component:* to transform the spoken words into text (speech-to-text) which can be further processed, and also analyze other properties of the speech, related to emotion and sentiment analysis which can be useful to determine the level of attention. It is responsible for operations which belong to speech processing path (Figure 2).
- *Head movement processing component*: in order to analyze the speed of head movement and point of view (looking at certain object or not, for attention level estimation) in order to detect specific behaviour which is useful for creating the feedback (sharp moves could indicate fear, which could lead to the termination of the application in case etc.). It is responsible for operations which belong to head movement processing path (Figure 3).
- *Logic engine:* which performs actions corresponding to the current state in order to produce the feedback to the patient, based on speech analysis, movement analysis and story progress. For example, it can create or destroy objects in the virtual environment by sending the commands to the graphic component within the current context, or generate spoken response by using text-to-speech component.
- *Speech output component:* the task of this component is to create spoken answer from the text generated as a response from the logic component. This is actually a text-to-speech block.
- *Graphical component:* creates virtual reality environment based on the input from the logic component, by placing 3D objects and models in the current context. It is actually based on Unity 3D API built-in capabilities.
- *Story engine:* the component which is responsible for reading the story file in a proper way, sending the right instructions to the logic engine which can further invoke graphic component to create the VR environment, make transitions or perform changes of the state. The file includes also reference to the 3D models which appear as objects in the virtual environment. The stories are created by experts (doctors, psychologists) and contain certain scenario where the patient has to listen to the story and provide correct answers in order to proceed.
- *Model retrieval:* the component which has a role to retrieve the right models for the story by browsing the database of the models which can be either offline (locally stored) or online (Unity 3D model store or any other source of models), and forward them to the graphical component. The focus is on using 3D models which can be downloaded for free and embedded into environment "on the fly",

whenever the situation in the story needs a model which is not already available in the local database. For example, if at certain point patients are asked to continue story using their own imagination, they can mention some objects whose models might not already be available in the database.

- *Model database:* keeps the models locally in order to make the loading times faster. Each time new object taking a role in the story is mentioned, the model retrieval component looks for appropriate model – first within the local database, and then looks up online resources. This way, new models are transferred to local storage and available for the next session.
- *Activity tracking and analysis:* this component is monitoring the patient behavior during the therapy session and recording the data inside the log, which can be later useful to therapists for patient progress evaluation and evaluation of the therapy effectiveness by therapy domain experts, such as psychologists. Typically, it contains the time necessary for the patient to respond to the given question, time spent to solve the given task, number of wrong answers given or even number of times the patient was showing unexpected behavior, reaction or decision to quit.

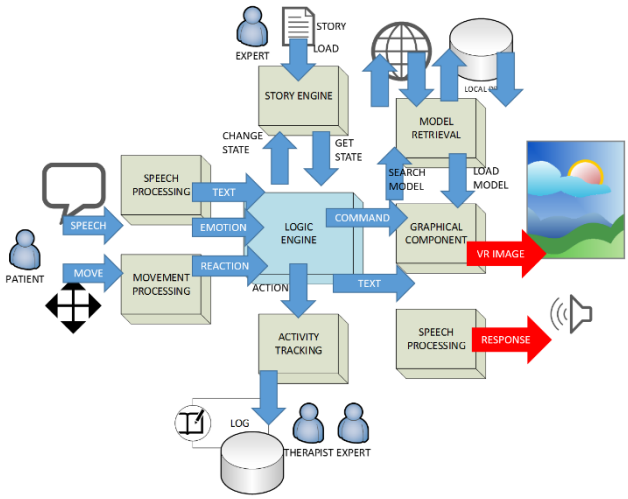



Fig. 4. System architecture overview.

## IV. TECHNOLOGY

Considering the state-of-art wearable VR platforms (Oculus Rift, Play Station VR, HoloLens, Google Cardboard), we decided to focus on smartphone-oriented solution supported by Google Cardboard VR platform.

Google Cardboard is a virtual reality (VR) platform developed by Google for use with a head mount for Android smartphone. A smartphone is inserted in the back of the device and held in place by the selected fastening device. A Google Cardboard–compatible app splits the smartphone display image into two, one for each eye, while also applying barrel distortion to each image to counter pincushion distortion from the lenses. The result is a stereoscopic ("3D") image with a wide field of view. In our system, patient is wearing VR kit, which gives her/him perception of being virtually present in 3D scene. The main reasons for the choice of this technology are:

- Cost-effectiveness of the solution
- Wide compatibility
- Continous support

As it is obvious, smartphones are everywhere these days, so it is not necessary to buy additional dedicated hardware which is executing the application (which could be quite expensive if we consider some of the solutions mentioned). So, any gyroscope-enabled smartphone running Android, Windows or iOS could be used for that purpose. As for now, a simple prototype has been made for Android, but we plan porting it to other platforms as well in future. Also, Google Cardboard has solid support (taking into account that it is Google product) and is compatible with many game development-oriented and 3D environment modelling frameworks, like Unity.

Taking into account the software platform behind the application and requirements, we can cocnlude that the following hardware components and devices are necessary:

- *Gyroscope-enabled Android smartphone (recommended with Android OS 5.0 or greater):* the device executing the application.
- *Cardboard-compatible glasses:* the patient wears a phone inside cardboard-compatible glasses in order to perceive the virtual environment. When it comes to glasses selection, there are many different solutions provided by both Google and 3rd party manufacturers, and all of them are relatively cheap (ranging from 5 to 50 euros). The original Google Cardboard glasses are very cheap and made of paper material. However, considering the usage context of our application (children, therapy) we found more comfortable, durable and higher quality solution (by 3rd party manufacturer) made of plastic and easy to mount, with hole for headphone connector (important for our application).
- *Headphones:* not obligatory, but recommended, in order to avoid speech-to-text interference with the sound which is played back).
- *Bluetooth remote control:* used by therapists, in order to control the therapy if necessary.

During the application testing and prototype development, we were using gyroscope-enabled LG G3 Android smartphone with Android OS version 6 with VR BOX Cardboard glasses.

## V. SCENARIOS

As a proof of the concept, two types of story scenarios [4] were implemented, each of them using different type of story engine which instructs logic. More stories are planned to be added in future.

In what follows, these scenarios are going to be described, in order to present the idea of the system

features and goals it has to achieve and which kind of user interaction is needed for each of them.

*Scenario 1 - Multimodal interaction:* The VR environment appears set up as a virtual transposition of a paperbased fantasy, fairy tale-alike story, similar to ones that the are used in classroom activities with pre-school children. To proceed in the story, the child is required to reply to simple questions that are aurally asked by the conversational agent, which could appear as a fantasy character, such as knight, magician or fairy, for example, to focus their eyes on the main character, or on other story elements or the objects in the environment, and to maintain their eyes focused on these items for a pre-defined minimum time. This character has the role of a facilitator to suggest what to do and when to do it. Also, it could play a role of rewarding agent, to provide feedback and positive reinforcement to the child's interactions. In what follows, an example is going to be provided (Figure 5 displays the screenshot of the scenario).



Fig. 5. Multimodal interaction scenario screenshoot.

The main character (knight) appears in the virtual environment similar to fantasy video game weapon/item shop, and the voice of the narrator says: "The knight, entered in the shop, saw a shopkeeper near some objects: potions, weapons, wands, and many other thing stored on the shelves. But the knight needs precisely few items. These items are written on a list, so the knight approaches to take it".

Now, the child can do the following things: "summon" the list by saying "my list" in a phrase; take a potion saying "bottle"; take a weapon saying "weapon"; take a shield saying "shield"; finish the dialogue with the shopkeeper saying "no thanks" (or "no thank you"). In order to take the item, the child has to maintain her eye focus on the item for at least 2 seconds and pronounce its name.

After saying "list", near the shopkeeper's desk, a white rectangular screen will appear, with some names of objects. This is the knight's list and the nouns are the objects that the knight needs. In this case, the objects are: potion, weapon, shield. If the patient says one of this words, the shopkeeper will say: "Yes sir! Here it is. Do you need something else?"

When the patient finishes pronouncing all the items that are contained in the list and says "No, thanks", the answer from the shopkeeper is: "Farewell. Be careful, your quest will not be easy!"

In some situations, the child may also use verbal commands to control characters' behavior, for example saying "Disappear!" to remove a character from the current scene. Using a remote controller, the educator can control the progression of the story and interact with the VR items to help the children sustain attention and engagement.

*Scenario 2 - Storytelling dynamic contextualization and story generation:* The child is asked to "tell a story" while the application provides a visual and auditory context for the story in the VR environment. This is achieved by dynamically populating the VR space with multimedia contents (2D or 3D scenes or characters, videos, or images) that are contextualized to the child's narrative, and offering verbal clues to stimulate the child to proceed with the story. If eventually the child does not continue in spite of the system's prompts, a continuation of the story is automatically generated. For example, when the child says "Once upon a time…there was a goblin in a forest", the images of a goblin inside forest appear in the VR space, underlined by the spoken comments like "Do you want to attack the goblin?". If the child does not react, the main character passes by (without attacking), while in the case of response „Attack!", the main character enters the battle against the goblin. Figure 6 displays the screenshot of this scenario.

It should be taken into account that in this scenario it might happen that not all multimedia content (such as 3D models corresponding to the mentioned objects) are internally available, so the component that is able to search certain online resources (Unity store, for example) in order to retrieve the corresponding content should be generated in this case.

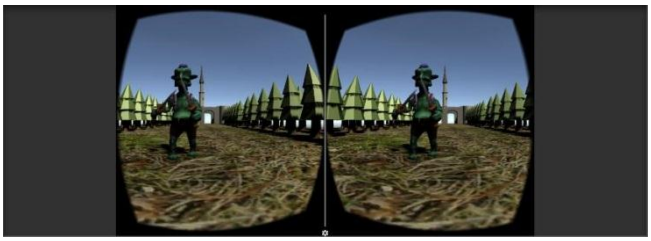


Fig. 6. Dynamic storytelling contextualization scenario screenshoot.

Demo video presenting the application capabilities in live action is available online: https://www.youtube.com/watch?v=30_v01R52Vc.

## VI. Conclusion

The proposed solution is portable and cost effective. It is designed to be compatible with wide variety of smartphones and no dedicated device is needed. Any smartphone satisfying the requirement in terms of compatible operating system and gyroscope presence could be used for this purpose. Cardboard glasses are affordable, available in different forms and shops, in most cases

lightweight which is also important, as the targeted group of patients are actually children.

Multimodality and adaptivity in combination with storytelling approach is what makes this solution stand out. Dynamic response tailored to the characteristics of specific patient makes this approach even more effective.

Finally, what makes this project interesting for research is the fact that it includes results and methods from different areas of computer science and related areas – such as natural language processing, computer graphics, image processing, audio signal analysis, machine learning, artificial intelligence, statistics, information retrieval, even data mining, machine learning and many others. Furthermore, it involves the conclusions and results from psychology. As it is based on strong multimedia capabilities of modern smartphones – there is a space for work of sound and graphic designers also.

To sum up, this approach is innovative and interesting way to support traditional therapy methods targeting children with NDD, as it uses state-of-the-art technology in an entertaining way to keep attention of the patient, which is an important factor when it comes to dealing with this kind of diseases, but is also a relevant and challenging research area, due to its novelty and interdisciplinarity.

## ACKNOWLEDGEMENT

The development of the application was carried out as an assignment related to the course Advanced User Interfaces (academic year 2016/17) at Politecnico di Milano, thanks to the course teacher, professor Franca Garzotto, teaching assistant Daniele Occhiuto and my colleagues Simone Lazzaretti and Mattia Crippa, who helped me during the project implementation phases.

## REFERENCES

[1] N. Newbutt, C. Sung, H.-J. Kuo, M. Leahy, C.-C. Lin, B. Tong, „Brief Report: A Pilot Study of the Use of a Virtual Reality Headset in Autism Populations“, in Journal of Autism and Developmental Disorders, 2016, pp. 1-11.

[2] F. Garzotto, M. Gelsomini, F. Clasadonte, D. Montesano, D. Occhiuto, „Wearable Immersive Storytelling for Disabled Children“, in Proceedings of the International Working Conference on Advanced Visual Interfaces AVI'16, Italy, 2016, pp. 196-203.

[3] Y. Cheng, C.-L. Huang, C.-S. Yang, „Using a 3D Immersive Virtual Environment System to Enhance Social Understanding and Social Skills for Children With Autism Spectrum Disorders“, in Focus on Autism and Other Developmental Disabilities 2015. Vol. 30, USA, 2015, pp. 222-236.

[4] F. Garzotto, „Multimodal Adaptive Interaction with Wearable Immersive Virtual Reality for Children's Learning (MAWI)“, in Advanced User Interfaces (Politecnico di Milano) 2016/17, Italy, 2016, pp. 1-4.

# Tool for Modelling and Automatic Deployment of Data Intensive Applications

Nenad Petrović

*Abstract* – Data-intensive applications (DIAs) and the Big data assets behind them are key to innovation in many industrial sectors and scientific areas. However, they require much effort not only in design, but there are also struggles how to deploy, test, optimize, redeploy and operate such complex systems and the infrastructure behind them. In this paper we focus on automating the deployment and redeployment activities starting from simple UML-alike deployment diagrams and developing a tool based on the infrastructure-as-a-code approach, which bridges the deployment design with fully automated, model-driven deployment.

## I. INTRODUCTION

Data-intensive applications are rapidly becoming key enablers for any industry, from automotive and entertainment to aerospace. However, they require continuous architectural design, framework/infrastructure configuration and deployment-testing to fine tune software and underlying resources by means of (re-)configuration.

However, the adoption of big data technologies is not simple to achieve. At the moment there are no simple-to-use tools that support designers and operators in configuring a data intensive application to use one or more of the available big data frameworks, and in deploying it on proper resources. The result is that such designers and operators have to go through a cumbersome trials and errors continuous architecting process where they iteratively tune the configurations of data intensive applications and of their underlying infrastructure until they reach satisfactory results. These exercises are, at the same time, very expensive (ad-hoc infrastructure needs to be rented and exploited for thousands of euros), very complex (they have to combine the complexity of the technological framework with concerns such as privacy, legal issues, data quality, etc.) and, finally, very slow (the execution of said exercises is reportedly still carried out mostly manually and by trial-and-error) [1] [2]. Of course, things become even more complicated when we need to re-design and re-configure our application to change the underlying used technology.

The goals behind this project are to support the above scenarios with model-driven, DevOps-fashioned tool and methods for actionable and continuous data-intensive design and deployment, which aims to speeds up the key phases of data intensive continuous architecting by aiding the configuration of data intensive frameworks and underlying infrastructures while fully automating the generation of deployment blueprints from design models [2]. The tool produces actionable blueprints in TOSCA, the "Topology and Orchestration Specification for Cloud Applications" to guarantee executability on any TOSCA-enabled orchestrator.

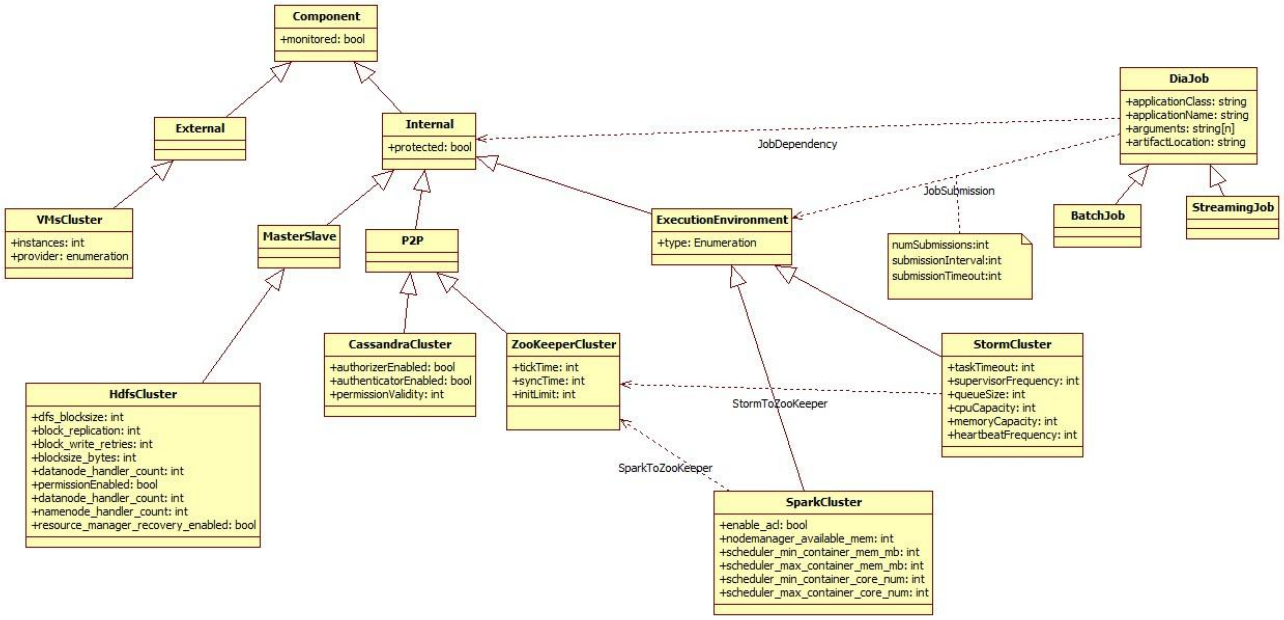



Fig. 1. UML diagram of the metamodel behind the tool.

Nenad Petrović, M.Sc. in Computer Science and Engineering, Politecnico di Milano, Piazza Leonardo da Vinci, 32, 20133 Milano, Italy, E-mail: penenadpi@gmail.com

## II. BACKGROUND

The tool is based on following principles [1]:

- *Model-Driven Engineering:* principles and tools may allow designers to use models for analyzing and optimizing their architectural configuration, and to abstract from the specific details of big data technologies delegating to some automation steps the generation of needed code and scripts.

- *Infrastructure-as-a-code:* approach where the code (possibly automatically generated from the model) is able to provide precise instructions to management tools that support the automatic deployment and configuration of applications on the specified resources [2]. A predefined set of TOSCA-compliant [3] elements defining the deployment and configuration instructions for a representative set of Big Data frameworks. This set of elements can be easily combined and instantiated to compose the code to deploy complex data-intensive applications.

- *UML metamodel:* behind the modelling tool, there is a metamodel (Figure 1) which consists of concepts common for data-intensive applications and relations between them. Among these concepts, we have both hardware and software abstractions. A data-intensive application is typically composed of multiple computational components (we call them *jobs*), acquiring data from one or more data sources and producing data to one or more data sinks. Data sources and data sinks can be of various kinds (for instance, GUIs, files, databases, streams, other systems). Data-intensive applications can work in batch, stream or interactive mode. This has an impact on how data is acquired and produced from sources/sinks and on individual component lifetime. Several big data technologies exist that aim at providing specific frameworks where modern DIAs can live and operate. In our modelling environment, several of them are considered: HDFS [4], Cassandra [5], ZooKeeper [6], Spark [7] and Storm [8]. These technologies are typically distributed systems running on top of clusters of computational resources -virtual machines (VMs) within a Cloud infrastructure. Therefore, a concept of virtual machine cluster is introduced. It includes various parameters related to configuration in order to give more flexibility (such as number of CPUs, memory allocated, storage size, provider etc.)

In what follows, the most important metamodel elements [1] are presented and described in Table 1.

TABLE I
METAMODEL ELEMENTS OVERVIEW

| Name | Description |
|---|---|
| Component | The highest-level abstraction of the application element – virtual machine or application. Model is a set of Components. |
| External Component | Components owned by Cloud provider. |
| Internal Component | Components owned by application provider. Each of the internal components needs to be inside one of the external components. |
| VMsCluster | It is a specific External Component, representing the well-known concept of virtual machine. It has been customized in context of DICER -represented by a collection of virtual machines on a particular provider platform. |
| MasterSlave | A specialization of InternalComponent, which includes application platforms based on master-slave architecture. |
| P2P | A specialization of InternalComponent, which includes application platforms based on peer-to-peer architecture. |
| Execution Environment | A specialization of InternalComponent which includes both master-slave and peer-to-peer components, with addition that they are able to be the execution environment, meaning that data-intensive application jobs can be submitted for execution to them. |
| DiaJob | Represents a concrete instance of data-intensive application, which is submitted to ExecutionEnvironment. |
| BatchJob | A specific type of DiaJob, covering the cases where we have batch processing. |
| StreamingJob | A specific type of DiaJob, covering the cases where we have streaming processing. |
| JobSubmission | Assigning a DiaJob to an InternalComponent which belongs to ExecutionEnvironment. |
| JobDependency | A relationship to denote a dependency between any DiaJob and InternalComponent (including the one that is not ExecutionEnvironment) |
| HdfsCluster  | Special type of master-slave InternalComponent designed specifically for storing and analysing huge amount of unstructured data in distributed computing environment. |
| CassandraCluster  | Apache Cassandra is a free and open-source distributed NoSQL database management system designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure. It is an InternalComponent which belongs to P2P category. |

| | |
|---|---|
| ZooKeeperCluster  | ZooKeeper is a Distributed Coordination Service for Distributed Applications. ZooKeeper allows distributed processes to coordinate with each other through a shared hierarchal namespace which is organized similarly to a standard file system. It belongs to P2P category of Internal Components. |
| SparkCluster | Apache Spark is a fast and general engine for big data processing, with built-in modules for streaming, SQL, machine learning and graph processing. It is an InternalComponent, based on master-slave architecture which is also an ExecutionEnvironment and is able to accept job submissions. |
| StormCluster | Apache Storm is a free and open source distributed realtime computation system. Storm makes it easy to reliably process unbounded streams of data, doing for realtime processing what Hadoop did for batch processing. It is an InternalComponent, based on master-slave architecture. which is also an ExecutionEnvironment and is able to accept job submissions. |
| StormToZooKeeper | StormCluster and ZooKeeperCluster dependency relationship. Each StormCluster needs to be connected to a ZooKeeperCluster. |
| SparkToZooKeeper | SparkCluster and ZooKeeperCluster dependency relationship. Each SparkCluster needs to be connected to a ZooKeeperCluster. |

## III. IMPLEMENTATION

The tool is utilizing model-to-model transformations and infrastructure-as-a-code approach in order to automatically deploy the data-intensive application represented by user-created diagram. Figure 2 presents the overview of components involved and operations performed to deploy the data-intensive application given by a model, using the tool we developed.

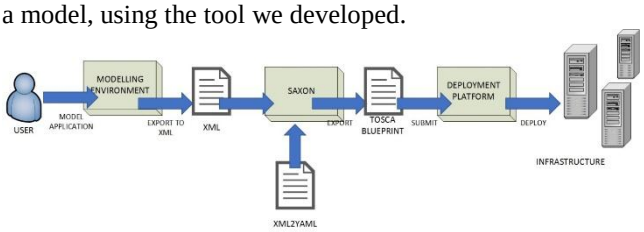



Fig. 2. Working principle overview.

First, user models a data-intensive application using a set of given concepts and relationships between them inside the modelling environment. The modelling environment is developed using ADOxx meta-modelling platform [9] which provides the necessary capabilities to develop a GUI-enabled modelling tool based on UML-alike representation. The modelling primitives are described using ADOxx Definition Language (a meta-modelling language used by ADOxx platform).

Once the modelling is done, it is possible to check if the model is valid. A set of model verification rules based on the constraints of certain technologies are implemented in order to achieve this. After that, if the considered model is valid, the necessary transformations can be done and model deployed. The constraints are also described using ADOxx Definition Language. Figure 3 displays the screenshot of the modelling environment.

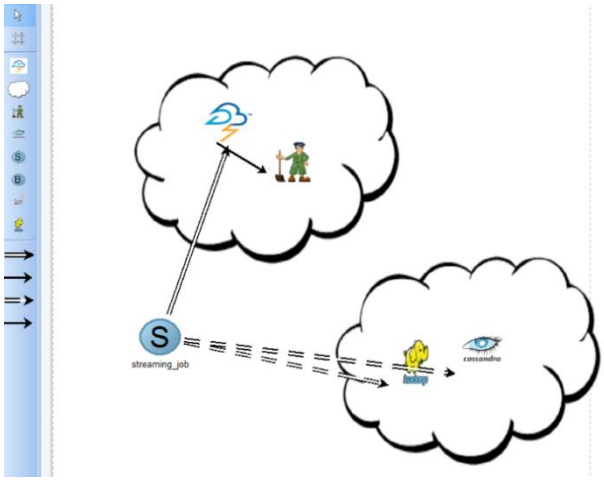



Fig. 3. Modelling environment screenshot.

Before the deployment phase, the model is internally exported to XML format. Then, it is converted to TOSCA YAML blueprint using XSLT [10] model-to-model transformation (from XML to TOSCA YAML). Saxon engine [11] is responsible for performing the transformation. In what follows, the pseudocode of the XSLT transformation is given (Listing 1).

```
foreach(ExternalComponent external in model)
{
    generateHostNodeTemplate(external);
    foreach(InternalComponent node in
external.nested)
    {
    if(node.isProtected)
        {
            generateFirewall(external,node);
        }
    }
}
foreach(InternalComponent node in model)
{
    switch(node.type)
    {
            case 'StormCluster':
               generateSlavesNode(node);
               generateMasterNode(node);
        generateAdditionalStormNodes(node);
```

```
        case 'ZooKeeperCluster':
            generatePeersNode(node);
    generateAdditionalZooKeeperNodes(node);
        ...
    }

}
foreach (DiaJob job in model)
{
    generateJobConfiguration(job);
}
```

Listing 1. Pseudo-code of the XSLT transformation.

Finally, the generated TOSCA blueprint is submitted to the deployment platform and application is deployed to the infrastructure. The deployment platform is developed by XLAB.

## IV. Evaluation

To check that tool works correctly without introducing a significant overhead, we have experimented with various data-intensive applications that exploit different combinations of the currently supported technologies. Furthermore, we compared the times obtained for setting up the same application with and without tool to figure out how much is the speed up of the whole procedure (from deployment diagram to the actual application running).

Table 2 shows a summary of results obtained during the experiments. In all cases, we developed the deployment model for the corresponding DIA, relying on the consistency and validation check controls embedded in tool, and then we generated and deployed the resulting blueprint on our internal OpenStack infrastructure. In all considered cases, the process led to a correct deployment fulfilling our expectation.

The first column in Table 2 includes the names we have assigned to the considered data-intensive application and the second lists the used technologies. The *Max depth* column represents the length of the longest path in a blueprint's dependency graph, not counting any virtual resources that are created almost instantly. For example, level 1 represents an application with services that are all peers and where none of the services depends on any other service. Level 2 represents a two-tier application, where a service (for example, a database) needs to be deployed before another one (the main application) can be deployed and configured. Thus, in general, *Max depth* in conjunction with the used technologies gives an idea of the complexity of the system under deployment. The most complex case we considered features four layers and includes three different and distributed DIA technologies. To quantify the time needed to generate and deploy blueprints, we repeated the experiments 10 times and computed the average duration for generating the blueprint (fourth column in the table), for deploying it (fifth column), and for tearing it down (sixth column, this is the time from when we started the teardown procedure to when the tool deployment service reported termination).

From the Table 2, we can see that blueprint generation time strictly depends on the number of used technologies, the size of a model, while the deployment times are roughly correlated with the max level of dependence.

Table 2.
Metamodel elements overview

| Blueprint | Tech. | Max Depth | Gen. Time [s] | Deploy time [s] | Tear down [s] |
|---|---|---|---|---|---|
| Pi Calculus | Spark | 3 | 1.13 | 452 | 58 |
| Word Count | ZooK. Storm | 3 | 1.32 | 460 | 72 |
| Wiki Stats | Storm ZooK. Cassandra | 4 | 1.87 | 482 | 86 |

After that, we compared the time spent to completely set up the application using the tool and without it. The application used in this experiment was WikiStats. It took 4 hours to set it up manually [1], without any tool intervention. Note that this estimate assumes that user is experienced and using and some form of assisted installation procedure/script. Furthermore, the same exercise carried out by students without any prior knowledge, required up to a full day of effort (around 12 hours) [1].

However, using the tool in this procedure, we have a different situation: installing the same cluster, running the application and modifying after several runs takes around 30 minutes and requires little or no user interaction at all, which is up to 8x times faster [1].

## IV. Conclusion

This approach helps users in designing the deployment of DIAs that exploit and combine different Big Data frameworks, supporting the DevOps cycle (deploy, test, optimize, redeploy and operate) by enabling the automated creation of executable blueprints from deployment diagrams, which makes the whole process faster and more intuitive.

The evaluation has shown reduces up to 8x the times related to (re-)deployment and (re-)configuration of DIAs. Also, the tool introduces negligible performance overheads connected to modelling, generation, and deployment of infrastructure-as-code for DIAs.

Currently, the tool supports some of the most well known big data frameworks. Adding new frameworks is possible and we have been constantly doing this during the research, however, it requires an in-depth knowledge of the

approach and tool. As a future work, support for even more technologies is planned.

## ACKNOWLEDGEMENT

The development of the tool was carried out as a result of activities related to *150 hours tutoring* research position at Politecnico di Milano (second semester of academic year 2016/17), under supervision of professor Elisabetta di Nitto and PhD student Michele Guerriero, as a part of DICE H2020 research and innovation project [12].

## REFERENCES

[1] E. Di Nitto, D. Tamburi, M. Guerierro, M. Artac, T. Borovsak, „Developing Data-Intensive Cloud Applications with Iterative Quality Enhancements – 2.4 Deployment Abstractions“, [On Line]. Available on: http://wp.doc.ic.ac.uk/dice-h2020/wp-content/uploads/sites/75/2017/07/D2.4_Deployment-abstractions-Final-version.pdf

[2] E. Di Nitto, E. Di Nitto, D. Tamburi, M. Guerierro, M. Artac, T. Borovsak, "DevOps: introducing infrastructure-as-code", in Proc. 39th International Conference on Software Engineering Companion ICSE-C'17, Argentina, 2017, pp. 497-498.

[3] Apache ARIA TOSCA Orchestration Engine, [On Line]. Available on: http://ariatosca.incubator.apache.org/

[4] Apache HADOOP, [On Line]. Available on: http://hadoop.apache.org/

[5] Apache Cassandra, [On Line]. Available on: http://cassandra.apache.org/

[6] Apache ZooKeeper, [On Line]. Available on: https://zookeeper.apache.org/

[7] Apache Spark, [On Line]. Available on: https://spark.apache.org/

[8] Apache Storm, [On Line]. Available on: http://storm.apache.org/

[9] ADOxx Metamodelling Platform, [On Line]. Available on: https://www.adoxx.org/live/meta-modelling-platforms-hierarchy

[10] XSLT intro, [On Line]. Available on: https://www.w3schools.com/xml/xsl_intro.asp

[11] Saxon XSLT, [On Line]. Available on: http://saxon.sourceforge.net/

[12] Developing Data-Intensive Cloud Applications with Iterative Enhancements, [On Line]. Available on: http://www.dice-h2020.eu/

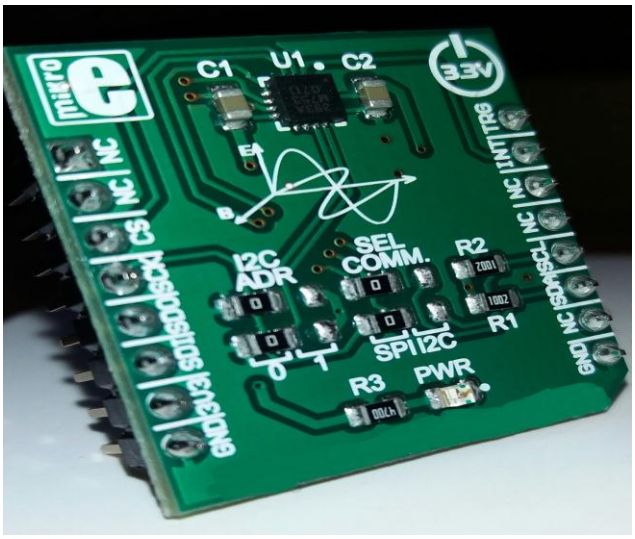
# Senzor za merenje magnetne indukcije

Luka Spahić

*Sadržaj* – Senzor opisan u ovom radu služi za merenje magnetne indukcije pomoću integrisanog kola MLX90393. Rad senzora je baziran na principu Hall-ovog efekta. Prenos izlaznih signala (mereni podaci temperature i magnetne indukcije za X, Y, Z osu) je obezbeđen preko brzih I$^2$C protokola ili preko SPI protokola.

## I. UVOD

Ukoliko je potrebno izmeriti intenzitet magnetne indukcije u okolini električnih instalacija, električnih uređaja (TV-a, računara...), automobila ili prostorije, senzor za merenje magnetne indukcije predstavlja jednostavno i praktično rešenje. Značaj ovog senzora je u tome što se njegov rad zasniva na primeni Hall-ovog efekta i što je celokupan senzor izrađen u integrisanoj tehnici i time je obezbeđena ušteda prostora. Pored merenja magnetne indukcije, postoji i mogućnost merenja temperature uz pomoć temperaturnog senzora koji se nalazi u samom integrisanom kolu. Senzor, takođe ima mogućnost odabira dve vrste protokola za komunikaciju, I$^2$C ili SPI ,što omogućava slobodan odabir načina komunikacije od strane korisnika. Celokupan senzor (Slika1) se sastoji od integrisanog kola MLX90393, jumpera, tri otpornika, dva kondezatora i jedne zelene LE diode za indikaciju rada. Senzor je povezan na SPI bus.

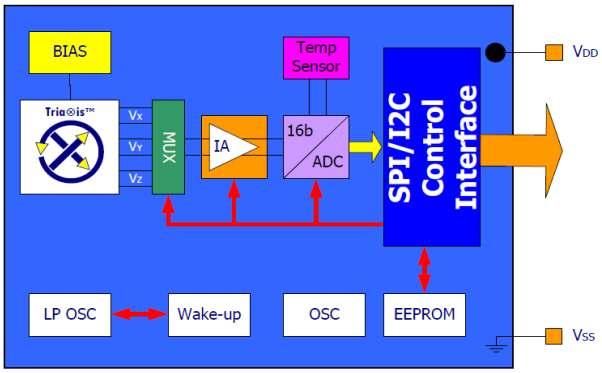



Slika 1. Izgled senzora za merenje magnetne indukcije.

Luka Spahić, student III godine osnovnih akademskih studija na Elektronskom fakultetu, smer Elektronske komponente i mikrosistemi, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: luka.spahic@gmail.com

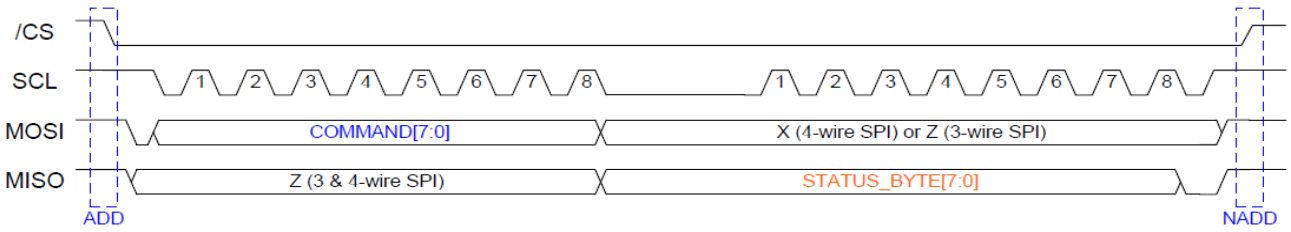
## II. PRINCIP RADA

Kao što je već navedeno, rad samog senzora baziran je na principu Hall-ovog efekta. Prilikom ubacivanja senzora u strano magnetno polje, unutar integrisanog kola dolazi do promene potencijala $V_X$, $V_Y$, $V_Z$ koji je proporcialan promeni magnetnog fluksa. Nadalje se analogna informacija obrađuje i šalje na AD konvertor. Na slici 2 je predstavljen blok dijagram samog integrisanog kola MLX90393. Na osnovu dijagrama možemo primetiti da se u sklopu MLX90393 nalazi i senzor temperature koji je takođe povezan sa analogno-digitalnim konvertorom.



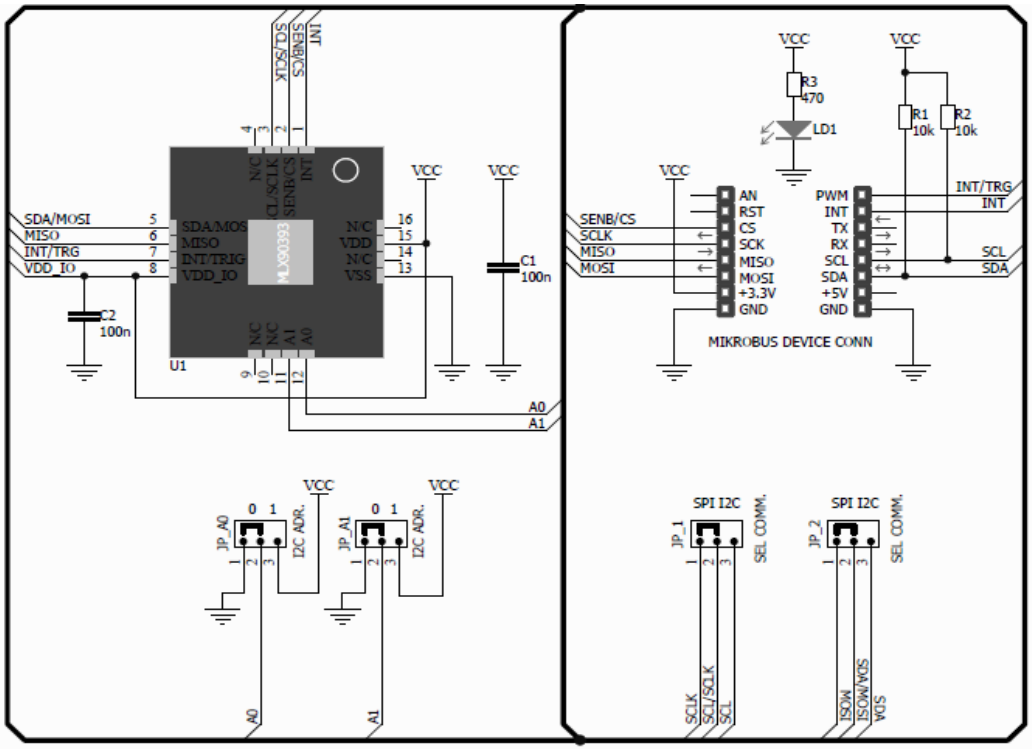

Slika 2. Blok dijagram integrisanog kola MLX90393.

Analogno-digitalnom konverzijom, na izlazu integrisanog kola dobijamo 16-bitni podatak koji se prosleđuje kontrolnom interfejsu. Trenutno, je senzor podešen tako da se komunikacija odvija preko SPI linija (SDI, SDO, SCK, CS). MLX90393 podržava SPI komunikaciju brzine do 10 MHz. Možemo reći da SPI komunikacija, u ovom slučaju, pokazuje velike sličnosti sa I$^2$C komunikacijom, s tim što se adresiranje odvija preko CS (*Chip Select*) pina, a ne preko arbitraže bus-a. Da bi izvršili komunikaciju sa senzorom, pin CS mora biti na niskom naponskom nivou. Komunikacija se odvija slanjem paketa bajtova, jednako MSB first (Bit najveće težine) i MSByte first. Informacija koja se šalje, može sadržati i više od jednog bajta, kao i povratna informacija u formi višestrukih bajtova koji se redaju iza statusnog bajta. Na slici 3 je prikazan primer SPI komunikacije. Kada je CS na niskom naponskom nivou, aktivira se SCL (*clock* signal) što znači da je omogućeno slanje 8-bitne komande ili više 8-bitnih komandi.
Odmah po slanju komande, preko SPI busa, senzor prosleđuje statusni bajt mikrokontroleru.




Slika 3. Primer SPI komunikacije.

U zavisnosti od vrste komande koju je mikrokontroler prosledio senzoru, zavisi i to da li će se iza statusnog bajta nalaziti dodatni paketi bajtova koji bi nosili podatke o izmerenim vrednostima magnetne indukcije ili temperature. Kada se završi ciklus pisanja/čitanja pin CS se setuje na visok naponski nivo. Eksterni pull-up otpornici, $R_1$ i $R_2$ otpornosti 10 kΩ vezani za pinove SDA i SCL, služe da obezbede stalni visoki naponski nivo i mogućnost rada samog senzora prilikom $I^2C$ komunikacije. Da bi se prešlo sa SPI-a na $I^2C$ komunikaciju, dovoljno je na pločici podesiti jumpere SEL COMM. na naznačeno mesto za $I^2C$ (u desno). Jumperi I2C ADR služe za selektovanje adrese. Kondenzatori $C_1$ i $C_2$ kapacitivnosti 0.1 μF, nalaze se na gornjoj strani pločice i služe za stabilan rad integrisanog kola. Otpornik $R_3$ otpornosti 470 Ω služi da ograniči struju kroz LE diodu (LD1). LE dioda (LD1) služi kao indikator da je senzor pravilno priključen na napajanje. Na slici 4 se može videti električna šema senzora i kompletan raspored komponeti. Treba napomenuti da će senzor ispravno raditi isključivo pri naponu od 3.3 V. Veće ili manje vrednosti napona mogu dovesti do nepravilnog rada samog senzora, a napon veći od 4 V dovodi do oštećenja integrisanog kola.



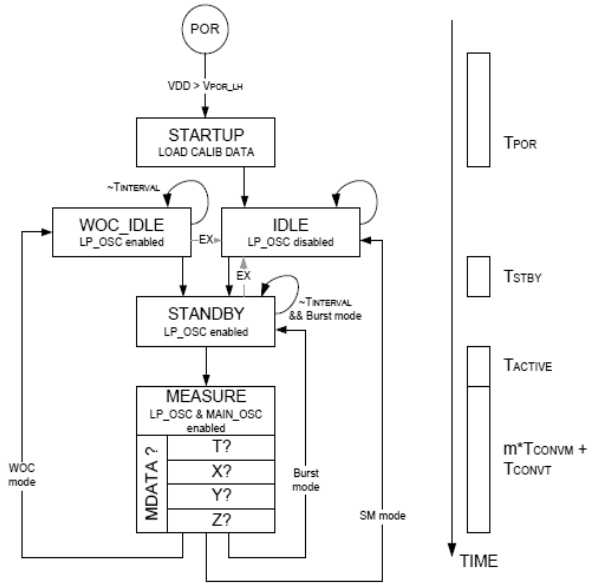

Slika 4. Električna šema senzora.

## III. FUNKCIONALNE SPECIFIKACIJE SENZORA I PISANJE NAREDBE ZA MERENJE PODATAKA

Senzor može funkcionisati u tri režima rada: *Burst mode, Single Measurement mode* i *Wake-up on Change mode.* Kada je podešen u *Single Measuremant mode-u*, preko bus-a mikrokontroler od senzora zahteva da izvrši merenje, slanjem komande ili slanjem okidnog signala. Kod *Burst mode-a*, senzor iz neaktivnog stanja prelazi u stanje čekanja na komandu za merenje. Posle izvršene komande za merenje vraća se u stanje čekanja. Ovakav radni ciklus je programabilan. I konačno, *Wake-up on Change mode* je isti kao i *Burst mode*, s tim što kad god je pin INT na visokom naponskom nivou (logička jedinica),

znači da vrednost poslednjeg merenja odstupa od vrednosti praga prethodnog merenja ili vrednosti merenja odmah posle ulaska u *Wake-up on Change mode*. Slika 5 predstavlja vremenski tok izvršavanja naredbi za sva tri mode-a.

Naredbe za startovanje željenog režima rada kodovane su heksadekadnim ciframa oblika:
Za *Burst mode*: 0x1<zyxt>,
Za *Wake-up on Change mode*: 0x2<zyxt>,
Za *Single Measurement mode*: 0x3<zyxt>,
gde <zyxt> predstavlja naredbu kojom se senzoru govori koje tačno merenje treba da izvrši. Na primer, ukoliko bismo mu preko SPI-a poslali naredbu 0x3F, mi bismo senzoru naredili da startuje *Single Measurment mode* i izvrši merenje za x, y, z osu i temperaturu. Treba napomenuti da se merenje izvršava obrnutim redosledom od <zyxt>, što znači da će se prvo meriti temperatura, pa vrednost x, zatim y i na kraju z ose. Na slici 6 nalazi se deo koda koji jasno pokazuje slanje naredbe preko SPI-a da bi se izvršilo merenje u Single Measurment modu.




Slika 5. Vremenski tok izvršavanja.

```
Chip_Select = 0;
SPI3_Write(0x3F);
take1 = SPI3_Read(0);
Chip_Select = 1;
```

Slika 6. Deo koda napisan u mikroC-u for PIC32.
Slanje naredbe za merenje vrednosti T, X, Y, Z.

Posle naredbe pisanja, senzor vraća povratnu informaciju u vidu status bajta koji govori da li se komanda uspešno izvršila. U „take1“ promenljivu tipa integer smešta se podatak status bajta. Vrednost status bajta treba da bude binaran (00100011), odnosno dekadan broj koji izosi 35. Statusni bajt nosi 8-bitnu informaciju o tome koji je mod aktivan, da li se komanda uspešno izvršila, da li je došlo do resetovanja senzora, koliko se bajtova šalje iza statusnog bajta. Slika 7 pokazuje izgled statusnog bajta i ulogu svakog bita. Sedmi, šesti i peti bit govore o tome koji je režim aktivan. Kada je aktivan *Single Measurement mode-u* petom bitu se upisuje „1“, dok u šestom i sedmom bitu „0“. Ukoliko dođe do greške u slanju naredbi ili čitanju merenih vrednosti, u četvrtom bitu će biti upisana „1“, a ako je jedan bit greške ispravljen, u trećem bitu biće takođe upisana „1“. Drugi bit služi za indikaciju resetovanja. U zavisnosti da li smo preko komande za čitanje izmerenih vrednosti odabrali čitanje x ,y ,z ose ili temperature, u nultom i prvom bitu će biti upisana nula ili jedinica.

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| Burst_mode | WOC_mode | SM_mode | ERROR | SED | RS | D1 | D0 |

Slika 7. Statusni bajt.

Prema formuli:

$$2 \cdot D[0:1] + 2 \qquad (1)$$

možemo izračunati koliko bajtova treba da očekujemo posle statusnog bajta. Na primer, kada želimo da pročitamo samo y osu i temperaturu, u prvom bitu će biti upisana nula, a u nultom bitu jedinica. Ovakva binarna kombinacija daje dekadni broj jedan. Primenom formule (1) umesto promenljive D[0:1] pišemo broj 1. U nastavku dobijamo konačnu vrednost 4. Vrednost četiri govori da posle status bajta očekujemo još četiri bajta podataka.

## IV. ČITANJE PODATAKA I OBRAĐIVANJE PRIMLJENIH INFORMACIJA

Čitanje izmerenih vrednosti za x, y, z osu i temperaturu se odvija naredbom *Read Measurement* heksadekadnog oblika 0x4< zyxt >. Na primer, ukoliko bi smo preko SPI-a poslali kodovanu naredbu 0x4F (pod uslovom da se pre toga izvršila naredba za merenje) senzor bi kao povratnu informaciju poslao statusni bajt, a iza njega niz paketa podataka, posebno za temperaturu i x, y, z osu. Slanje podataka odvija se istim redom kao i redosled merenja navedenih veličina. Za svaku veličinu posebno, šalju se dva bajta podataka, Bajt najveće težine(MSB) i Bajt najniže težine(LSB). Slika 8 ilustruje niz povratnih paketa podataka koje senzor šalje mikrokontroleru konketno za gore navedenu naredbu.

| Statusni Bajt | MSB temperatura | LSB temperatura | … | LSB Z-osa |
|---|---|---|---|---|

Slika 8. Niz bajtova koji se prosleđuju mikrokontroleru posle naredbe čitanja merenih podataka.

Informacija koju senzor, posle naredbe *Read Measurement*, šalje mikrokontroleru se smešta u niz. Slika 9 prikazuje primer koda za čitanje podataka i smeštanje podataka u niz.

```
Chip_Select = 0;
SPI3_Write(0x4F);
MES[0]=SPI3_Read(0);
MES[1]=SPI3_Read(0);
MES[2]=SPI3_Read(0);
MES[3]=SPI3_Read(0);
MES[4]=SPI3_Read(0);
MES[5]=SPI3_Read(0);
MES[6]=SPI3_Read(0);
MES[7]=SPI3_Read(0);
MES[8]=SPI3_Read(0);
Chip_Select = 1;
```

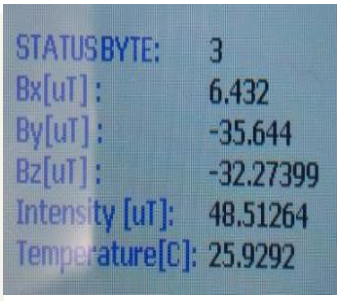
Slika 9. Čitanje podataka sa senzora i smeštanje podataka u niz.

U daljem tekstu sledi konverzija i spajanje podataka. Svaki neparni (MSB) element niza pomeramo (množimo sa 256) za 8 bita u levo i operacijom ILI (sabiranjem) spajamo sa parnim elementom niza (LSB). Tako spojene podatke smeštamo u odgovarajuće promenljive. Prema podacima iz tehničke dokumentacije iz odgovarajućih tabela očitavamo odgovarajuću rezoluciju i osetljivost. Očitanim vrednostima množimo promenljive. Ovo važi samo za promenljive u kojima su smešteni podaci x, y, z ose (Slika10). Za tačan prikaz temperature potrebno je izvršiti temperaturnu kompenzaciju pomoću sledeće formule:

$$T[°C] = 35 + \frac{Tmereno + 46996}{45.2} \qquad (2)$$

gde *Tmereno* predstavlja vrednost koja je pročitana iz niza podataka (nulti i prvi element niza). Formula se nalazi u tehničkoj dokumentaciji. Vrednosti za x, y, z osu, temperatura i intenzitet magnetne indukcije se ispisuju na TFT ekranu (Slika10).

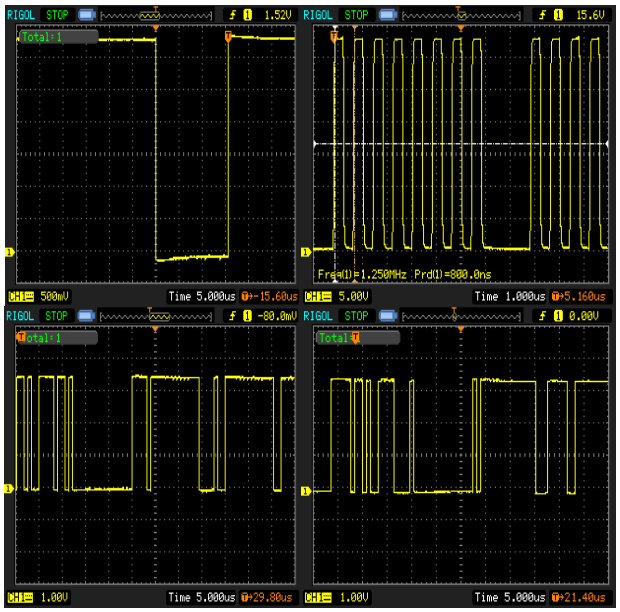
```
Tmereno=((MES[1]*256) + (MES[2]));
x1= ((MES[3]*256) + (MES[4]));
y1= ((MES[5]*256) + (MES[6]));
z1= ((MES[7]*256) + (MES[8]));
x=x1*0.268;
y=y1*0.268;
z=z1*0.489;
Tdigital=((Tmereno-46696));
Tdigitalfloat=Tdigital/45.2;
Temperatura=35 + Tdigitalfloat;
Bkvad=((x*x)+(y*y)+(z*z));
B=sqrt(Bkvad);
```




Slika 10. Kod za obradu podataka i prikaz merenja magnetne indukcije i temperature u prostoriji na TFT-u ekranu.

## V. PRIKAZ SIGNALA SA SPI PINOVA

Na slici 11 prikazani su signali redom CS, SCK, MOSI i MISO pinova snimljeni digitalnim osciloskopom. Izgled signala je snimljen kada je senzor podešen na SPI komunikaciju. Senzor je testiran pomoću razvojnog okruženja mikroC PRO for PIC32 i mikrokontrolera P32MX795F512L.



Slika 11. Prikaz signala CS, SCL (clock) , MOSI, MISO (sa leva na desno).

## VI. ZAKLJUČAK

Senzor je moguće realizovati i na sasvim drugačiji način. Zbog veličine samog integrisanog kola celokupan senzor je kompaktan i praktičan. Takođe sam senzor ima veoma malu potrošnju. Dalje, senzor se može reprogramirati i iskoristiti u razne svrhe kao džojstik, elektronska poluga, indikator u raznim kućnim i industrijskim objektima i druga naučna istraživanja.

## ZAHVALNICA

Autor se zahvaljuje kompaniji MikroElektronika.

## LITERATURA

[1] https://www.melexis.com/en/documents/documentation/datasheets/datasheet-mlx90393
[2] Hall Effect sensors. Theory and Applications by Edward Ramsden. Dostupno na: https://books.google.rs
[3] An introduction to I2C and SPI protocols by Frederic Leens. Dostupno na: http://ieeexplore.ieee.org
[4] https://makezine.com/2009/03/05/understanding-pullup-and-pulldown-r/
[5] https://learn.mikroe.com/ebooks/8051programming/chapter/spi-system-serial-peripheral-interface/
[6] S. Stojković, N. Stojanović, D. Stojanović, Uvod u računarstvo, Elektronski fakultet Niš, 2014.

# Meteorološka stanica – Data Logger

Miloš Đorđević, Branislav Jovičić, Bojan Milosavljević

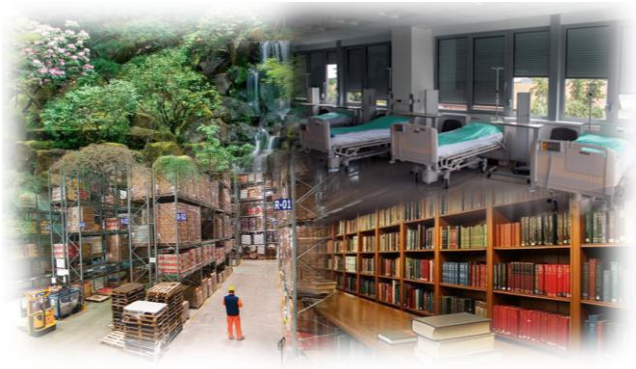
*Sadržaj* – U ovom radu je predstavljen sistem za skladištenje podataka dobijenih merenjem različitih fizičkih parametara u atmosferi. Realizovani uređaj je prevashodno namenjen korišćenju u poljoprivredi i meteorološkim stanicama, ali se njegova primena tu ne ograničava.

## I. UVOD

Meteorološka stanica – Data Logger je uređaj namenjen za meteorološka merenja i osmatranja i skladištenje podataka, svuda gde je neophodno merenje i osmatranje ambijentalnih parametara, kao što su proizvodne hale, rudnici, bolnice, biblioteke, muzeji, botaničke bašte i sl. (slika 1.). Uređaj ima široku primenu, na osnovu meteoroloških/ambijentalnih parametara koje meri:

- Temperatura [°C] ili [°F],
- Vlažnost vazduha [%],
- Atmosferski pritisak [mBar],
- Nadmorska visina [m],
- Brzina vetra (Anemometar) [m/s], [km/h],
- Osvetljenje [lux],
- Detekcija i merenje koncentracije gasa plina [%],
- Vreme [hh:mm:ss].

Podaci koji su izmereni, zajedno sa trenutnim vremenom se skladište na SD (Secure Digital) karticu u vidu datoteke .txt formata. Na osnovu toga, korisnik ima uvid u kom vremenskom trenutku je došlo do promene nekog od merenih parametara.



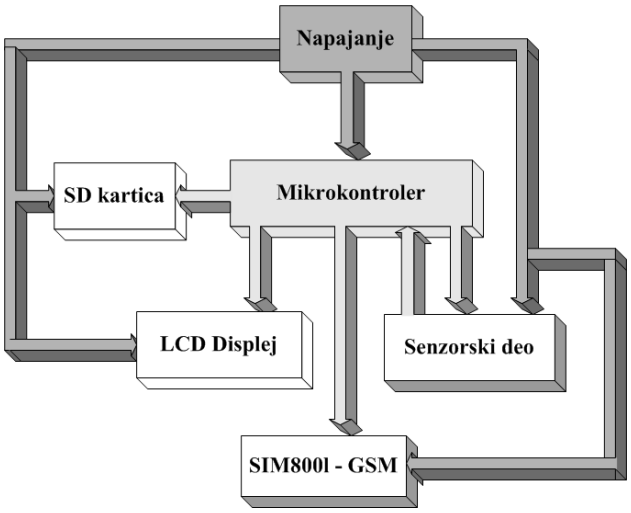
Slika 1. Oblasti primene Meteorološke stanice – Data Loggera.

Miloš Đorđević, student doktorskih studija na modulu Nanotehnologije i mikrosistemi, Elektronski fakultet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija. E-mail: milos.djordjevic@elfak.ni.ac.rs

Branislav Jovičić i Bojan Milosavljević, studenti osnovnih studija na smeru Računarstvo i Informatika, Elektronski fakultet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija. E-mail: badza.jo@gmail.com

## II. PRINCIP RADA

Meteorološka stanica – Data Logger je realizovan tako da se sastoji iz 6 segmenata (blokova (slika 2.)). Blok napajanja opslužuje sve ostale blokove, osim u slučaju napajanja pomoću solarne ćelije, gde se ne koristi LCD displej. Mikrokontroler PIC18F45K22 [1], koji predstavlja srž celokupnog uređaja, upravlja senzorskim blokom, koji služi za meteorološka merenja i osmatranja. Takođe, GSM blokom, realizovanog pomoću SIM800l modula [2], upravlja se gore navedenim mikrokontrolerom.

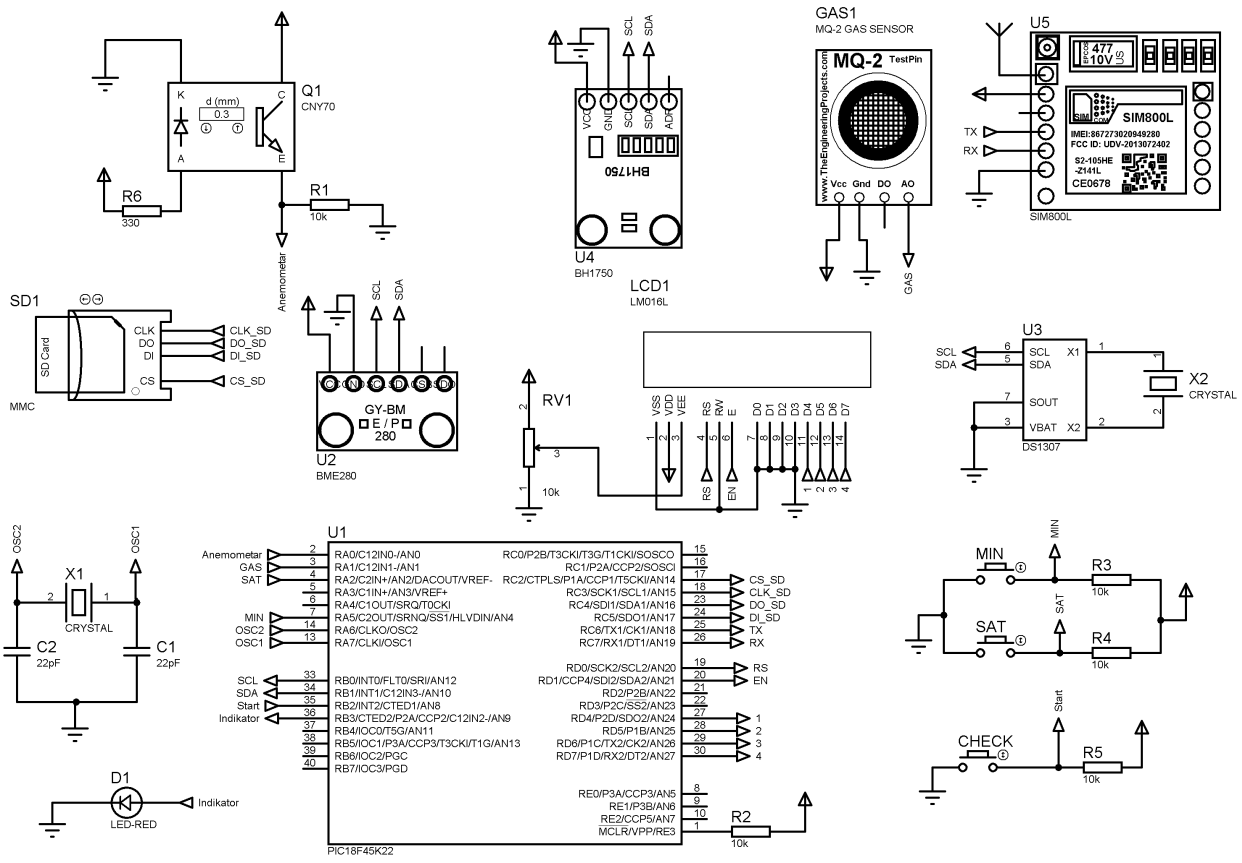



Slika 2. Blok šema Meteorološke stanice – Data Loggera.

Senzorski deo Meteorloške stanice – Data Loggera sastoji se iz sledećih senzora:

- BME280 [3]:
  - Temperatura [°C] ili [°F]
  - Vlažnost vazduha [%],
  - Atmosferski pritisak [mBar],
  - Nadmorska visina [m],
- Anemometar [4]:
  - Brzina vetra (Anemometar) [m/s], [km/h],
- BH1750 [5]:
  - Osvetljenje [lux],
- MQ-2 [6]:
  - Detekcija i merenje koncentracije gasa plina [%],
- DS1307 [7]:
  - Vreme [hh:mm:ss].

Električna šema Meteorološke stanice – Data Logger-a je data na slici 3.




Slika 3. Električna šema Meteorološke stanice – Data Logger-a.

Mikrokontroler korišćen za realizaciju Meteorološke stanice – Data Logger-a, radi na frekvenciji od 8 MHz, korišćenjem eksternog oscilatora. Za brzinu rada mikrokontrolera PIC18F45K22 moguće je koristiti visokofrekventni interni oscilator čija je najviša frekvencija 16 MHz, sa mogućnošću povećanja frekvencije 4 puta (*Primary Clock Frequency 4x multiplier* (PLLEN)) – na 64 MHz. Takođe, pruža se mogućnost korišćenja eksternog oscilatora, koji takođe ima mogućnost rada na frekvenciji od 64 MHz, uz korišćenje PLLEN množioca radne frekvencije oscilatora. Master reset (MCLR - *Master Clear Pin External Reset*) je softverski isključen, ali je u samoj realizaciji postavljen i pull-up otpornik vrednosti 10 kΩ, kako bi se videla mogućnost hardverskog isključenja MCLR pina na mikrokontroleru. Ovaj mikrokontroler poseduje 2 USART (*Universal Synchronous Asynchronous Receiver Transmitter*) modula [8]. Uz pomoć ovog modula je moguće hardverski upravljati modulima koji komuniciraju sa mikrokontrolerom preko I2C ili SPI magistrale, a takođe i preko RX/TX serijskog terminala.

Senzor BME280 (*BOSCH Sensortec*) kao što je navedeno korišćen je za merenje vlažnosti vazduha, temperature i atmosferskog pritiska, a na osnovu atmosferskog pritiska je određivana i nadmorska visina. Mikrokontroler PIC18F45K22 komunicira sa ovim senzorom preko I2C magistrale [9]. I2C magistrala je softverski realizovana koristeći RB.1 i RB.2, porta B korišćenog mikrokontrolera. BME280 barometarski senzor pruža široki opseg merenja:

- vlažnost vazduha od 0 do 100 %RH,
- temperatura od -40° do +85°C i
- atmosferski pritisak od 300 do 1100 hPa.

Anemometar, senzor za merenje brzine vetra, realizovan je korišćenjem senzora za merenje brzine obrtaja. Komunikacija ovog senzora sa mikrokontrolerom iskorišćen je RA.0 pin porta A mikrokontrolera. Za rad ovog senzora bila je neophodna kalibracija, kako bi se dobila vrednost brzine vetra.

Merenje osvetljenosti vrši se senzorom BH1750 (*ROHM Semiconductor*), koji sa mikrokontrolerom komunicira preko softverski realizovane I2C magistrale. Korišćeni senzor pruža široki opseg merenja visoke rezolucije (0-65535 lx).

Za detekciju gasa plina korišćen je MQ-2 senzor. Ovaj senzor pruža mogućnost digitalnog i analognog izlaza. U ovom slučaju korišćen je analogni izlaz i samim tim za rad sa ovim senzorom iskorišćen je pin RA.1 mikrokontrolera. Na portu A mikrokontroler poseduje A/D konvertor, koji se koristi za određivanje procenata gasa plina u okolini. MQ-2 senzor takođe nudi široku primenu, tj. mogućnost merenja

više različitih parametara, kao što su: TNG (LPG) – tečni naftni gas, CH4 – metan, CO – ugljen monoksid, alkohol, dim i propan.

DS1307 je integrisano kolo koje se koristi kao sat realnog vremena (*RTC – Real Time Clock*). Ovo kolo radi na frekvenciji od 32,768 kHz, korišćenjem eksternog oscilatora. Takođe, ovo kolo za komunikaciju sa mikrokontrolerom koristi softverski realizovanu I2C magistralu. RTC DS1307 se može koristiti kao sat i kao kalendar. Računanje vremena kreće od 00:00:00 u formatu hh:mm:ss. Podešavanje vremena se vrši uz pomoć tastera za sate i minute, realizovanih preko RA.2 (sati) i RA.5 (minuti) pinova mikrokontrolera. Ovi tasteri se takođe koriste za podešavanje rada Meteorološke stanice – Data Logger-a, gde se koristi još jedan dodatni taster realizovan preko RB.2 pina mikrokontrolera. Ovaj taster se koristi za odabir podešavanja režima rada Meteorološke stanice – Data Logger-a.

GSM/GPRS modul SIM800l služi za slanje SMS poruke korisniku sa informacijom da je merenje završeno. Ovaj modul komunicira sa mikrokontrolerom preko RX/TX UART serijskog terminala korišćenjem AT komandi. RX/TX UART je realizovan preko prvog UART modula mikrokontrolera na pinovima RC.6 (TX) i RC.7 (RX).

Modul SD kartice za rad koristi mikro SD karticu. Komunikacija ovog modula sa mikrokontrolerom realizovana je preko SPI magistrale [10] softverski realizovane na pinovima:

- RC.2 SPI CS --> SD CS (SD pin 1) CS
- RC.3 SPI CLK --> SD CLK (SD Pin 2) SCK
- RC.4 SPI DI <-- SD DO (SD Pin 4) MISO
- RC.5 SPI DO --> SD DI (SD Pin 3) MOSI

Za interakciju sa korisnikom tokom rada sa Meteorloškom stanicom – Data Logger-om i kada se napajanje iste ne vrši solarnom ćelijom, koristi se LCD displej 2x16 karaktera [11]. Prilikom napajanja solarnom ćelijom, podešavanje uređaja se vrši kada je uređaj priključen na DC izvor napajanje. Kada je uređaj podešen i spreman za rad, podešavanja se pamte u EEPROM memoriji mikrokontrolera. Prilikom pokretanja uz pomoć solarne ćelije, podešavanja se pozivaju iz EEPROM memorije mikrokontrolera i uređaj nastavlja sa radom.

Sama Meteorološka stanica – Data Logger nudi meni i rad na 4 (četiri) jezika: engleski, srpski, nemački i francuski. Na svakom od navedenih jezika moguće je podešavanje same Meteorološke stanice – Data Logger-a, a takođe i na osnovu izabranog jezika nazivi fizičkih veličina koje se skladište na SD kartici će biti na izabranom jeziku, kako bi korisnicima bilo olakšano korišćenje Meteorološke stanice – Data Logger-a. Napajanje uređaja moguće je vršiti na više načina:

- Pomoću ispravljača (5V – 12V),
- Solarno napajanje, preko ugrađene solarne ćelije, gde nije moguće korišćenje LCD displeja tokom rada, osim u toku podešavanja Meteorološke stanice – Data Logger-a gde se napajanje vrši putem gore pomenutog ispravljača (5V – 12V).
- Pomoću baterija (mogućnost korišćenja punjivih baterija koje se pune na licu mesta korišćenjem solarnog napajanja).

Pri uključenju uređaja, na ekranu LCD displeja se pojavljuje poruka sa pratećom animacijom SD kartice i nazivom samog uređaja. Nakon toga sledi izbor jezika koji će se koristiti za dalje podešavanje i rad. Zatim sledi podešavanje vremena, koje se sastoji u sledećem:

- Podešavanje trenutnog vremena kada se počinje sa merenjem;
- Podešavanje vremena kada korisnik želi da Meteorološka stanica – Data Logger završi sa merenjem.

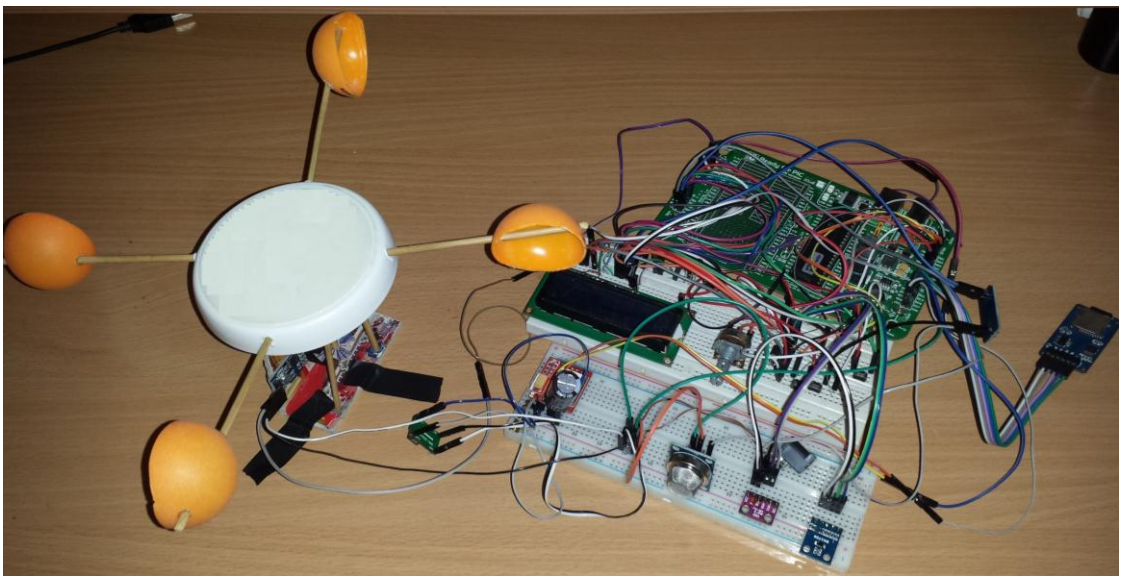
Kada je vreme početka i kraja merenja podešeno, sledi unošenje naziva datoteke u koju će biti smeštani rezultati merenja. Naziv datoteke sastoji se od 5 karaktera, gde prva četiri karaktera predstavljaju slovne karaktere (A-Z), a peti karakter po izboru samog korisnika može biti broj (0-9) ili kao i kod prethodnih karaktera slovo. Nakon toga, sledi unos broja telefona na koji će biti poslata SMS (*Short Message Service*) poruka. Na kraju, sledi provera same memorijske SD kartice, informacija korisniku o nazivu datoteke u koju će se smeštati rezultati merenja, da bi na samom kraju podešavanja počelo odbrojavanje koje prethodi početku merenja.

Na početku merenja uključuje se LED dioda kao indikator početka merenja Meteorološke stanice – Data Loggera. Tokom merenja na LCD ekranu korisniku se pruža mogućnost uvida u preostalo vreme merenja, tj. koliko je preostalo do kraja merenja, u vidu progress bar-a, praćenog informacijom u procentima. Interval merenja 5 minuta (podesiv).

Na kraju merenja na LCD ekranu se ispisuje poruka o završenom merenju, a kada se Meteorološka stanica – Data Logger napaja putem ispravljača, pruža se mogućnost da sam uređaj korišćenjem GSM modula pošalje SMS poruku. Korisnik će na svom mobilnom telefonu, čiji je broj prethodno unet tokom podešavanja Meteostanice – Data Logger-a, primiti SMS poruku o završenom merenju. Tekst poruke zavisiće od prethodno izabranog jezika za rad uređaja. Na ovaj način sam korisnik nije u obavezi da bude tokom rada pored uređaja. Kada je SMS poruka poslata korisniku sa informacijom o završenom merenju, indikatorska LED dioda se gasi. Nakon isključivanja Meteorološke stanice – Data Logger-a, korisnik može dalje manipulisati rezultatima merenja putem PC računara.

## III. PRAKTIČNA REALIZACIJA

Prototip Meteorloške stanice – Data Logger-a je realizovan na protoploči u kombinaciji sa razvojnim okruženjem „Ready for PIC“ [12], proizvod firme Mikroelektronika. Opremljeno je mikrokontrolerom PIC18F45K22 i pomoćnom elektronikom. Slika praktične realizacije prikazana je na slici 4.



Slika 4. Praktično realizovana prototip Meteorološke stanice – Data Logger-a.

Kako bi rad sa uređajem bio olakšan i zanimljiv korisniku, projektovane su animacije i slike koje animiraju podešavanja i rad Meteorloške stanice – Data Logger-a. Na slici 5. prikazane su neke od animiranih poruka tokom podešavanja i tokom rada uređaja.




Slika 5. Poruke na LCD displeju tokom rada sa Meteorološkom stanicom – Data Logger-om.

Moguća je implementacija Wi-Fi mrežnog modula (npr. ESP8266), zajedno sa SQL bazom podataka i Web serverom. Ovom implementacijom, Meteorološka stanica – Data Logger, postaje uređaj baziran na IoT (Internet of Things). Pomoću Wi-Fi modula, uređaj može tokom rada slati podatke merenja i smeštati ih u bazu podataka na serveru. Na ovaj način korisnik može imati uvid u promene meteoroloških ili ambijentalnih parametara u dužem vremenskom periodu na jednom mestu.

## IV. Zaključak

Meteorološka stanica – Data Logger predstavlja uređaj koji pruža široku primenu. Prednost ovako realizovanog uređaja je osim malih dimenzija, laka prenosivost i različiti načini napajanja. Takođe, mogućnost izbora očitavanja kako meteorloških, tako i ambijentalnih parametara, zavisno od oblasti primene. Sledeća prednost je što uređaj ne zahteva brisanje prethodno uskladištenog merenja, jer pruža mogućnost izbora novog imena datoteke u koju se smeštaju podaci merenja. Širinu primene pruža i mogućnost rada sa ovim uređajem na četiri različita jezika.

## Zahvalnica

Autori rada se zahvaljuju kompletnom osoblju katedre za Mikroelektroniku, posebno doc. dr Ljubomiru Vračaru i doc. dr Danijelu Dankoviću na nesebičnoj pomoći i sugestijama. Autori se takođe zahvaljuju finansijskoj podršci Ministarstva prosvete, nauke i tehnološkog razvoja. Ovaj rad je deo projekta TR-33035. Meteorološka stanica – Data Logger je prijavljena na Konkurs Saveza inženjera i tehničara Srbije „Stazama Nikole Tesle“, za najbolju inovativnu ideju mladih istraživača i inovatora.

## Literatura

[1] http://www.microchip.com/wwwproducts/en/PIC18F45K22
[2] http://simcom.ee/documents/SIM800/SIM800_Hardware%20Design_V1.08.pdf
[3] https://cdn-shop.adafruit.com/datasheets/BST-BME280_DS001-10.pdf
[4] https://www.omega.com/prodinfo/anemometers.html
[5] http://rohmfs.rohm.com/en/products/databook/datasheet/ic/sensor/light/bh1721fvc-e.pdf
[6] https://www.pololu.com/file/download/MQ2.pdf?file_id=0J309
[7] https://datasheets.maximintegrated.com/en/ds/DS1307.pdf
[8] http://es.elfak.ni.ac.rs/pld/Materijal/UART.pdf
[9] https://www.nxp.com/docs/en/application-note/AN10216.pdf
[10] http://es.elfak.ni.ac.rs/es/Materijal/Pog.24-SPI.pdf
[11] http://www.farnell.com/datasheets/305035.pdf
[12] https://shop.mikroe.com/ready-pic

# M2M uređaj za merenje temperature i vlažnosti vazduha

Milan Stojanović

*Sadržaj* - U ovom radu predstavljen je uređaj koji se koristi za merenje temperature i vlažnosti vazduha. Pogodnost njegovog korišćenja je mogućnost komunikacije sa korisnikom putem SMS poruke. Centralni deo uređaja, koji obrađuje informacije i upravlja ostalim delovima je platforma Arduino Uno. Za merenje temperature iskorišćen je senzorski modul DHT11, a razmena poruka sa korisnikom ostvaruje se preko GSM modula SIM800L.

## I. UVOD

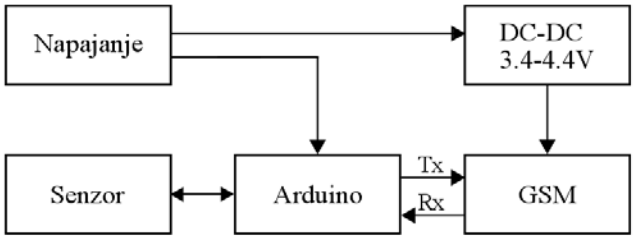
Sistemi pametnih kuća (*Smart Home Systems*) su poslednjih godina jedna od najpopularnijih tema istraživanja. Prvi pametni sistem konstruisao je inženjer Emil Matias 1950. godine, i ugradio ga u svoju kuću [1]. Neke od funkcionalnosti ovog sistema su bile: alarmni sistem, automatsko uključivanje osvetljenja, otvaranje prozora i pomeranje zavesa preko prekidača. Njegov rad predstavljao je temelj za dalji razvoj ovih sistema. Najveće dostignuće 20. veka, na ovom polju, bila je pametna kuća u Watford-u („*Integer Millenium House*“), otvorena 1998. godine. Imala je pametno osvetljenje, sistem za grejanje, sigurnosni sistem i dr.

Danas se na unapređivanju i implementaciji sistema pametnih kuća radi gotovo u celom svetu. Sistem pametne kuće se sastoji od uređaja integrisanih u jednu celinu, koja zahvaljujući tome ima veću funkcionalnost nego svaki od uređaja ponaosob. Jedan deo te celine predstavlja uređaj za merenje i kontrolu temperature. Ovaj uređaj, preko veze sa sistemom za grejanje/hlađenje, može regulisati temperaturu unutar prostorije. Takođe, može se koristiti kao deo bezbedonosnog sistema, tako što može da signalizira da li je kod nekog drugog uređaja došlo do neželjenog pregrevanja. Ovakav uređaj najčešće sadrži displej preko koga korisniku prikazuje određenu poruku. Na osnovu te poruke korisnik može ukoliko je potrebno da isključi/uključi neki drugi uređaj i zaštiti se od neželjenih posledica. Međutim, ovo nije moguće ako korisnik nije u svojoj kući. Iz tog razloga je, kao dodatak ovim uređajima, razvijena mogućnost komunikacije sa korisnikom na daljinu. Zahvaljujući tome, korisnik u bilo kom trenutku može da ima uvid u temperaturu i vlažnost vazduha u svojoj kući. Komunikacija se može obavljati na više načina, a uređaj opisan u ovom radu vezu sa korisnikom uspostavlja preko SMS poruka.

Milan Stojanović, student na smeru Elektronske komponente i mikrosistemi, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija,
E-mail: milanstojanovic@elfak.rs

## II. KOMPONENTE UREĐAJA

Pod nazivom komponente uređaja misli se na njegove funkcionalne celine, odnosno centralne blokove. Uređaj se sastoji od pet blokova, koji su međusobno povezani kao na slici 1.




Slika 1. Blok šema uređaja.

### *A.Napajanje*

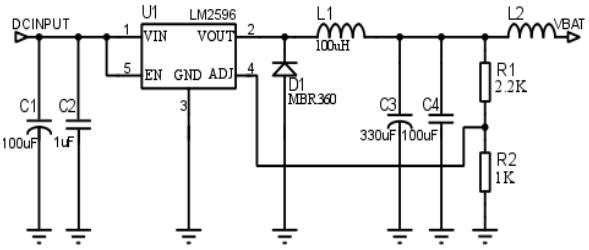
Napajanje je deo neophodan za rad ostalih blokova uređaja. Pomoću njega se mrežni naizmenični napon tranformiše u jednosmerni napon odgovarajuće amplitude. Za realizaciju uređaja koji je tema ovog rada iskorišćen je prekidački izvor napajanja sledećih karakteristika:

- Napon na ulazu: 230 V;
- Napon na izlazu: 12 V;
- Struja na izlazu: 2 A;
- Snaga 24 W.

### *B. DC-DC konvertor*

Ovaj blok uređaja obezbeđuje napajanje za rad GSM modula. Potrebno je da na izlazu daje jednosmerni napon vrednosti od 3.4 V do 4.4 V (preporučljivo je 4 V). Uz to je neophodno da u trenutku konekcije GSM-a na mrežu (slanje poruke, uspostavljanje poziva) obezbedi struju od 2 A. Jasno je da ovaj blok ustvari predstavlja jedan naponski regulator. U zavisnosti od tipa glavnog napajanja može se koristiti i linearni i prekidački naponski regulator [2]. Ukoliko postoji velika razlika između vrednosti napona izvora napajanja i napona potrebnog za rad GSM-a, na samom konvertoru se javlja veliki pad napona. U tom slučaju treba iskoristiti prekidački izvor napajanja, kako bi se smanjila disipacija i neželjeno grejanje celokupnog uređaja.

Za realizaciju ovog dela uređaja iskorišćen je *step-down* regulator (konvertor) zasnovan na integrisanom kolu LM2596 [3]. Na slici 2 je prikazana električna šema, a na slici 3 izgled štampane ploče ovog konvertora.

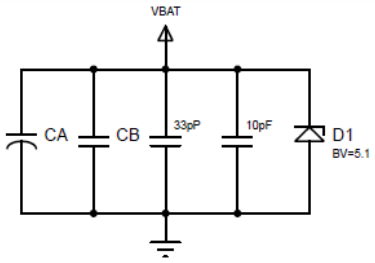



Slika 2. Električna šema konvertora sa integrisanim kolom LM2596.



Slika 3. Štampana ploča konvertora sa integrisanim kolom LM2596 [4].

Oznaka DCINPUT na šemi predstavlja napon na izlazu bloka Napajanje, dok je VBAT oznaka napona za napajanje GSM-a. Na liniji za napajanje GSM-a je potrebno dodati komponente sa slike 4 da bi se osigurao njegov pouzdan rad i uklonili šumovi.




Slika 4. Dodatne komponente na liniji napajanja GSM-a.

Kondenzator $C_A$ je tantalni elektrolitski kondenzator kapacitivnosti 100 µF. Veoma je značajno da njegova ekvivalentna serijska otpornost (ESR) bude što manja, kako bi u kratkom vremenskom intervalu mogao da isporuči visoku vrednost električne energije. Keramički kondenzator $C_B$ se koristi kako bi vrednost ESR-a bila dodatno smanjena. Preostala dva kondenzatora služe za uklanjanje visokofrekventnih šumova, a Zener dioda štiti sam modul od previsokog ulaznog napona.

*C. Arduino*

Arduino je zajednički naziv za familiju mikrokontrolerskih platformi, namenjenih razvoju aplikacija kod kojih je potrebno na jednostavan način ostvariti vezu između hardverskih I softverskih komponenti. Postoji nekoliko različitih vrsta Arduino-a. To su: Arduino Nano, Arduino Uno, Arduino Due, Arduino Leonardo, Arduino Mega i dr. Osnovna razlika između njih je u naponu napajanja, kapacitetu raspoložive memorije i broju izlaznih i ulaznih pinova. Pri projektovanju uređaja koji se opisuje upotrebljen je Arduino Uno, prikazan na slici 5

.




Slika 5. Izgled Arduino Uno platforme [5].

Arduino Uno se napaja jednosmernim naponom u opsegu od 7-12 V, koji se preko konektora dovodi na platformu. Dovedeni napon se regulatorom koji je integrisan u samu platformu, spušta na 5 V što je naponski nivo neopodan za rad Arduino-a.

Centralni deo svakog Arduino-a je mikrokontroler. Arduino Uno platforma ima AtmelATmega328/P mikrokontroler. Ovaj mikrokontroler sadrži 32 registra opste namene i 32KB (kilobajta) progamabilne fleš memorije sa mogućnošću istovremenog čitanja i upisa. Takođe ima 1 KB EEPROM-a, 2 KB RAM-a i 23 ulazno/izlaznih pinova opšte namene. Poseduje i generator taktnog signala u realnom vremenu (RTC), tri podesiva brojača (jedan 8-bitni i dva 16-bitna) i PWM, kao i 10-bitni AD konvertor i dva UART porta. Sadrži set od 131 instrukcije koje se izvršavaju brzinom od 20 MHz [6].

Arduino Uno ima 6 analognih pinova. Ovi pinovi su ulazni i na njih se najčešće povezuju senzori, ili komponente čiji izlazni napon dalje treba obraditi. Pored analognih, ova platforma ima 14 digitalnih pinova koji se mogu koristiti i kao ulazni i kao izlazni. Od toga se dva pina (0-Rx, 1-Tx) koriste za serijsku (UART) komunikaciju.

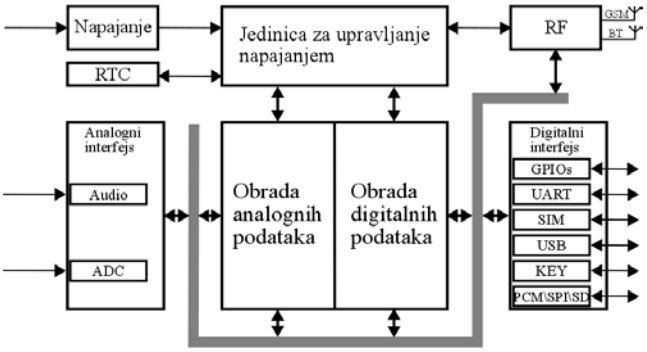
Pisanje programskog kôda vrši se u Arduino IDE softveru. Program se sastoji od dve funkcije, *void setup()* i *void loop()*. Kôd koji je napisanu *void setup()* izvršava se samo jednom. Tu se obično definišu ulazni i izlazni pinovi i zadaju početne vrednosti promenljivama. Kada se izvrši ovaj deo kôda, nastavlja se sa kôdom u funkciji *void loop()*, u kojoj se izvršava u beskonačnoj petlji.

Upisivanje koda vrši se povezivanjem Arduino-a sa računarom, pri čemu se podaci prenose USB vezom. USB port za programiranje se takođe može koristiti za napajanje Arduino-a.

*D. GSM molul*

GSM je specijalizovana vrsta modula, koja sadrži SIM karticu i radi preko mobilne mreže. Gledano iz perspektive operatera, GSM modul i mobilni telefon izgledaju isto. GSM se najčešće koristi za mobilni internet, kao i za slanje SMS i MMS poruka [7]. Ideja za kontrukciju GSM-a razvijena je u Bell laboratoriji 1970. godine, a danas je on najrasprostanjeniji sistem za mobilnu komunikaciju.

U ovom radu iskorišćen je GSM modul SIM800L, kompanije SIMCOM, koji pripada seriji modula SIM800. Blok šema ovog modula prikazana je na slici 6.




Slika 6. Blok šema GSM modula SIM800L.

Napajanje za ovaj modul kreće se u opsegu od 3.4 V-4.4 V kao što je ranije navedeno. Jedinica za upravljanje napajanjem kontroliše potrošnju električne energije modula. RTC obezbeđuje taktni signal i na taj način omogućava da se komunikacija sa modulom obavlja u realnom vremenu. Blok obeležen sa RF ima ulogu da procesira radio-frekventni signal. Na ovaj blok se nadovezuju priključci za antene. SIM800L ima GSM antenu preko koje se dobija mrežni signal neophodan za komunikaciju, i BT antenu koja se koristi za *Bluetooth* komunikaciju.

Deo analognog interfejsa je Audio blok. U okviru njega ovaj modul ima jedan audio ulaz i jedan audio izlaz. Na ulaz se može priključiti mikrofon, a na izlaz potrošač (zvučnik impedanse 32 Ω). Drugi deo analognog interfejsa čini 10-bitni analogno-digitalni konvertor.

Digitalni interfejs ovog modula je nešto složeniji i pruža više mogućnosti za ostvarivanje komunikacije. Jedna od njih je preko pinova opšte namene (GPIOs), koji mogu biti i ulazni i izlazni. Preko odgovarajuće komande se može podesiti izlazni napon ili očitati vrednost ulaznog napona.
Ovaj modul ima i jedan UART port koji sadrži sedam linija. Zadavanje instrukcija modulu od strane mikrokontrolera koji njime upravlja odvija se preko linija Rx i Tx ovog porta. Interfejs za SIM karticu modulu omogućava komunikaciju preko SMS-a. SIM800L podržava kartice koje se napajaju sa 1.8 V i 3 V. Napajanje za SIM karticu obezbeđuje interni regulator u modulu. USB port se koristi kako bi se modul povezao sa računarom. Ova veza se uglavnom koristi za nadogradnju softvera modula ili za otklanjanje grešaka u postojećem softveru. KEY interfejs omogućava prikjučivanje numeričke tastature. Maksimalna veličina tastature koja se može priključiti je matrica 5x5. Ovaj modul sadrži PCM interfejs pomoću koga može prihvatati i slati analogne signale. Pri ovom procesu se amplituda analognih signala predstavlja kao binarna vrednost. Ovaj interfejs je namenjen i za SPI komunikaciju. Ova komunikacija se obično koristi zapovezivanje LCD-a. Na istom interfejsu se može priključiti i SD kartica.

GSM izvršava određene instrukcije tako što prihvata komande koje mu šalje mikrokontroler ili računar. Svaka od tih instrukcija počinje sa AT (*Attention*), pa se skup ovih instrukcija naziva set AT komandi. Postoje četiri vrste AT komandi:

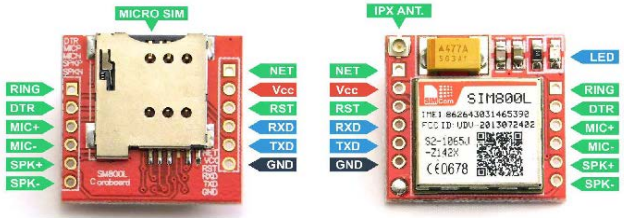
- Test komande, sintakse AT+<x>=?, pri kojima GSM vraća listu parametara (tipične za izabrani modul i mrežu) koji se mogu podešavati kao i njihove vrednosti;
- Komande za čitanje, sintakse AT+<x>?, pri kojima GSM vraća trenutno postavljene vrednosti parametara;
- Komande za upis, sintakse AT+<x>=<…>, kojima se podešavaju vrednosti izabranih parametara;
- Komande za izvršavanje, sintakse AT+<x>, pri kojima GSM izvršava intrukcije zadate izabranom komandom.

U tabeli 1. dat je opis nekih AT komandi koje su korišćene pri projektovanju uređaja koji se opisuje.

TABELA I
KORIŠĆENE AT KOMANDE [8].

| **Sintaksa komande** | **Opis komande** | **Odgovor** |
|---|---|---|
| AT | Aktiviranje modula. | OK |
| AT+CSMINS? | Ispitivanje da li ima SIM kartice. | +CSMINS:<x>,<y> y-označava da li ima kartice. y=0-nema, 1-ima. |
| AT+CMGR=X | Čitanje SMS-a sa adrese X. | +CMGR: status, br. tel., datum i vreme, tekst poruke. status:pročitana, nepročitana. |
| AT+CMGS=,,br.tel.“ Tekst poruke Ctrl+Z | Slanje SMS poruke. | +CMGS:<x> OK x-br.poslatih poruka |

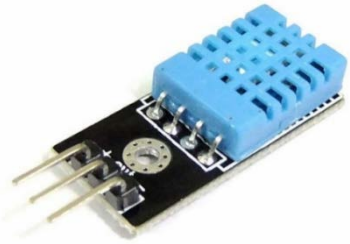
Na slici 7 prikazan je ovaj modul kao integrisana celina.




Slika 7. Izgled GSM modula SIM800L [9].

*E. Senzor za merenje temperature i vlažnosti vazduha*

Za merenje temperature i vlažnosti vazduha, pri projektovanju uređaja koji se opisuje, upotrebljen je digitalni senzorski modul (senzor) DHT11 (slika 8). Merenje vlažnosti vazduha vrši se pomoću komponente čija se otpornost menja sa promenom količine vodene pare u vazduhu, a merenje temperature preko NTC komponente. Obe komponente su povezane sa 8-bitnim mikrokontrolerom, koji obrađuje rezultate merenja i na izlazu senzora daje digitalni signal.



Slika 8. Digitalni senzorski modul DHT11 [10].

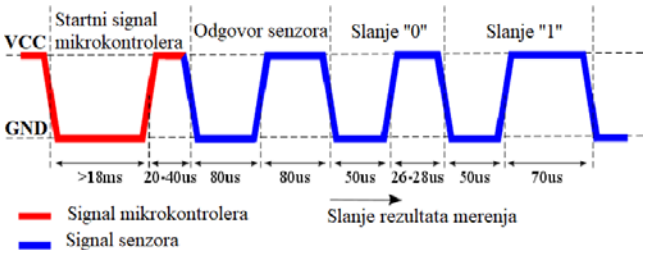
U tabeli 2 date su neke od karakteristika senzora DHT11. Zbog osobina kao što su niska cena, dugotrajna stabilnost, brzi odziv, precizna kalibracija, niska potrošnja struje i prenos signala na daljinu, DHT11 se koristi u velikom broju aplikacija. Neke od njih su: automobilska industrija, automatska kontrola, meteorološke stanice, kućni aparati, i medicinska oprema.

TABELA 2.
KARAKTERISTIKE SENZORA DHT11 [11].

| **Temperatura** | | **Vlažnost** | |
|---|---|---|---|
| **Parametar** | **Vrednost** | **Parametar** | **Vrednost** |
| Rezolucija | $1^0$C | Rezolucija | ±1 %RH |
| Preciznost | ±$1^0$C | Preciznost | ±4 %RH |
| Opseg | 0-$50^0$C | Opseg | 30-90 %RH |
| Histerezis | | Histerezis | ±1 %RH |
| Stabilnost | | Stabilnost | ±1 %RH/god. |
| Napon napajanja | | 3.5-5.5 V | |
| Struja pri merenju | | 0.3 mA | |
| Struja u neaktivnom režimu | | 60 uA | |

Senzor DHT11 ima četiri pina, od kojih se jedan ne koristi (NC). Zbog toga se često na samom senzoru nalaze tri pina za povezivanje. Na pin obeležen VCC dovodi se napon napajanja, a na pin GND se povezuje masa. Podaci, odnosno vrednosti merenih veličina, se na mikrokontroler prenose preko pina OUTPUT. Između linije za prenos podataka i napajanja postavlja se otpornik. Ako je rastojanje od senzora do komponente koja obrađuje rezultate merenja (mikrokontroler) manje od 20 m, otpornik treba da ima vrednost 5.1 kΩ. Sa povećanjem rastojanja treba smanjiti vrednost ovog otpornika.

DHT11 sa mikrokontrolerom komunicira serijskom komunikacijom preko jedne linije (*one wire communication*). Na slici 9 prikazan je komunikacioni proces između senzora i mikrokontrolera [12].




Slika 9. Vremenski dijagram komunikacije senzora sa mikrokontrolerom.

Kada nema komunikacije, a zbog otpornika, na pinu mikrokontrolera na kome je vezan senzor, je logička jedinica. Komunikacija počinje tako što mikrokontroler senzoru šalje startni signal koji predstavlja prelazak sa logičke jedinice na logičku nulu. Da bi senzor to prepoznao kao startni signal, bitno je da logička nula traje najmanje 18 ms. Nakon toga, mikrokontroler na liniji za komunikaciju postavlja logičku jedinicu. Ovaj naponski nivo treba da traje 20-40 µs, i predstavlja vreme u toku koga se čeka odgovor senzora.

Odgovor senzora se takođe sastoji od logičke jedinice i logičke nule. Glavni deo odgovora predstavlja logička nula i traje 80 µs. Logička jedinica traje isti vremenski period, i to vreme senzor koristi kao pripremu za dalju komunikaciju.

Zatim sledi slanje podataka, odnosno 40 bitova ("0" ili "1"), koji predstavljaju rezultat merenja vlažnosti vazduha i temperature. Svaki bit podataka, koji senzor pošalje mikrokontroleru počinje logičkom nulom, koja traje 50 µs. Nakon logičke nule sledi naponski nivo logičke jedinice, koji ukazuje na to da li je podatak "0" ili "1". Kada je trajanje logičke jedinice 26-28 µs mikrokontroler smatra da je senzor poslao "0". Ako sa senzora nakon logičke nule stigne logička jedinica u trajanju od 70 µs, mikrokontroler to detektuje kao "1".

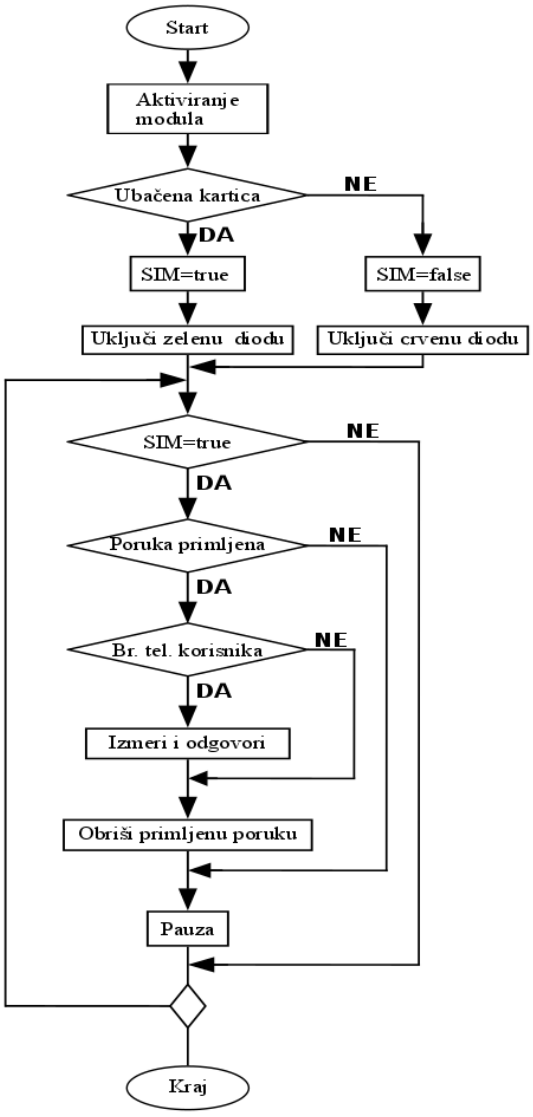
Komunikacija se završava kada senzor pošalje 40 bitova podataka. Bitovi podataka se mogu podeliti u pet grupa po osam bitova, pri čemu svaka grupa ima određeno značenje. Prva i druga grupa bitova predstavljaju vrednost vlažnosti vazduha, s tim što prva grupa opisuje celobrojnu vrednost, a druga grupa razlomljenu vrednost. Treća grupa predstavlja celobrojnu, a četvrta razlomljenu vrednost

temperature. Poslednja grupa bitova služi za proveru vrednosti primljenih rezultata.

## III. PRINCIP RADA UREĐAJA

Na slici 10 predstavljen je algoritam rada projektovanog uređaja. Kada korisnik pošalje poruku, uređaj je prihvata pomoću GSM modula, i ona se smešta na SIM karticu. Uređaj treba korisniku da uzvrati SMS u kome će pisati kolika je vrednost temperature i vlažnosti vazduha u prostoriji gde je uređaj postavljen. Algoritam sa slike ustvari opisuje kôd koji je upisan u Arduino, a čije izvršavanje počinje odmah nakon uključivanja uređaja.

Na samom početku Arduino šalje GSM-u odgovarajuće AT komande kojima se modul aktivira.




Slika 10. Algoritam rada projektovanog uređaja.

Kako bi komunikacija između uređaja i korisnika bila moguća, neophodno je da GSM modul ima SIM karticu. Iz tog razloga se najpre mora proveriti da li je ona postavljena. Ukoliko je odgovor negativan promenljiva SIM će dobiti vrednost *false* i uključiće se crvena dioda. Ako u modulu ima SIM kartice pomenuta promenljiva dobija vrednost *true*, a uključiće se zelena dioda. Ovaj deo algoritma se izvršava u funkciji *void setup().*

Nastavak algoritma predstavlja deo koji se izvršava u funkciji *void loop().* Ukoliko je vrednost promenljive SIM prethodno postavljena na *false*, Arduino smatra da u GSM-u nema kartice i u nastavku neće raditi ništa. Ako je vrednost promenljive SIM *true,* Arduino, a samim tim i uređaj, nastavlja sa radom.

Arduino ispituje da li na kartici ima primljenih poruka. Kada korisnik pošalje više poruka one se smeštaju na sukcesivnim memorijskim lokacijama (adresama) kartice, tako da Arduino najpre ispituje da li ima poruke na prvoj adresi. Ako nema primljene poruke program izlazi iz petlje i čeka određeno vreme do sledećeg čitanja (pauza). Ukoliko ima primljene poruke prelazi se na njenu obradu.

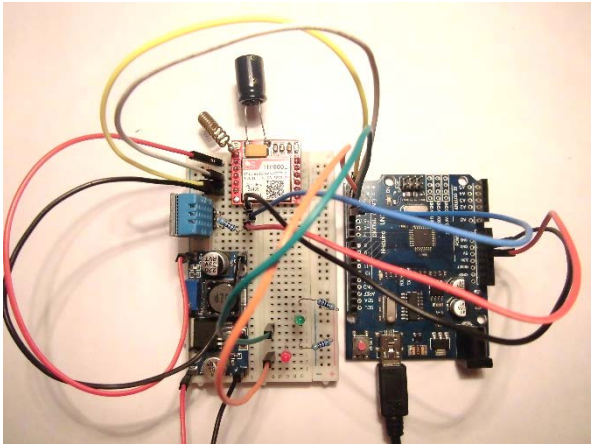
Iz primljene poruke se izdvaja broj telefona i proverava da li je on odgovarajući, odnosno isti kao broj telefona korisnika uređaja. Kada broj nije odgovarajući poruka se briše i prelazi na pauzu. Ako je broj odgovarajući, aktivira se senzor, vrši merenje temperature i vlažnosti vazduha i rezultati šalju korisniku.

Nakon toga sledi brisanje poruke i pauza do čitanja sledeće. Primljene poruke se brišu kako bi na kartici uvek bilo slobodne memorije za prijem nove poruke.

Kada je primljeno više poruka one će se redom čitati, menjanjem vrednosti određene promenljive unutar programa. Kada se u tom procesu naiđe na prvu adresu na kojoj nema poruke, vrednost te promenljive se vraća na početnu vrednost, tako da se pri sledećem čitanju, opet čita poruka sa prve adrese.

## IV. PRAKTIČNA REALIZACIJA

Uređaj opisan u ovom radu je praktično realizovan na nivou prototipa, kao što je prikazano na slici 11.



Slika 11. Prototip uređaja.

Kako bi se obezbedilo napajanje Arduino platforme ona se povezuje na računar. Može se iskoristiti i neki drugi izvor za njeno napajanje (punjač za mobilni telefon), ali treba voditi računa da to napajanje bude stabilno i ne veće od 5 V, kako ne bi došlo do oštećenja kola za programiranje Arduino-a.

Na slici 12. prikazana je komunikacija između uređaja i korisnika.




Slika 12. Primer komunikacije korisnika sa uređajem.

## V. ZAKLJUČAK

Na osnovu svega navedenog u radu, očigledno je da ovakav uređaj može biti deo sistema pametnih kuća ili se može koristiti samostalno. Zahvaljujući malim dimenzijama može se implementirati u bilo kom delu kuće ili nekog drugog objekta. Sve komponente uređaja su komercijalno i ekonomski dostupne, tako da je proizvodnja ovakvih uređaja moguća.

Kako je prikazano na slici 11 uređaj je razvijen na nivou prototipa. Dalje treba projektovati štampanu ploču na kojoj bi se smestile sve komponente uređaja. Pri tome umesto Arduino platforme treba iskoristiti mikrokontroler Atmel ATmega328/P, kako bi se smanjile dimenzije i cena izrade. Uz to treba voditi računa o dimenzijama kućišta za pakovanje uređaja.

Nedostatak, koji uređaj trenutno ima, je potreba za dva priključka za napajanje. Jedan, u koji se uključuje blok za napajanje i drugi u koji se uključuje Arduino (Arduino se napaja preko konektora za programiranje). To se može izmeniti korišćenjem još jednog naponskog regulatora LM2596. Izlaz bloka “Napajanje” treba povezati na prvi naponski regulator, koji bi napajao Arduino i drugi regulator. GSM modul bi bio vezan na izlaz drugog naponskog regulatora. Drugi način je korišćenjem jednog regulatora kao sada, koji bi napajao Arduino, dok bise za napajanje GSM-a koristila baterija.

Ovaj uređaj poseduje i mogućnost nadogradnje i povećanja funkcionalnosti. Može se isprogramirati tako da u slučaju niske ili visoke temperature, korisniku automatski pošalje SMS. Pored senzora DHT11 jednostavno se mogu dodati i neki drugi senzori. Na primer senzor koji detektuje gas, ili senzor koji detektuje da li ima nekoga u prostoriji. Uz to, na sam Arduino se preko releja mogu povezati razni uređaji. Te uređaje bi korisnik mogao da kontroliše, na osnovu vrednosti izmerenih veličina, slanjem parametara koje upiše u SMS poruku.

## ZAHVALNICA

Ovaj rad je realizovan kao završni rad na smeru Elektronske komponente i mikrosistemi. Autor se zahvaljuje mentoru, doc. dr Ljubomiru Vračaru na izdvojenom vremenu, sugestijama i pomoći pri realizaciji rada.

## LITERATURA

[1] http//www.appcessories.co.uk/evolution-of-smart-home-and-iot-through-hisstory/.
[2] Z. Prijić, „Analogna Mikroelektronika“, Predavanja, Elektronski fakultet, Niš
[3] "SIM800 Hardware Design V1.08", SIMCom A Company of Sim tech
[4] http://www.hobbyelectronics.net/review_dctodcmodule.html
[5] https://store.arduino.cc/arduino-uno-rev3
[6] "8-bit AVR Microcontrollers, ATega328/P, DATASHEET COMPLETE", Atmel
[7] https://www.nowsms.com/faq/what-is-a-gsm-modem
[8] "SIM800 Series AT Command Manual V1.01", SIMCom A Company of Sim tech
[9] https://nettigo.eu/products/sim800l-gsm-grps-module
[10] https://drotek.com/shop/en/sensors/369-temperature-and-humidity-sensor-dht11.html
[11] "Temperature and humidity module, DHT11 Product Manual", AOSONG
[12] "DHT11 Humidity & Temperature Sensor", D-Robotics UK (www.droboticsonline.com)

# Akustična sklopka

Marko Mihajlović

*Sadržaj* – Akustična sklopka (zvučni prekidač) predstavlja uredjaj pomoću kojeg možemo uključivati i isključivati svetlo pljeskom dlana o dlan. Prikazana je i detaljno objašnjena praktična (fizička) realizacija električnog kola akustične sklopke, prikazane su upotrebljene komponente i instrumenti neophodni za realizaciju zvučnog prekidača, kao i simulacija električnog kola u programima LTspiceIV i Fritzing.

## I. Uvod

Moderne tehnologije su danas pre svega tu kako bi savremenom čoveku olakšale svakodnevne potrebe. Komunikacija, prenos podataka, upravljanje uređajima na daljinu, korišćenje raznih uređaja, kućnih aparata i dr. sada je znatno lakše zahvaljujući savremenim tehnologijama, kolima i komponentama, digitalnim kolima i sl.

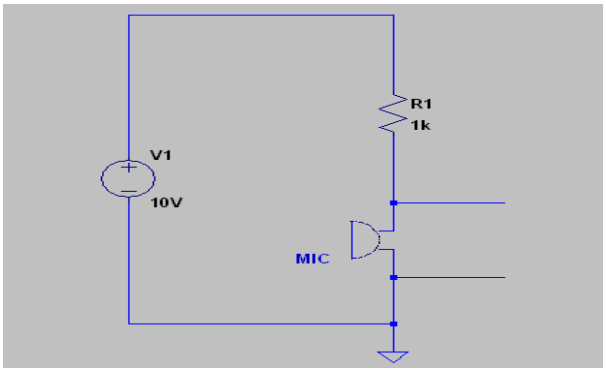
U okviru ovog rada predstavljen je jedan od načina na koji možemo olakšati svakodnevne potrebe, tačnije, predstavljeno je kolo pomoću kog možemo realizovati uređaj za uključivanje i isključivanje svetla pljeskom dlana o dlan. U okviru rada prikazana je praktična (fizička) realizacija akustične sklopke, kao i njena simulacija u programima LTspiceIV [1] i Fritzing [2]. Akustična sklopka može funkcionisati kako na visokim naponima (230V), tako i na niskim naponima reda 5-10V, u zavisnosti od iskorišćenih komponenata. Tako je npr. neophodno koristiti odgovarajući rele, ukoliko kolo funkcioniše na mrežnom naponu, uskladiti vrednost svih komponenata i potrošača.

## II. Simulacija akustične sklopke u LTSpice IV

### *A. Neophodne izmene za potrebe simulatora LTSpice IV*

Kako bi ispitali određene parametre i simulirali rad električnog kola u LTspiceIV, neophodno je pojedine komponente zameniti onim koje nam simulator dozvoljava, tj. potrebno je električnu šemu akustične sklopke prilagoditi simulatoru LTspiceIV. Tako smo u našem slučaju, mikrofon morali zameniti nekim komponentama (u zavisnosti od vrste mikrofona), ili da napravimo novu komponentu, što je teži način. Na slici 1 prikazan je deo kola koji je potrebno modifikovati.

Marko Mihajlović je student master studija na modulu Elektronika i mikrosistemi, Katedra za mikroelektroniku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: mihajlovic9933@yahoo.com




Slika 1. Deo kola koji je potrebno izmeniti u simulatoru LTSpiceIV.

### *B. Funkcionalna električna šema realizovana u LTSpiceIV*

Kao što je već rečeno, da bismo snimili sve karakteristike koje nas interesuju, bilo je neophodno prilagoditi šemu simulatoru, pa smo tako dobili električnu šemu koja je funkcionalna, prikazanu na slici 2. Kako bismo dobili kratkotrajne pozitivne impulse (simulacija pljeska rukama), koji su potrebni za aktivaciju kompletnog kola, iskorišćen je pulsni generator V1 sa vrednošću od 0-10V. Mikrofon, koji ima ulogu senzora, u realnim uslovima daje određeni električni signal, usled promene kapacitivnosti, otpornosti ili nekog drugog parametra, u zavisnosti od sklopa i principa rada, tj. da li je reč o kondenzatorskom ili dinamičkom mikrofonu. Nama je za simulaciju najlakše bilo da mikrofon zamenimo jednim otpornikom i jednim generatorom, pa će tako doći do promene otpornosti usled pljeska rukama, odnosno usled pozitivnog impulsa sa generatora V1.

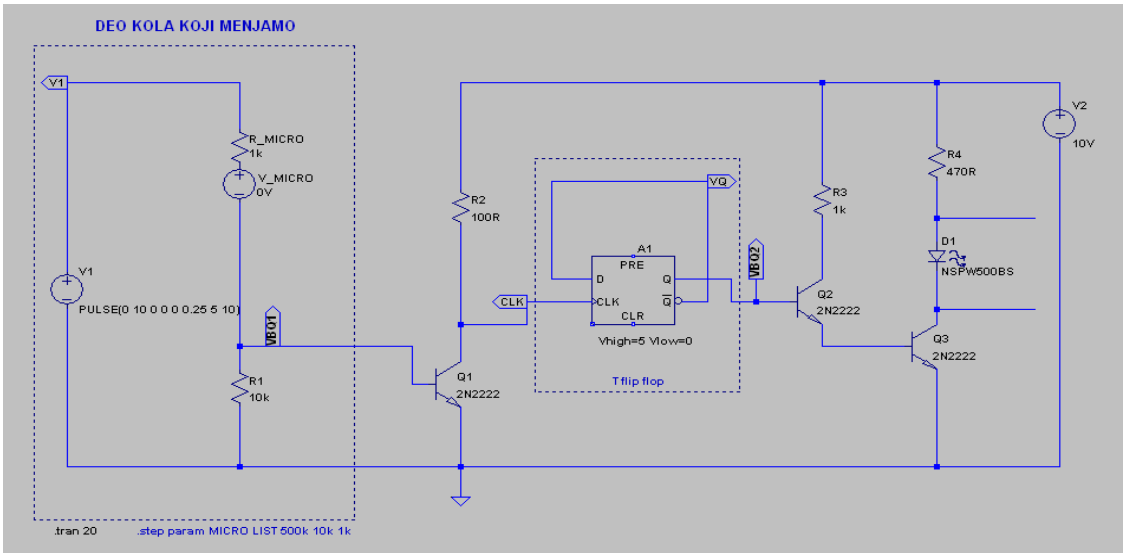
Da bi sve funkcionisalo, neophodno je da napon između baze i emitora Vbe tranzistora Q1, bude veći od 0.7V, posmatran u našoj simulaciji kao napon VBQ1. Taj napon se kontroliše naponskim razdelnikom, a njegovu vrednost možemo odrediti pomoću sledeće formule:

$$VBQ1 = \frac{R_1}{R_1 + R_micro} * V_1 \quad (1)$$

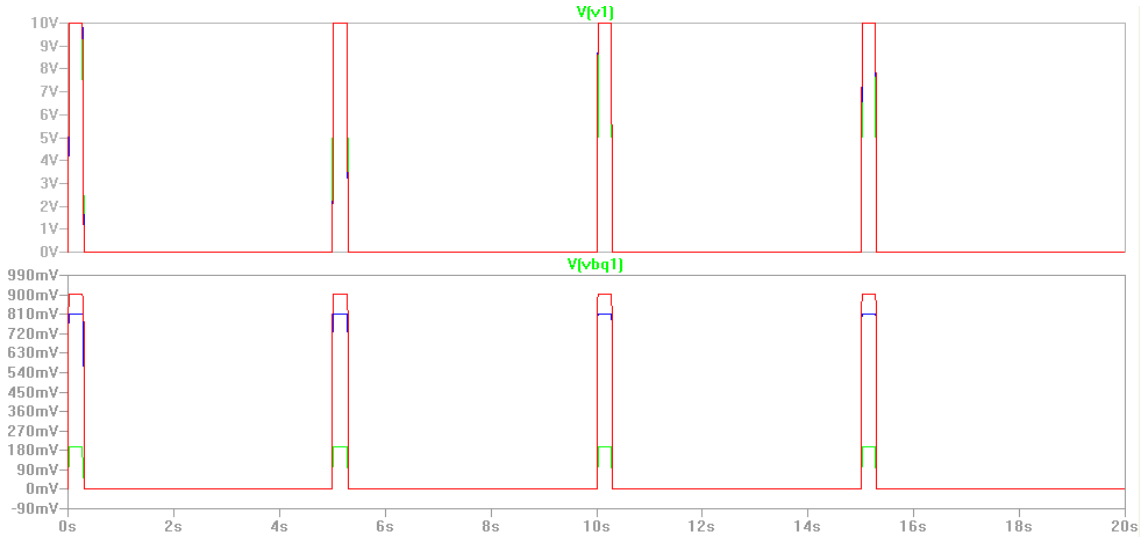
pri čemu je R1 otpornik sa vrednošću 10 kΩ, V1 je gore pomenuti pulsni generator, dok otpornikom R_micro (otpornost mikrofona) možemo direktno kontrolisati napon VBQ1. Na početku, kako bi simulaciju pojednostavili, otporniku R_micro zadaćemo vrednost od 1 kΩ, pa će tada naš uslov biti ispunjen, što je prikazano sledećom formulom:

$$VBQ1 = \frac{R_1}{R_1 + R_micro} * V_1 = \frac{10k\Omega}{10k\Omega + 1k\Omega} * 10V = 9.0909V \quad (2)$$

Kao što vidimo, dobijeni napon VBQ1 je više nego dovoljan da tranzistor Q1 provede, pa će i kompletno kolo funkcionisati. Međutim, napon VBQ1 možemo podešavati i pomoću pulsnog generatora V1, tako što će kada naiđe pozitivni impuls (10V), napon VBQ1 imati pokazanu vrednost, a za vreme negativnog impulsa tranzistor Q1 će biti zakočen. U simulatoru možemo da posmatramo šta se dešava sa naponom VBQ1 ukoliko podešavamo vrednost otpornika R_micro. Za to se može iskoristiti komanda .step param MICRO LIST 500k 10k 1k. Otporniku R_micro zadavali smo redom vrednost 500kΩ, 10kΩ i 1kΩ




Slika 2. Funkcionalana električna šema akustične sklopke u simulatoru LTSpice IV.

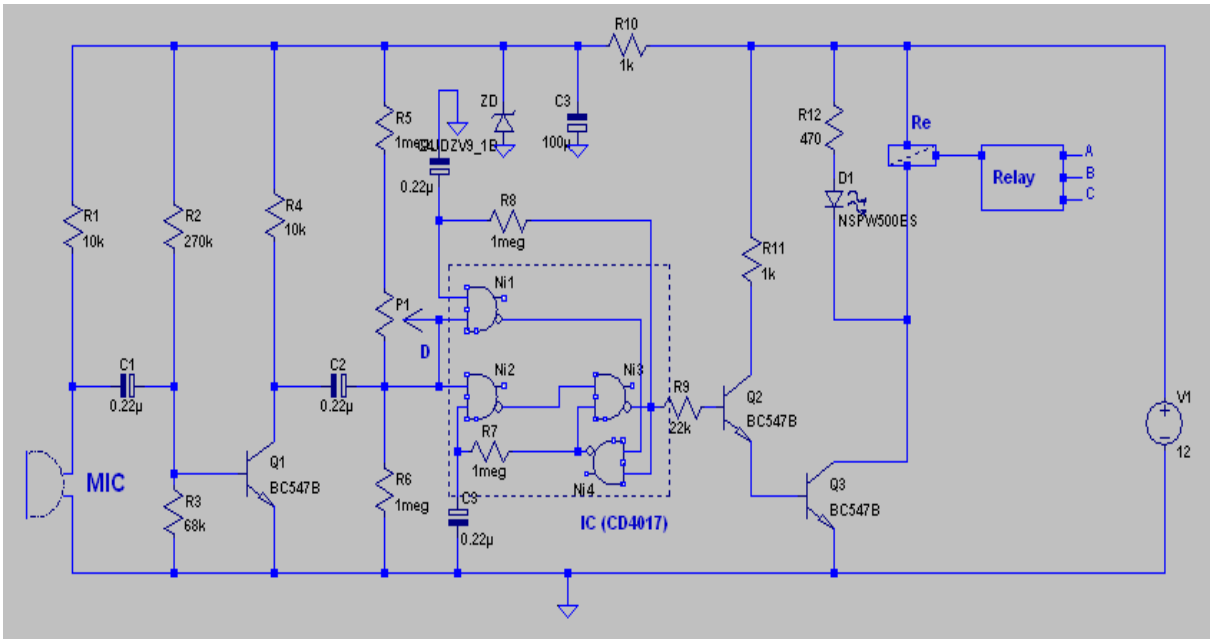



Slika 3. Naponi V[V1] i V[VBQ1].

Uočljivo je na slici 3 da se pri određenim vrednostima otpornika R_micro, napon baze VBQ1 menja, pa će i tranzistor Q1 biti zakočen za velike vrednosti otpornika R_micro, dok će pri dovoljno malim vrednostima provesti (1kΩ). Upravo ta promena otpornosti simulira promenu otpornosti mikrofona usled pljeska rukama.

*C. Električna šema akustične sklopke za visoke napone, projektovana u simulatoru LtspiceIV*

Na slici 4, prikazana je električna šema akustične sklopke za visoke napone. Upotrebljeni rele ima kontakte za 3A i moguće je koristiti ga pri mrežnom naponu (230V). Ukoliko se rele koristi za napone veće od 40V, štampanu pločicu je iz sigurnosnih razloga neophodno ugraditi u odgovarajuću plastičnu kutiju. Tokom provere rada, uređaj ne sme biti priključen na mrežni napon.




Slika 4. Električna šema akustične sklopke za visoke napone u LTSpice IV.

## III. PRAKTIČNA REALIZACIJA AKUSTIČNE SKLOPKE

### *A. Instrumenti i komponente*

U tabeli 1 navedeni su iskorišćeni instrumenti i komponente za praktičnu realizaciju uređaja.

TABELA 1
SPISAK INSTRUMENATA I KOMPONENATA POTREBNIH ZA REALIZACIJU UREĐAJA

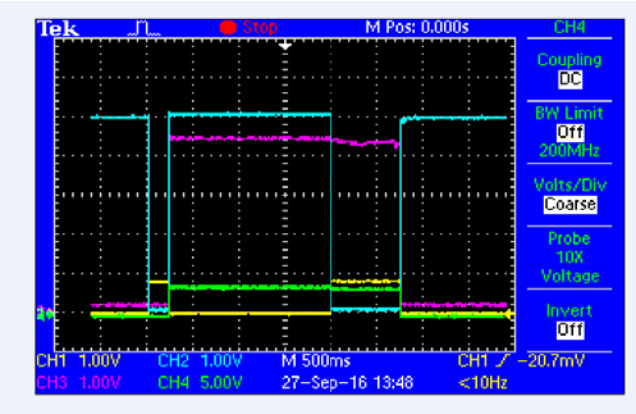
| Oznaka | Opis | Komada |
|---|---|---|
| GW INSTEK - PSP-2010 [3] | Izvor napajanja | 1 |
| TEKTRONIX TDS2024B [4] | Osciloskop | 1 |
| 74LS74 [5] | D flip-flop | 1 |
| Taster | Taster | 1 |
| Q1-Q3 | BC547 | 3 |
| R1 | 10kΩ | 1 |
| R2 | 100Ω | 1 |
| R3 | 1kΩ | 1 |
| R4 | 470Ω | 1 |
| LED | LE dioda | 1 |
| Protoploča | Protoploča | 1 |

### *B. Snimanje signala na osciloskopu*

Signali koje smo posmatrali na osciloskopu [4], prikazani su na slici 5. To su: Napon na bazi tranzistora Q1, označen na šemi kao napon VBQ1 (signal žute boje), napon koji dolazi na CLK flip-flopa [5] (signal plave boje), napon na spoju ulaza flip-flopa „D" i izlaza (signal roze boje) i napon na kolektoru tranzistora Q3, tj. na katodi LE diode (signal zelene boje).

U prvom slučaju, snimljen je napon VBQ1 (signal žute boje) i napon na CLK flip-flopa [5] (signal plave boje). Ako posmatramo napon VBQ1, jasno se primećuju impulsi koje smo „slali", tačnije simulirali pritiskom tastera.

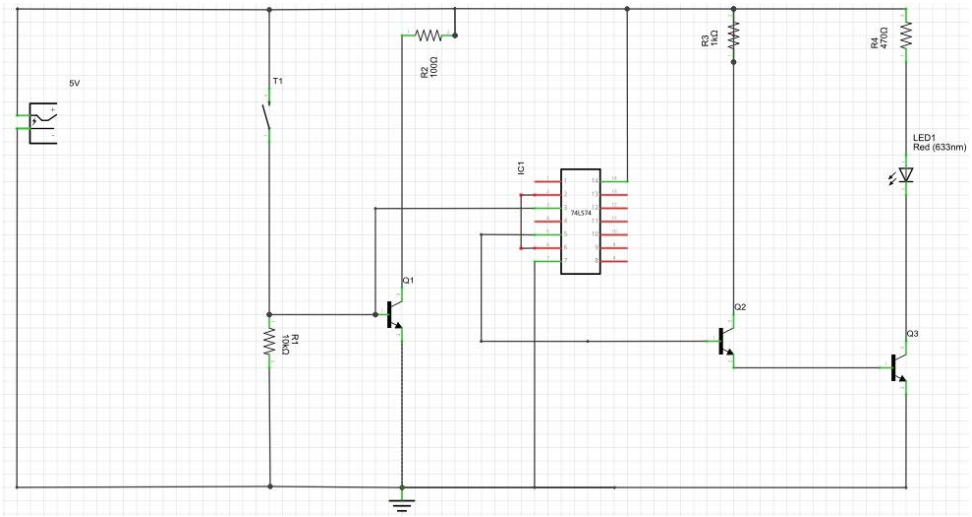
Reč je o onim kratkotrajnim pozitivnim impulsima, o kojima je bilo reči u teorijskom delu i koje bi mikrofon detektovao (pljesak rukama ili glasniji povik). CLK signal imaće suprotnu vrednost, jer negativni delovi CLK signala aktiviraju izlaz flip-flopa [5]. Stanje na izlazu flip-flopa menja se dolaskom prednje ivice CLK signala. Posmatrali smo šta se dešava na izlazu flip-flopa, odnosno snimljen je napon na spoju ulaza flip-flopa „D" i izlaza (signal roze boje). Kada je aktivan izlaz $\overline{Q}$=1, tada će izlaz Q biti na niskom naponskom nivou Q=0, pa tako tranzistori Q2 i Q3 neće voditi i LE dioda ne svetli. Kada se pojavi sledeća prednja ivica CLK signala, izlaz Q dobija vrednost „1", tranzistori Q2 i Q3 provedu i LE dioda svetli. Posmatraćemo šta se dešava na kolektoru tranzistora Q3, tj. na katodi LE diode. Signal koji nam to pokazuje je signal zelene boje. Kada je izlaz $\overline{Q}$=1, tada će napon na kolektoru tranzistora Q3 i katodi LE diode imati vrednost 5V, što znači da dioda neće svetleti. Kada je $\overline{Q}$=0, izlaz Q=1, tranzistori Q2 i Q3 će provesti, a napon na katodi LE diode će pasti na 0V i dioda će svetleti.



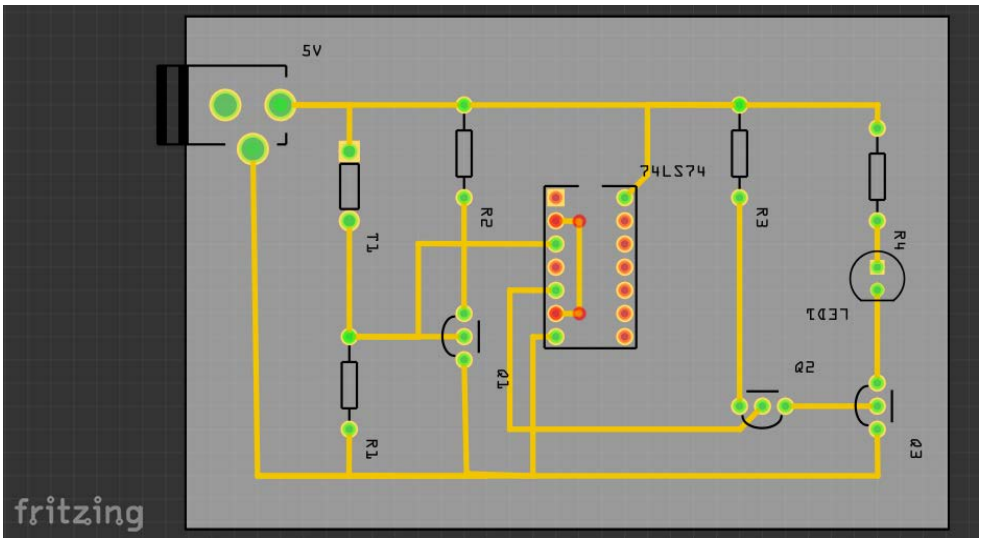

Slika 5. Napon VBQ1 (žuto), CLK signal (plavo), izlaz $\bar{Q}$ (roze) i napon na kolektoru tranzistora Q3 (zeleno).

## IV. PROJEKTOVANJE U PROGRAMU FRITZING

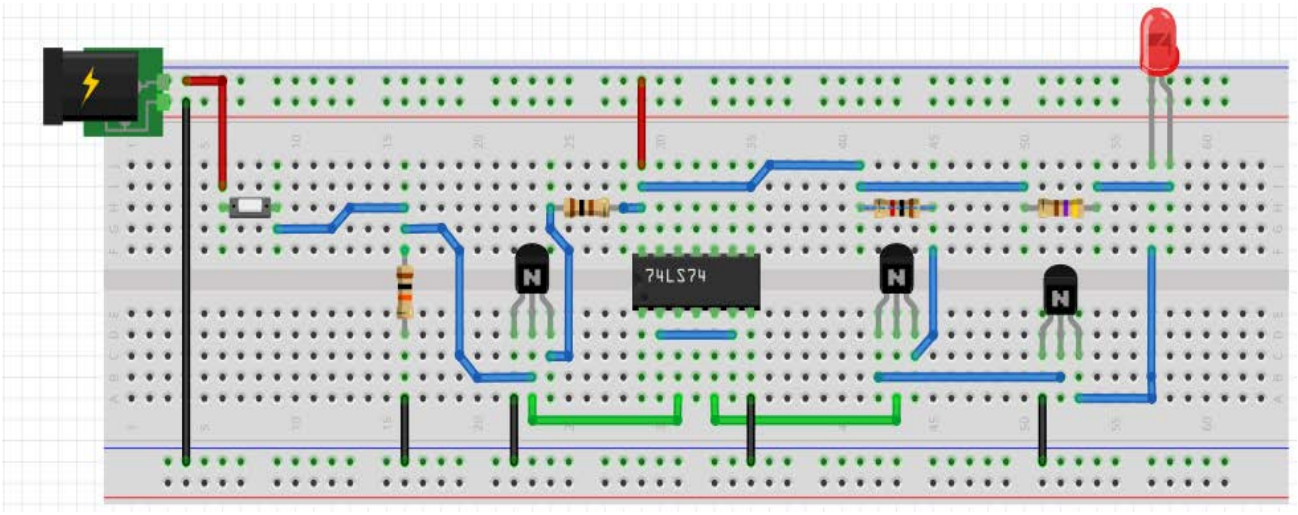
Program Fritzing, prikazuje nam kako će naša električna šema i raspored komponenata izgledati fizički na protoploči, kao i PCB projekat. Olakšava nam praktičnu realizaciju, na taj način što unapred možemo dizajnirati PCB projekat i videti potpuno realan prikaz na protoploči. Iz ovog programa možemo iskoristiti fajlove za izradu štampane pločice. Upotreba Fritzing-a je krajnje jednostavna, a program se može preuzeti besplatno preko interneta [2]. Na slici 6. ilustrovana je naša električna šema u programu Fritzing. PCB projekat akustične sklopke prikazan je na slici 7. Potpuno realan prikaz protoploče prikazan je na slici 8, a njena izrada je gotovo identična praktično realizovanoj.




Slika 6. Električna šema akustične sklopke u Fritzing-u.




Slika 7. PCB projekat akustične sklopke u Fritzing-u.




Slika 8. Izgled realizovane protoploče Fritzing-u.

## V. ZAKLJUČAK

U radu je dat osvrt na kolo akustične sklopke, poznatom još kao zvučni prekidač, kao i na detaljan princip rada. Jasno i konkretno je opisan način funkcionisanja zvučnog prekidača, prikazane su upotrebljene komponenate i instrumenti za praktičnu realizaciju istog. Prikazani su rezultati merenja, teorijski pregled, simulacija u programu LTspiceIV, praktično realizovan projekat, snimljeni važni signali na osciloskopu, kao i projektovanje u programu Fritzing. Ovo kolo može imati veoma široku primenu, jednostavnim modifikacijama. Tako smo mi za potrebe simulatora mikrofon zamenili generatorom i otpornikom, dok je u praktičnom delu zamenjen tasterom. Na mesto mikrofona moguće je postaviti bilo koju komponentu promenljive vrednosti, kao npr. fotootpornik, termistor, laser itd., dok je na izlazu moguće postaviti bilo koji signalizator umesto sijalice ili LE diode. Ukoliko na izlazu postavimo zvučnik ili elektromotor, naše kolo dobija sasvim drugi koncept i namenu.

## ZAHVALNICA

Veoma sam zahvalan svom mentoru, profesoru Danijelu Dankoviću, na pomoći u realizaciji ovog rada.

## LITERATURA

[1] „Uvod u LTspiceIV”, dostupno: www.viser.edu.rs/download.php?id=16346, (pristupljeno 23. septembra 2016.).
[2] „Fritzing”, dostupno na: http://fritzing.org/home/, (pristupljeno 23. septembra 2016.).
[3] „GW INSTEK - PSP-2010”, dostupno na: http://www.gwinstek.com/en global/products/DC_Power_Supply/Programmable_Single_Channel_DC_Power_Suppli/PSP-Series, (pristupljeno 27. septembra 2016.).
[4] „TEKTRONIX TDS2024B”, dostupno na http://www.testequipmentdepot.com/tektronix/pdf/tds1000_tds2000series.pdf, (pristupljeno 27. septembra 2016.).
[5] „DM74LS74A”, dostupno na: http://www.futurlec.com/74LS/74LS74.shtml, (pristupljeno 26. septembra 2016.).

# Mobilna android aplikacija za pomoć pri vožnji na autoputevima u Srbiji

Petar Lakčević, Bojan Martinović, Rastko Damnjanović

*Sadržaj* – U ovom radu je predstavljena mobilna android aplikacija za pomoć pri vožnji na autoputevima u Srbiji – "Prosečna brzina". U prvom delu rada opisana je arhitektura aplikacije i način funkcionisanja. Drugi deo rada objašnjava korisnički interfejs i praktičnu primenu aplikacije. U poslednjem delu se razmatra komercijalna vrednost aplikacije i ostvareni rezultati.

## I. UVOD

Od 1. septembra 2017. godine na autoputevima u Srbiji se u okviru zatvorenog sistema naplate vrši merenje prosečne brzine vožnje između dve naplatne stanice, a na osnovu toga se vrši i potencijalno kažnjavanje vozača koji premaše dozvoljenu brzinu. Aplikacija "Prosečna brzina" je nastala kao odgovor na novonastalu situaciju i služi da pomogne vozačima da voze u zakonskim okvirima i time poveća bezbednost na autoputevima u Srbiji. Ključne funkcionalnosti aplikacije su izračunavanje prosečne brzine i obaveštavanje u realnom vremenu na osnovu prikupljenih podataka. Aplikacija je funkcionalna na svim verzijama operativnog sistema „Android" od verzije 4.1.

## II. RAZVOJ I FUNKCIONALNOSTI APLIKACIJE

Razvoj inicijalne verzije aplikacije trajao je od 19. do 31. avgusta 2017. Cela aplikacija je razvijena u okviru integrisanog razvojnog okruženja za android aplikacije "Android studio". Aplikacija se sastoji od 4 xml *layout-a* i tri java klase koje sadrže više metoda. Zasniva se na Google Maps API kao i "Location" klasi. Mnoge metode Google Maps API-a se ne mogu koristiti u *offline mode-u* kao što je distanca između trenutne lokacije i određenog markera pa je potreban parameter računat na način koji će biti opisan kasnije u radu. Takođe, ostale ključne metode, poput metoda *OnMapready*, *OnCreate*, *Distance*, *OnLocationChange*, *OnMarkerClick*, će biti opisane u radu.

Prilikom pokretanja aplikacije poziva se "override" metoda "onCreate" klase "MapsActivity", koja vrši sledeće:

- Pri prvom pokretanju zahteva dozvolu za korišćenje Lokacije,
- Ukoliko korisnik ne želi da aplikaciji omogući pristup GPS servisima, ona isključuje aplikaciju,
- Postavlja "onClickListener" za dugmad koji služe za izbor kategorije vozila,
- Inicijalizuje reklame,
- Inicijalizuje zvuk upozorenja za trenutnu brzinu veću od 120km/h,
- Isključuje se automatsko zaključavanje ekrana.

Druga bitna "override" metoda klase "MapsActivity" je "onMapReady". Kao što se može videti pri pokretanju aplikacije (Slika 2), korišćena je Google mapa (API V2) sa kastomizovanim markerima. Ova metoda ima sledeću ulogu:

- Učitavaju se baze distanci i cena putarina,
- Inicijalizuju i deklarišu se lokacije svih naplatnih stanica kao instance klase "LocationPN",
  - Klasa "LocationPN" sadrži naziv naplatne stanice koje je tipa "String" kao i njenu tačnu lokaciju u formatu "LatLong" (Latitude, Longtitude),
- Inicijalizuju i deklarišu se kastomizovani markeri na Google mapi na osnovu pomenutih instanci klase "LocationPN",
- Poziva se metoda "distance" koja meri vazdušnu razdaljinu između trenutne lokacije korisnika i svih naplatnih stanica u Srbiji i najbližu postavlja kao ulaznu. To se radi pod pretpostavkom da će se korisnik uključiti na autoput na naplatnoj stanici najbližoj njegovoj trenutnoj lokaciji,
  - Ukoliko korisnik želi da promeni ulaznu naplatnu stanicu to može uraditi klikom na dugme "Ulazna rampa" i izborom željene naplatne stanice. Ukoliko se na taj način izabere ulazna naplatna stanica, metoda "distance" više neće biti pozvana do sledećeg pokretanja aplikacije ili resetovanja na ekranu za obaveštavanje korisnika,
- Metoda distance:

Petar Lakčević je student na smeru Informacioni sistemi i tehnologije, Fakultet organizacionih nauka, Univerzitet u Beogradu, Jove Ilića 154, 11000 Beograd, Srbija, e-mail: petar.lakcevic@gmail.com

Bojan Martinović je student master studija na Politehničkom Univerzitetu u Milanu, Piazza Leonardo da Vinci 32, 20133 Milano, Italija, e-mail: martinovic.bojan@yahoo.com

Rastko Damnjanović je student na smeru Menadžment, Fakultet organizacionih nauka, Univerzitet u Beogradu, Jove Ilića 154, 11000 Beograd, Srbija, e-mail: rastko94@gmail.com

```
private double distance(LatLng a, LatLng b)
{
   double lon1 = a.longitude;
   double lon2 = b.longitude;
   double lat1 = a.latitude;
   double lat2 = b.latitude;


   double theta = lon1 - lon2;
   double dist = Math.sin(deg2rad(lat1))
         * Math.sin(deg2rad(lat2))
         + Math.cos(deg2rad(lat1))
         * Math.cos(deg2rad(lat2))
         * Math.cos(deg2rad(theta));
   dist = Math.acos(dist);
   dist = rad2deg(dist);
   dist = dist * 60 * 1.1515;
   return (dist);
}
```

- Postavlja se “onMarkerClickListener” koji omogućuje korisniku da na samoj mapi izabere izlaznu naplatnu stanicu,
- Postavlja se “onClickListener” za svu dugmad prvog ekrana. To su “Ulazna rampa”, “Izlazna rampa”, “Start”,
    - Dugmad “Ulazna rampa” i “Izlazna rampa” pozivaju listu svih naplatnih stanica u kojoj se u zavisnosti od dugmeta biraju ulazna/izlazna rampa. Za elemente liste postavljen je “onItemClickListener” koji to omogućuje. Pritiskom na dugme “Start” čuvaju se podaci bitni za merenje prosečne brzine i vremena pauze kao što su trenutno vreme, udaljenost ulazne i izlazne naplatne stanice i cena putarine. Ti podaci prenose se na novi ekran u kome se vrši ispis parametara bitnih korisniku (prosečna brzina, odgovarajuće vreme stizanja, potrebna pauza, potrebna brzina, ostatak puta i ruta),

Sledeća bitna “override” metoda je “onLocationChanged”. S obzirom da aplikacija ima mogućnost korišćenja u offline modu, za merenje prosečne brzine korišćene su baze rastojanja, proteklo vreme (od ulaska na naplatnu stanicu tj. pritiska na dugme start) kao i metode “getSpeed” klase “location”. Od GPS-a se pomenutom metodom “getSpeed” na svake dve sekunde dobija trenutna brzina data u metrima u sekundi. Dobijene brzine se čuvaju u kumalativu a potom dele sa brojem upisivanja u isti kako bi se dobila prosečna brzina. U ovoj metodi se izračunava i pređeni put (prosečna brzina se množi proteklim vremenom konvertovanim iz sekunda u sate), dužina potrebne pauze, preostala kilometraža kao i na koliko treba smanjiti brzinu da bi korisnik na naplatnoj stanici imao prosečnu brzinu od 120km/h.

Formula kojom se izračunava potrebna pauza je sledeća:

$$PP = \frac{du}{120km/h} - \frac{du}{vp},$$

Gde su:

- du – ukupna distanca,
- vp – prosečna brzina do tog trenutka.

Formula kojom se izračunava potrebna brzina je sledeća:

$$PB = \frac{du - dt}{\frac{du}{120km/h} - \frac{dt}{vp}},$$

Gde su:

- du – ukupna distanca,
- vp – prosečna brzina do tog trenutka,
- dt – pređena distanca.

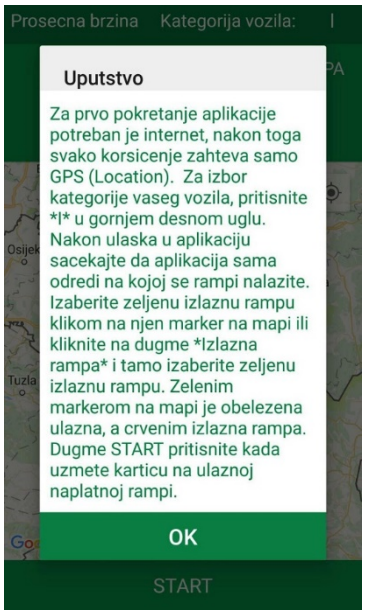
## III. Korisnički interfejs i primena aplikacije

Korisnički interfejs je razvijan u okviru integrisanog razvojnog okruženja za android aplikacije “Android studio”. Ključni elementi za izradu korisničkog intefejsa su objekti, boje, font i tekst. Objekti koji su implementirani u aplikaciju “Prosečna brzina” odgovaraju standardima „Material design“ razvoja korisničkog interfejsa, koji je u skladu sa celokupnom „Android“ platformom, i u ovom slučaju to su: dugme, lista, mapa i reklamni prostor. Boje i font koji su primenjeni u aplikaciji su odabrani sa namerom da budu slični ostalim aplikacijama za pomoć pri vožnji (npr. navigacionim sistemima) kako bi se korisnici što lakše privikli na iterfejs. Na kraju, tekst koji se nalazi u okviru objekata korisničkog iterfejsa je osmišljen na način da korisnike intuitivno nauči korišćenju aplikacije

Mogućnost prilagođavanja korisničkog interfejsa različitim dimenzijama uređaja koji koriste „Android“ operativni sistem omogućava „Relative layout“ opcija u okviru razvojnog okruženja „Android studio“ koja sama izvršava pomenuto prilagođavanje.

### *A. Inicijalno uputstvo*

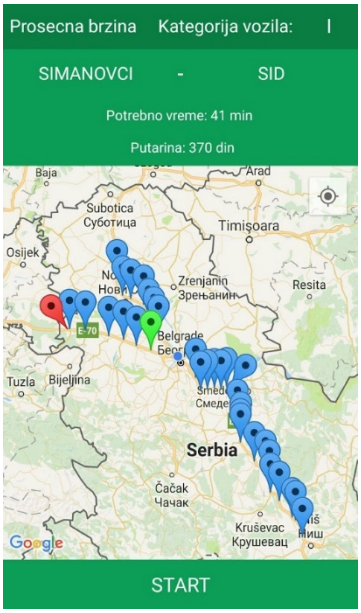
Pri pokretanju aplikacije “Prosečna brzina” kratko uputstvo o načinu korišćenja iste se otvara preko celog ekrana kako bi korisnik imao mogućnost da se detaljno upozna sa svim mogućnostima. Uputstvo se zatvara jednim klikom da na dugme “OK” nakon čega se korisnik prebacuje na početni ekran (slika 1).




Slika 1. Početno uputstvo za korišćenje aplikacije.

*B. Početni ekran*

Početni ekran u aplikaciji služi da korisnik podesi parametre svog putovanja (Slika 2).



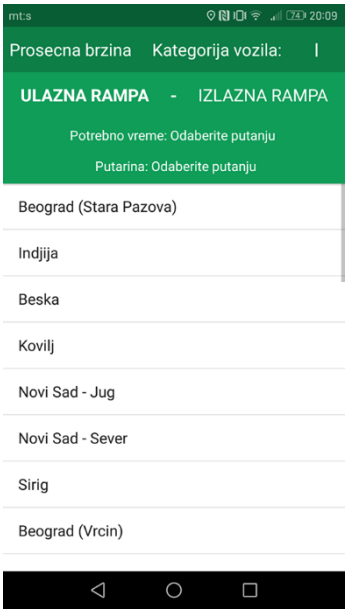

Slika 2. Početni ekran za podešavanje parametara putovanja.

U gornjem desnom uglu se klikom na ikonicu otvara lista putem korisnik bira kategoriju vozila (Slika 3).




Slika 3. Ekran za biranje kategorije vozila.

U zavisnosti od odabrane kategorije vozila i obeleženih ulaznih i izlaznih naplatnih stanica aplikacija korisniku prikazuje cenu putarine i minimalno potrebno vreme putovanja za njegov put. Aplikacija automatski obeležava ulaznu naplatnu stanicu za korisnika tako što uz pomoć GPS tehnologije pronalazi najbližu njegovoj trenutnoj lokaciji, a korisnik može i sam da odabere ulaznu naplatnu stanicu otvaranjem liste klikom na dugme „ULAZNA RAMPA“. Zatim, korisnik bira izlaznu naplatnu stanicu putem liste klikom na dugme „IZLAZNA RAMPA“, odnosno odabirom željene naplatne stanice na mapi (Slika 4).




Slika 4. Ekran za biranje ulazne i izlazne naplatne stanice.

*C. Ekran obaveštavanja korisnika*

Klikom na dugme start korisnik započinje merenje odgovarajućeg vremena stizanja, preostale distance, prosečne brzine, potrebnog vremena za pauzu i potrebnog usporavanja (slika 5).




Slika 5. Ekran za obaveštavanje korisnika.

U zavisnosti od toga da li je prosečna brzina iznad dozvoljenog nivoa ili nije, korisnik u cilju efikasnijeg obaveštavanja prima vizuelnu notifikaciju gde prikaz prosečne brzine menja boju (slika 6), kao i glasovnu notifikaciju kako bi ostao fokusiran na put i vožnju.




Slika 6. Primer promene boje teksta radi obaveštavanja korisnika o prekoračenju dozvoljene brzine.

## IV. KOMERCIJALNA VREDNOST APLIKACIJE I OSTVARENI REZULTATI

Pored toga što je aplikacija razvijena radi povećanja bezbednosti korisnika na autoputevima u Srbiji, još jedan od razloga je i komercijalna korist od iste. Kako korisniici ne bi bili obeshrabreni da preuzmu aplikaciju, preuzimanje je besplatno, a komercijalna korist se ostvaruje putem odvajanja dela interfejsa za reklamni prostor. Kao servis za naplatu reklamnog prostora je odabran "Google AdMob". Ovaj servis omogućava najefikasnije plasiranje i naplatu reklamnog sadržaja.

Ključni indikatori performansi kojima se meri uspešnost aplikacije "Prosečna brzina" i njihove vrednosti u periodu od 02.09.2017 do 28.09.2017 su:

- Broj preuzimanja aplikacije – 18.568
- Broj aktivnih korisnika aplikacije – 14.452
- Prosečna ocena zadovoljstva korisnika – 4.49
- Broj prikaza reklame – 31.810
- Broj klikova na reklamu - 290
- Zarada od reklame – EUR 33.23

## V. ZAKLJUČAK

Aplikacija opisana u radu služi svim korisnicima koji žele da se na što bolji način prilagode novim uslovima na autoputevima u Srbiji. Ovom aplikacijom korisnici mogu da najefikasniji način prate prosečnu brzinu i druge parametre koji će im pomoći da budu bolje informisani i ostanu u okviru zakona prilikom vožnje. Efekat ove aplikacije je povećana bezbednost na autoputevima u Srbiji.

## ZAHVALNICA

Autori bi ovim putem želeli da se zahvale Tomislavu Damnjanoviću koji im je ukazao na inicijalnu ideju i obezbedio im smernice, kao i Danilu Radoviću za savete oko dizajna i korisničkog interfejsa.

## LITERATURA

[1] C. Stein, C. E. Cormen, Ronald L. Rivest, T. H. Cormen, Introduction to algorithms, Massachusetts, MIT press, 2009.
[2] B. Tomić, J. Jovanović, N. Milikić, Z. Ševarac, D. Đurić, Principi programiranja, Beograd, Fakultet organizacionih nauka, 2013.
[3] https://www.google.com/admob, datum pristupa 25.09.2017. godine.
[4] https://play.google.com/store/apps/details?id=com.droidnip.nemanja.naplatnerampe, datum pristupa 27.09.2017. godine.
[5] https://play.google.com/apps/publish, datum pristupa 26.09.2017. godine.
[6] https://developer.android.com, datum pristupa 26.09.2017. godine.

# Laboratorijska maketa za proučavanje kretanja tela po strmoj ravni zasnovana na *Raspberry Pi* mini računaru

Milan Danković, Snežana Janković

*Sadržaj* – U ovom radu prikazana je laboratorijska maketa za proučavanje kretanja tela po strmoj ravni. Dat je kratak teorijski pregled kretanja tela niz strmu ravan, zatim je prikazan izgled laboratorijske makete i na kraju su predložene ideje kako se ova maketa može koristiti u obrazovne svrhe.

## I. UVOD

Kretanje tela niz strmu ravan ima ogromnu praktičnu primenu i skoro svakodnevno se susrećemo sa tim kretanjima u svakodnevnom životu, pa bi iz tih razloga bilo dobro da se ova materija što više razjasni. Princip strme ravni ljudi su koristili od davnina i to da podignu teret znatno manjom silom od težine tereta, tako što su teret vukli po strmoj ravni. Tako je opšteprihvaćena teorija da su milioni blokova od kojih su sagrađene piramide u Egiptu morali da budu podizani uz strmu ravan, koja je u tom slučaju korišćena kao prosta mašina. Da bi se blokovi pri vrhu piramida podigli na tu visinu bilo je potrebno da se postave strme ravni čija je dužina iznosila i više stotina metara, a prostor između blokova i peska je ispunjavan posebnim materijalima kako bi se smanjio koeficijent trenja, a samim tim i gubici energije.

Problemi trenja u tehničkoj fizici rešavaju se makropristupom sa eksperimentalnim određivanjem koeficijenta trenja za aktuelne uslove kretanja [1]. Ovaj rad predstavlja nastavak prethodnih istraživanja [2], ali je sada uz razvijeni simulator napravljena i savremena maketa strme ravni koja je grafički ilustrovana na slici 1. Kretanje tela u ovakvom sistemu detaljno je analizirano i opisano odgovarajućim zakonima fizike i matematike, što će biti opisano nizom pratećih jednačina. Izvedene jednačine predstavljale su odličnu osnovu za razvoj simulatora [2], a ovde će poslužiti u svrhu određivanja koeficijenta trenja.

## II. TEORIJSKA POSTAVKA LABORATORIJSKE MAKETE

Pod strmom ravni podrazumeva se ravna površina koja sa horizontalnom podlogom zaklapa ugao α, kao što je ilustrovano na slikama 1 i 2. U idealnom slučaju strma ravan visine *h*, osnove *d* i dužine *l* napravljena je od čvrstog materijala, a deformacije usled uzajamnog delovanja tela sa njom su zanemarljive.

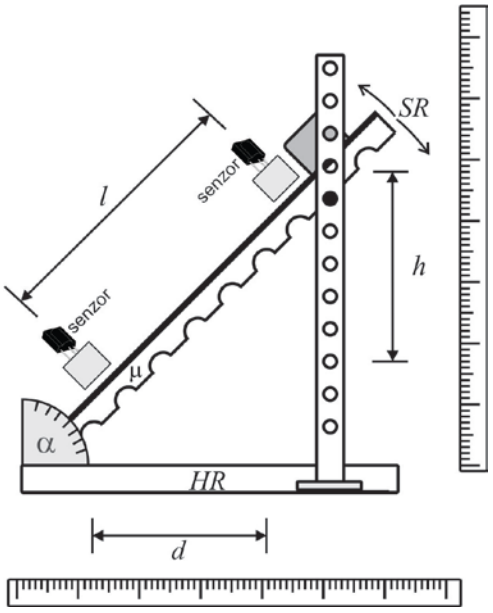
Sile koje deluju na telo, kada je ono na strmoj ravni, su sila Zemljine teže, sila trenja i sila reakcije podloge, kao što je prikazano na slici 2. Sila Zemljine teže $\vec{F}_g$ deluje vertikalno na dole, pa je pogodno da ovu silu razložimo na komponentu normalnu na strmu ravan ( $\vec{F}_N$ ) i komponentu paralelnu sa dužinom strme ravni ( $\vec{F}_P$ ). Sila trenja ( $\vec{F}_{tr}$ ) deluje u ravni dodirne površine tela i podloge u smeru suprotnom od smera kretanja tela. Pošto telo pritiska podlogu silom $\vec{F}_N$, istom tolikom silom, ali u suprotnom smeru deluje podloga na telo – reakcija podloge ( $\vec{N}$ ).

Analizom slike 2 možemo doći do jednačine kojom se određuje koeficijent trenja između tela koje se kreće po strmoj ravni i same podloge. Sile paralelne sa pravcem kretanja tela utiču na njegovo kretanje i to je opisano sledećim jednačinama:

$$m \cdot a = F_p - F_{tr} \quad (1)$$

$$F_p = Q \cdot \sin\alpha = mg \cdot \sin\alpha \quad (2)$$

$$F_{tr} = \mu \cdot N = \mu \cdot Q \cdot \cos\alpha = \mu \cdot mg \cdot \cos\alpha \quad (3)$$




Slika 1. Grafički prikaz idejnog rešenja makete za proučavanje kretanja tela po strmoj ravni.

Milan Danković je učenik četvrtog razreda Gimnazije "Bora Stanković", Vožda Karađorđa 27, 18000 Niš, Serbia, E-mail: dankovicmilan@gmail.com

Snežana Janković je profesor Fizike u Gimanziji "Bora Stanković", Vožda Karađorđa 27, 18000 Niš, Serbia, E-mail: nenajankovic62@gmail.com

$$m \cdot a = mg \cdot \sin\alpha - \mu \cdot mg \cdot \cos\alpha \quad (4)$$

$$\mu = \frac{g \cdot \sin\alpha - a}{g \cdot \cos\alpha} \quad (5)$$

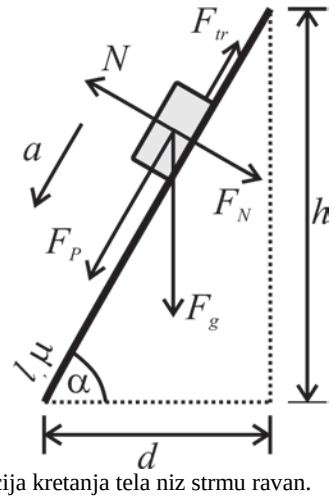
Na osnovu jednačine 5 zaključujemo da je za određivanje koeficijenta trenja ($\mu$) neophodno poznavanje ubrzanje tela na strmoj ravni ($a$) i ugla strme ravni ($\alpha$). Ugao strme ravni određuje se iz geometrije trougla ili jednostavnim očitavanjem vrednosti sa uglomera. Ubrzanje određujemo iz jednačina koje opisuju kretanje tela niz strmu ravan. Većina eksperimenata sa kretanjem tela niz strmu ravan odvija se bez početne brzine. U tom slučaju telo se kreće ravnomerno ubrzano, što je opisano sledećim jednačinama za brzinu i pređeni put:

$$v = a \cdot t \quad (6)$$

$$S = \frac{a \cdot t^2}{2} \quad (7)$$

Na osnovu jednačine 7 zaključujemo da je za određivanje ubrzanja tela na strmoj ravni potrebno odrediti rastojanje između dva senzora ($l$, naznačeno na slici 1) i vreme za koje telo pređe taj put ($t_{mereno}$, za šta će biti upotrebljen mikroračunar *Raspberry Pi* [3]) Na osnovu ovih podataka dobija se:

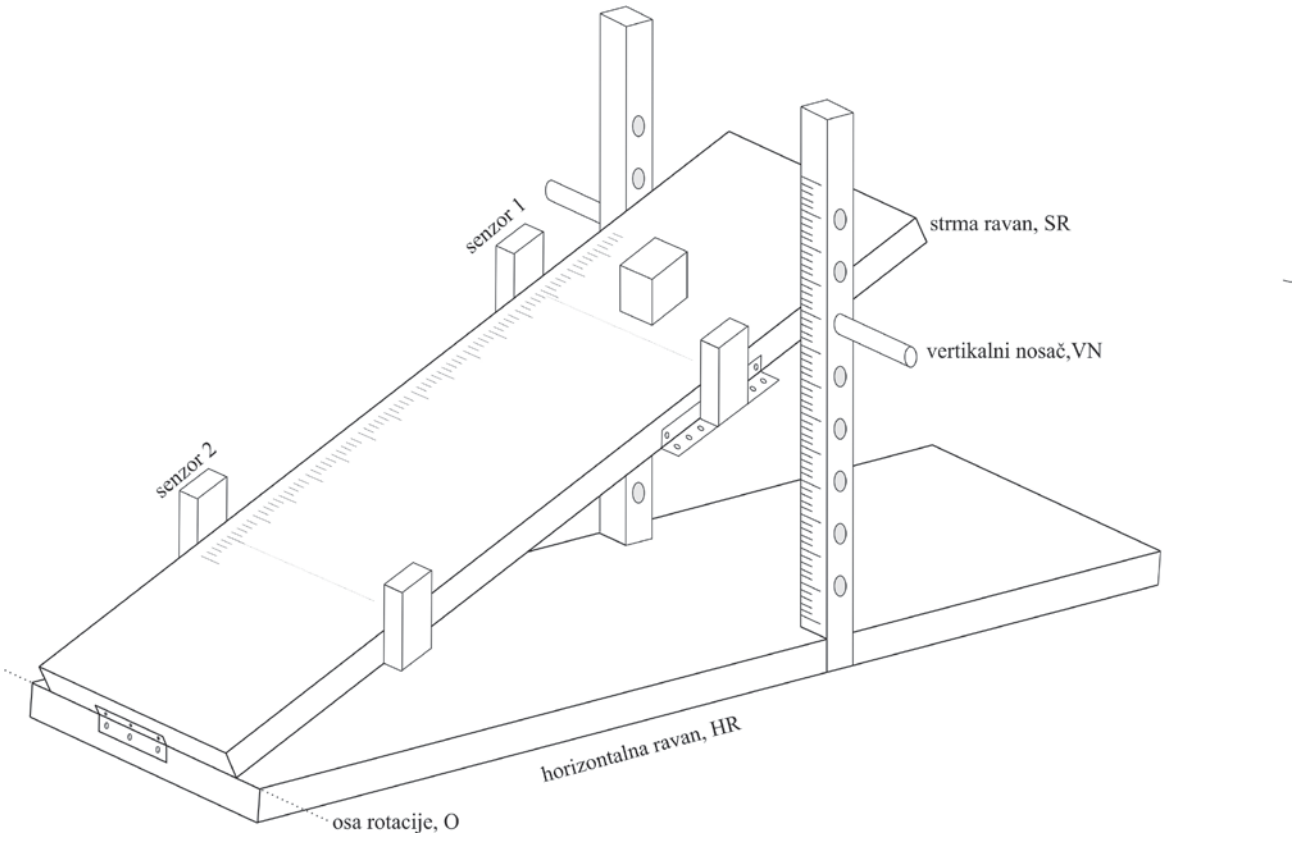
$$a = \frac{2 \cdot l}{t_{mereno}^2} \quad (8)$$




Slika 2. Ilustracija kretanja tela niz strmu ravan.

## III. PRAKTIČNA REALIZACIJA LABORATORIJSKE MAKETE

Napred navedene jednačine služe kao teorijska postavka ove vežbe, a kompletno kretanje najbolje je eksperimentalno proveriti (slika 3), pri čemu treba imati u vidu da će zbog brojnih zanemarivanja uvek biti odstupanja između teorije i eksperimenta. Ovo je realan problem koji postoji gotovo u svim eksperimentima.



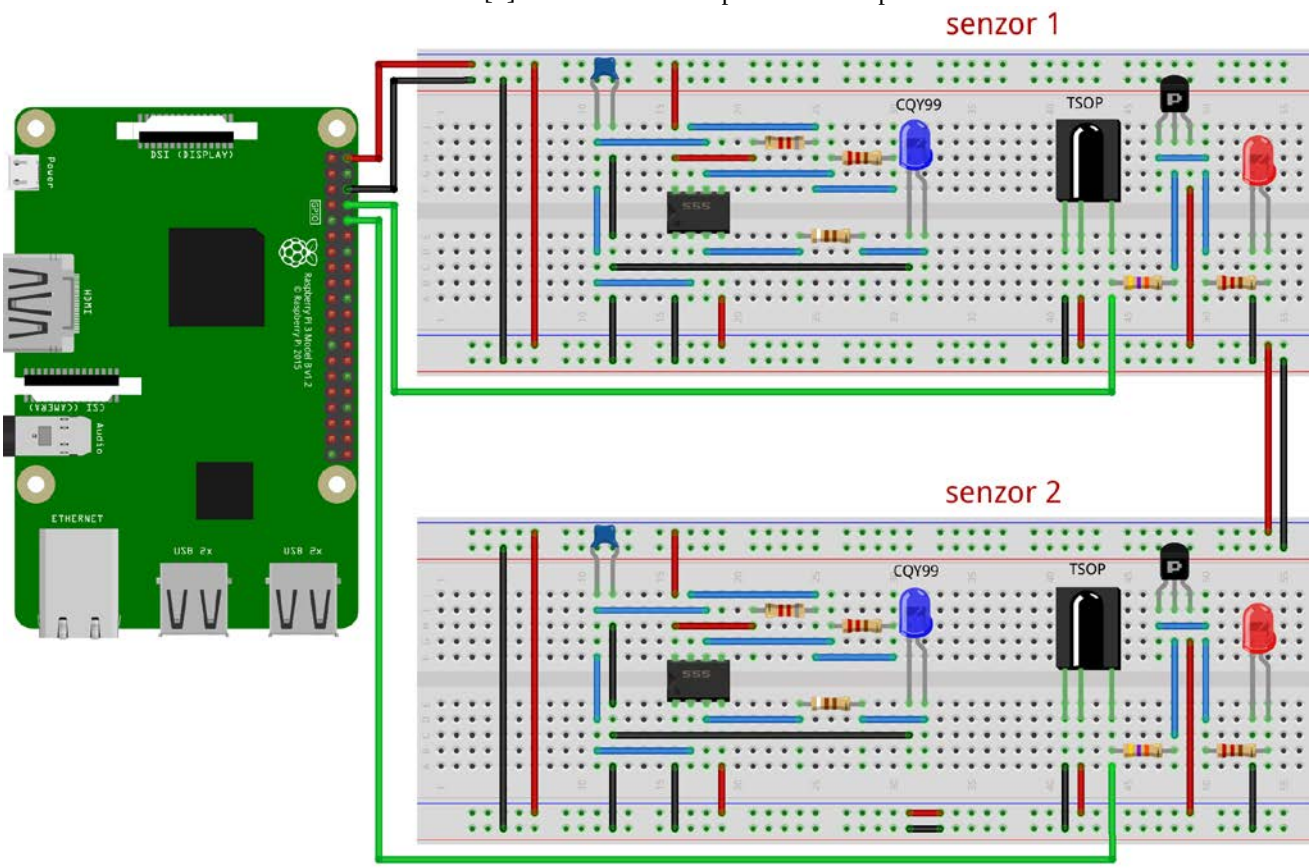

Slika 3. Eksperimentalna postavka laboratorijske vežbe.

Laboratorijska maketa ima mogućnost podešavanja svih dužina, ugla i koeficijenata trenja između podloga i tela koje se kreće po njima, što omogućava realizaciju čitavog niza različitih eksperimenata i donošenje velikog broja zaključaka. Na primer, rotiranjem strme ravni SR oko ose rotacije O, što se postiže pomeranjem vertikalnog nosača VN, može da se podešava ugao strme ravni, što se jednostavno očitava uglomerom. Takođe, pomeranjem vertikalnog nosača VN može da se menja visina strme ravni, a sa pomeranjem pokretnog senzora 1 može da se menja i dužina strme ravni. Konačno, lepljenjem različitih podloga (guma, drvo, plastika, staklo, metal…) na telo koje se spušta niz strmu ravan, odnosno lepljenjem podloga na strmu ravan moguće je podešavati i koeficijente trenja.

Sa aspekta elektronike srž ove makete predstavljaju senzorski blokovi (senzor 1 i senzor 2) čija je uloga očitavanje vremena ($t_{mereno}$) neophodnog za određivanje koeficijenta trenja (jednačine 8 i 5). Senzorski blok se sastoji iz dva dela, predajne i prijemne strane. Prijemnu stranu čini IR prijemni modul za daljinsku kontrolu TSOP31236 [4]. Na osnovu teničke dokumentacije jasno se zaključuje da TSOP31236 reaguje na IR signal frekvencije 36 kHz, pa je sa predajne strane generisan signal čija je učestanost 36 kHz i koji se emituje sa IR diode CQY99 [5]. Za generisanje signala frekvencije 36 kHz iskorišćeno je NE555 kolo u astabilnom režimu rada [6]. Proračun pokazuje da je pored NE555 kola potrebno uzeti i otpornike R1=47 kΩ, R2=91 kΩ i kondenzator C=390 pF.

Osnovna ideja je da se kretanjem tela niz strmu ravan vrši presecanje IR snopa i detektuje vreme koje protekne između presecanja snopa kod senzorskog boka 1 i kod senzorskog bloka 2 ($t_{mereno}$). U ovu svrhu iskorišćen je Raspberry Pi mini računar. Ovaj računar veličine kreditne kartice ima sve ono to što je potrebno: četiri USB porta, za povezivanje miša i tastature, HDMI izlaz za video signal i prikazivanje na monitoru, 26 GPIO pinova za prijem signala sa senzora i pinove koji daju napon od 5 V, kojim se napajaju senzorski blokovi. Na slici 4 prikazano je kompletno povezivanje *Raspberry Pi* računara i senzorskih blokova, pri čemu treba napomenuti da su na *Raspberry Pi* računar priključeni i miš, tastatura i monitor.

Treba napomenuti da je očitavanje signala sa TSOP senzora urađeno direktno sa pina OUT. Međutim, da bi imali vizuelnu indikaciju kada telo koje se kreće niz strmu ravan vrši presecanje IR snopa iskorišćena je i jedna LED dioda. Imajući u vidu da TSOP31236 preko pina OUT daje struju od najviše 5mA bilo je neophodno izvršiti pojačanje struje sa pina OUT. Kada je u pitanju TSOP31236 ovaj problem i njegovo rešenje su vrlo dobro poznati u literaturi, pa je rešenje preuzeto u originalu. Korišćenjem ove dve LED diode imamo mogućnost i vizuene indikacije kada telo preseca IR snopove kod senzorskih blokova 1 i 2.




Slika 4. Povezivanje Raspberry Pi mini računara i senzorskih blokova.

## IV. PROGRAMSKI KOD ZA PRORAČUNAVANJE KOEFICIJENTA TRENJA

Na *Raspberry Pi* mikroračunaru instaliran je operativni system Raspbian [7]. Za izradu programa koji meri vreme ($t_{mereno}$) korišćen je programski jezik Python 3 [8]. Izvorni kod kompletnog programa dat je u nastavku.

```
import RPi.GPIO as GPIO
import time, math

G = 9.81 #gravitaciono ubrzanje m/s^2

GPIO.setmode(GPIO.BCM)

senzor1 = 14
senzor2 = 15

GPIO.setup(senzor1, GPIO.IN)
GPIO.setup(senzor2, GPIO.IN)

p = True

d = float(input("Unesi d: "))
h = float(input("Unesi h: "))
l = math.sqrt(d*d + h*h)
print("l = %f m" % l)
print("Pusti telo!")
time_lasted = 0.0

while(p):
    if(GPIO.input(senzor1)):
        start = int(round(time.time() * 1000))
        time.sleep(0.3)
        while(not(GPIO.input(senzor2))):
            time.sleep(0.001)
        end = int(round(time.time() * 1000))
        time_lasted = int(end - start)
        print("t = %.5f s" % (time_lasted/1000))
        p = False
    time.sleep(0.001)

a = (2*l)/((time_lasted/1000) * (time_lasted/1000))
m = (G*(h/l) - a)/(G*(d/l))

print("a = %.5f m/s^2" % a)
print("Koeficijent trenja iznosi %.5f" % m)

GPIO.cleanup();
```

Laboratorijska maketa i program testirani su pri sledećim uslovima: visinska razlika između senzora iznosila je 0.7 m, dok je horizontalna udaljenost iznosila 1m. Nakon unosa ovih podataka (*h* i *d*) program automatski izračunava put koji telo prelazi (*l*) i obaveštava da može da se pusti telo niz strmu ravan tako što ispiše informaciju: “Pusti telo!“. Ukoliko telo počne da se kreće niz strmu ravan ono preseče IR snop kod prvog senzorskog bloka gde se startuje merenje vremena, a kada preseče IR snop kod drugog senzorskog bloka tu se završava merenje vremena i odmah se ispisuje izmereno vreme, ubrzanje tela i tražena vrednost koeficijenta trenja. Za ovakvu eksperimentalnu postavku dobijeni su sledeći rezultati.

```
Unesi d: 1
Unesi h: 0.7
l = 1.220656 m
Pusti telo!
t = 0.84400 s
a = 3.42719 m/s^2
Koeficijent trenja iznosi 0.27356
```

Moguće je napraviti verziju programa gde korisnik bira koja dva od moguća četiri parametra (*d*, *h*, *l* i $\alpha$) želi da unese, a preostala dva se automatski izračunavaju. Takođe, moguće je dodati još jedan senzorski blok (senzor 3) i time eliminisati grešku početne brzine. Naime, sadašnjom postavkom eksperimenta telo bi moralo da se pusta bez početne brzine i odmah pored senzora 1. Korišćenjem tri senzora telo može da se pušta sa proizvoljne visine i sa početnom brzinom.

## V. ZAKLJUČAK

U radu je detaljno objašnjeno kretanje tela niz strmu ravan i izvedene su formule koje predstavljaju osnovu za razvoj laboratorijske makete. Za realizaciju laboratorijske makete iskorišćen je *Raspberry Pi* mikro računar. Osmišljena laboratorijska maketa omogućava određivanje koeficijenta trenja, ali njena modifikacija omogućava realizaciju čitavog niza različitih eksperimenata iz oblasti fizike koji bi mogli da motivišu učenike, da pospeše sticanje znanja iz fizike i da podstaknu korišćenje eksperimenata u nekim drugim predmetima, naučnim disciplinama ili svakodnevnom životu.

## ZAHVALNICA

Ovaj rad realizovan je u okviru Projekta “Računarske aplikacije za Galilejeve zakone fizike” koji realizuju Snežana Janković i Milan Danković. Projekat je finansiran od strane Centra za promociju nauke.

## LITERATURA

[1] Trenje. Dostupno na: https://svejefizika.wordpress.com/video-7/trenje/
[2] M. Danković, M. Stanković, “Laboratorijska maketa za proučavanje kretanja tela po strmoj ravni”, IEEESTEC 8th Student projects conference, Niš 2015, pp. 211-216.
[3] Raspberry Pi 2 model B mikro računar. Dostupno na: https://www.raspberrypi.org/products/raspberry-pi-2-model-b/
[4] IR prijemni modul TSOP31236 Dostupno na: https://www.vishay.com/docs/82492/tsop312.pdf
[5] IR dioda CQY99. Dostupno na: http://www.datasheet-pdf.com/PDF/CQY99-Datasheet-ETC-520832
[6] Integrisano kolo NE555. Dostupno na: http://www.ti.com/lit/ds/symlink/ne555.pdf
[7] Operativni system Raspbian. Dostupno na: https://www.raspbian.org/
[8] Programski jezik Python 3. Dostupno na: https://www.python.org/

# Analiza paketskog saobraćaja taktičke VVF radio mreže

Danilo Stanojević, Milica Kostić

*Sadržaj* – Vojne bežične taktičke mreže za prenos paketski uokvirenih podataka zasnivaju se na brzom formiranju, sa zaštićenim komunikacionim tokovima, što je osnovna odlika ovakvih mreža. Ovakve mreže se formiraju u VVF frekvencijskom opsegu primenom frekvencijskog skakanja i tehnike pristupa medijumu zasnovane na CSMA/CD (Carrier Sense MultipleAccess – Collision Detection) principu. Komunikacije je moguće ostvariti na udaljenosti do nekoliko desetina kilometara. Prenos je zaštićen primenom modulacionih tehnika za prenos i ključevima koji se implementiraju u same uređaje za prenos informacija. Ovakve mobilne zaštićene ad-hoc mreže imaju brzine prenosa svega nekoliko desetina kb/s, što je često ograničavajući faktor prilikom prenosa paketa.

Za realizaciju ovog istraživanja izrađena je aplikacija u programskim paketima Microsoft Visual Studio 2013 i Microsoft SQL Server 2012. U radu je prikazan i analiziran jedan realni mrežni scenario zaštićene ad-hoc radio mreže u kojoj su ispitane mogućnosti mreže sa aspekta prenosa fajlova u novoizrađenoj aplikaciji. Testirana je radio mreža pronalaženjem odgovarajuće optimalne količine informacija koja može da se prenese u realnom vremenu kroz radio mrežu.

## I. UVOD

Korišćenjem vrlo pristupačnih bežičnih mrežnih uređaja kao što su kartice za bežični pristup, bežične pristupne tačke, bežični ruteri, zatim korišćenjem javnog sistema mobilne telefonije sa svojim modernim baznim stanicama, koje podržavaju inteligentnu raspodelu spektralnih resursa, jednostavno se ostvaruju velike brzine prenosa i željeni kvalitet prenosa paketa u bežičnim komunikacijama. Od najšire rasprostranjenih standardizovanih tehnika radio pristupa, ističu se WiFi (IEEE 802.11), WiMAX (IEEE 802.16), 3G mobilna telefonija (CDMA 2000, UMTS, HSPA), i tehnike 4G bežičnog pristupa (HSPA +, IEEE 802.16e i LTE). Ostvarivanje željenih kvaliteta servisa postiže se oslanjanjem na stacionarnu infrastrukturu [1].

Međutim, kada je nemoguće oslanjanje na stacionarne elemente usled elementarnih nepogoda, gubitka napajanja, uništenja prenosnih puteva itd. najefikasnije rešenje jeste formiranje bežične radio ad-hoc mreže. Ovakve mreže u Vojsci Srbije formiraju se u VVF opsegu, od 30 MHz do 88 MHz, uz primenu radio uređaja TRC-9210, čiji je proizvođač francuska firma Thales.

Danilo Stanojević je sa Vojne Akademije, Univerzitet odbrane u Beogradu, Generala Pavla Jurišića Šturma 33, 11000 Beograd, Srbija, E-mail: dstanojevic94@gmail.com.

Milica Kostić je student na Elektronskom fakultetu, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: kosticka94@hotmail.com

## II. RADIO UREĐAJ TRC-9210

Radio uređaj TRC-9210 (slika 1) je prenosni, primopredajni, softverski definisani radio uređaj VVF opsega male snage. U radio uređaj je ugrađen GPS prijemnik i IP ploča. Namenjen je za održavanje radio veze na kraćim odstojanjima u poljskim uslovima rada u mestu i pokretu. Omogućava prenos nezaštićenog ili zaštićenog govora i podataka u različitim komunikacionim režimima i vrstama rada [2]. Radio uređaj može da radi u sledećim komunikacionim režimima:

1. Borbena radio mreža (CNR - Combat Network Radio) za simpleksni prenos govora i podataka,
2. IP-MUX namenjen za istovremen prenos govora i IP podataka i
3. IP-PAS namenjen samo za prenos IP podataka.

Za realizaciju ovog istraživanja radio uređaj TRC-9210 konfigurisan je da radi u IP-PAS komunikacionom režimu.



Slika 1. VVF radio uređaj TRC-9210.

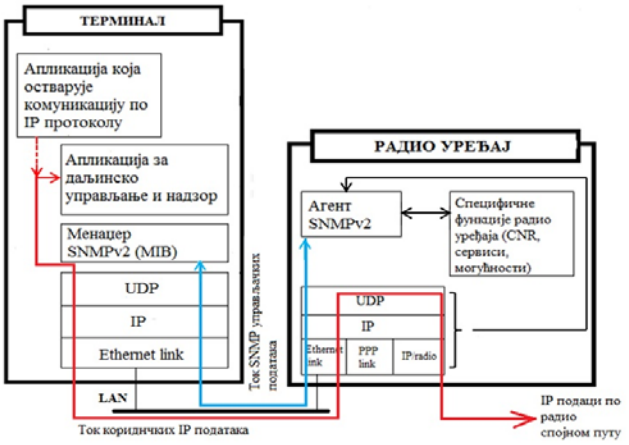
## III. PAKETSKI PRENOS U IP-PAS KOMUNIKACIONOM REŽIMU

U komunikacionom režimu IP-PAS, radio uređaj TRC-9210 vrši prenos paketa u samoformirajućoj ad-hoc mreži primenom frekvencijskog skakanja do 300 skokova u sekundi, odnosno vreme zadržavanja na svakoj frekvenciji je 3ms. U ovakvoj mreži moguće je prenositi samo paketski uokvirene podatke i računar mora biti povezan sa radio stanicom preko IP interfejsa. Maksimalan broj učesnika u mreži je 32, a prenos paketa se obavlja u ramovima

veličine 4kB. Maksimalna brzina prenosa je do 19200 b/s. Međutim ona se menja u zavisnosti od uslova propagacije. U mrežama koje rade u IP-PAS režimu, moguće je ostvariti prenos preko radio kanala brzinama 2400b/s, 4800b/s, 9600b/s i 19200b/s.

Sinhronizacija između radio stanica u jednoj mreži održava se slanjem hello paketa i detekcijom susednih mreža. U IP-PAS komunikacionom režimu mora se definisati stanica koja će biti inicijator sinhronizacije i kao takva prilikom formiranja mreže uspostaviti sinhronizaciju slanjem referentnog takta ostalim stanicama u mreži.

Svaki radio uređaj ima jedinstvenu adresu u mreži i ona je prevedena IP radio adresa. Svaka stanica ima u svakom trenutku ostvaren pristup medijumu i mogućnost iščitavanja podataka o prisutnim stanicama u mreži iz MIB (Management Information Base) baze uređaja upotrebom SNMP (Simple Network Menagment Protocol) protokola. U određenim vremenskim trenucima stanica distributivno emituje poruke koje se tiču mrežne topologije, određuje listu susednih i stanica koje joj nisu susedne. SNMP upravljanje (slika 2), u načelu omogućava konfiguraciju VVF generacije radio uređaja. Funkcionalnosti koje se odnose na rutiranje paketa u okviru uređaja omogućene su u IP komunikacionim režimima (IP-MUX i IP-PAS), a odnose se na UDP i TCP transfere.

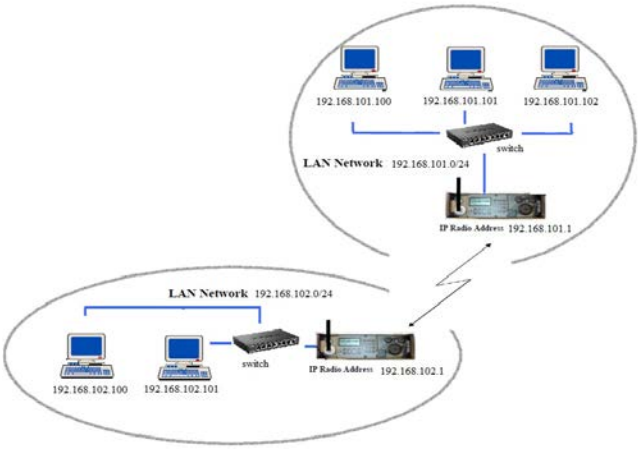



Slika 2. SNMP upravljanje radio uređajem.

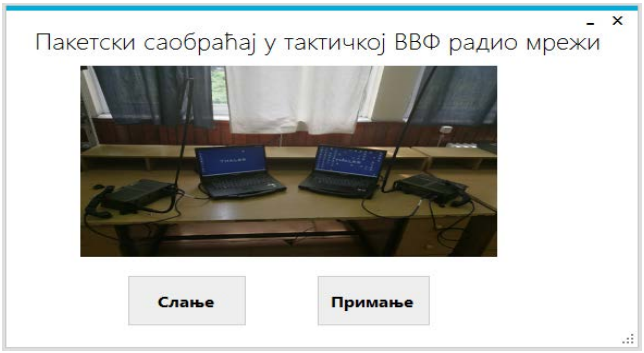
## IV. ANALIZA REZULTATA

Na slici 3 prikazan je izgled mrežnog scenarija koga čine dve LAN mreže. To su LAN mreža 192.168.101.0/24 i LAN mreža 192.168.102.0/24. Od mrežne opreme korišćeno je pet računara sa mrežnom karticom, dva sviča i dva radio uređaja TRC-9210 sa IPC2-EC karticom. Na računarima je instalirana aplikacija koja je napravljena i korišćena za realizaciju prenosa paketa. Prikaz glavnog prozora aplikacije dat je na slici 4. Aplikacija radi po UDP protokolu i ima dva moda rada:

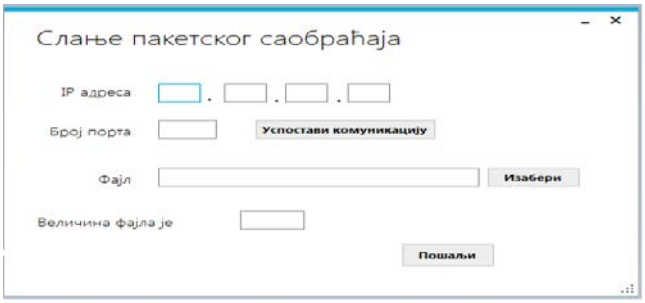
- Slanje – Mod za slanje paketa (slika 5),
- Primanje – Mod za prijem paketa (slika 6).



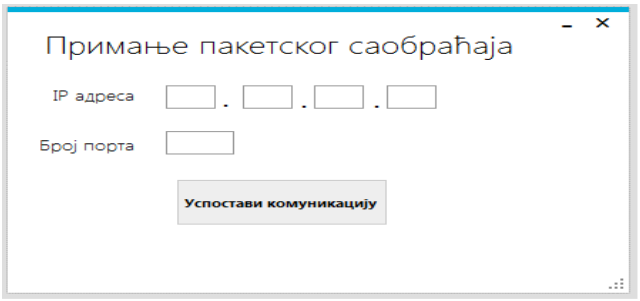

Slika 3. Mrežni scenario za istraživanje.




Slika 4. Izgled glavnog prozora aplikacije.

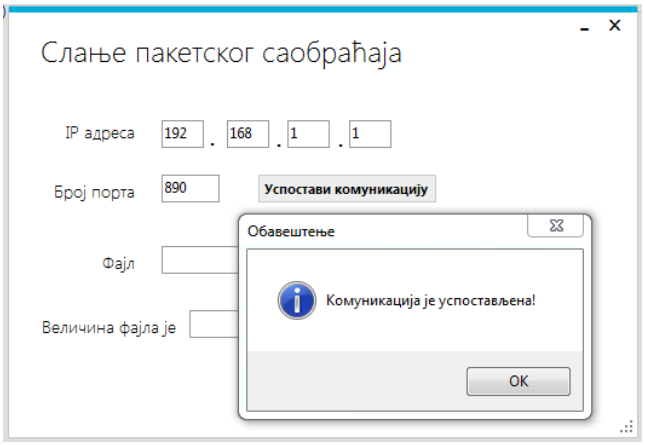



Slika 5. Izgled moda za slanje paketa.



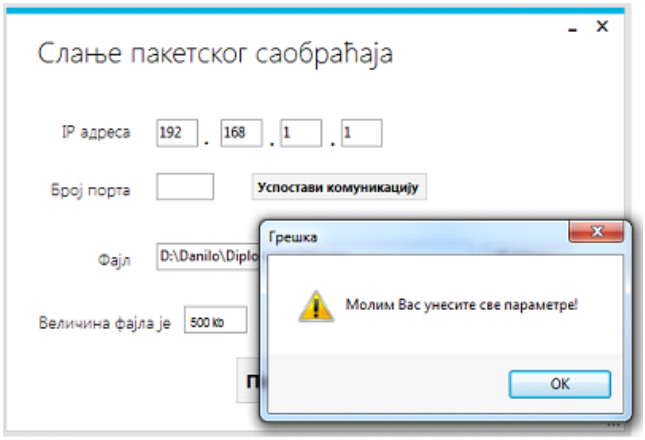

Slika 6. Izgled moda za prijem paketa.

Kada korisnici aktiviraju aplikacije na krajnjim računarima i popune polja IP adresa (polje za unos IP adrese krajnjeg korisnika sa kojim treba uspostaviti komunikaciju) i broj porta (polje za unos broja porta po kome se vrši prenos paketa. Mora biti isti i na računaru za slanje i na računaru za prijem paketa), klikom na dugme „uspostavi komunikaciju“ uspostavlja se komunikacija između njih (slika 7). U procesu uspostave komunikacije najpre se vrši prenos pozivne sekvence preko zajedničkog radio kanala za sve učesnike. Ona omogućava sinhronizaciju stanica i zauzima jedan vremenski slot (oko 30 frekvencijskih skokova).




Slika7. Obaveštenje o uspostavljenoj komunikaciji.

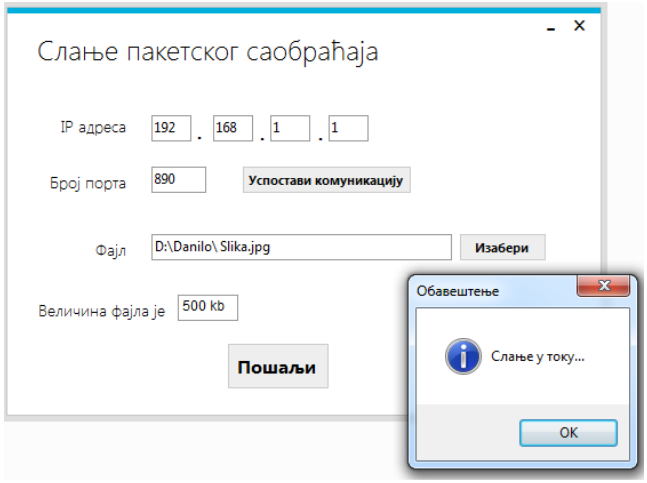
Takođe postoji i obaveštenje ukoliko korisnik ne popuni sva polja u aplikaciji, koja su neophodna za uspostavu komunikacije i prenos informacija (slika 8).




Slika 8. Prikaz prozora kada nisu uneti svi parametri.

Kada je komunikacija uspostavljena odabirom fajla i klikom na dugme „pošalji“ počinje proces slanja paketa u ramovima veličine 4 kB koji je prikazan na slici 9.

Radio uređaj TRC-9210 konfigurisan je da radi u vrsti rada frekvencijsko skakanje u dva podopsega od 30-35 MHz i 50-55 MHz sa nominalnom brzinom prenosa 19200 b/s i snagom 0.5W. Rezervna brzina prenosa se primenjuje u slučaju kada je kvalitet prenosa narušen i ona je podešena na 9600 b/s.




Slika 9. Prikaz prozora u toku slanja.

Izvršeno je merenje kojim je posmatran uticaj promene veličine fajla koji se šalje u odnosu na vreme potrebno za prenos. Dobijeni rezultati prikazani su u tabeli 1. Na osnovu rezultata može se zaključiti da se sa povećanjem veličine informacije povećava i vreme potrebno za prenos, što je i bilo očekivano. Informacije veličine do 50 kb pogodne su za prenos u VVF radio mreži. Informacije veće od 50 kb zahtevaju preveliko vreme za prenos, odnosno uzrokuju kašnjenje. Za informacije veće od 50 kb potrebno je prethodno izvršiti neposrednu obradu takvih informacija, odnosno kompresiju.

Neke od trenutnih mana aplikacije su te što vreme potrebno za prenos fajla nije mereno u okviru aplikacije. Takođe, ne postoji izveštaj u aplikaciji o tome da je fajl primljen, odnosno da je informacija poslata s kraja na kraj mrežne topologije.

TABELA I
REZULTATI ISTRAŽIVANJA

| Veličina fajla [kb] | Vreme potrebno za prenos [s] |
|---|---|
| 10 | 22.44 |
| 20 | 38.01 |
| 30 | 58.90 |
| 40 | 68.86 |
| 50 | 71.69 |
| 80 | 158.12 |
| 100 | 193.88 |
| 200 | 282.11 |
| 500 | 652.01 |
| 800 | 921.40 |
| 1000 | 1256.62 |

## V. ZAKLJUČAK

Dosadašnje iskustvo u radu sa radio uređajima VVF opsega pokazuje da su mreže ovog tipa vrlo povoljne za ostvarivanje komunikacije između korisnika čiji aplikativni zahtevi nisu preveliki, što je i potvrđeno rezultatima ovog istraživanja. Razvijanjem aplikacije koja vrši prenos paketa između korisnika omogućen je prenos fajlova po UDP protokolu u VVF ad-hoc radio mreži. Na osnovu prikazanih rezultata može se doneti zaključak o vezi količine prenetih podataka i vremena potrebnog za prenos poslatih podataka. Zaključeno je i da upotreba VVF radio uređaja nije preporučljiva za servise koji zahtevaju visoke brzine prenosa i konstantan protok. Naime, ovakve mobilne radio mreže pogodne su za nepouzdan prenos i zbog toga treba izbegavati prenos fajlova koji sadrze bitne informacije zbog gubitaka paketa i nemogućnosti ponovnog slanja, sto je odlika UDP protokola.

Dalja istraživanja bice usmerena na razvijanju aplikacije, dodavanju informacije o vremenu prenosa kao izveštaja nakon uspešnog prenosa fajla. Za potrebe prenosa trebalo bi programirati kompresor informacije koji bi omogućio kompresiju informacije pre samog prenosa i na taj način smanjio potrebno vreme za prenos. S obzirom da se kompresijom narušava kvalitet informacije bilo bi poželjno programirati u aplikaciji takav kompresor koji korisniku omogućava da sam podešava parametre kao što su maksimalna brzina prenosa i kvalitet same informacije.

## LITERATURA


[1] Bajčetić J., Pavlović B., „Analiza e-mail servisa kroz IP radio VVF radio mrežu“, Konferencija PosTel 2011, Beograd, 2011.

[2] Devetak S., Bajčetić J., „Pravilo prenosni radio uređaj TRC-9210-3“, Vojnoizdavački zavod, Beograd, 2010.

[3] Stanojević D., Bondžulić B., Pavlović B., Petrović V., „Uticaj parametara kvaliteta servisa na subjektivnu i objektivnu procenu kvaliteta videa“, Konferencija ETRAN, Kladovo, 2017.

[4] Wang X., „Mobile Ad-Hoc Networks: Protocol Design“, InTech, 2011.

# Poređenje vremena izvršenja funkcije kloniranja objekata bez upotrebe i sa upotrebom refleksije u programskom jeziku C#

Petar Mijović

*Sadržaj* – U ovom radu predstavljeno je poređenje vremena izvršenja funkcija kloniranja objekata korišćenjem refleksije, naspram korišćenja hardkodiranja. Za test primere uzeto je sedam različitih klasa. Testovi su pisani u programskom jeziku C#.

## I. UVOD

U informatici, refleksija je sposobnost nekog računarskog programa za ispitivanje i izmenu svoje strukture i ponašanja (konkretno vrednosti, meta-podatke, osobine i funkcije) za vreme izvršenja programa (*runtime*) .[1] Pored navedenih osobina, refleksija u programskom jeziku C# omogućava i kreiranje potpuno novih klasa kao i njihovih instanci.

Neke od primena refleksije su u metaprogramiranju pri testiranju softvera (kreiranje test objekata u *runtime-u*), kod implementacije fleksibilnih komponenti (*property editor*, u ovom slučaju implementacija editora bi bila jedinstvena, a omogućila bi da se editoru prosledi instanca bilo koje klase, svaka instanca bi bila korektno prikazana).

## II. OPISI TEST KLASA

Za svrhu testiranja vremena izvršenja funkcija kloniranja korišćenjem refleksije i hardkodiranja korišćeno je sedam klasa koje se mogu podeliti u dve test grupe. Prva test grupa služi za poređenje porasta vremena izvršenja funkcija kloniranja sa porastom broja atributa, dok se druga grupa test klasa koristi za poređenje promene vremena izvršenja funkcije kloniranja u zavisnosti od tipova podataka atributa klasa.

U prvu test grupu, grupa koja služi za poređenje porasta vremena izvršenja funkcija kloniranja sa porastom broja atributa, spadaju sledeće klase:

1. TestClassSize1 – 2 atributa
2. TestClassSize2 – 25 atributa
3. TestClassSize3 – 50 atributa
4. TestClassSize4 – 20 atriubta kompleksnog tipa (10 tipa TestClass3)

Kompleksni tip je klasa koja se sastoji iz više atributa, TestClassSize3 koja je navedena kao primer sastoji se iz atriuba tipa int, bool, double gde su ti tipovi podataka elemntarni, nisu kompleksni.

U drugu test grupu koja se koristi za poređenje promene vremena izvršenja funkcije kloniranja u zavisnosti od tipova podataka atributa klasa spadaju sledeće klase:

1. TestClassType1 – 20 atriubuta vrednosnih elementarnih tipova
2. TestClassType2 – 20 atributa kompleksnih referentnih tipova
3. TestClassType3 – 10 atributa vrednosnih elementarnih tipova i 10 atributa kompleksnih referentnih tipova

Razlika između referentnih i vrednosnih tipova jeste u ugrađenom načinu na koji operatori interaguju sa podacima datog tipa. Na primer operator dodele se u slučaju vrednosnog tipa ponaša tako da kopira vrednost atriubta promenljive sa desne strane operatora i tu vrednost dodeljuje elementu sa leve strane operatora. U slučaju referentnog tipa događaji su drugačiji, konkretno, vrši se dodela adrese (reference) koja u memoriji pokazuje na reprezentaciju objekta sa desne strane operatora dodele.

TABELA 1
BROJ I TIPOVI ATRIBUTA U TEST KLASAMA

| Grupa | Klasa | BA | BPA | BKA | BRA | BVA |
|---|---|---|---|---|---|---|
| Test grupa 1 | TestClassSize1 | 2 | 2 | 0 | 0 | 2 |
| | TestClassSize2 | 25 | 21 | 4 | 4 | 21 |
| | TestClassSize3 | 50 | 40 | 10 | 10 | 40 |
| | TestClassSize4 | 20 | 0 | 20 | 20 | 0 |
| Test grupa 2 | TestClassType1 | 40 | 40 | 0 | 0 | 40 |
| | TestClassType2 | 40 | 0 | 40 | 40 | 0 |
| | TestClassType3 | 40 | 20 | 20 | 20 | 20 |

- B A – Broj atributa
- B P A – Broj atributa prostih tipova
- B K A – Broj atributa kompleksnih tipova
- B R A – Broj atributa referentnih tipova
- B V A – Broj atributa vrednosnih tipova

## III. OPIS TESTOVA

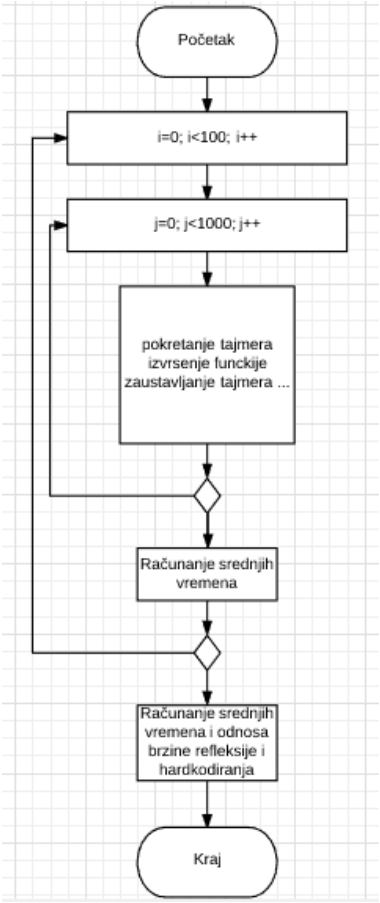
Proces testiranja sastoji se iz poziva funkcija za kloniranje za svaku klasu. Prvo je pozivana hardkodirana

Petar Mijović je student četvrte godine na smeru Računarstvo i informatika, Elektronski Fakultet Niš, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: mijovic.petar@elfak.rs

funkcija, a zatim i funkcija kloniranja iskodirana korišćenjem refleksije. Pre poziva svake funkcije startovan je tajmer koji je stopiran nakon izvršenja funkcije. Vrednosti tajmera nakon svakog izvršenja za oba tipa funkcije za svaku klasu, pamćene su u posebnim promenljivama kako bi kasnije bilo moguće izračunati prosečno vreme izvršenja.

Pozivi funkcija za kloniranje nalaze se unutar *for* petlje koja se izvršava 1000 puta, koja se nalazi unutar *for* petlje koja se izvršava 100 puta. Nakon svakog hiljaditog izvršenja funkcija računato je srednje vreme izvršenja za svaku od funkcija, a nakon svakog ciklusa gornje for petlje (ona koja se izvršavala 100 puta) računa se srednje vreme srednjeg vremena kao i odnos vremena izvršenja funkcija koje koriste refleksiju i koje imaju hardkodirane atribute. Testiranje je ponovljeno 10 puta. Dijagram toka predstavljen je na Slici 1.

Cilj velikog broja ponavljanja je dobijanje što je moguće preciznijih rezultata jer vreme izvršenja funkcija varira pri svakom ponavljanju.




Slika 1. Dijagram toka testiranja.

Predstavljene funkcije (Slika 1 i Slika 2) kloniranja koje koriste refleksiju i koje koriste hardkoiranje atributa su za klasu TestClassType1.

```
public TestClassSize1 Clone()
{
    TestClassSize1 result = new TestClassSize1();

    result.Number = this.Number;
    result.Text = this.Text;

    return result;
}
```

Slika 2. Kod funkcije za kiloniranje koja ima hardkodirane atribute.

```
public static TestClassSize1 CloneWithReflection(TestClassSize1 forCLoning)
{
    var properties = typeof(TestClassSize1).GetProperties();

    TestClassSize1 result = new TestClassSize1();
    foreach (var property in properties)
    {
        property.SetValue(result, property.GetValue(forCLoning));
    }

    return result;
}
```

Slika 3. Kod funkcije za kloniranje koja koristi refleksiju.

Bitno je napomenuti da veličina funkcije koja koristi hardkodiranje atributa direktno zavisi od broja atributa i da je veća što više atributa klasa ima.

## IV. Rezultat testiranja

TABELA 2
REZULTATI TESTIRANJA PO TEST KLASAMA

| | | TCS1 | TCS2 | TCS3 | TCS4 | TCT1 | TCT2 | TCT3 |
|---|---|---|---|---|---|---|---|---|
| 1 | tH | 0.072 | 0.004 | 0.0041 | 0.003 | 0.003 | 0.0025 | 0.0045 |
| | tR | 1.023 | 0.047 | 0.080 | 0.037 | 0.065 | 0.63 | 0.0628 |
| | O | 14.19 | 10.22 | 19.14 | 12.90 | 22.76 | 25.009 | 13.878 |
| 2 | tH | 0.077 | 0.0047 | 0.0048 | 0.0029 | 0.003 | 0.0026 | 0.0047 |
| | tR | 1.128 | 0.053 | 0.088 | 0.0399 | 0.072 | 0.07 | 0.0697 |
| | O | 14.43 | 11.225 | 18.459 | 13.601 | 23.86 | 26.67 | 14.602 |
| 3 | tH | 0.055 | 0.0033 | 0.0033 | 0.0022 | 0.0023 | 0.0022 | 0.0038 |
| | tR | 0.712 | 0.0379 | 0.0637 | 0.029 | 0.0528 | 0.053 | 0.0052 |
| | O | 12.926 | 11.45 | 19.27 | 12.739 | 22.34 | 24.38 | 13.78 |
| 4 | tH | 0.058 | 0.0037 | 0.0036 | 0.0025 | 0.0027 | 0.0021 | 0.0041 |
| | tR | 0.822 | 0.039 | 0.068 | 0.031 | 0.057 | 0.054 | 0.055 |
| | O | 14.09 | 10.599 | 18.74 | 12.53 | 20.75 | 25.6 | 13.44 |
| 5 | tH | 0.0636 | 0.0038 | 0.00367 | 0.025 | 0.0028 | 0.0022 | 0.004 |
| | tR | 0.846 | 0.0419 | 0.069 | 0.0316 | 0.0586 | 0.05575 | 0.0558 |
| | O | 13.285 | 11.016 | 19.02 | 12.43 | 20.403 | 25.21 | 13.78 |
| | t | 0.056 | 0.003 | 0.003 | 0.002 | 0.00 | 0.00 | 0.003 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 6 | H | 6 | 2 | 2 | 2 | 23 | 20 | 7 |
| | tR | 0.693 | 0.035 | 0.065 | 0.028 | 0.0529 | 0.049 | 0.052 |
| | O | 12.25 | 10.71 | 20.085 | 12.621 | 22.174 | 24.068 | 13.72 |
| 7 | tH | 0.059 | 0.0034 | 0.0034 | 0.0023 | 0.0026 | 0.0022 | 0.0038 |
| | tR | 0.761 | 0.039 | 0.062 | 0.029 | 0.054 | 0.054 | 0.051 |
| | O | 12.86 | 11.27 | 18.23 | 12.34 | 21.01 | 24.90 | 13.34 |
| 8 | tR | 0.065 | 0.0039 | 0.0037 | 0.0025 | 0.0027 | 0.0025 | 0.0043 |
| | tH | 0.917 | 0.045 | 0.076 | 0.034 | 0.0632 | 0.060 | 0.0632 |
| | O | 14.07 | 11.53 | 20.11 | 13.11 | 22.94 | 23.61 | 14.84 |
| 9 | tH | 0.057 | 0.0035 | 0.0034 | 0.0024 | 0.0024 | 0.0020 | 0.0040 |
| | tR | 0.756 | 0.039 | 0.069 | 0.028 | 0.053 | 0.055 | 0.054 |
| | O | 13.21 | 10.98 | 19.79 | 11.63 | 22.21 | 27.25 | 13.7 |
| 10 | tH | 0.064 | 0.0037 | 0.00373 | 0.0025 | 0.0027 | 0.0022 | 0.004 |
| | tR | 0.819 | 0.0397 | 0.068 | 0.0308 | 0.054 | 0.05548 | 0.0548 |
| | O | 12.85 | 10.882 | 18.256 | 12.324 | 20.368 | 24.667 | 13.56 |

Jedinica mere svih podataka osim odnosa(O) je ms.
tR – vreme izmereno funkciji koja koristi refleksiju
tH – vreme izmereno funkciji koja koristi hardkodiranje atributa
O – tR/tH
TCS1 – TestClassSize1
TCS2 – TestClassSize2
TCS3 – TestClassSize3
TCS4 – TestClassSize4
TCT1 – TestClassType1
TCT2 – TestClassType2
TCT3 – TestClassType3

Na osnovu rezultata prezentovanih u tabeli testiranja po test klasama, moguće je doći do sledećih srednjih vrednosti:

TABELA 3
SREDNJI REZULTATI TESTIRANJA PO TEST KLASAMA

| | TCS1 | TCS2 | TCS3 | TCS4 | TCT1 | TCT2 | TCT3 |
|---|---|---|---|---|---|---|---|
| tH | 0.063 | 0.0037 | 0.0037 | 0.0048 | 0.0027 | 0.0023 | 0.0041 |
| tR | 0.848 | 0.042 | 0.0709 | 0.0318 | 0.0582 | 0.1137 | 0.0523 |
| O | 13.42 | 10.99 | 19.11 | 12.62 | 21.89 | 25.14 | 13.86 |

Ovi rezultati pokazuju da je refleksija uvek sporija i to od 10-25 puta u odnosu na hardkorianje atributa kod funkcija kloniranja.

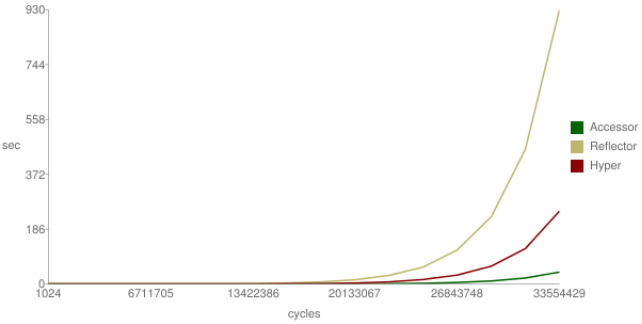
Konkretni zaključci do koje se došlo su sledeći:

1. Refleksija je u odnosu na hardkodiranje atributa kod funkcija kloniranja vremenski najbliža kod raznovrsnih klasa (klasa sa atributima vredonosnog i referentnog tipa) što pokazuju rezultati za TestClassSize2 i TestClassType3 koje su u svojim grupama najraznovrsnije. Raznovrsna klasa je u stvarnom životu i najrealnija klasa.
2. Hardkodiranje atributa je brže kada su atributi referentnog tipa nego kada su vredonosni. Kombinacija ove dve vrste tipova usporava rad hardkodirane funkcije sto se vidi u redu tH za klase TCT1, TCT2, TCT3.
3. Obe vrste funkcija su najsporije radile za najmanju klasu. (Vrednosti u koloni TCS1 su najveće).

## V. ZAKLJUČAK

Refleksija je moćan alat koji ima svoju cenu korišćenja. Iako se iz priloženog vidi da je vreme izvršenja funkcija koje koriste refleksiju 10-25 puta duže od klasičnih funkcija i dalje pričamo o vremenima izvršenja koje se i dalje mere u milisekundama.

Uz adekvatno korišćenje može se koristiti bez vidljive razlike u performansama. Ozbiljne razlike je moguće uočiti ako je kompleksnost funkcija velika (>O($n^2$)) to jest ako postoji mnogo ugnježdavanja. Grafik [3] koji pokazuje ovu zavisnost nalazi se na slici 4.




Slika 4. Performanse refleksije zavisno od broja ciklusa.

## LITERATURA

[1] Reflection [OnLine]
Dostupno na:https://msdn.microsoft.com/en-us/library/f7ykdhsy(v=vs.71).aspx
[2] Reflection in .NET Framework [OnLine]
Dostupno na: https://docs.microsoft.com/en-us/dotnet/framework/reflection-and-codedom/reflection
[3] How costly is .NET reflection [OnLine]
Dostupno na: https://stackoverflow.com/questions/25458/how-costly-is-net-reflection

# Sistem za kontrolu garaže

Marko Nikolić

*Sadržaj* – U ovom radu biće opisan sistem za kontrolu garaže, realizovan korišćenjem savremenih hardverskih i softverskih komponenti i tehnologija, kao što su Android razvojno okruženje i Arduino razvojna platforma. Cilj projekta je pružanje mogućnosti korisnicima da uz pomoć svojih pametnih telefona lako upravljaju svojom garažom.

## I. UVOD

Razvojem modernih tehnologija neminovne su promene u svakoj sferi društva. Integracija modernih tehnologija u svakodnevnom životu trebalo bi da olakša obavljanje svakodnevnih aktivnosti. Danas su svima dostupni pametni telefoni [1]. Najčešće korišćena mrežna tehnologija u privatnim kućama je Wi-Fi. Osim pametnih telefona, na Wi-Fi mrežu mogu se povezati i drugi uređaji kao što su laptopovi, kućni računari ali i kućni aparati koji u sebi imaju ugrađen neki vid mikroračunara. Povezivanje je bežično što omogućava da se uređaji mogu postavljati bilo gde u kući i oko kuće. To je omogućilo razvoj takozvanih „Pametnih kuća“ [2] koje u najširem smislu podrazumevaju integraciju raznih, pre svega električnih, uređaja u domaćinstvu u sistem kojim se može upravljati preko Wi-Fi mreže, korišćenjem pametnih uređaja. U ovom radu biće opisan deo sistema „Pametne kuće“ koji se odnosi na „Pametnu garažu“. Korišćene su dobro poznate tehnologije, kao što su Arduino [3] razvojna platforma i savremeni programski jezici Java i Android [4] razvojnom okruženju i dijalekt programskog jezika C za mikrokontrolere u razvojnom okruženju Arduino IDE. Svaka od komponenata sistema biće detaljnije objašnjena u nastavku ovog rada.

Drugo poglavlje ovog rada odnosi se na opis problema koji je rešavan u ovom projektu. Nakon trećeg poglavlja koje opisuje Arduino razvojnu platformu, u četvrtom je opisan Wi-Fi modul ESP8266. U petom poglavlju opisano je na koji su način u sistemu iskorišćeni senzori i aktuatori, dok je u šestom opisana električna šema, kreirana korišćenjem mikrokontrolera i ostalih hardverskih komponenti.

Poglavlja sedam, osam i devet odnose se na softverski deo sistema i opisuju program koji se izvršava na mikrokontroleru i Wi-Fi modulu, kao i android aplikaciju koja je u direktnom kontaktu sa korisnikom.

Na kraju, u zaključku, dat je kratak osvrt na celokupni sistem i opisane su glavne prednosti i nedostaci kod ovog rešenja.

Marko Nikolić je student na smeru Računarstvo i informatika, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: nikolicmarko93@gmail.com

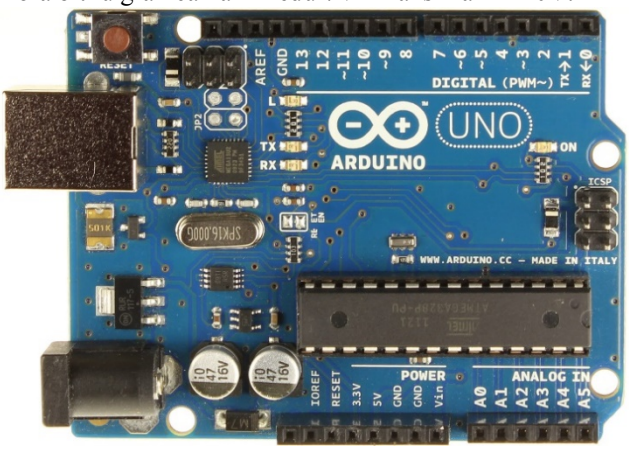
## II. OPIS PROBLEMA

Projektovani sistem pruža mogućnost korisnicima da lako upravljaju svojom garažom bez izlaska iz svog automobila, ali i sprečavaju opasne situacije poput visoke koncentracije gasa u vazduhu ili provale u garažu. Takođe, tu je i svetlo izvan garaže koje se pali automatski ukoliko je detektovan pokret.

Sistem se sastoji iz tri komponente: Arduino sistem koji meri vrednosti očitane senzorom i kontroliše aktuatore, Android mobilne aplikacije koja šalje zahteve za promenu ili proveru stanja elemenata u garaži i Wi-Fi modul ESP8266 konfigurisan kao „access point“ koji služi kao veza između prethodne dve komponente. Kreirana je aplikacija koja predstavlja interfejs ka sistemu i poseduje intuitivan i jednostavan dizajn. Između Wi-Fi modula i mikrokontrolera uspostavljena je serijska komunikacija. Kao razvojna platforma korišćen je Arduino Uno, zbog svoje jednostavnosti, ali istovremeno dovoljno velike procesorske moći, kao i dobre prateće dokumentacije.

## III. ARDUINO RAZVOJNO OKRUŽENJE

Arduino predstavlja razvojni sistem otvorenog koda zasnovan na 8-bitnom Atmel AVR ili 32-bitnom Atmel ARM mikroprocesoru. Arduino Uno zasniva se na Atmelovom ATmega328 mikrokontroleru. Poseduje 14 digitalnih ulazno-izlaznih pinova, 6 analognih pinova i 8 specijalnih pinova, od kojih možemo izdvojiti naponske regulatore vrednosti 5 V, 3.3 V i GND. Takođe poseduje i USB port, AC/DC *jack*, keramički rezonator na 16 MHz i reset dugme. Osim eksternog napajanja može se napajati i preko USB porta. Napajanje neophodno za funkcionisanje mora biti u granicama između 7V i maksimalnih 20V.




Slika 1. Izgled Arduino Uno okruženja.

Softversko razvojno okruženje Arduino IDE je potpuno besplatno, otvorenog je koda i može se preuzeti za Mac OS X, Windows i Linux operativni sistem.

## IV. Wi-Fi modul ESP8266

Wi-Fi modul ESP8266 [5], prikazan na slici 2, je samostalan SOC (sistem na čipu) sa integrisanim TCP/IP protokolom koji omogućava pristup bilo kom mikrokontroleru na Wi-Fi mrežu. ESP8266 može ili da čuva aplikaciju ili da povuče sve Wi-Fi mrežne funkcije sa drugog procesora. Svaki ESP8266 modul dolazi pre-programiran sa AT skupom firmware komandi, što znači da se može jednostavno spojiti Arduino uređaj i dobiti onoliko Wi-Fi mogućnosti koliko Wi-Fi šild nudi. ESP8266 modul je povoljna ploča sa velikim i stalno rastućim brojem korisnika. Ovaj modul ima moćne mogućnosti obrade i skladištenja podataka na ploči, koje mu dozvoljavaju da se integriše sa senzorima i drugim specifičnim uređajima preko njegovih GPIO pinova sa minimalnim razvojem *up-front* i minimalnim unosom tokom priključenja. Visok stepen integracije na čipu sa minimalnim dodatnim kolima, uključujući *front-end* module, omogućava ovom modulu da zauzme minimalno PCB područje. ESP8266 podržava APSD za VoIP aplikacije i Bluetooth interfejse, sadrži samopodešavajući RF dozvoljavajući mu da radi pod svim radnim uslovima i ne zahteva dodatne RF delove. Postoji skoro neograničena biblioteka informacija raspoloživih za ESP8266, koje su dobijene neverovatnom podrškom zajednice.



Slika 2. Wi-Fi modul ESP8266.

## V. Senzori i aktuatori

Arduino sistem meri vrednosti očitane senzorom i kontroliše aktuatore. Senzori korišćeni u sistemu su: senzor za detekciju pokreta, gasni senzor, senzor svetlosti i senzor Holovog efekta. Aktuatori su zujalica i 4 LED diode. U daljem tekstu biće objašnjeno na koji način su oni iskorišćeni.

### *A. Alarmni sistem*

Alarmni sistem može detektovati provalu kao i visoku koncetraciju gasa u garaži. Koristi se zujalica, senzor Holovog efekta i gasni senzor. Ako je alarm uključen, svaka povećana vrednost koncentracije gasa i svaki prekid senzora Holovog efekta će aktivirati zujalicu.

### *B. Garažna vrata*

Garažna vrata mogu se otvarati i zatvarati mobilnom aplikacijom. Vrata su simulirana pomoću dve LED diode. Kada su otvorena svetli zelena LED dioda, a kada su zatvorena svetli crvena LED dioda.

### *C. Svetlo unutar garaže*

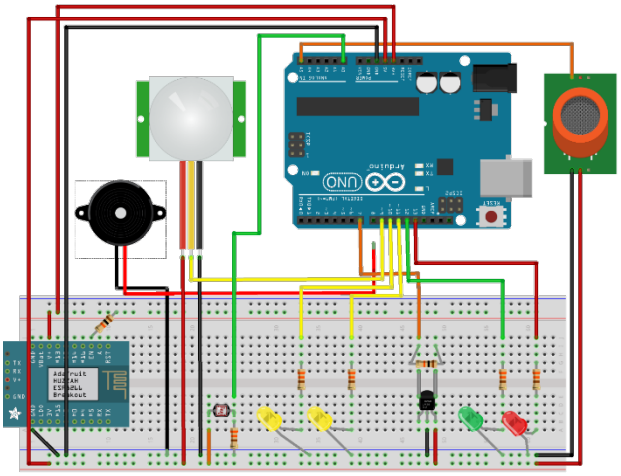
Svetlo unutar garaže može se uključivati i isključivati od strane korisnika korišćenjem mobilne aplikacije. Korisnik pored toga može menjati intenzitet svetlosti. Za unutrašnje svetlo iskorišćena je LED dioda.

### *D. Svetlo izvan garaže*

Kada je neki pokret detektovan u neposrednoj blizini garaže u toku noći, sistem automatski pali svetlo izvan garaže. Ovaj sistem se sastoji od senzora za detekciju pokreta, senzora svetlosti i LED diode. Dok senzor za detekciju pokreta registruje pokret, senzor svetlosti proverava da li je dovoljno mračno napolju da bi se upalilo svetlo. Ako su oba uslova ispunjena, svetlo će biti uključeno nekoliko sekundi nakon zadnjeg pokreta.

## VI. Električna šema

Na slici 3 predstavljena je električna šema veza između Arduina, Wi-Fi modula i senzora i aktuatora.



Slika 3. Električna šema.

Osnovu sistema predstavlja mikrokontroler Arduino Uno, na čije pinove su povezane dodatne komponente. Za komunikaciju mikrokontrolera sa Android pametnim telefonima iskorišćen je Wi-Fi modul ESP8266. Komunikacija između Arduina i ESP8266 modula se bazira na serijskoj komunikaciji. RX modula je konektovan na TX

Arduina i obrnuto. Zahtevi su definisani kao stringovi. Svaki zahtev korisnika se prenosi sa Wi-Fi modula na Arduino, zatim Arduino postavlja aktuatore u odgovarajuće stanje i prosleđuje odgovor nazad. Wi-Fi modul zatim šalje obaveštenje korisniku.

## VII. ARDUINO PROGRAM

Program koji se izvršava na mikrokontroleru, napisan je na mikro C-u, dijalektu programskog jezika C, korišćenjem razvojnog okruženja Arduino IDE [6] [7]. U nastavku, biće objašnjene neke od glavnih funkcija.

Funkcija „forceLight“ koja se nalazi na slici 4, služi za uključivanje svetla unutar garaže. Ako je zahtev drugačiji od trenutnog stanja svetla, promena će biti izvršena. U suprotnom, šalje se obaveštenje korisniku da je stanje isto kao i zahtev, tako da ostaje nepromenjeno.

```
void forceLight(bool lightState){
  if(lightState!=lightStateMem){
    if(lightState){
      digitalWrite(ledLightIn, HIGH);
      transmitter = "lightTurnedOn";
    }else{
      digitalWrite(ledLightIn, LOW);
      transmitter = "lightTurnedOff";
    }
  }else{
    transmitter = "stateSame";
  }
  lightStateMem=lightState;
}
```

Slika 4. Funkcija forceLight.

Na slici 5 prikazana je funkcija „moveDoors“ koja služi kao simulacija otvaranja i zatvaranja garažnih vrata, tako što se uključuju zelena ili crvena LED dioda. Ako je zahtev drugačiji od trenutnog položaja vrata, promena će biti izvršena: ako se vrata otvaraju uključiće se zelena dioda a crvena ugasiti i obrnuto. U suprotnom, šalje se obaveštenje korisniku da je stanje isto kao i zahtev, tako da ostaje nepromenjeno.

```
void moveDoors(bool doorState){
  if(doorState!=doorsStateMem){
    if(doorState){
      digitalWrite(ledDoorsClosed, LOW);
      digitalWrite(ledDoorsOpened, HIGH);
      transmitter="doorOpened";
    }else{
      digitalWrite(ledDoorsClosed, HIGH);
      digitalWrite(ledDoorsOpened, LOW);
      transmitter="doorClosed";
    }
  }else{
    transmitter="stateSame";
  }
  doorsStateMem=doorState;
}
```

Slika 5. Funkcija moveDoors.

Funkcija „setAlarm“ postavlja alarm na uključeno ili isključeno i prikazana je na slici 6. Ako je zahtev drugačiji od trenutnog stanja alarma, izvršiće se promena. Sa isključivanjem alarma, takođe se isključuje zujalica ako je u tom trenutku aktivna. Ako je zahtev isti kao trenutno stanje, šalje se obaveštenje korisniku i ono ostaje nepromenjeno.

```
void setAlarm(bool alarmState){
  if(alarmState!=alarmStateMem){
    if(alarmState){
      transmitter = "AlarmTurnedOn";
    }else{
      transmitter = "AlarmTurnedOff";
      stopAlarm();
    }
  }else{
    transmitter = "stateSame";
  }
  alarmStateMem=alarmState;
}
```

Slika 6. Funkcija setAlarm.

Na slici 7 je prikazana funkcija „checkGasLevel“ koja očitava vrednost koncentracije gasa u vazduhu. Ako koncentracija gasa prelazi gornju granicu, alarm će se oglasiti.

```
bool checkGasLevel(){
  gasLevel = analogRead(gasSensor);
  sensorVolt=gasLevel*(5.0/1023.0);
  gasValue = ((5.0*150.0)*sensorVolt)-10.0;

  if(gasValue>2000){makeSomeNoise();}
  Serial.println(gasValue);
}
```

Slika 7. Funkcija checkGasLevel.

Funkcija „checkLightLevel“, koja je prikazana na slici 8, služi za uključivanje svetla izvan garaže. Svetlo će biti uključeno ukoliko su ispunjena dva uslova: pokret je registrovan od strane senzora za detekciju pokreta i jačina svetlosti napolju koja se očitava sa senzora svetlosti je dovoljno niska. Ukoliko je detektovan pokret ali je jačina svetlosti veća, svetlo će se isključiti. Ako senzor za detekciju pokreta ne registruje nikakav pokret, svetlo se neće uključiti.

```
void checkLightLevel(){
  if (analogRead(lightSensor)<lightLimit
          and digitalRead(PIR)==HIGH){
    digitalWrite(ledLightOut, HIGH);
  }else if(analogRead(lightSensor)>lightLimit+30
          and digitalRead(PIR)==HIGH){
    digitalWrite(ledLightOut,LOW);
  }else{
    digitalWrite(ledLightOut,LOW);
  }
}
```

Slika 8. Funkcija checkLightLevel.

Funkcija „checkBurglary“ služi za otkrivanje pokušaja pljačke ukoliko je postavljen alarm (slika 9). Kada je postavljen alarm i senzor Holovog efekta pokazuje vrednosti veće od dozvoljenih, alarm će se aktivirati.

```
bool checkBurglary(bool alarmMem){
  if(alarmMem){
    if(digitalRead(halleffect)){
      makeSomeNoise();
    }
  }
}
```

Slika 9. Funkcija checkBurlgary.

Na slici 10 je prikazana funkcija „changeLightValue“ koristi se za podešavanje intenziteta svetlosti. Intenzitet svetlosti je između 0 i 255. Isključeno je kada je intenzitet na 0.

```
void changeLightValue(String lightLimit){
  int lightVal=255*lightLimit.toInt()/100;
  analogWrite(ledLightIn, lightVal);
  if (lightVal==0){
    lightStateMem=false;
  }
  transmitter=lightLimit;
}
```

Slika 10. Funkcija changeLightValue.

## VIII. PROGRAM ZA WI-FI MODUL

Komunikacija preko Wi-Fi modula se odvija uz pomoć dve funkcije. U funkciji „handleRequest“ svaki zahtev korisnika se prenosi sa Wi-Fi modula na Arduino.

```
void handleRequest(WiFiClient client){
  request = client.readString();
  client.flush();
  if (request == "lightON"){Serial.print("turnLightOn");}
  else if(request == "lightOFF"){Serial.print("turnLightOff");}
  else if(request == "doorCLOSE"){Serial.print("closeDoors");}
  else if(request == "doorOPEN"){Serial.print("openDoors");}
  else if(request == "AlarmON"){Serial.print("setAlarmOn");}
  else if(request == "AlarmOFF"){Serial.print("setAlarmOff");}
  else if(request == "GAS"){Serial.print("checkGas");}
  else if(request == "doorSTATE"){Serial.print("checkDOOR");}
  else if(request == "lightSTATE"){Serial.print("checkLIGHT");}
  else if(request == "alarmSTATE"){Serial.print("checkALARM");}
  else if(isNumber(request)){Serial.print(request);}
  Serial.print("*");
}
```

Slika 11. Funkcija handleRequest.

Nakon što Arduino izvrši odgovarajuću komandu, on šalje odgovor Wi-Fi modulu o izvršenju i u funkciji „handleAnswer“ ako se taj odgovor poklapa sa nekom od komandi, korisniku šalje obaveštenje o promeni stanja. Ukoliko se ne poklapa, šalje se izveštaj o grešci.

```
void handleAnswer(WiFiClient client){
  while (!Serial.available()){delay(1);}
  transmitter=Serial.readString();

  if (transmitter == "lightTurnedOn"){client.print("lightON");}
  else if(transmitter == "lightTurnedOff"){client.print("lightOFF");}
  else if(transmitter == "doorClosed"){client.print("doorCLOSED");}
  else if(transmitter == "doorOpened"){client.print("doorOPENED");}
  else if(transmitter == "AlarmTurnedOn"){client.print("AlarmON");}
  else if(transmitter == "AlarmTurnedOff"){client.print("AlarmOFF");}
  else if(transmitter == "stateSame"){client.print("stateSAME");}
  else if(isNumber(transmitter)){client.print(transmitter);}
  else{client.print("error");}

  transmitter="";
}
```

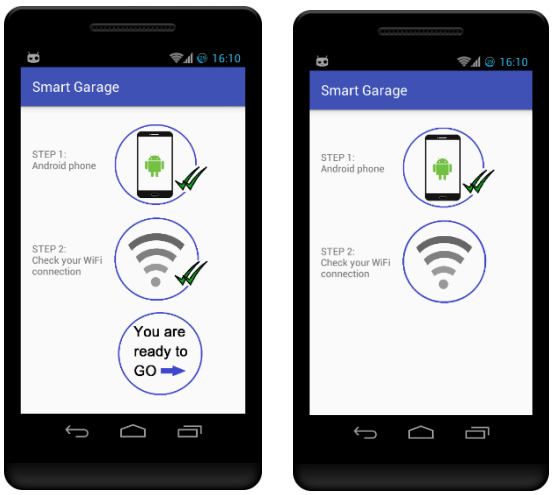
Slika 12. Funkcija handleAnswer.

## IX. ANDROID APLIKACIJA

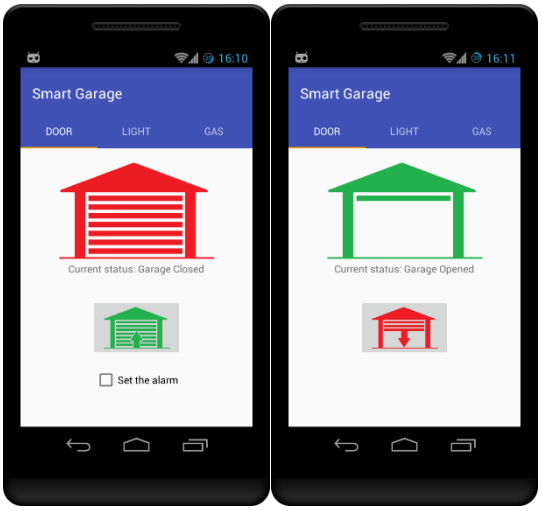
Za implementaciju mobilne aplikacije korišćene su dve aktivnosti (eng. *activity*), tri fragmenta i četiri pomoćne klase. Prva aktivnost je “ConnectionActivity” za konekciju sa Arduinom preko Wi-Fi modula. Druga aktivnost je aktivnost koja koristi tabove i koja u aplikaciji služi kao glavna stranica. Ona se sadrži iz tri fragmenta koji su tri dela aplikacije (vrata, svetlo i gas). Za sve dugmiće u aplikaciji dodati su osluškivači (eng. *listener*) koji izvršavaju razne zadatke. Sva komunikacija, slanje zahteva i dobijanje odgovora od Arduina odvija se u pozadinskim nitima koristeći AsyncTask klase. Na taj način, glavna UI nit nikad nije stopirana. Za čitanje i smeštanje informacije o stanju aplikacije koristi se „PreferencesHelper“ klasa koja koristi SharedPreferences interfejs. U nastavku, biće prikazani i objašnjeni delovi aplikacije.

Na slici 13 je prikazan izgled dela aplikacije koji služi za povezivanje sa Arduinom. Kao što je ranije pomenuto, komunikacija ide preko Wi-Fi modula ESP8266. Prvi korak je već čekiran jer korisnik poseduje Android telefon. Drugi korak je provera da li je korisnik u istoj mreži kao i Arduino da bi komunicirao sa njim. Klikom na krug sa Wi-Fi simbolom, korisnik biva obavešten ako je u drugoj Wi-Fi mreži ili nije konektovan uopšte na Wi-Fi. Ako to nije slučaj, to znači da je sigurno u istoj mreži sa Arduinom i korisniku će biti prikazan još jedan krug koji kaže da je spreman za komunikaciju sa Arduinom.

Na slici 14 je prikazan izgled prvog dela glavne stranice aplikacije (Door). Ovo je prvi deo glavne stranice aplikacije koji se vidi nakon uspešne konekcije sa Arduinom. Jedan od dva statusa u ovom tabu je “Garage Closed” što znači da su vrata zatvorena. U ovom slučaju postoje dve mogućnosti: korisnik može jednostavno otvoriti vrata, dok je druga mogućnost postavljanje alarma. Postavljanje alarma će verovatno sprečiti provalnika da ukrade nešto nakon što su vrata od garaže otvorena. Drugi status je “Garage Opened” i u ovom slučaju postoji samo jedna mogućnost, a to je zatvaranje vrata.




Slika 13. Povezivanje sa Arduinom.




Slika 14. Glavna stranica – Door.

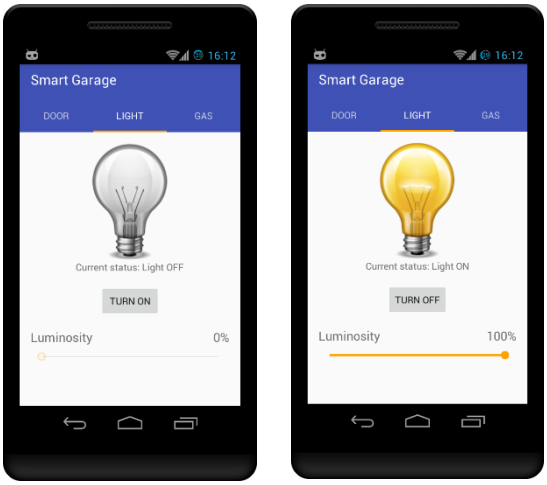
Na slici 15 je prikazan izgled drugog dela glavne stranice aplikacije (*Light*). U tom delu takođe postoje dva statusa. Jedan od njih je slučaj kada je svetlo već isključeno. Jedina opcija u ovom slučaju je da se uključi svetlo. Kada je svetlo uključeno, postoje dve mogućnosti koje korisnik može da odabere. Pored isključivanja svetla, korisnik može i da menja intenzitet svetla unutar garaže koristeći slajder, a rezultat će automatski biti prikazan po otpuštanju istog.

Na slici 16 je prikazan izgled trećeg dela glavne stranice aplikacije (*Gas*). U tom delu korisnik može videti koncentraciju gasa klikom na dugme za sinhronizaciju. Ako gas prelazi granične vrednosti, korisnik će biti obavešten.

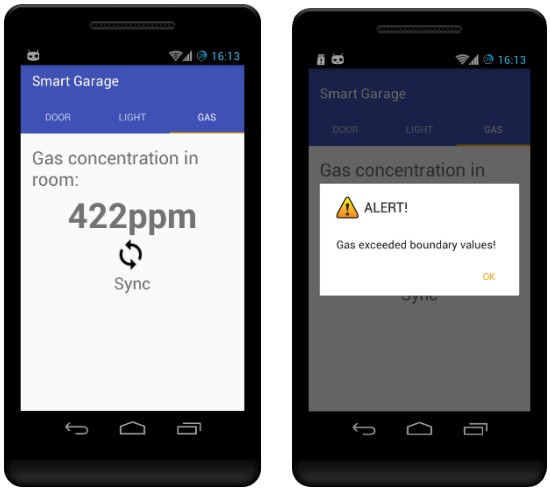
## X. ZAKLJUČAK

Sistemi kućne automatizacije postaju sve popularniji. Sve se češće grade kuće i zgrade koje već unapred imaju projektovanu mrežu senzora i za koje se planira da postanu pametne građevine. Danas su elektronske komponente dostupne svuda i svima po jeftinim cenama. Takođe, svima su dostupni pametni telefoni koji omogućavaju povezivanje preko Wi-Fi mreže sa tim komponentama. Sve ovo je dovelo do toga da je izrada takvog sistema moguća bez mnogo muke i resursa.

Ovaj projekat predstavlja praktičnu implementaciju dela takvog sistema koji pruža mogućnost korisnicima da uz pomoć svojih pametnih uređaja lako upravljaju svojom garažom. Glavna prednost ovog rešenja je skalabilnost. Pored toga, omogućeno je simultano korišćenje sistema od strane više različitih korisnika ali je sa druge strane rešen problem simultanog pristupa istom resursu. Kod ovog rešenja nedostatak je komunikacija sa korisnicima van lokalne mreže. Komunikacija bazirana na Wi-Fi nije dovoljna za alarmni sistem. Ali postoje i druga rešenja koja prevazilaze i taj problem i koja bi bila urađena u sledećoj verziji ovog projekta.




Slika 15. Glavna stranica – Light.




Slika 16. Glavna stranica – Gas.

## LITERATURA


[1] https://sr.wikipedia.org/sr-el/Pametni_telefon, Pametni telefon, 2017

[2] https://en.wikipedia.org/wiki/Home_automation, Home automation, 2017

[3] https://sr.wikipedia.org/wiki/Arduino, Arduino, 2017

[4] https://sr.wikipedia.org/sr-el/Android (operativni_sistem), Android (operativni sistem), 2017

[5 https://en.wikipedia.org/wiki/ESP8266, ESP8266, 2017

[6] https://www.arduino.cc/en/Reference/HomePage, zvanična Arduino dokumentacija, 2017

[7] John Baichtal, Arduino for Beginners: Essential Skills Every Maker Needs, Amazon, 2013

[8] https://www.arduino.cc/en/Guide/ArduinoUno, Getting Started with Arduino and Genuino UNO, 2017

[9] https://www.arduino.cc/en/Main/ArduinoBoardUno, zvanična Arduino specifikacija, 2017

[10] https://www.sparkfun.com/products/13678, WiFi Module - ESP8266, 2017

[11] https://learn.sparkfun.com/tutorials/esp8266-thing-hookup-guide/example-sketch-ap-web-server, ESP8266 Thing Hookup Guide, 2017

[12] https://developer.android.com/index.html, Android, 2017

# Digitalni termometar realizovan na principu analogno-digitalne konverzije

Nikola Mitrović

*Sadržaj* – U ovom radu će ukratko biti opisan proces A/D konverzije i osnovni A/D konvertori, a onda i komponente koje će biti korišćene u realizaciji. Zatim će uređaj biti praktično realizovan i u laboratoriji će se proveriti njegove karakteristike lineranost, preciznost i slično.

## I. UVOD

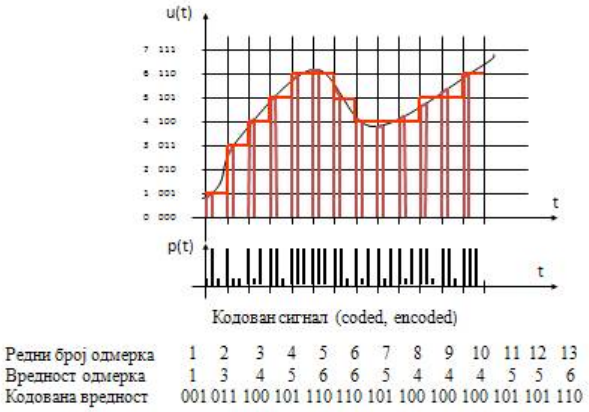
Analogno-digitalna konverzija predstavlja generisanje digitalnog kodovanog broja koji odgovara analognom ulaznom signalu. Elektronsko kolo koje vrši konverziju naziva se analogno-digitalni konvertor, ili skraćeno A/D konvertor (ADC).

Kod savremenih tehnika merenja, dominantnu grupu instrumenata i merne opreme, čine digitalni merni instrumenti. Oni često ne samo da poseduju naprednije performanse u odnosu na odgovarajuće analogne instrumente (brzina, propusni opseg, pouzdanost, otpornost na smetnje, minijaturizacija, itd.), već imaju i znatno nižu cenu, a omogućavaju i veoma jednostavnu automatizaciju procesa merenja.

## II. PROCES A/D KONVERZIJE

Sam proces analogno-digitalne konverzije može se podeliti na tri osnovne faze. To su diskretizacija, kvantizacija i kodiranje [1].

Diskretizacija analognog signala je proces gde se od analognog signala dobija diskretizovan signal. Proces počinje uzorkovanjem (sampling). Uzorkovanje je proces prilikom koga se, u ekvidistantnim vremenskim trenucima „uzima uzorak signala„ odnosno, očitava se njegova vrednost. Na osnovu očitanih vrednosti, formira se diskretan izlazni signal, koji ima očitanu vrednost u datim ekvidistantnim trenucima (slika 1). Dakle, rezultat ovog procesa je povorka impulsa promenljive veličine, koji odgovaraju ulaznom signalu izmerenom u trenucima odmeravanja. Ovi impulsi nazivaju se odmerci (samples). Dodeljivanje vrednosti ovim odmercima naziva se kvantizacija. Vrednost odmeraka jednaka je ulaznom signalu i ima bilo koju vrednost iz ulaznog intervala. Bilo koju znači da, teorijski, postoji beskonačno mnogo različitih vrednosti. Pošto digitalni merni instrumenti i računari skladište i obrađuju brojeve u binarnom obliku, pri pretvaranju je uvek unapred određen broj bitova unutar kojih se vrednost mora nalaziti (dužina binarne reči). Taj broj bitova određuje ukupan broj mogućih različitih kombinacija (konačan broj) i svakoj kombinaciji odgovara jedna vrednost analognog signala. Zato je ulazni interval podeljen na manje delove kojih ima tačno onoliko koliko ima različitih binarnih reči. Veličina ovih delova naziva se rezolucija analogno digitalnog pretvaranja ili korak kvantizacije.




Slika 1. Postupak A/D konverzije.

Prilikom kvantizacije odmerci su dobili samo „dozvoljene“ vrednosti iz skupa diskretnih veličina, ali oni i dalje predstavljaju napon (u V ili mV) koji se ne može digitalno obrađivati. Kodiranje je proces u kojem se svakoj diskretnoj vrednosti napona dodeljuje jedna i samo jedna binarna reč.

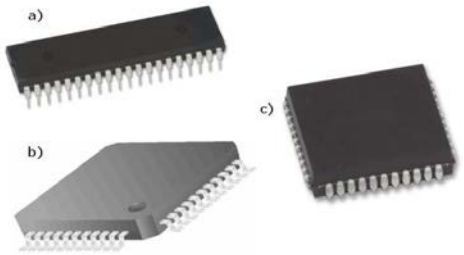
## III. KOMPONENTE ZA REALIZACIJU

### *A. PIC16F877a*

PIC mikrokontroler 16F877A [2] je jedan od najprisutnijih mikrokontrolera u modernoj industriji. Vrlo je jednostavan za korišćenje, pa postoje odgovarajuća softverska okruženja sa kojima je relativno lako kodirati i programirati ga. Jedna od glavnih prednosti ovog mikrokontrolera je ta što je on “piši-briši” mikrokontroler, jer koristi tehnologiju fleš memorije. Na taj način, moguće je koristiti ga više puta, svaki put sa različitom funkcijom. Ima ukupno 40 pinova, od kojih su 33 ulazno-izlazni. Poseban akcenat bi trebalo staviti na port A, jer je to jedini

Nikola Mitrović je student četvrte godine na Katedri za mikroelektroniku na Elektronskom fakultetu u Nišu, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: nikola.mitrovic.nis@gmail.com

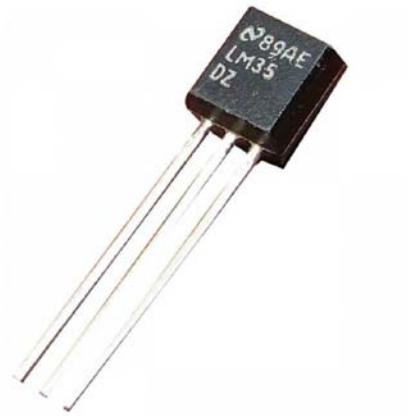
port koji može da prepozna analogni ulazni signal. Ulaz porta A zapravo predstavlja desetobitni A/D konverotor koji ulaze može poslati na neki od drugih portova.




Slika 2. Pakovanja mikrokontrolera.

*B. Senzor temperature LM35*

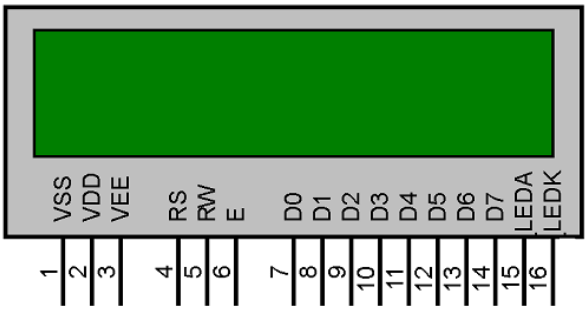
LM35 je temperaturni senzor visoke preciznosti čiji je izlazni napon linearno proporcionalan Celzijusovoj temperaturnoj skali [3]. Senzor ne zahteva nikakvu posebnu spoljašnju kalibraciju, i pri sobnoj temperaturi ostvaruje preciznost od ± ¼ ˚C , a ukupnu preciznost od ± ¾ ˚C na nivou pune skale. Mala izlazna impedansa, linearni izlaz i precizna fabrička kalibracija čine ovaj senzor izuzetno jednostavnim za korišćenje. Zbog toga, ovaj senzor se uglavnom sa lakoćom može implementirati u veliki broj kola. Senzoru je potrebno eksterno napajanje. Prilikom korišćenja, LM35 izvlači samo 60 μA struje iz napajanja, pa to uzrokuje i da se senzor jako slabo greje, i pri temperaturama oko sobne se greje samo do 0.1˚C.




Slika 3. Senzor temperature LM35.

*C. 16x2 LCD Display*

LCD (*Liquid Crystal Display*) [4] ekran je elektronska komponenta koja nalazi jako širok spektar primena. 16x2 displej je jedan od najjednostavnijih modula i veoma je često korišćen u uređajima. *B,* ...). To što je displej 16x2 znači da se na njemu mogu prikazivati po 16 karaktera u 2 reda. Svaki karakter na displeju prikazuje se kao matrica od 5x7 piksela. Takođe, displej ima dva registra, komandni registar i registar podataka.

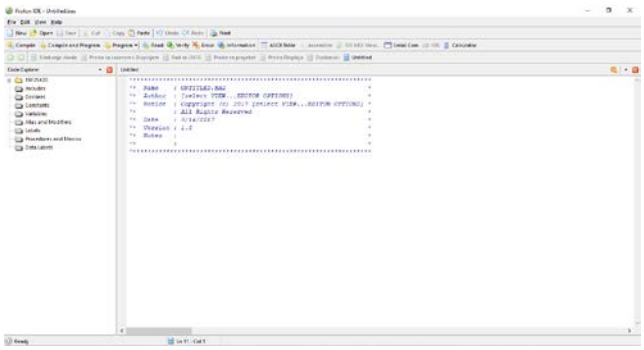



Slika 4. Pinout displeja.

## IV. SIMULACIJA I PRAKTIČNA REALIZACIJA

*A. Pisanje koda*

Proton IDE [5] omogućava da se korišćenjem programskog jezika BASIC napišu instrukcije za mikrokontroler. Proton IDE će onda to prevesti na jezik kontrolera, i formiraće .hex fajl koji će se učitavati u mikrokontroler korišćenjem PIC Kit-a. Sada će biti objašnjene celine sintakse kojima je programiran mikrokontroler.



Slika 5. Okruženje Proton IDE.

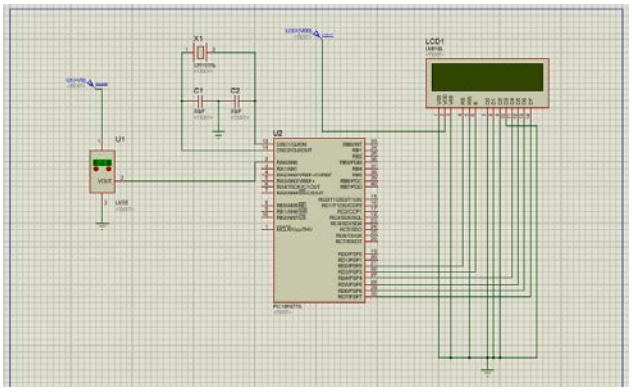
*B. Simulacija softverskim alatom Proteus 8*

U simulatoru treba napraviti šemu, i onda je celu simulirati. Za taj zadatak iskorišćen je softverski alat Proteus 8 [6]. Šema treba da bude verna onome što je namera da se praktično realizuje, s tim što će napajanje biti fiktivno.

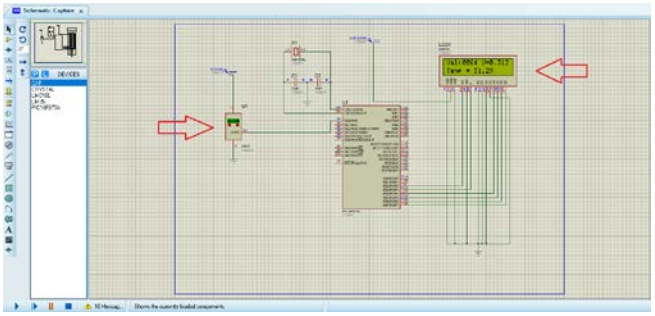
Kada se pokrene simulator, potrebno je konstatovati da sve radi onako kako je planirano.

Leva strelica ukazuje na to koja vrednost temperature je u simulaciji dovedena do senzora. Desna strelica ukazuje na displej, odnosno na vrednosti koje se iščitavaju. Provereno je da se vrednosti poklapaju. Dakle, podsetimo se, senzor je kalibrisan tako da se broj stepena dobija kada se broj milivolti pomnoži sa 100. U simulaciji, senzor je savršen (prema datasheetu), nema gubitaka ni kod senzora, ni kod mikrokontrolera, pa su i vrednosti idealne.

Sada će se pristupiti praktičnoj realizaciji ovog sistema, i videti koliko se vrednosti koje treba da dobijemo (i dobijemo u simulacijama) razlikuju od vrednosti koje će biti dobijene u praksi.



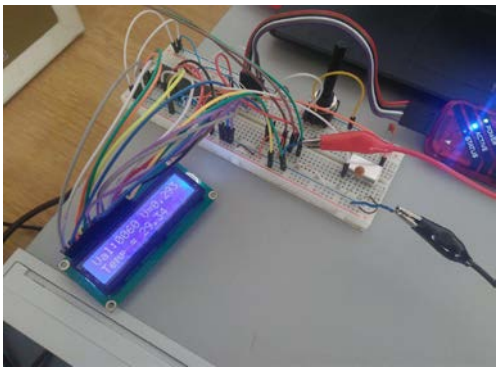
Slika 6. Realizacija šeme alatom Proteus 8.



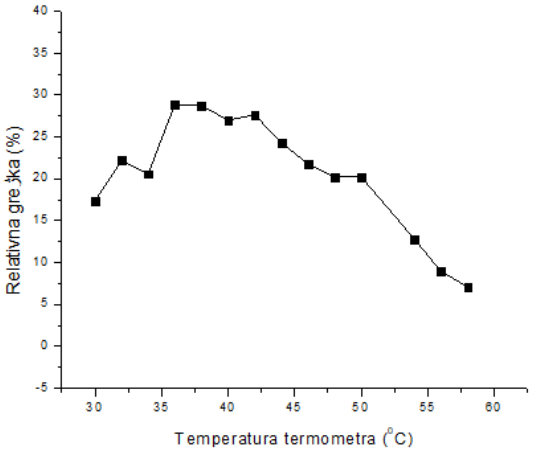
Slika 7. Pokrenuta simulacija.

*C. Praktična realizacija*

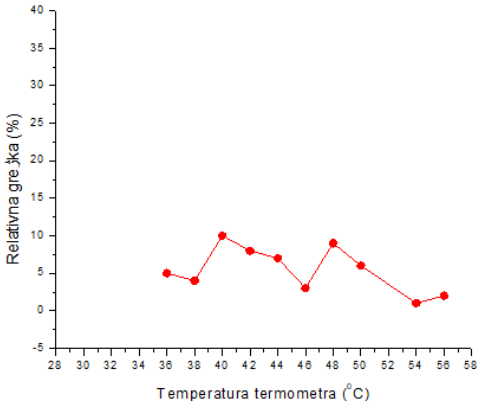
Nakon praktične realizacije kola, potrebno je ispitati i praktično njegove karakteristike, kako bi se proverilo da li kolo radi ispravno. Za testiranje će biti korišćeni eksterni termometar, eksterni voltmetar i pećnica u kojoj će se zagrevati sensor. Prvo će se posmatrati ponašanje senzora prilikom zagrevanja od 30 ˚C do 60 ˚C, a onda i prilikom hlađenja na istom temperaturnom opsegu. U nastavku će biti dati grafici merenja relativne greške.



Slika 8. Izgled praktično realizovanog kola.




Slika 9. Relativna greška pri zagrevanju.




Slika 10. Relativna greška pri hlađenju.

Sa ovih grafika možemo primetiti da je relativna greška merenja koju pokazuje senzor dosta manja prilikom hlađenja, nego prilikom zagrevanja. Glavni razlog ovakvog ponašanja uglavnom se krije u brzini rada. Prilikom zagrevanja, grejači pećnice su bili uključeni i temperature je relativno brzo rasla. Prilikom hlađenja, nije bilo pospešivanja opadanja temperature, pa je proces tekao sporije. Zbog toga, senzor je imao više vremena da se stabilizuje, i samim tim da verodostojnije opiše temperaturu

## V. ZAKLJUČAK

Na ovaj način, završeno je sa proverom karakteristika senzora. Merenjima, došlo se do nekih nesavršenosti koje ukazuju na nesavršenosti i karakteristike konkretno korišćenog senzora, koji je samo jedan u ovoj seriji senzora. Ova merenja ne znače da će odstupanja biti ista i za bilo koji drugi LM35, jer bitnu ulogu ima i tehnologija izrade. Takođe, sa odstupanjima se može boriti i softverski. U skladu rezultata sa grafika, programski mogu se uneti odstupanja u zavisnosti od toga u kom opsegu se nalazi temperatura koja se meri. Na ovaj način bio bi dobijen kompletniji i tačniji termometar. No ipak, onda će taj termometar biti primenljiv samo na konkretno taj senzor, a

ne na čitavu seriju i samim tim biće neupotrebljiv za industrijsku proizvodnju. Stoga, treba prihvatiti dozvoljena odstupanja data u datasheetu i biti spreman na njih prilikom rada.

## ZAHVALNICA

Autor se ovom prilikom zahvaljuje profesoru Danijelu Dankoviću na podsticanju ideje za projekat, kao i za pomoć prilikom praktične realizacije. Takođe, zahvalnost ide saradniku Milošu Đorđeviću za pomoć pri praktičnoj realizaciji.

## LITERATURA

[1] Z. Prijić, D. Danković, Analogna mikroelektronika (A/D konverzija), Predavanja, Niš, 2016..Dostupno na: http://mikroelektronika.elfak.ni.ac.rs/analogna/pred_sr.php
[2] PIC16F87xA datasheet, Microchip, 2003. Dostupno na: http://ww1.microchip.com/downloads/en/DeviceDoc/39582b.pdf
[3] LM35 datasheet, Texas Instruments, 2016. Dostupno na: http://www.ti.com/lit/ds/symlink/lm35.pdf
[4] QAPASS 1602a datasheet, Hitachi, 2007. Dostupno na: https://funduino.de/DL/1602LCD.pdf
[5] Proton IDE, Crownhill Associates, 2016. Dostupno na: http://www.protonbasic.co.uk
[6] Proteus 8, Labcenter Electronics, 2014. Dostupno na: https://www.labcenter.com

# Projektovanje solarnog punjača za baterije

Nikola Mitrović

*Sadržaj* - U ovom radu je opisana praktična realizacija solarnog punjača za baterije. U prvom delu rada opisan je princip funkcionisanja fotonaponskih ćelija i sistema, u drugom delu je predstavljena simulacija sistema softverskim alatom Proteus 8.

## I. UVOD

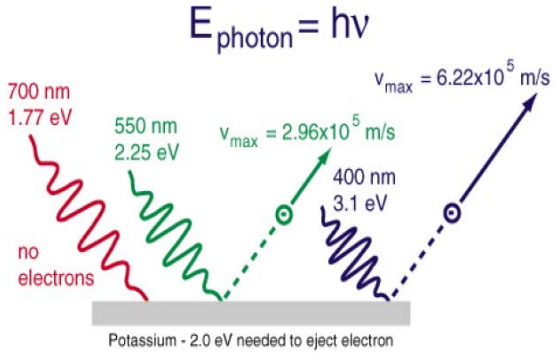
Solarna energija je energija sunčevog zračenja koju primećujemo u obliku svetla i toplote koju primamo od najvećeg izvora energije, Sunca. Sunčevo zračenje je odgovorno i za stalno obnavljanje energije vetra, morskih struja, talasa, vodenih tokova i termalnog gradijenta u okeanima. Već decenijama se solarna energija koristi za generisanje toplote u smislu zagrevanja vode, životnog prostora, a takođe i za hlađenje. Upotreba solarne energije ima višestruke prednosti. To je tih, čist i pouzdan izvor energije. Zbog rastuće cene fosilnih goriva kao i zbog jačanja svesti o potrebi očuvanja životne sredine sve više raste interes za korišćenje sunčeve energije.

## II. PRINCIP FUNKCIONISANJA

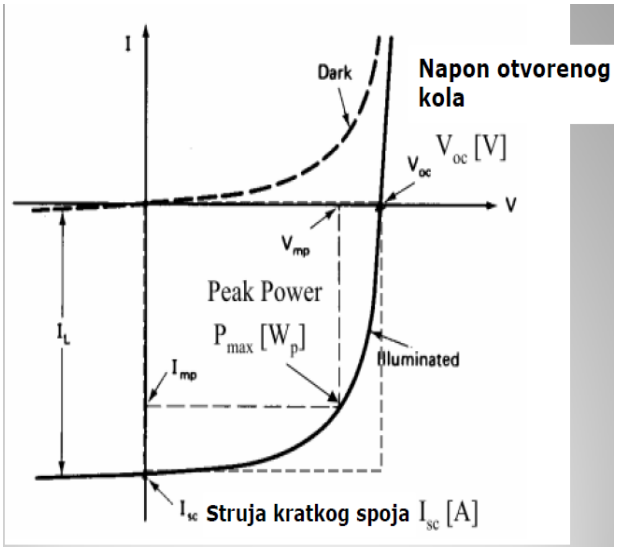
Fotonaponske ćelije su poluprovodnički elementi koji direktno pretvaraju energiju sunčevog zračenja u električnu energiju [1]. Efikasnost im je, od 10% za neke jeftinije, do 25% za skuplje sisteme. Fotonaponska ćelija je osnova fotonaponskih modula. Kao samostalni izvor energije koriste se npr. na satelitima, znakovima na putu i udaljenim objektima koji zahtevaju dugotrajni izvor energije. U svemiru je i snaga sunčevog zračenja puno veća jer Zemljina atmosfera apsorbuje veliki deo zračenja pa je i dobijena energija veća. Kao dodatni izvori energije fotonaponske ćelije mogu se na primer povezati na električnu mrežu. Princip rada fotonaponskih ćelija se zasniva na fotoelektričnom efektu. Fotonaponska ćelija je PN spoj koji je izložen sunčevoj svetlosti. Fotoni sunčeve svetlosti dovoljno jake energije imaju moć da izazovu eksitaciju elektrona (slika 1). Ekscitovani elektroni postaju nosioci naelektrisanja. Na ovaj način, posledica delovanja fotona je kretanje ekscitovanih elektrona. Ova pojava naziva se fotoelektrični efekat.

Naravno, ne može se očekivati da električna energija dobijena na ovakav način ima neke velike vrednosti, i ovo jeste možda i najveća mana ovog izvora energije. Koeficijent iskorišćenosti energije prosečne moderne ćelije je oko 20%. Grafik rada solarne ćelije diskutuje se u četvrtom kvadrantu, a primer je dat na sledećem grafiku koji je prikazan na slici 2.

Nikola Mitrović je student četvrte godine na Katedri za mikroelektroniku na Elektronskom fakultetu u Nišu, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: nikola.mitrovic.nis@gmail.com




Slika 1. Fotoelektrični efekat.




Slika 2. Grafik rada solarne ćelije.

Više solarnih ćelija, matrično organizovane, čine solarne module. Više solarnih modula čine solarne panele. Dobro projektovani solarni paneli mogu da proizvedu i jako veliki napon.

Za odgovarajuće napajanje za bateriju, potrebno je napon sa solarnog panela prilagoditi karakteristikama baterije. Potrebno je formirati linearni izvor napajanja [2]. Kod ovakvih izvora napajanja, u opštem slučaju, ulazni napon vrednosti $V_{in}$ se transformiše u izlazni napon vrednosti $V_{out}$ na principu promenljive provodnosti elemenata koji se nalazi između ulaza i izlaza. Nakon što napon sa ulaza dovodemo na isti red veličine kao i napon koji nam je potreban na izlazu (ovde se kod većih energija koriste transformatori, mada kod solarne energije, najčešće

nema potrebe za ovakvim korakom), potrebno je odabrati odgovarajući regulator napona. Naponski regulator je kolo koje na izlazu daje stabilisani napon određene vrednosti. Na ulazu i izlazu ove komponente postavljeni su filtarski elektrolitski kondenzatori. Prilikom odabira regulatora, treba voditi računa o opsegu ulaznog, opsegu izlaznog napona i minimalnom padu napona (dropout voltage). Takođe, uz filtarske kondenzatore valja postaviti i bypass kondenzatore koji imaju zadatak da eliminišu neželjene signale visoke učestanosti. Na ovaj način na bateriji se može dobiti željeni napon.

## III. KOMPONENTE ZA REALIZACIJU

Komponente koje su iskorišćene za realizaciju ovog sistema:

### *A. PV modul ZW80x55*

Ovaj solarni panel (prikazan na slici 3) ima solarnu efikasnost od oko 17%. Polikristalni je, i ima radni napon od 6V. Na poleđini ima izvučene krajeve za pozitivnu i negativnu elektrodu koji omogućavaju lako lemljenje, kao i kombinovanje više ovakvih panela.

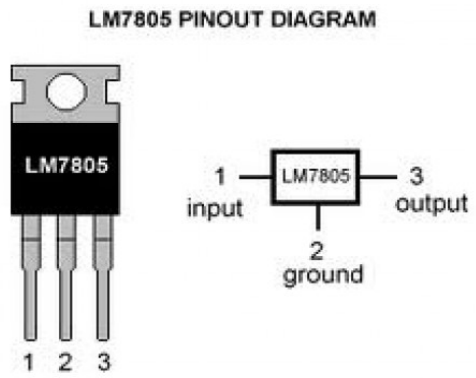


Slika 3. PV modul ZW80x55.

### *B. Naponski regulator LM7805*

Ovaj naponski regulator [3] je jedan od standardnih naponskih regulatora (slika 4). Tipični izlazni napon ovog regulatora je 5V, a ulazni može da ide i do 18V. Minimalni pad napona je 2V. Pri sobnoj temperaturi (25˚C) nema većih odstupanja.

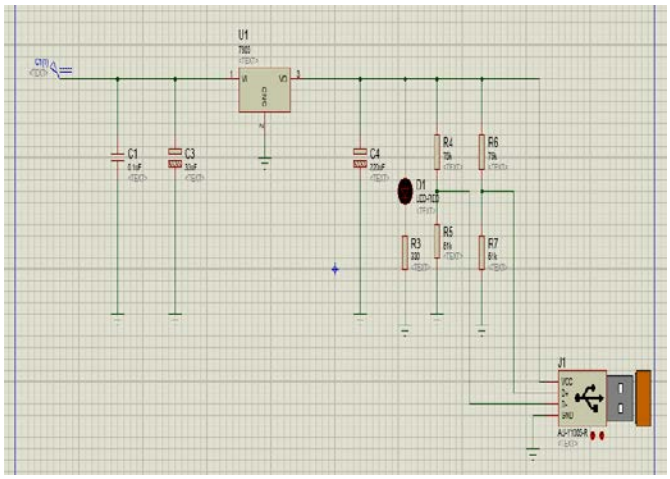
Pored ovih glavnih komponenti, pri realizaciji su iskorišćeni otpornici i kondenzatori.




Slika 4. Pinout diagram regulatora LM7805.

## IV. SIMULACIJA SOFTVERSKIM ALATOM PROTEUS

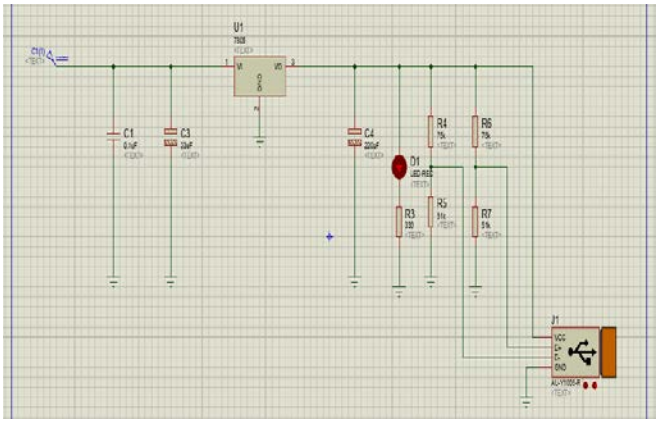
Simulacija punjača je realizovana softverskim alatom Proteus 8 [4]. Kao solarni panel korišćen je generator napona od 12V. Ovo se dobija serijskim vezivanjem dva PV modula ZW80x55.




Slika 5. Simulacija u programu Proteus 8.

Na slici 5. možemo primetiti da su odabrane tablične vrednosti za filtarske (C3 i C4) i za bypass kondenzatore (C1). Povezan je i regulator LM7805. Kako je ranije objašnjeno kod komponenata, ukoliko se na ovaj regulator dovede napon od 5V – 18V, na izlazu ćemo dobiti 5V.

Takođe, u kolu se mogu primetiti i naponski razdelnici koji imaju zadatak da prilagode napon izlaznom priključku za usb punjač. Odabir ovih razdelnika zavisi od toga koju tačno bateriju punimo i skladno tome, treba ga prilagođavati. To može znatno da utiče na životni vek baterije. Dodata je i grana sa LED diodom, koja služi za proveru rada kola.




Slika 6. Kolo nakon pokrenute simulacije.

Kada pokrenemo simulaciju i stavimo pokazivač na izlaz, možemo primetiti da dobijamo željenu vrednost izlaznog jednosmernog napona. Kolo nakon pokrenute simulacije prikazano je na slici 6. Takođe, primećujemo da dioda svetli, što je indikator toga da kolo radi. Na ovaj način, dobili smo odgovarajući punjač od solarnog panela.

Naravno, ono čega moramo biti svesni je da nije moguće bilo kojim punjačem puniti bilo koju bateriju. Odabir i projektovanje punjača (konkretno kombinovanje naponskih razdelnika) najviše zavisi od vrste baterije koju želimo da napunimo. Takođe, treba voditi računa o kapacitetu baterije. Ukoliko se baterija ne puni dovoljno često, ili se puni suviše često i duže nego što treba, moguće je značajno skratiti radni vek baterije. Odnos punjenja i pražnjenja baterije u odnosu na njen kapacitet u literaturi se često može sresti kao C-odnos (C-rate) [5]. Definiše se kao količnik struje punjenja i kapaciteta baterije, i ima jedinicu $h^{-1}$.

## V. Predlozi za unapređenje

Ovako projektovan punjač ograničen je na punjenje jedne vrste baterije, jednim načinom (preko usb-a). Način na koji bi se ovaj sistem mogao unaprediti je da se doda prekidač kojim bi se birala jedna od mreža razdelinka. Svaka od njih imala bi svoj izlaz, i karakterističan konektor na kraju koji bi omogućio punjenje odgovarajućeg tipa baterije. Ovakvi uređaji su komercijalno vrlo zastupljeni i predstavljaju jednu od modernih ,,gadget-a,,.

## VI. Zaključak

Solarni punjač za baterije može vrlo lako naći svoju primenu u budućnosti. Ovakav sistem predstavlja obnovljiv izvor energije koji ne zagađuje okolinu, ima širok dijapazon upotrebe i vrlo malo ograničavajućih faktora. Zbog današnje masivne upotrebe mobilnih telefona ovaj punjač može biti spas u određenim situacijama gde korisnik nema drugi način da napuni bateriju na svom telefonu. Sistem je takođe ekonomičan, ne zahteva preterana ulaganja i može biti dostupan za komercijalnu upotrebu.

## Zahvalnica

Autor se zahvaljuje profesoru Zoranu Prijiću na ustupanju literature koja je pomogla pri projektovanju bloka za napajanje. Takođe, zahvalnost se upućuje i profesoru Draganu Pantiću na teorijskim osnovama funkcionisanja solarnih elemenata.

## Literatura

[1] D. Pantić, Solarne komponente i sistemi, 2016. Dostupno na: http://mikroelektronika.elfak.ni.ac.rs/members_sr.php
[2] Z. Prijić, D. Danković, Analogna mikroelektronika, Predavanja, Niš, 2016. Dostupno na: http://mikroelektronika.elfak.ni.ac.rs/analogna/pred_sr.php
[3] LM7805, Fairchild Semiconductor, 2001. Dostupno na: http://pdf1.alldatasheet.com/datasheet-pdf/view/82833/FAIRCHILD/LM7805.html
[4] Proteus 8, Labcenter Electronics, 2014. Dostupno na: https://www.labcenter.com
[5] ,,A Guide to Understanding Battery Specifications,, - MIT Electric Vehicle Team , December 2008. Dostupno na: http://web.mit.edu/evt/summary_battery_specifications.pdf

# Praktična realizacija bezbedonosnog sistema

Nikola Mitrović, Mladen Radivojević

*Sadržaj* – U ovom radu predstavljena je praktična realizacija bezbedonosnog sistema uz pomoć lasera, kako analogna, tako i digitalna realizacija. U prvom delu rada opisan je princip funkcionisanja bezbedonosnog sistema prilikom realizacije, a u drugom delu je data simulacija sistema.

## I. UVOD

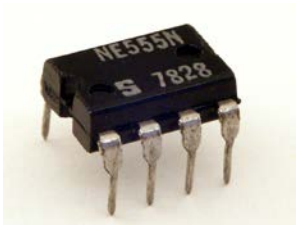
Bezbedonosni sistemi predstavljaju veoma bitan deo u funkcionisanju mnogih današnjih bitnih ustanova kao što su: banke, pošte, muzeji i drugo. Imaju vrlo široku primenu kod svih ustanova gde je bezbednost objekata i predmeta smeštenih u njima na prvom mestu. Naravno da je cilj banke da recimo, zaštiti svoj kapital ili muzeja da zaštiti svoje vredne eksponate. To se upravo postiže instaliranjem bezbedonosnih sistema. Jedan ovakav bezbedonosni sistem jeste dosta ekonomičan, lak je za održavanje a doprinosi dosta samoj bezbednosti jer omogućava brzu i pravovremenu reakciju u nepoželjnim situacijama.

## II. KOMPONENTE ZA REALIZACIJU

Glavne komponente koje su iskorišćene za realizaciju :

### *A. NE555 tajmer*

NE555 (slika 1) [1] je integrisano kolo koje spada u najkorišćenija kola na svetu, i predstavlja srce ovog sistema. NE555 se u komplikovanijim kolima može koristiti na čak 4 načina, odnosno u 4 režima rada. Postoje astabilni režim rada, monostabilni režim rada, bistabilni režim rada i režim rada invertujućeg bafera. Od ova 4 režima rada, u ovom radu iskorišćen je rad NE555 kola u astabilnom i monostabilnom režimu rada.




Slika 1. Tajmer NE555.

Nikola Mitrović je student četvrte godine na Katedri za Mikroelektroniku na Elektronskom fakultetu u Nišu, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: nikola.mitrovic.nis@gmail.com

Mladen Radivojević je student četvrte godine na Katedri za Mikroelektroniku na Elektronskom fakultetu u Nišu, univerzitet u Nišu, Aleksandra medvedeva 14, 18000 Niš, Serbia, Email:mladenradivojevic389@gmail.com

### *B. Fotootpornik VT900*

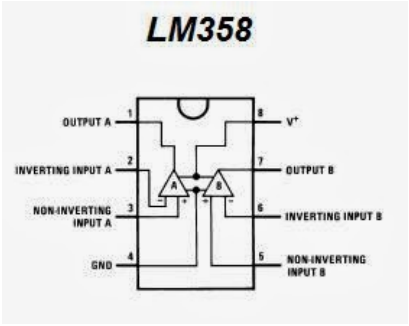
Fotootpornik (slika 2) [2] je otpornik čiji se otpor menja pod uticajem svetlosti koja pada na njega. Zbog te osobine, nalazi primenu kao senzor osvetljenja.



Slika 2. Fotootpornik VT900.

### *C. Komparator LM358*

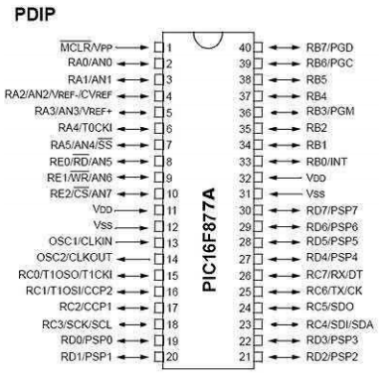
Komparator LM358 (slika 3) [3] u svojoj strukturi ima operacioni pojačavač koji radi kao komparator signala. Ako je signal koji je na neinvertujućem ulazu veći od signala na invertujućem ulazu, ovaj komparator na svom izlazu generiše signal logičke jedinice, a ukoliko to nije slučaj, na izlazu se dobija signal logičke nule.




Slika 3. Komparator LM358.

### *D. Mikrokontroler PIC16F877A*

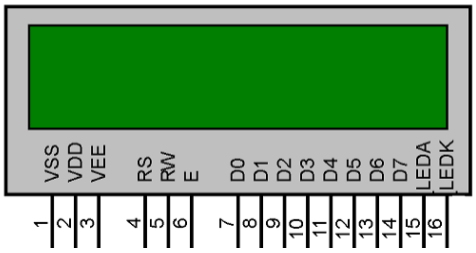
PIC mikrokontroler PIC16F877A (slika 4) [4] je jedan od najprisutnijih mikrokontrolera u modernoj industriji. Vrlo je praktičan za korišćenje, pa postoje odgovarajuća softverska okruženja sa kojima je relativno lako kodirati i programirati ga. Jedna od glavnih prednosti ovog mikrokontrolera je ta što je on ,,piši-briši,, mikrokontroler, jer koristi tehnologiju fleš memorije. Na taj način, moguće je koristiti ga više puta, svaki put sa različitom funkcijom. Ima ukupno 40 pinova, od kojih su 33 ulazno-izlazni.




Slika 4. Pinout mikrokontrolera PIC16F877A.

*E. 16x2 LCD Displej*

LCD (Liquid Crystal Display) [5] ekran je elektronska komponenta koja nalazi jako širok spektar primena. 16x2 displej je jedan od najjednostavnijih modula i veoma je često korišćen u uređajima (slika 5).




Slika 5. Dijagram pinova LCD.

Uz sve ove glavne komponente, iskorišćeni su i otpornici, kondenzatori, LED diode i izvor konstantog snopa svetlosti (laser).

## III. Princip funkcionisanja

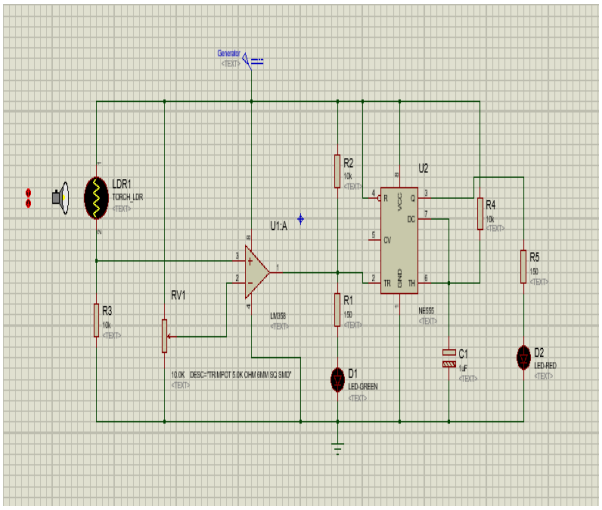
Princip funkcionisanja ovog bezbedonosnog sistema se zasniva na osobini fotootpornika da provodi struju dok je osvetljen. U ovom sistemu, fotootpornik je konstantno osvetljen laserskim snopom. Sistem je realizovan tako da sve dok je fotootpornik osvetljen, sistem miruje. Prilikom prekidanja konstantnog laserskog zraka sistem javlja promenu koja je podešena da u ovom sistemu signalizira uzbunu. Kod analogne realizacije sistema fotootpornik je preko razdelnika napona povezan na neinvertujući ulaz komparatora. Na invertujućem ulazu je potenciometar kojim je podešavan referentni napon i time je definisana osetljivost sistema. Komparator upoređuje ova dva signala, i ukoliko je signal sa neinvertujućeg ulaza veći, na svom izlazu daje logičku jedinicu. Signal sa komparatora ide na tajmer NE555, tačnije ulaz TR, koji daje na Q izlazu logičku nulu. Signal sa izlaza komparatora takođe pobuđuje zelenu LED diodu koja služi kao signalizacija da je sve u redu.

Kada se prekine laserski snop, fotootpornik ne provodi struju i na neinvertujući ulaz komparatora stiže signal koji je manji od referentnog, tako da on na svom izlazu daje logičku nulu. Logička nula ne pobuđuje zelenu diodu, ali aktivira tajmer NE555 koji na svom izlazu daje signal koji pobudjuje crvenu LED diodu koja signalizira uzbunu. Što se tiče digitalne realizacije, glavnu komponentu sistema predstavlja mikrokontroler PIC16F877A koji je isprogramiran da kada fotootpornik provodi struju sistem signalizira da je sve u redu, a kada se snop laserskog zraka prekine sistem signalizira uzbunu. Sistem signalizira uzbunu blinkanjem crvene LED diode i određenim natpisom na displeju.

## IV. Simulacija softverskim alatom Proteus

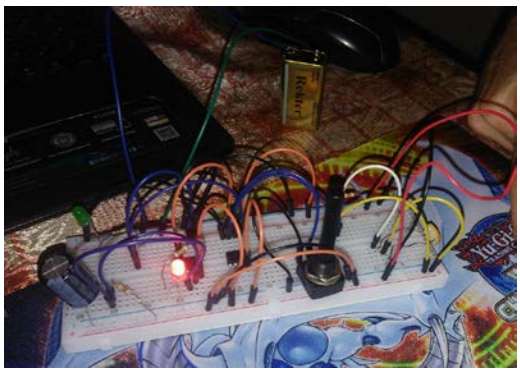
U ovom delu opisan je rad bezbedonosnog sistema u softverskom alatu Proteus 8. Softverska simulacija obuhvata analognu i digitalnu realizaciju sistema.

*A. Simulacija analogne realizacije*



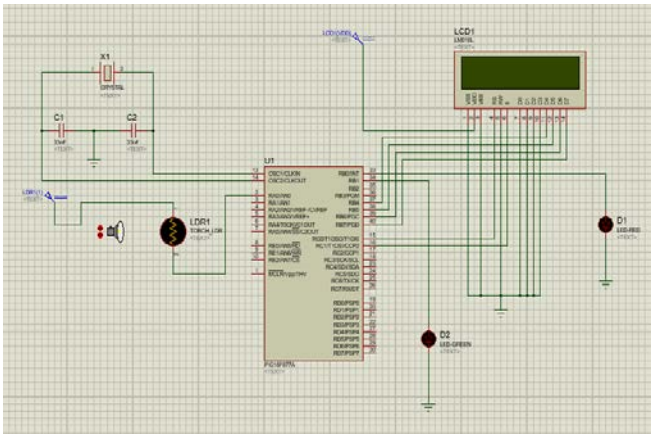

Slika 6. Šema analogne realizacije sistema.

Na slici 6 prikazan je preciznije način rada sistema prilikom analogne realizacije. Fotootpornik koji je osvetljen konstantnim snopom svetlosti je preko razdelnika napona povezan na komparator LM358, čiji je izlaz povezan na tajmer NE555. Zelena i crvena LED dioda služe kao indikatori, zelena diode signalizira da je sve u redu, crvena je znak za uzbunu. Zelenu LED diodu pobuđuje signal iz komparatora LM358 (logička jedinica), dok crvena LED dioda biva pobuđena signalom iz tajmera NE555 koji u ovoj konfiguraciji biva aktiviran signalom logičke nule na ulazu.



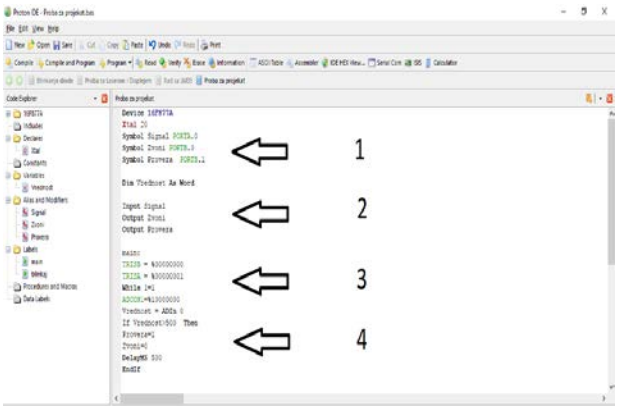
Slika 7. Praktičan prikaz analogne realizacije.

*B. Simulacija digitalne realizacije*

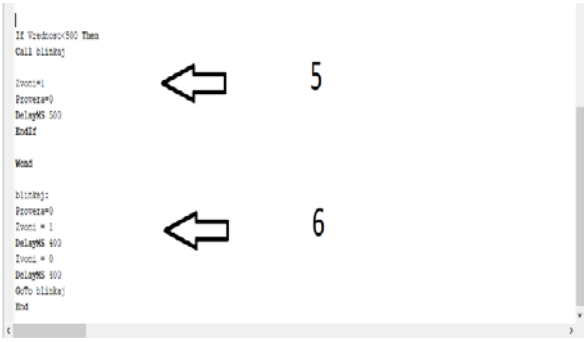


Slika 8. Šema digitalne realizacije sistema.

Na slici 8 prikazana je šema digitalne realizacije sistema. Princip je sličan, razlika je u tome što ovde glavnu ulogu ima mikrokontroler PIC16F877A koji je programiran da kada je snop svetlosti konstantan signalizira da je sve u redu uz pomoć zelene LED diode. Kada je snop svetlosti prekinut, mikrokontroler uključuje crvenu LED diodu koja blinka i ispisuje natpis na displeju koji je povezan na mikrokontroler.

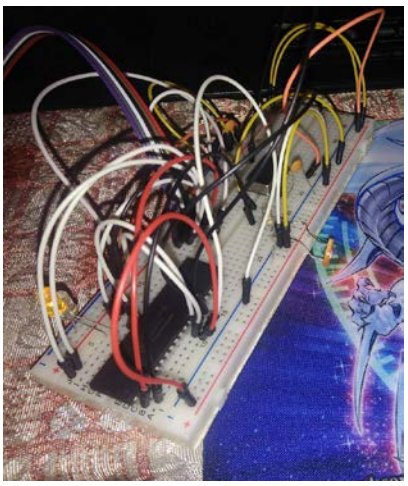



Slika 9. Kod za programiranje mikrokontrolera.




Slika 10. Deo koda koji se realizuje kada je snop svetlosti prekinut.

Kod je realizovan u softverskom alatu Proton IDE. Na slici 9 vidimo deo koda koji je podešen port B mikrokontrolera da bude izlazni, a jedan pin porta A je podešen da bude ulazni (broj 3 na slici 9). If petljom proverava se da li je snop svetlosti prekinut (broj 4 na slici 9). Ukoliko nije, sistem je stabilan i zelena LED svetli (broj 5 na slici 10). Ukoliko jeste, pokreće se deo koda koji je prikazan na slici 10. On aktivira crvenu LED i ispisuje proizvoljan natpis na displeju.



Slika 11. Praktičan prikaz digitalne realizacije.

## V. ZAKLJUČAK

Ideja analogne i digitalne realizacije je upravo međusobno upoređivanje ova dva načina realizacije. Analogna realizacija je dosta ekonomičnija, ne zahteva prevelika ulaganja ali je ograničena, ne postoji previliki spektar mogućnosti optimizovanja sistema, veoma je zahvetno ubaciti dodatni displej ili na bilo koji način proširiti ovaj sistem. Što se tiče digitalne realizacije, ona jeste skuplja jer podrazumeva kupovinu mikrokontrolera i displeja koji spadaju u skuplje komponente, ali je realizacija lakša i mogućnost optimizacije je velika. Moguće je unaprediti ovaj sistem jer mikrokontroler podržava rad sa velikim brojem komponenata koje bi doprinele daljoj optimizaciji realizovanog sistema.

## ZAHVALNICA

Projekat je razvijen kao unapređena praktična realizicija na predmetu Laserska elektronika na Katedri za mikroelektroniku na Elektronskom fakultetu u Nišu. Autori se zahvaljuju profesorki Vesni Paunović za podsticanje ideje. Takođe se zahvaljuju saradniku u nastavi Milošu Đorđeviću, za pomoć pri praktičnom radu.

## LITERATURA

[1] Tajmer NE555, Texas Instruments, 2014. Dostupno na: http://www.ti.com/lit/ds/symlink/ne555.pdf

[2] Fotootpornik VT900, Farnell, 2014. Dostupno na: http://www.farnell.com/datasheets/919043.pdf

[3] Komparator LM358, Texas Instruments, 2017. Dostupno na: http://www.ti.com/lit/ds/symlink/lm258a.pdf

[4] PIC16F87xA datasheet, Microchip, 2003. Dostupno na: http://ww1.microchip.com/downloads/en/DeviceDoc/39582b.pdf

[5] QAPASS 1602a datasheet, Hitachi, 2007. Dostupno na: https://funduino.de/DL/1602LCD.pdf

[6] Proteus 8, Labcenter Electronics, 2014. Dostupno na: https://www.labcenter.com

[7] Proton IDE, Crownhill Associates, 2016. Dostupno на: http://www.protonbasic.co.uk

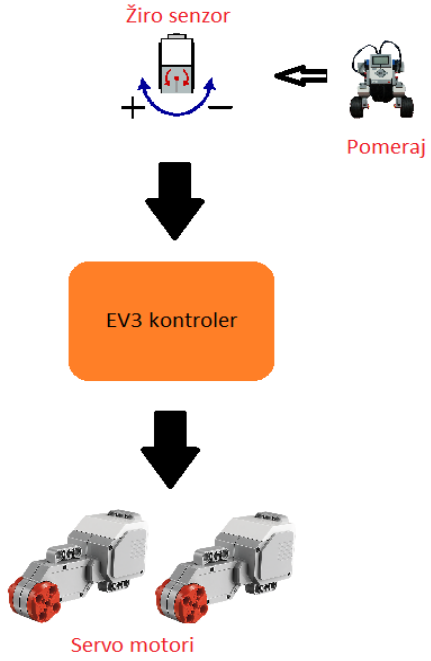
# Primena žiro senzora za upravljanje elektro motorima na primeru Lego platforme

Aleksa Rajković, Ognjen Ilić

*Sadržaj* - Ovaj rad ima za cilj da predstavi jednu od mogućih primena žiro senzora. Osnovu sistema čine: Lego EV3 kontroler, žiro senzor i elektro motori za kretanje, iz Lego paketa [1]. Programiranje se vrši u softverskom paketu Lego mindstorms EV3.

## I. UVOD

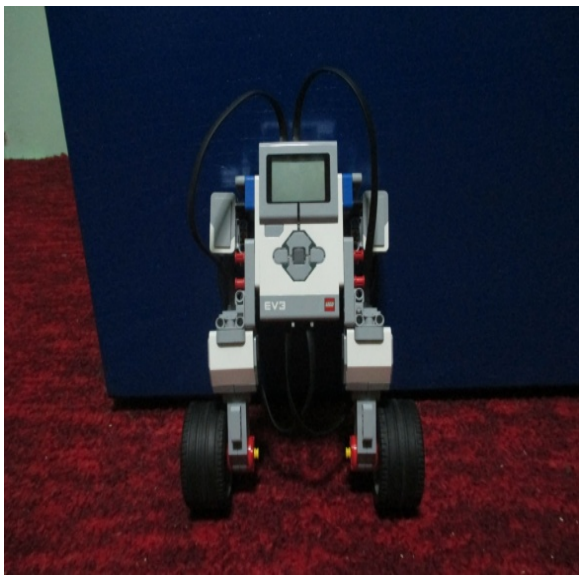
Sistem za upravljanje funkcioniše tako što žiro senzor detektuje promenu stanja, tj. pomeraj konstrukcije pod dejstvom gravitacije i informacije prenosi u EV3 kontroler. EV3 kontroler obrađuje podatke na osnovu kojih odlučuje o smeru i intenzitetu okretajnja motora. Osnovna šema sistema prikazana ja na slici 1.




Slika 1. Osnovna šema sistema.

Konačan rezultat rada je Lego platforma prikazana na slici 2 .

Ognjen Ilic je učenik završnog razreda smera Elektrotehnicar mehatronike, dok je Aleksa Rajković učenik treće godine smera: Elektrotehničar informacionih tehnologija – ogled u „Tehničkoj školi Zaječar“, Knjeginje Ljubice 3-5, 19000 Zaječar. Email: arajkovic22@gmail.com



Slika 2. Lego platforma.

## II. PRINCIP RADA ŽIRO SENZORA

Sistem funkcioniše tako što EV3 kontroler prikuplja informacije od žiro senzora, obrađuje podatake i kontroliše rad motora. Osnovna namena sistema je balansiranje konstrukcije na dva točka i njegovo kretanje u prostoru.

### *A. Predstavljanje komponenata i priprema za izradu*

EV3 kontroler [2] je upravljačka jedinica iz Lego paketa. Hardver se sastoji od mikrokontrolera pod oznakom ARM 9 koji poseduje 16MB flash memorije, kao i 64MB RAM memorije. Na jedinici se nalazi ukupno 8 portova od kojih se 4 upotrebljavaju za povezivanje motora, a ostala 4 za povezivanje senzora. Softver se sastoji od razvojnog okruženja koje čine standardni kompajler i bootloader koji se nalazi na samoj ploči sa Linux operativnim sistemom otvorenog tipa.

EV3 kontroler se povezuje na racunar, preko koga se i programira, putem USB kabla, Bluetooth-a ili WiFi-a. Na slici 3. prikazan je EV3 kontroler koji je korišćen u razvoju ovog projekta.



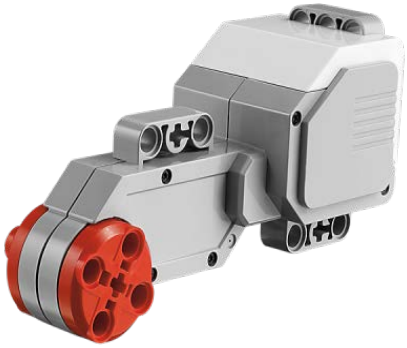

Slika 3. EV3 kontroler.

Žiro senzor [3] je prikazan na slici 4. Žiro senzor je uređaj za merenje promene ugla tela. Funkcioniše na principa ugaonog momenta. Najveći pomeraj koji žiro senzor može da zabeleži je 440 stepeni u sekundi, podatke šalje 1000 puta u sekundi i radi na frekvenciji od 1kHz. Preciznost žiro senzora je (+/-3 stepena) i može da meri pomeraj u sve tri ose (x y z).



Slika 4. Žiro senzor.

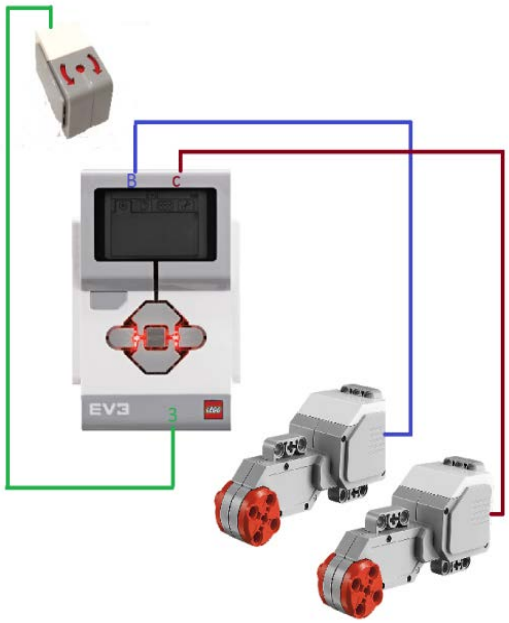
Servo motor [4] koristi povratne informacije sa EV3 kontrolera za preciznu kontrolu kretanja Lego platforme. Korišćenjem ugrađenih senzora rotacije, motori se mogu sinhronizovati tako da se kreću ravnomerno istom brzinom. Na slici 5. je prikazan servo motor.



Slika 5. Servo motor.

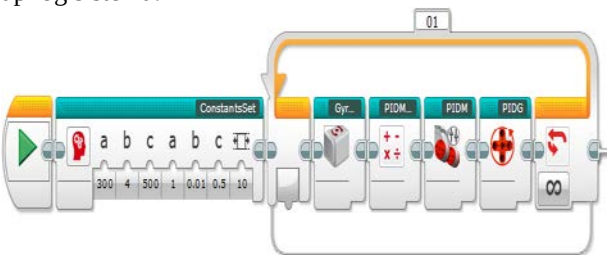
*B. Izrada sistema*

Povezivanje senzora i motora sa Lego EV3 se vrši pomoću kablova sa RJ12 konektorima. Žiro senzor se povezuje na ulaznom portu broj 3 EV3 kontrolera. Motore povezujemo pomoću izlaznih portova tako da levi motor povežemo na port B, a desni na port C EV3 kontrolera. Šema povezivanja je na slici 6.




Slika 6. Šema za povezivanje.

Nakon povezivanja, napisan je kod u softveru Lego Mindstorms EV3 [5] koji se preko bluetooth-a ili USB kabla prebacuje u EV3 kontroler. Nakon pokretanja kontrolera žiro senzor posebnom komandom kalibrišemo. Kalibracija se vrsi u stanju mirovanja senzora. Senzor konstantno šalje podatke o uglu ose konstrukcije. Nakon obrade podataka EV3 kontroler po potrebi aktivira motore. Na taj način se omogućava da konstrukcija ostane u vertikalnom položaju. U softverskom paketu EV3 mindstorms programiranje se vrši pomoću blokova. Zeleni blokovi su izlazne naredbe, žuti blokovi su izlazni podaci senzora, narandžasti blokovi su programske petlje, crveni blokovi su matematičke operacije i plavi blokovi su funkcije (skup više blokova). Na slici 7. prikazan je kod celokupnog sistema.

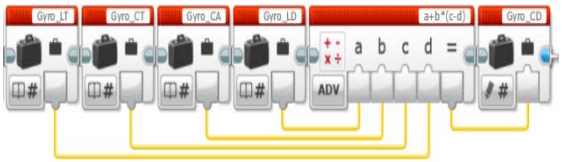



Slika 7. Kod celokupnog sistema.

Prvi blok služi za podešavanja svih parametra u sistemu i podesavanje PID[6] regulatora. U petlji se nalazi: blok za očitavalje žiro senzora, blok za računanje “greške”, blok za PID kontrolisanje ugla praćenja i blok za PID kontrolu motora.

PID regulator izvršava fukciju obrade “greške” u realnom vremenu koja upravlja smerom i intenzitetom okretanja motora.

Najbitniji blokovi koda očitavanja žiro senzora:




Slika 8. Kod očitavanja žiro senzora.

Na slici 8. je prikazan kod očitavanja žiro senzora. Zbog razloga što žiro sensor nije u mogućnosti da očita apsolutni ugao, ugao se izračunava u odnosu ugla prethodnog merenog stanja, brzine promene ugla i vremenskog intervala.

## III. Zaključak

Celokupan projekat je osmišljen od strane učenika, a realizovan uz pomoć profesora-mentora. Bilo je potrebno primeniti znanje ne samo iz informatike već i iz elektronike, elektrotehnike, mehatronike i praktičnog rada sa elektro materijalima. Ovaj projekat pruža i mogućnost za dalja istraživanja kao što je upravljanje kretanja preko Bluetooth-a, WiFi-a ili mobilne aplikacije.

## Zahvalnica

Autori rada zahvaljuju se Tehničkoj školi u Zaječaru radi pružanja tehničkih i materijalnih sredstava takođe velika zahvalnost i profesorima elektro aktiva škole na pruženoj podršci, a posebno mentorimu dip.inž.elek. Dejanu Nikoliću.

## Literatura

[1] Lego Page
https://www.lego.com/en-us/mindstorms/about-ev3)
Septembar, 2017
[2] EV3 kontroler
(http://www.legama.si/en/pregled-produktov-za-ev3)
Septembar, 2017
[3] Žiro sensor
(https://education.lego.com/en-us/products/ev3-gyro-sensor-/45505), Avgust, 2017
[4] Servo motor
(https://en.wikipedia.org/wiki/Servomotor), Septembar, 2017
[5] Lego Mindstorms EV3
(https://www.lego.com/en-us/mindstorms/learn-to-program), Avgust, 2017
[6] PID
(https://en.wikipedia.org/wiki/PID_controller), Avgust, 2017

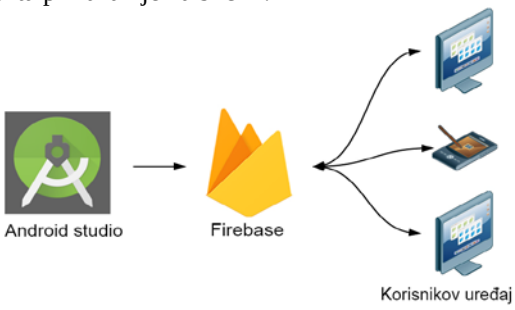
# Android aplikacija za muzički festival "Outhide"

Marko Mihajlović

*Sadržaj* - Ovaj rad ima za cilj da predstavi aplikaciju za uređaje koji koriste Android oprativni sistem. Aplikacija je razvijena u softerskom paketu: *Andriod Studio* [1] i servis za baze podataka *Firebase* [2].

## I. UVOD

Aplikacija je uslužni servis za mobilne telefone i sve vrste uređaja koji koriste *Android* operativni sistem. Za izradu aplikacije korišćena je uslužna baza podataka *Firebase* koja u realnom vremenu omogućava ažuriranje podataka. Postupak izrade aplikacije celog projekta prikazan je na slici 1.




Slika 1.Postupak izrade aplikacije.

Konačan rezultat rada prikazan je na slici 2. gde je prikazan jedna stranica screenshot aplikacije "*Outhide*" tj. jedna stranica o vestima.




Slika 2. *Screenshot* aplikacije "*Outhide*".

## II. PRINCIP RADA

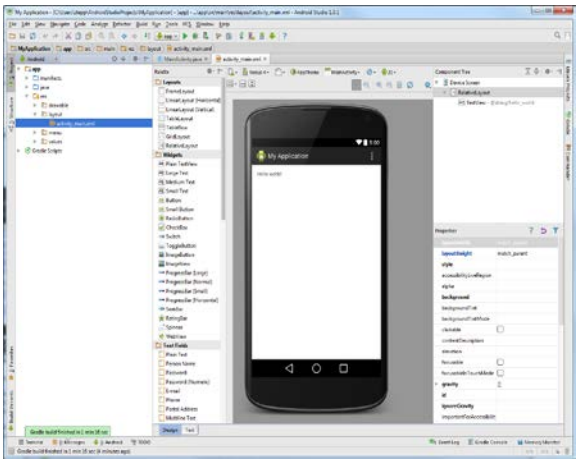
Sistem funkcioniše pomoću servera koji u realnom vremenu kontrolše novosti u vezi festivala i koji direktno šalje željena obaveštenja korisnicima aplikacije.

Marko Mihajlović je učenik Tehničke škole Zaječar, Knjeginje Ljubice 3-5, 19000 Zaječar. Email: markomihajlovicfm@gmail.com

Takođe korisnik aplikacije ima mogućnost da ostavi svoje mišljenje koje se prosleđuje organizatorima. Osnovna namena aplikacije je interakcija između orginizatora festivala i ljudi koji želi da posete festival.

### *A. Predstavljanje softverskih alata i priprema za izradu*

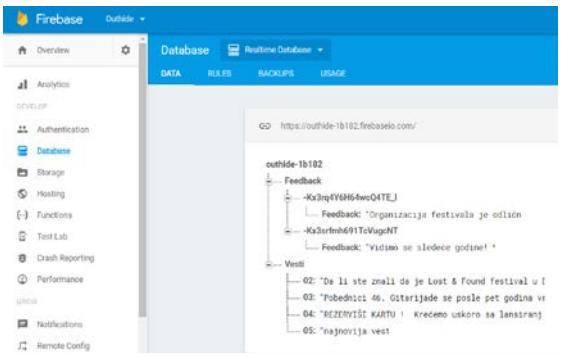
*Android Studio* je platforma otvorenog koda koja povezuje Javu [3] i XML [4]. Java služi za osnovne funkcije i kontrolisanje aplikacije. Dok XML služi za dizajn aplikacje i korisnički interfejs. Testiranje aplikacije je izvršeno preko virtuelnog uređaja koji je ugrađen u samoj platformi.



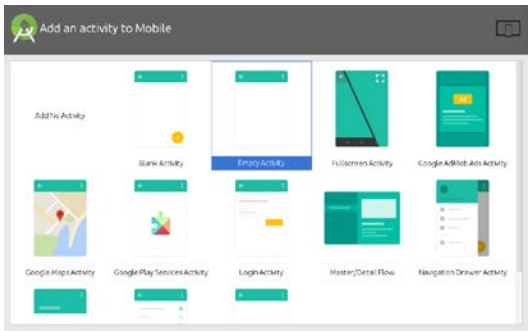
Slika 3. *Android Studio*.

Na samom početku pravljenja aplikacije potrebno je da se napiše kod i pripremi aplikacija za rad u *Android Studio-u* (slika 3). Potom se pomoću *Firebase-a* (slika 4) kontrolišu odgovarajući podaci koji se prikazuju na uređaju korisnika. Kada korisnik pregleda aplikaciju može ostaviti svoje mišljenje o unapređenju organicacije festivala koje se skladišti u databazu. Na taj način se vrši interakcija između organizatora i korsinika. Aplikacija koristi *Firebase Cloud Messaging* (FCM) [5], pomoću njega, preko *FireBase-a* u mogućnosti smo da šaljemo obaveštenja svim korisnicima aplikacije. Obaveštenje praćeno zvukom će se pojaviti na ekranima svih korisnika kada budu imali pristup internet konekciji.

*Android Studio* omogućava kreiranje šablona za osnovu aplikacije i sadrži kodove za odgovarajuće kategorije. Glavni šabloni su prikazani na slici 5.




Slika 4. *Firebase.*




Slika 5. Glavni šabloni.

*B. Izrada aplikacije*

Programiranje se vrši u programskom jeziku Java koji ima svoju klasu u programu *Android Studio*, dizajniranje aplikacije rađeno je XML *layoutu.* Svaki korišćeni *class* ima svoj odgovarajući *layout. Java class* i XML layout povezani u jednu celinu čine aktivnost (*Activity*). Pomoću odgovarajuće opcije u programu automatski se kreira jedna prazna aktivnost. Realizacijom prve aktivnost možemo krenuti sa pisanjem koda. Na slici 6. je prikazan kod u XML-u za ubacivanje linkovane slike ka drustvenim mrezama Facebook i Instagram.

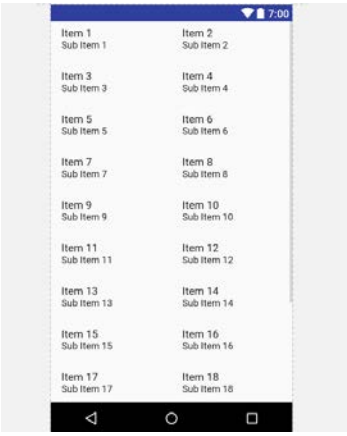
Fragment [6] je deo korisničkog interfejsa u aktivnosti koji čini najvažniji deo ove aplikacije zato što on omogućava prevlačenje između stranica. Aktivnost kontroliše sve fragmente jer oni nemaju mogućnost da funkcionišu sami. Fragmenti su orginalno osmišljeni za bolji dizajn korisničkog interfejsa većih ekrana, kao što je tablet. Zbog veličine ekrana postoji više prostora za kombinovanje i razmenu komponenti korisničkog interfejsa. U fragmentu je omogućen ovakav dizajn aplikacije bez kompleksnog programiranja. Deljenjem layout-a na fragmente dizajn korisnčkog interfejsa postaje znatno profesionalniji. Za kretanje kroz fragmente korišćene su SWITCH i CASE komande.

U Android aplikaciji koristimo funkciju *GridView*, (slika 7). *GridView* funkcija omogućava ubacivanje više sadržaja na jednoj stranici i kreira prostor za odgovarajući sadržaj. Zbog velikog memorijsog sadržaja slika, koristiti se biblioteka *Glide* [7] kao pomoćni alat. On obezbeđuje brži pristup zahtevnim slikama i omogućava bezbedno korišćenje aplikacije.

```
<ImageButton
    android:layout_width="210dp"
    android:layout_height="210dp"
    android:scaleType="fitCenter"
    android:id="@+id/facebook"
    android:src="@drawable/facebook"
    style="?android:attr/borderlessButtonStyle"
    />

<ImageButton
    android:layout_width="210dp"
    android:layout_height="210dp"
    android:scaleType="fitCenter"
    android:id="@+id/insta"
    android:src="@drawable/ig"
    style="?android:attr/borderlessButtonStyle"
    />
```

Slika 6. XML layout, primer ubacivanja facebook i instagram slike.




Slika 7. GridView funkcija.

Na slici 8. prikazan je Java kod koji determiniše rad *GridView* funkcije. Ovaj kod omogućava zamenu slike i teksta, kao i pregled detaljnih informacija o izvođaču. Zamena se radi zbog boljeg korišćenja memorijskog prostora.

```
gridView.setOnItemClickListener((parent, view, position, id) → {
        toast.setVisibility(View.VISIBLE);
        toast.setText(textList[position] + ": \n\n" + textMore[position]);
        naslov.setVisibility(View.INVISIBLE);
        gridView.setVisibility(View.INVISIBLE);
});

toast.setOnClickListener((OnClickListener) (v) → {
        naslov.setVisibility(View.VISIBLE);
        gridView.setVisibility(View.VISIBLE);
        toast.setVisibility(View.INVISIBLE);
});
```

Slika 8. Determinisanje *GridView* funkcije.

*C. Najbitnije linije koda*

*Linija koda za ugrađivanje layout-a u class:* `setContentView(R.layout.a_layout);`
*Linija koda za vertikalnu (portret) orijentaciju ekrana:* `this.setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);`

*Kod za ubacivanje slika:*

```
<ImageView
android:layout_width="match_parent"
android:layout_height="match_parent"
android:src="@drawable/slika"/>
```

*Linija koda za greške u aplikaciji:*

```
Toast.makeText(Activity.this,
"Error...Please restart app.",
        Toast.LENGTH_LONG).show();
```

*Linija koda za dodavanje nove aktivnosti:*

```
Intent i = new
Intent(SplashActivity.this,
MainActivity.class);
startActivity(i);
```

*Linija koda za postavljenje ikonice u Tab:*

```
tabLayout.addTab(tabLayout.newTab().se
tIcon(R.drawable.ic_home));
```

*Linija koda za postavljanje slike pomoću Glide-a:*

```
Glide.with(context).load(file).into(Im
g);
```

## III. ZAKLJUČAK

Celokupan projekat je osmišljen od strane učenika, a realizovan uz pomoć profesora-mentora. Bilo je potrebno primeniti znanje iz programiranja i Web dizajna. Aplikacija je našala svoju primenu na održanom festivalu.

Ovaj projekat pruža i mogućnost za dalja poboljšanja kao što je na primer naručivanje karte za festival i rezervisanje smeštaja. To bi zahtevalo povećanu sigurnost transakcija, autentifikacija i e-banking.

## ZAHVALNICA

Autor rada zahvaljuju se Tehničkoj školi u Zaječaru radi i profesorima elektro-informatičkog aktiva škole na pruženoj podršci, a posebno mentorima dipl.inž. Dejanu Nikoliću i dipl. inž. Draganu Manojloviću.

## LITERATURA

[1] Offical Android Studio page, https://developer.android.com, Septembar, 2017
[2] Offical FireBase page, https://firebase.google.com, Septembar, 2017
[3] Offical Java page, https://www.java.com, Septembar, 2017
[4] XML introduction https://www.w3schools.com/xml, Septembar, 2017
[5] Firebase Cloud Messaging (FCM) https://firebase.google.com/docs/cloud-messaging/, Septembar, 2017
[6] Fragments information https://developer.android.com/guide/components/fragments.html Septembar, 2017
[7] Glide information https://futurestud.io/tutorials/glide-getting-started, Septembar, 2017

# Area Calculator - Softver za merenje površine objekata na slici

Nemanja Petrović, Jelena Tasić

*Sadržaj* - Osnovna namena softvera Area Calculator je efikasno određivanje stvarne površine objekata na slici. Predstavlja softver koji omogućava lakše izračunavanje površine objekata u stvarnom svetu.

## I. UVOD

Potreba za ovim softverom se javila kod osoba koje se bave rekonstrukcijom istorijski značajnih objekata, slika, mozaika i građevina. Sama rekonstrukcija zahteva određenu količinu materija koji se koristi za popravke. Pre same rekonstrukcije potrebno je odrediti o kojoj količini materijala se radi. Za određivanje količine vrši se merenje površine objekta nad kojim se vrši popravka. Zbog visoke cene materijala javila se potreba za ovakvim softverom kojim se jednostavno određuje površina slikanog objekta.

## II. PRISTUP

### *A. Detekcija markera*

Za određivanje stvarne veličine objekata na slici koristi se pristup detekcije markera. Pronalaskom markera na slici dobijamo vrednost jednog piksela u milimetrima. Kao primer prepoznavanja markera koristićemo sledeće markere:

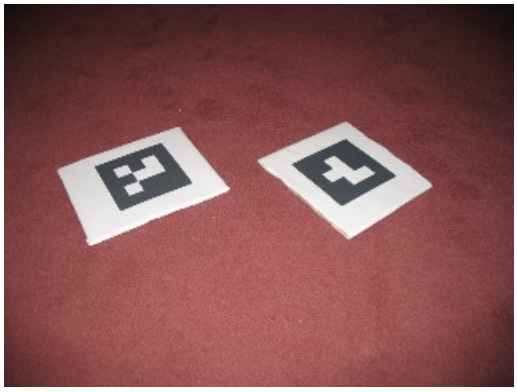


Slika 1. Vrste markera.

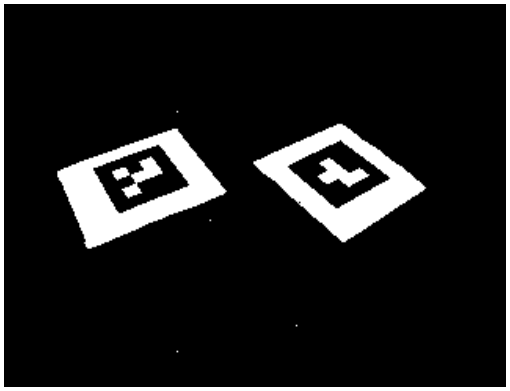
Algoritam zahteva da markeri budu striktne veličine, da sadrže po 5 kvadrata u smeru X i Y koordinate i da ivice markera uvek budu u crnoj boji radi prepoznavanja.

Prvi korak pri prepoznavanju markera je prepoznati na slici nešto što liči na marker, zbog toga prvo treba primeniti filtere „grayscale“ i „threshold“ kojim se dobja crno bela slika.

Nemanja Petrović i Jelena Tasić su studenti Elektronskog fakulteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija. Email: nemanjapetrovic1794@gmail.com

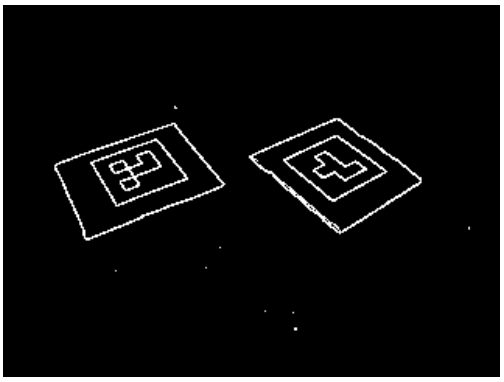


Slika 2. Primeri markera.



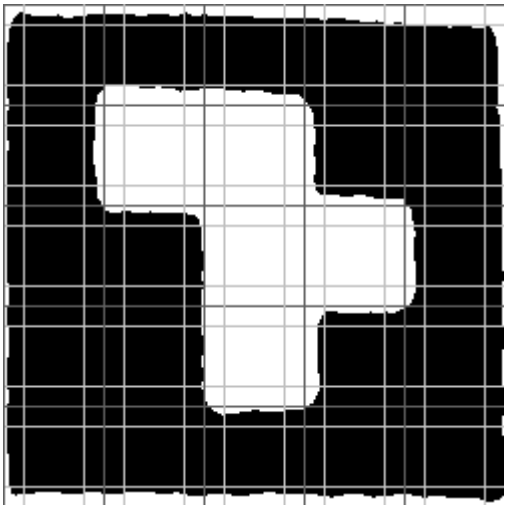
Slika 3. Primena filtera „grayscale“ i „threshold“.

Nakon primene filtera u provom koraku, potrebno je izvršiti prepoznavanje ivica putem algoritma „edge detection“ kojim se dobija rezultat kao na sledećoj slici:



Slika 4. Primena algoritma „edge detection“.

Preostaje samo prepoznati unutrašnje ivice markera i prebrojati kvadrate unutar markera, time se dobija konačna pozicija markera.



Slika 5. Pronalazak kvadrata u slici.



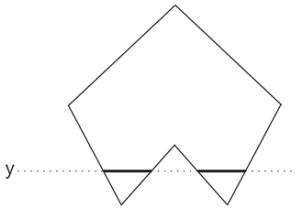
Slika 6. Pronađeni markeri.

*B. Određivanje povšine poligona zadatog listom temena*

Algoritam za određivanje površine poligona zadatog listom temena određuje njegovu površinu u pikselima. Svako teme zadato je svojom x i y koordinatom na ekranu, a lista temena može da čuva temena u redosledu kretanja kazaljke na satu, ili suprotnom kretanju kazaljke na satu. Na osnovu dva susedna temena u listi temena moguće je dobiti podatke o stranici koju ta temena formiraju, kao sto su y koordinate najviše i najniže tačke stranice i nagib stranice. Ove podatke algoritam koristi pri radu, pa ih pre početka samog određivanja površine izračunava i smesta u listu u kojoj pamti podatke za svaku stranicu.

Algoritam određuje površinu poligona krenuvši od temena poligona čija je y koordinata najmanja (najviše teme poligona na ekranu). Za svaku narednu vrednost y koordinate, potrebno je ustanoviti koje su to stranice koje poseduju tačke sa tom y koordinatom i odrediti broj piksela između tih stranica, tačnije između tačaka sa istom y koordinatom koje pripadaju tim stranicama. Postupak prebrojavanja piksela se ponavlja sve dok se ne dostigne vrednost y koordinate za koju poligon nema definisano teme.

Zbog mogućnosti da je zadat konkavni poligon, za jednu vrednost y koordinate, moguće je da postoje više od dve stranice koje sadrže tačke sa tom y vrednošću. Zbog toga je potrebno prebrojati piksele između svake dve stranice.




Slika 7. Slika površine opisane listom temena.

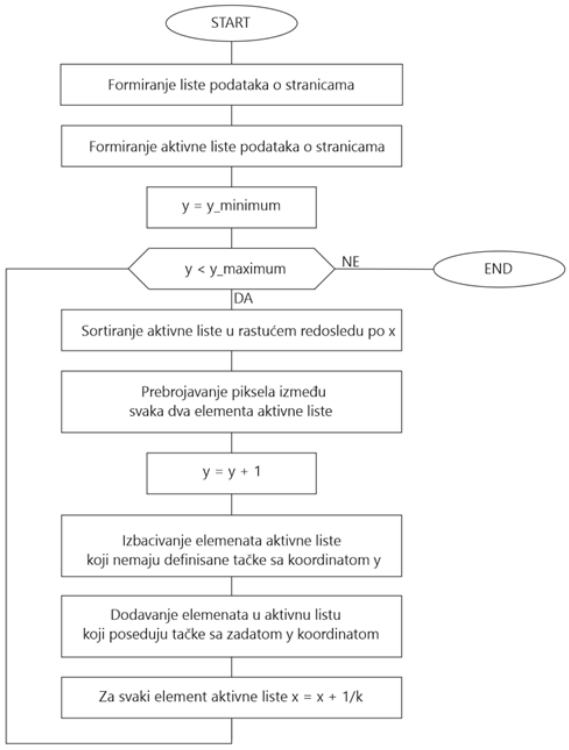
Kao što je rečeno, algoritam koristi podatke o stranicama koje dobija na osnovu temena poligona I smešta ih u listu podataka o stranicama. Za svaku stranicu pamte se:

- y_minimum – y koordinata najviše tačke stranice
- y_maximum – y koordinata najniže tačke stranice
- x – vrednost x koordinate za posmatranu y vrednost (na početku algoritma y = y_minimum)
- k – nagib stranice

Nagib stranice se koristi za izračunavanje x koordinate za $y_{(i+1)}$.

$$x_{i+1} = x_i + \frac{1}{k}$$

Određivanje površine poligona opisuje sledeći dijagram:




Slika 8. Dijagram algoritma za dređivanje povšine poligona zadatog listom temena.

## III. ZAKLJUČAK

Nakon što se iskoriste algoritmi za prepoznavanje markera i algoritam za određivanje povšine poligona jednostavnom proporcijom možemo izračunati fizičku površinu poligona:

$$FS_0 = FS_1 * \frac{LS_0}{LS_1}$$

- $FS_0$ – Fizička površina selektovanog poligona u $cm^2$.
- $FS_1$ – Fizička površina markera u $cm^2$
- $LS_0$ – Logička površina selektovanog poligona u pikselima
- $LS_1$ – Logička površina markera u pikselima

## LITERATURA

[1] Glyph recognition using OpenCV and Python. https://rdmilligan.wordpress.com/2015/07/19/glyph-recognition-using-opencv-and-python/
[2] Aforge.NET GRAFT http://www.aforgenet.com/projects/gratf/code_samples.html

# Uređaj za hvatanje grabljivica

Nemanja Radić

*Sadržaj* - Cilj predstavljenog uređaja je smanjenje ili sprečavanje štete po domaće životinje i poljoprivredna dobra od strane divljih životinja. Uređaj služi za humano i bezbedno hvatanje i premeštanje divljih životinja u njihova prirodna staništa ili prirodne rezervate. Princip rada uređaja baziran je na imitaciji živog mamca-zvuka lažnog plena koji se koristi kako bi se privukle i u kavez zarobile divlje životinje. Nakon zarobljavanja predatora uređaj gasi emitovani zvuk lažnog mamca i vlasniku šalje SMS poruku da je u zamci zarobljena divlja životinja. Primenom ovog kaveza smanjio bi se broj brutalno ubijenih divljih životinja, smanjio bi se broj domaćih životinja korišćenih u svrhu mamca kao kod konvencionalnih zamki, a takođe bi se omogućio human način za hvatanje divljih životinja zalutalih na poljoprivredna dobra.

## I. Uvod

Uređaj predstavlja spoj oblasti informatike, elektronike, telekomunikacija i zoologije. Svrha predstavljenog uređaja je humano hvatanje divljih životinja koje mogu da načine ogromnu štetu poljoprivrednim gazdinstvima, kako domaćim životinjama tako i poljoprivrednim kulturama. Za razliku od drugih, po zakonu često zabranjenih metoda, ovaj uređaj služi za hvatanje živih divljih životinja i njihovo udaljavanje u njihova prirodna staništa od prostora gde mogu načiniti štete. Rezultat primene ovog uređaja je i smanjenje broja ubijenih divljih životinja kako se ne bi prekinuo prirodni lanac ishrane. Upotreba uređaja održava prirodni balans i omogućava da se ispoštuju zakonski okviri vezani za dobrobit divljih životinja.

## II. Alati i metodika rada

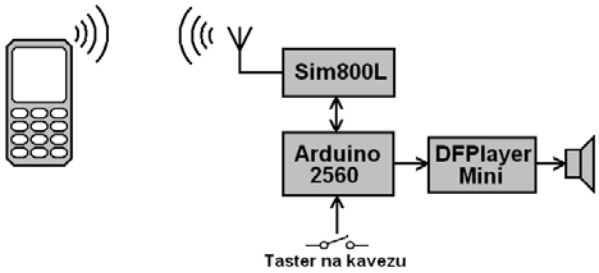
### *A. Hardver*

Fizički deo uređaja čine metalni kavez (Slika 1), zvučnik, MP3 plejer, GSM modul Sim800L, prekidački izvor napajanja, baterijsko napajanje 2x4,5V(3R12) i Arduino Mega 2560. Električna blok šema uređaja predstavljena je na slici 2. MP3 plejer emituje odgovarajući zvuk koji oponaša zvuk životinje koja bi se inače koristila kao mamac preko zvučnika. Kada divlja životinja, privučena zvucima lažnog mamca uđe u kavez, tada dođe do mehaničkog zatvaranja kaveza, zarobljavanja životinje i signaliziranja Arduino Mega 2560 da se u kavezu nalazi životinja. Arduino Mega 2560 prekida emitovanje zvuka i preko GSM modula šalje SMS poruku vlasniku da je u kavezu uhvaćena životinja.



Slika 1. Izgled kaveza.

Kavez ima ulogu da zarobi divlju životinju koja se našla u istom. U uređaju korišćen je GSM modul SIM800L (Slika 3.). Uloga GSM modula je komunikacija između uređaja i vlasnika. Preko SMS poruka obavlja se komunikacija između uređaja i vlasnika, kontrola rada uređaja i obaveštavanje vlasnika kada je uhvaćena životinja. GSM modul SIM800L komunicira sa mikrokontrolerom Arduino Mega 2560 preko serijskog interfejsa. Napajanje GSM modula ostvareno je sa DC/DC prekidačkim konvertorom 9/5 V. S obzirom na prirodu i mesto primene uređaja, napajanje uređaja je baterijsko. Za emitovanje zvuka korišćen je DFPlayermini koji u sebi ima i ugrađen audio pojačavač (slika 4.). Zvukovi koji se koriste za mamljenje životinja snimljeni su u mp3 formatu na memorijskoj SD kartici. Za detekciju uhvaćene divlje životinje koristi se mikrotaster koji signalizira kada se vrata metalnog kaveza zatvore. Za kontrolu rada svih hardverskih komponenti uređaja korišćen je mikrokontroler Arduino Mega 2560.

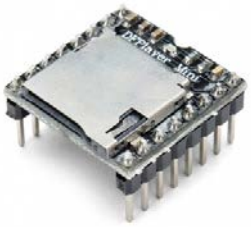



Slika 2. Blok šema uređaja.

Nemanja Radić, obrazovni profil elektrotehničar računara, Tehnička škola Loznica, Trg Jovana Cvijića 3, 15300 Loznica, Srbija, E-mail: nemanjaradic1999@gmail.com




Slika 3. GSM modula SIM800L.



Slika 4. Mp3 plejer DFPlayer Mini.

*B. Softver*

Za programiranje mikrokontrolera korišćen je besplatan program Arduino IDE [6] i programski jezik Arduino C/C++, koji ima mogućnost serijske komunikacije sa kontrolerom bez drugih dodatnih programa. U programiranju rada GSM modula korišćena je biblioteka <Sim800l.h> [4]. Program u mikrokontroleru vrši dekodovanje SMS poruke i na osnovu dekodovanog sadržaja reprodukuje zvuk ili menja reprodukovanu melodiju. Softver očitava stanje na mikrotasteru. Ako je mikrotaster pritisnut prekida se reprodukovanje zvuka i preko GSM modula šalje SMS poruku vlasniku da se u zamka aktivirana. Na slici 5. prikazan je deo programskog koda napisanog u Arduinu koji vrši čitanje primljene SMS poruke i reprodukcija zvuka.

```
textSms=Sim800l.readSms(1);

if (textSms.indexOf("OK")!=-1)
    {
    if (textSms.length() > 7)
        {

            numberSms=Sim800l.getNumberSms(1);
            textSms.toUpperCase();

            if (textSms.indexOf("PUSTI")!=-1){


                digitalWrite(pustanje,0);
                delay(100);
                digitalWrite(pustanje,1);
            }

            else if (textSms.indexOf("SLEDECA")!=-1){
                    digitalWrite(pustanje,LOW);
                    delay(100);
                    digitalWrite(pustanje,HIGH);

            }
```

Slika 5. Deo programa.

## III. Rezultati

S obzirom na mesto primene uređaja, napajanje uređaja je baterijsko. Životni vek baterije, odnosno vreme rada uređaja zavisi od kapaciteta baterije, odnosno akumulatora. U fazi testiranja uređaja merene su struje pražnjenja baterije u različitim režimima rada uređaja. U isključenom stanju struja baterije iznosila je 75 mA. Prilikom slanja i primanja SMS poruke struja baterije dostizala je vrednost 130 mA (na mestu gde je vršeno predstavljeno merenje GSM signal imao je dobar nivo). Struja baterije kada se uključi MP3 plejer zavisi od melodije koja se reprodukuje. Maksimalna izmerena vrednost struje baterije prilikom reprodukovanja melodije kreće se do 300 mA. Struja pražnjenja baterije merena je analognim ampermetrom. Prilikom izvšenih merenja struje napon baterije iznosio je 7,5 V. Napajanje uređaja ostvareno je sa dve baterije 2x4,5 V (3R12).

## IV. Zaključak

Efikasnost i humanost ovog uređaja u odnosu na druge metode hvatanja životinja je velika zbog upotrebe lažnog mamca korišćenjem MP3 plejera. S obzirom da je pokrivenost GSM signalom velika ovaj uređaj je primenjiv na skoro svakom terenu. Predstavljeni uređaj konstruisan je za male grabljivice ali se može prepraviti kako bi se hvatale i neke druge životinje ili velike grabljivice. Uređaj se takođe može upotrebiti i u hvatanju odbeglih životinja iz zoo-vrta. Iako uređaj lišava slobode divlje životinje na neki određen vremenski period, obezbeđuje im život u svom prirodnom staništu ili u rezervatu prirode.

## Zahvalnica

Posebnu zahvalnost pri izradi ovog projekta pridajem Marti Avramović, bivšoj učenici Tehničke škole u Loznici.

## Literatura

[1] https://cdnshop.adafruit.com/datasheets/sim800h_hardware_design_v1.00.pdf
[2] http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf
[3] http://www.picaxe.com/docs/spe033.pdf
[4] https://github.com/cristiansteib/Sim800l
[5] http://www.zzps.rs/novo/index.php?jezik=sr&strana=naslovna
[6] https://www.arduino.cc/en/Main/Software

# Sistem za detekciju pokreta kroz nepomične objekte

Luka Matović

*Sadržaj* - Cilj predstavljenog uređaja je detektovanje pokreta kroz neku prepreku (npr. kroz zid). Uređaj koristi senzor čiji je rad baziran je na Doplerovom efektu. Senzor emituje elektromagnetni talas frekvencije 10,5 GHz, prima reflektovani signal i množi ga sa frekvencijom predajnog signala. Ukoliko dođe do pomeranja objekta doći će do razlike između predajne i prijemne frekvencije a senzor će tu razliku frekvencija da detektuje. Mereći razliku frekvencija meri se brzina objekta. Zbog velike frekvencije elektromagnetnog talasa upotrebljeni senzor omogućava detektovanje pokreta i iza masivnih prepreka kao što su zid i sl.

## I. UVOD

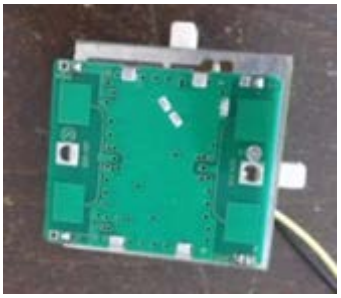
Uređaj je namenjen za detekcija pokreta kroz objekte. Osnovna uloga uređaja je detekcija pokreta i merenje brzine tih objekata. Senzor radi na principu Doplerovog efekta [1]. Ukoliko se objekat ne pomera, nema razlike između poslate i primljene frekvencije. Kada se objekat pomera javlja se razlika između poslate i primljene frekvencije. Što se objekat brže kreće veća je razlika frekvencija. Mereći razliku frekvencija moguće je odrediti brzinu objekta koji se kreće. Ova osobina Doplerovog efekta omogućava primenu u raznim oblastima kao što su: alarmni sistemi, saobraćaj itd. Radom uređaja upravlja mikrokontoler Arduino Mega 2560 [2] koji prikupljene i obrađene informacije prosleđuje računaru.

## II. ALATI I METODIKA RADA

### *A. Komponente sistema*

Uređaj čini senzor pokreta HB100 [3] (Slika 1), pojačavač signala [3] (Slika 2) i razvojno okruženje Arduino Mega 2560. Senzor pokreta HB100 emituje elektromagnetni talas frekvencije 10,5GHz, prima reflektovanu frekvenciju i množi ove dve frekvencije. Produkt množenja koji je od interesa je razlika ove dve frekvencije koja se dalje prosleđuje pojačavaču. Izlaz senzora HB100 daje napon male amplitude pa je upotrebljen pojačavač koji ima ulogu da poveća amplitudu signala kako bi ga Arduino Mega 2560 prihvatio. Pojačani signal od pojačavača Arduino Mega 2560 obrađuje i prosleđuje računaru. Pomoću programa (Slika 3) računar obaveštava korisnika o prikupljenim podacima, odnosno o izmerenoj brzini objekta.

Luka Matović, obrazovni profil elektrotehničar računara, Tehnička škola Loznica, Trg Jovana Cvijića 3, 15300 Loznica, Srbija, E-mail: lukamatovic45@gmail.com

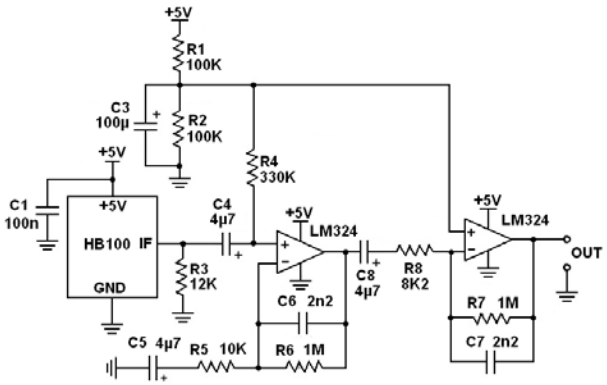


Slika 1. Senzor pokreta HB100.

Karakteristike senzora pokreta HB100 su:

- mala potrošnja struje (do 30 mA),
- napajanje +5V,
- ugrađen prijemnik, predajnik i množač signala,
- radna frekvencija 10,5 GHz (rad u nelicenciranom frekventnom opsegu),
- maksimalna snaga 20 dBm.

Za pojačavanje signala korišćen je pojačavački stepen sa operacionim pojačavačem LM324 [4] predstavljen na slici 2. Napajanje pojačavača je +5V.




Slika 2. Šema pojačavača [3].

### *B. Softver*

Za izradu ovog projekta korišćen je program Arduino IDE [5]. Program koji meri i ispisuje vrednost izmerene brzine prikazan je na slici 3.

```
int PINcitanje = 13;
long vreme0, vreme, vreme1;
double v, c, k;

void setup() {
Serial.begin(9600);
pinMode(PINcitanje, INPUT);
}

void loop() {
  vreme1 = pulseIn(PINcitanje, HIGH);
  vreme0 = pulseIn(PINcitanje, LOW);
  vreme=vreme0+vreme1;
  c=3*100000000;
  v=(-c/(2*10.5*1000*vreme));
  Serial.print("Brzina objekta \t");
  Serial.print(v);
  Serial.print(" m/s");
  Serial.println();
}
```

Slika 3. Program za merenje i ispis izmerene brzine objekta.

Formula upotrebljena u programu izvedena je na osnovu doplerovog efekta za izvor koji miruje i objekat koji se kreće uz aproksimaciju da je mala razlika između predajne i prijemne frekvencije (Formula 1).

$$V = \frac{c \cdot \Delta f}{2 f_o} \qquad \Delta f = f - f_0 \tag{1}$$

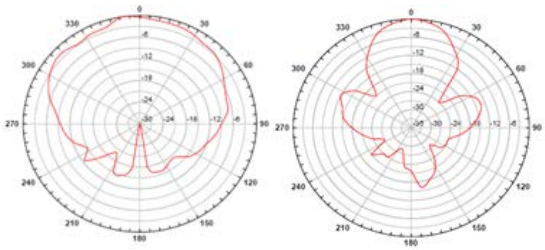
U prikazanoj formuli $\Delta f$ predstavlja razliku primljene $f$ i predajne frekvencije $f_0$. Brzina prostiranja elektromagnetnog talasa označena je sa $c$. Merenje prijemne frekvencije obavlja se posredno, merenjem trajanja periode signala. Vremensko trajanje signala koje da je funkcija *pulseIn* izraženo je u mikrosekundama.

## III. Rezultati

Amplituda signala na izlazu HB100 senzora reda je veličine 1 – 2 mV (Slika 4). Tokom testiranja rada uređaja merene su brzine kretanja čoveka koji se kreće ispred senzora. Izmerene brzine kreću se u opsegu od 1m/s do 5m/s. Maksimalno rastojanje pri kom senzor detektuje pokret je 5m. Kada se ispred senzora nađe prepreka rastojanje pri kom senzor detektuje pokret se smanjuje. U toku testiranja senzora korišćena je prepreka (drvena iverica debljine 15mm). Sa datom preprekom rastojanje pri kom senzor detektuje kretanje iznosilo je 1m. Karakteristike senzora (Slika 5) omogućava široki ugao detekcije pokreta ali ne i merenje brzine u tačno definisanom pravcu.




Slika 4. Izgled signala na izlazu senzora HB100.



Slika 5. Dijagram zračenja senzora HB100 po azimutu (slika levo) i elevaciji (slika desno) [3].

## IV. Zaključak

HB100 je jeftin senzor koji omogućava merenje različitih fizičkih veličina. Može se koristiti za detektovanje pokreta i u određenim uslovima kao merač brzine i sl. Zbog svoje radne frekvencije dobija na superiornosti u odnosu na druge detektore pokreta ili senzora brzine jer visoka frekvencija elektromagnetnog talasa lako prodire kroz prepreke kao što je zid i slično.

## Zahvalnica

Za pomoć pri izradi ovog projekta, podršku i stručnu pomoć, zahvalio bih se profesoru Tomislavu Mijiću i studentkinji Marti Avramović.

## Literatura

[1] https://en.wikipedia.org/wiki/Doppler_effect
[2] http://www.robotshop.com/media/files/pdf/arduinomega2560datasheet.pdf “Arduino MEGA Datasheet”
[3] http://www.kh-gps.de/HB100_Microwave_Sensor_Module_Datasheet.pdf “HB100 Datasheet”
[4] http://pdf1.alldatasheet.com/datasheet-pdf/view/17871/PHILIPS/LM324.html “LM324 Datasheet”
[5] https://www.arduino.cc/en/Main/Software

# Koncept Pametne kuće kroz nadgledanje ambijentalnih parametara i zaštitu kuće

Viktor Petrović

*Sadržaj* - U radu je opisan koncept Pametne Kuće i njen značaj za nadgledanje ambijentalnih parametara poput temperature i vlažnosti vazduha u zatvorenom prostoru, kao i važnost zaštite kuće od uljeza i havarija. Pored toga opisan je spoj Pametne kuće i Internet of Things platforme.

## I. UVOD

Dati tačnu definiciju Pametne kuće je jako teško, jer se radi o pojmu koji se svakog dana sve više razvija i samim tim dobija novo značenje. Uopšteno govoreći, Pametna kuća predstavlja nadgledanje i upravljanje sistemima u kući, poput sigurnosnog sistema, sistema za grejanje i hlađenje kao i sistema rasvete, ali i mnogobrojnim kućnim uređajima/aparatima. Zapravo, Pametna kuća povezuje sve ove sisteme i uređaje u jedan sistem, omogućavajući im međusobnu komunikaciju kao i komunikaciju same kuće sa vlasnicima. Dakle, korisnik ima pristup svim informacijama vezanim za kuću gde god da se nalazi u svetu.

Moderne Pametne kuće imaju visok nivo automatizacije i mogućnost učenja, „posmatrajući“ navike svojih vlasnika mogu predvideti različite situacije, pa pored toga što štede dragoceno vreme, pružaju i dodatni komfor i sigurnost. Pametne kuće postoje kako bi olakšale život savremenom čoveku koji živi mnogo brže.

## II. DELOVI PAMETNE KUĆE

Gledano sa tehničke strane Pametna kuća se sastoji iz pet blokova (Sl. 1), a to su:

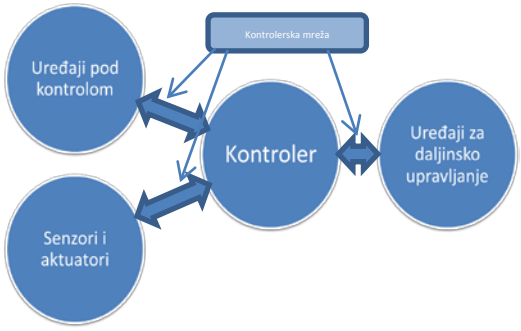
- Uređaji pod kontrolom - objekti upravljanja,
- Senzori i aktuatori,
- Kontroler,
- Uređaji za daljinsko upravljanje i
- Kontrolerska mreža [1].

Objekti upravljanja su kućni uređaji i sistemi, poput sigurnosnog sistema, sistema rasvete, sistema za grejanje hlađenje i ventilaciju. Ovi uređaji ili imaju ugrađenu mogućnost komunikacije sa kontrolerom ili im se ta mogućnost naknadno dodaje. Senzori nam omogućavaju povratnu spregu, odnosno omogućavaju da u svakom trenutku kontroler i korisnik imaju uvid u stanje kuće. Pomoću aktuatora kontroler sprovodi određene akcije nad objektima upravljanja. Aktuatori mogu biti električni i mehanički. Mehanički su različite pumpe ili motori, dok su elekrični aktuatori uređaji poput prekidača ili regulatora.

Kontroler se nalazi u samom središtu sistema, prikuplja informacije sa senzora i komande od uređaja za daljinsko upravljanje. Zadatak mu je da na osnovu predefinisanih pravila izvodi određene akcije pomoću aktuatora nad objektima upravljanja u kući. Takođe, potrebno je obaveštavati vlasnike o trenutnom stanju u kući. Karakteristično za ove kontrolere je da se nikad ne gase i da imaju beskonačne petlje.

Uređaji za daljinsko upravljanje predstavljaju sponu između vlasnika kuće i same kuće. Ulogu ovih uređaja nose aplikacije na pametnim telefonima ili tablet računarima. Ovi uređaji su uvek pri ruci, pa je komunikacija sa Pametnom kućom putem aplikacije značajno olakšana, takođe ove aplikacije se jednostavno održavaju i nadograđuju.

Kontrolerska mreža povezuje sve delove u jednu celinu omogućavajući im međusobnu komunikaciju. Postoje tri načina komunikacije: komunikacija preko energetske mreže, žičana komunikacija i bežična komunikacija. Često se načini komunikacije između kontrolera i korisnika i kontrolera i ostatka sistema razlikuju. Komunikacija sa korinsikom se odvija preko Bluetooth-a ili preko WiFi modula, a ponekad i preko GSM modula [1].




Slika 1. Delovi Pametne kuće.

Viktor Petrović, Katedra za automatiku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: viktor.petrovic@elfak.rs

## III. AMBIJENTALNI PARAMETRI I ZAŠTITA KUĆE

### *A. Važnost temperature i vlažnosti vazduha u zatvorenom prostoru*

Kvalitet vazduha u zatvorenom prostoru je jako važan, ukoliko padne ispod određenog nivoa može štetno uticati na ljude koji se u njemu nalaze. Većina ljudi kao uzrok lošeg kvaliteta vazduha vidi polen, prašinu i druge zagađivače, zanemarujući uloge koje imaju temperatura i vlažnost vazduha. Bez obzira na tip klime u kojoj živimo, nivo vlažnosti vazduha može uticati na osećaj udobnosti u kući. Podešavanje idealne vlažnosti u zatvorenom prostoru mora biti prioritet u svakom domu. Održavanje pravog nivoa vlažnosti vazduha u kući pomaže da se zimi osećamo toplije, a tokom leta hladnije.

Nizak nivo vlažnosti vazduha je jedan od vodećih uzroka za razvoj oboljenja u zatvorenom prostoru. Osim toga što ima uticaj na zdravlje ljudi i sama kuća može postati neugodno mesto za boravak, jer vlaga može imati negativne uticaje na kuću i nameštaj u njoj.

Optimalni nivo vlažnosti vazduha u kući zavisi, pre svega, od naših ličnih afiniteta, od odeće koju nosimo, ali i nivoa fizičke aktivnosti. Organizacija ASHRAE preporučuje, kao optimalni opseg relativne vlažnosti vazduha od 45% do 55% tj. kao najbolji za zdravlje, čak se mogu uzeti i nešto šire granice od 40 do 60%. U ovom opsegu imamo najmanji nivo pretnji od respiratornih infekcija, alergijskog rinitisa i astme, pored toga, ovaj opseg najmanje pogoduje razvoju bakterija, virusa, gljivica i grinja.

Kao što je to slučaj sa relativnom vlažnošću, tako i „savršena temperatura“ zavisi od mnogo faktora, te njeno podešavanje može biti jako teško, jer je ona individualna i zavisi od navika, načina odevanja, ali i fizičkih aktivnosti individue. Najveći broj istraživanja vezan za temperaturu u zatvorenom prostoru je posvećen kancelariji, dnevnoj i spavaćoj sobi, što je i logično jer upravo u ovim prostorijama provodimo najviše vremena. Najbolja temperatura za radni prostor i za dnevni boravak je oko 20 °C. Naravno, pored podešavanja temperature ne smemo zaboraviti na relativnu vlažnost vazduha, kao i na trenutno godišnje doba koje dosta utiče na ove parametre. Za spavaću sobu se temperatura kreće od 16 do 18℃, jer je ta temperatura pogodna za zdravlje, ubrzava metabolizam i tokom noći se troši više kalorija [2].

### *B. Zaštita kuće*

Kada je sigurnost kuće u pitanju potrebno je zaštititi kuću od lopova, ali i od havaraija koje mogu izazvati materijalnu štetu ili u najgorem slučaju mogu ugroziti život čoveka. Da bi se zaštitili od lopova koristimo uređaje poput senzora pokreta, kamera, senzora vrata i prozora, senzora lomljenja stakla itd.

Sa druge strane uređaji i instalacije u kući su podložne kvarovima, vlasnici treba da imaju sreće da se nađu u kući u trenutku kada dođe do kvara. Ovde je potrebno detektovati izlivanje vode (poplavu) do kog može doći usled kvara na instalacijama ili nekom uređaju koji je priključen na vodu. Takođe, može doći do curenja zapaljivih gasova pa je ovde potreban senzor zapaljivih gasova, osim toga potreban je i senzor dima i vatre, da bi se detektovao požar. Postoje i toksični gaosvi poput ugljen monoksida koji nema ni ukus ni miris ni boju, pa jedino uz pomoć senzora može da se detektuje [3].

## IV. PAMETNA KUĆA I INTERNET OF THNINGS

Internet Stvari (eng. Internet of Things - IoT) predstavljaju novi trend umrežavanja fizičkih objekata, vozila, zgrada i drugih *embedded* elektronskih uređaja, softvera, senzora i aktuatora što omogućava ovim objektima da prikupljaju i razmenjuju podatke. IoT omogućava da se objekat u Pametnoj kući nadgleda ili kontroliše iz daljine, preko postojeće mreže, čime se dozvoljava integracija fizičkog sveta u sisteme zasnovane na računarima, ovo doprinosi većoj efikasnosti, tačnosti, ali i ekonomskoj dobiti i to uz smanjenu ljudsku intervenciju.

IoT gura napred razvoj pametnih kuća, ali se ne zaustavlja samo na tome, već teži da olakša svaki aspekt života, pa se pored koncepta Pametne kuće javlja i koncept Pametnih gradova i on ima za cilj da gradski život prilagodi našim digitalnim navikama, ali i da zastareloj gradskoj infrastrukturi da nove funkcije.

Većinu IoT sistema možemo opisati na sledeći način. Sa jedne strane imamo pametne uređaje – stvari (things) ovi uređaji prikupljaju podatke sa različitih uređaja. U sredini ovog sistema imamo *cloud* gde se podaci sa mnogih izvora prikupljaju i analiziraju u realnom vremenu i to putem IoT platforme za analizu, projektovane specijalno za ovu namenu, jedna od tih platformi je ThingSpeak. Sa druge strane, imamo inženjera ili naučnika koji ima uvid u podatke koji su prikupljeni, u ovom slučaju podaci se sa platforme povlače u softver na desktop računaru. MATLAB i Simulink proizvodi podržavaju IoT sisteme tako što pomažu pri razvoju i testiranju pametnih uređaja, pružaju pristup i prostor za prikupljanje podataka na cloud-u i analizu tih podataka.

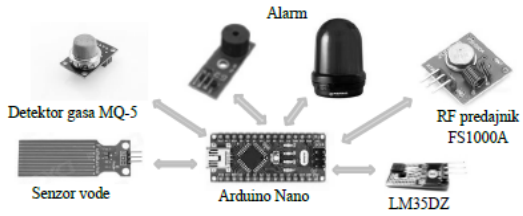
ThingSpeak je IoT platforma za analizu koja omogućava prikupljanje, vizuelizaciju i analizu tokova podataka na *cloud*-u. Kada se podatak pošalje na ThingSpeak sa nekog uređaja, može se napraviti trenutna vizuelizacija tog podatka preko dijagrama, a takođe postoji mogućnost slanja upozorenja na neki web servis poput Twitter-a.

## V. REALIZACIJA MODELA PAMETNE KUĆE

Pametna kuća realizovana u ovom radu se funkcionalno sastoji iz 4 zasebna dela, jedan je kontroler „Kuhinja“ sa senzorima i aktuatorima, drugi je glavni kontoroler „Dnevna soba“ takođe sa senzorima i aktuatorima, ova Pametna kuća potom komunicira preko

IoT platforme ThingSpeak, koja predstavlja treći deo i preko nje sa korisnikom (četvrti deo), gde on može imati uvid u stanje svoje Pametne kuće.

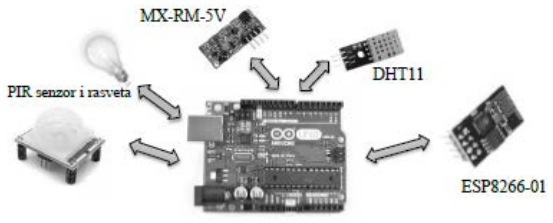
Prvi deo je Kuhinja (Slika 2) i ovde se kao kontroler koristi Arduino Nano, a od senzora imamo integrisani senzor temperature LM35DZ, poluprovodnički senzor prisustva zapaljivih gasova MQ-5 i detektor poplave. Takođe, ovde imamo i zvučni i svetlosni alarm, s tim što je kao zvučni alarm upotrebljena zujalica (buzzer), a kao svetlosni indikator LED dioda. Alarm se aktivira ukoliko dođe do poplave ili se detektuje curenje zapaljivih gasova, dovoljno je da se detektuje jedno od ova dva stanja da bi se alarm aktivirao. Za komunikaciju sa Dnevnom sobom odnosno sa glavnim kontrolerom koristi se jednosmerna RF veza koja radi na 433MHz (u Kuhinji se nalazi predajnik, a u Dnevnoj sobi prijemnik).




Slika 2. Funkcionalna šema Kuhinje.

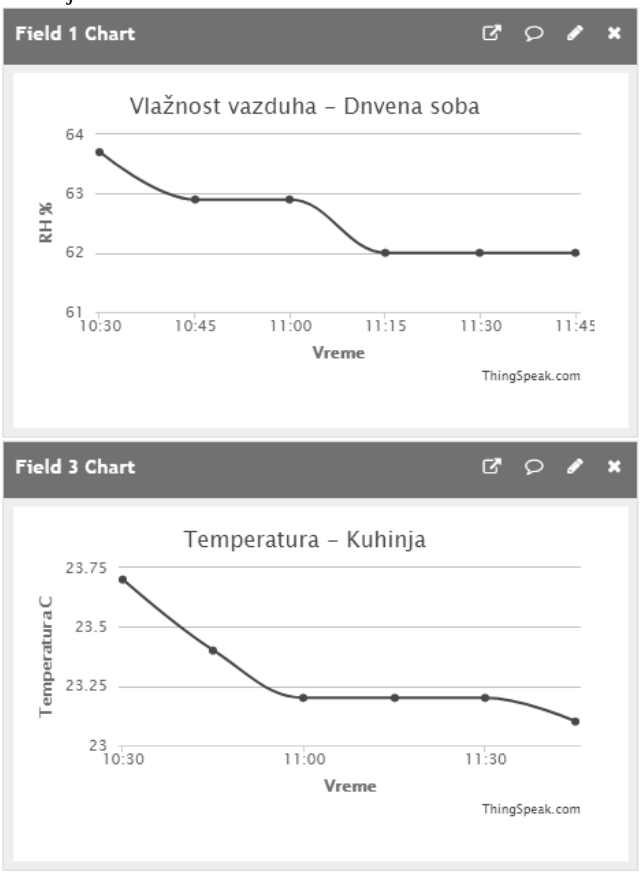
U drugom delu sistema imamo Dnevnu sobu u kojoj ulogu kontrolera ima Arduino Uno (Slika 3). Ovaj kontroler sa jedne strane prima podatke iz Kuhinje i to temperaturu i alarmna stanja (detekcija poplave i prisustva zapaljivih gasova), a sa druge strane na njega su povezani senzor temperature i relativne vlažnosti vazduha DHT11, PIR senzor pokreta i fotootpornik kao detektor svetlosti. Pomoću senzora pokreta pali se svetlo na ulazu u samu kuću, dok detektor svetlosti omogućava da se svetlo pali samo kada je noć.

Da bismo uspostavili vezu sa IoT platformom potrebna nam je veza sa internetom, ovde koristimo WiFi modul ESP8266-01. Preko ovog modula šaljemo temperaturu i relativnu vlažnost vazduha Dnevne sobe, potom temperaturu Kuhinje i dva alarmna stanja.




Slika 3. Funkcionalna šema Dnevne sobe.

Slanje podataka na IoT platformu ThnigSpeak obavljamo na svakih 20 sekundi, ovi podaci se upisuju na prethodno kreirani kanal koji ima pet polja za praćenje naših promenljivih. Izgled nekih promenljivih je dat na Slici 4. Preko ovih polja imamo mogućnost da pratimo kako se kreću promenljive, možemo vrlo lako i efikasno da steknemo uvid u to šta se dešava u kući i kada mi nismo u njoj i da blagovremeno reagujemo ako dođe do neke havarije.




Slika 4. Izgled polja na ThingSpeak kanalu.

## VI. ZAKLJUČAK

Ovaj rad omogućava da se dobije početna slika o Pametnim kućama i njenim mogućnostima, kao i o platformi za IoT ThingSpeak.

## ZAHVALNOST

Posebnu zahvalnost dugujem prof. dr Draganu B. Živanoviću na pomoći prilikom izrade ovog rada.

## LITERATURA

[1] Kyas, O. (2013). How To Smart Home. Nemačka: Key Concept Press e.K.
[2] Web-site: https://www.griffithenergyservices.com/article/the-importance-of-balancing-your-homes-humidity
[3] Elsenpeter, R., & Velte, T. (2003). Build Your Own Smart Home. McGraw-Hill
[4] Rangelov, N. (2016). Kratak uvod u Internet stvari – gde smo, kuda idemo i kako da oblikujemo budućnost uz IoT.

# Kolaborativni sistemi za preporučivanje sadržaja i primena NHSM metrike za određivanje sličnosti

Aleksandar Rajić

*Sadržaj* – U ovom radu biće dat pregled sistema za davanje preporuka, biće ukratno obrađene neke od popularnih metoda koje se danas koriste u praksi kao i detaljnije obrađena „novel heuristic similarity measure“. Takođe biće ukratko pomenute i neke od popularnih Web tehnologija, kojima se u praksi ovi sistemi mogu implemtirati. Cilj projekta je praktična implementacija navedene metode i njena upotrena upotreba u aplikaciji za personalizovanu pretragu ugostiteljskih objekata,

## I. UVOD

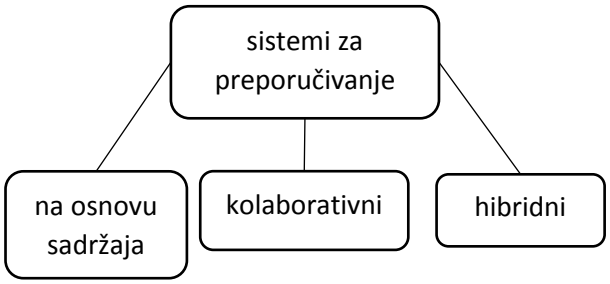
Sa razvojem Interneta izgled stranica se rapidno menjao i vremenom dodavane su nove mogućnosti što je povećalo količinu informacija prikupljenih od korisnika. Takođe, različiti tipovi korisnika zahtevali su personalizovanu prezentaciju sadržaja što je dovelo do razvoja tzv. sistema za davanje preporuka (eng. recommendation system[1]). Ovo zahteva praćenje aktivnosti korisnika, kao i korišćenje njegovih ličnih informacija, kako bi se filtrirali drugi korisnici i pronašli slični. Izdvajanjem sličnih korisnika dobija se informacija koji bi sadržaj mogao biti pogodan za datog korisnika. Sa druge strane moguće je davati preporuke i na osnovu samog sadržaja, bez ikakvih informacija o korisniku. Ova dva načina, na osnovu istorije sličnih korisnika (eng. collaborative filtering) i na osnovu sadržaja (eng. content-based filtering) predstavljaju osnovne pristupe u rešavanju problema preporuke sadržaja, ali su u praksi uglavnom u upotrebi hibridna rešenja. U ovom radu akcenat je na davanju preporuka na osnovu istorije sličnih korisnika i heurističkim metodama za određivanje koeficijenata sličnosti korisnika, na osnovu kojih se kasnije izdvajaju najsličniji korisnici i dalje vrši preporuka sadržaja, upoređivanjem istorija aktivnosti. Na početku, dat je pregled nekih od najpopularnijih metoda, upoređene su njihove karakteristike i naglašene loše strane svake. Kako je akcenat na NHSM[2] metodi, u posebnom poglavlju data je njena matematička formulacija, a zatim u nastavku rada i praktična implementacija modula za preporučivanje sa korišćenjem navedene metode. Na kraju ovaj modul je iskorišćen u aplikaciji za personalizovanu pretragu ugostiteljskih objekata.

Aleksandar Rajić je student na smeru Računarstvo i Informatika, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: aleksandar.v.rajic@gmail.com

## II. SISTEMI ZA PREPORUČIVANJE

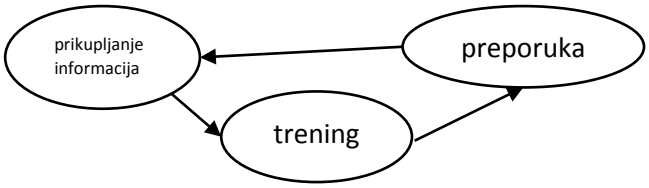
Sistem za davanje preporuka, nastoji da na osnovu prosleđenih parametara filtrira postojeći materijal predloži korisniku najpodobnije za njegove ličnost, što zahteva prikupljanje informacija o korisnicima praćenjem njihove aktivnosti, kreiranje različitih tipova korisnika, kao i korišćenje njigovih ličnih informacija. Davanjem preporuka povećava se efikasnost same Web aplikacije, ali i zadovoljnost korisnika njome. Jedna od danas najpoznatijih kompanija, Facebook, koristi ovaj sistem kako bi svojim korisnicima predložila ono što bi ih moglo najviše zanimati. O efikasnosti sistema za preporučivanje najviše govori činjenica da skoro sve napoznatije društvene mreže i aplikacije uspešno ih koriste.

Postoje dve grupe sistema za davanje preporuka: davanje preporuka na osnovu sadržaja (eng. content-based filtering[3]) i davanje preporuka na osnovu istorije sličnih korisnika (eng. collaborative filtering[1]). Prva se fokusira sam sadržaj pri tom uzimajući u obzir povratne informacije o prethodnim sadržajima koje je korisnik konzumirao. Dakle, ovde se javlja tzv. problem hladnog starta (eng. cold start problem) jer na početku korišćenja same aplikacije korisnik nije dao nikakvu povratnu informaciju, ili bar nije dao dovoljno povratnih informacija, da bi preporuka bila preciznija. Takođe, tu su još i problem ograničenja prilikom analize sdržaja i heterogenost podataka. Druga grupa, osim samog sadržaja i korisnikovih ocena, uzima u obzir i aktivnosti i preference drugih tzv. sličnih korisnika. Ovaj pristup se više koristi i najveći problem je identifikovanje sličnih korisnika za šta su razvijene različite metode. Kao i preporučivanje na osnovu sadržaja i preporučivanje na osnovu aktivnosti sličnih korisnika pati od problema hladnog starta, heterogenosti podataka i skalabilnosti. Postojanje ovih problema u mnogome smanjuje performanse samih sistema pa se u praksi obično koriste neke hibridne varijante sa karakteristikama obe grupe. Na slici 1 predstavljena je klasifikacija sistema za davanje preporuka, a u nastavku će biti nešto detaljnije objašnjene obe grupe sistema za preporučivanje.




Slika 1. Klasifikacija sistema za davanje preporuka.

Slika 2. prikazuje sekvencu faza u procesu davanja preporuka.




Slika 2. Faze u procesu davanja preporuka.

Na početku svakog procesa potrebno je prikupiti informacije od korisnika, kao što su lične informacije korisnika, njegovo ponašanje u aplikaciji ili materijal koji je posećivao, kako bi se generisao profil korisnika ili model predikcije. Količina informacija koju sistem sakupi o korisniku direktno je povezana sa stepenom preciznosti preporuke. Eksplicitno prikupljanje povratnih informacija podrazumeva da korisnik direktno odgovori na postavljena pitanja, napiše svoje mišljenje o materijalu ili da ocenu u okviru neke skale. Implicitno prikupljanje povratnih informacija podrazumeva prikupljanje informacija od strane sistema o aktivnosti korisnika u samoj aplikaciji. Hibridno prikupljanje informacija predstavlja kombinaciju prethodnih metoda. Sledeća faza u procesu preporučivanja je faza učenja kada sistem različitim algoritmima analizira prikupljene informacije i kreira profil korisnika ili model predikcije, koji će biti korišćeni u narednoj fazi. Krajnja faza jeste davanje preporuka, tj. predikcija na osnovu profila korisnika ili modela kreiranih u fazi učenja. U ovoj fazi se uzimaju u obzir i profili sličnih korisnika ukoliko je u pitanju kolaborativna metoda.

## III. METODE ZA ODREĐIVANJE KOEFICIJENTA SLIČNOSTI

U prethodnom poglavlju govoreno je o različitim metodama preporučivanja i prilikom obrade kolaborativnih metoda pomenuto je izračunavanje sličnosti izmeću korisnika ili materijala. U ovom poglavlju biće pomenute neke od funkcija sličnosti koje se koriste u praksi.

Najšire korišćene funkcije sličnosti su Pirsonov korelacioni koeficijent (eng. Pearson Correlation Coefficient (PCC)) i kosinusna sličnost (eng. cosine similarity (COS)) i definisane su sledećim formulama, respektivno:

$$sim(u,v)^{PCC} = \frac{\sum_{p\in I}(r_{u,p}-\bar{r}_u)(r_{u,p}-\bar{r}_v)}{\sqrt{\sum_{p\in I}(r_{u,p}-\bar{r}_u)^2}*\sqrt{\sum_{p\in I}(r_{u,p}-\bar{r}_v)^2}} \quad (1)$$

$$sim(u,v)^{COS} = \frac{\overrightarrow{r_u}*\overrightarrow{r_v}}{\|\overrightarrow{r_u}\|*\|\overrightarrow{r_v}\|} \quad (2)$$

gde su:

- u,v – korisnici,
- I – skup materijala,
- r – materijal iz skupa I čija se ocena koristi,
- ru,rv – prosečne vrednosti ocene korisnika u i v, respektivno,
- ru,p, rv,p – ocene korisnika u i v materijala p, respektivno.

Kako u obema funkcijama postoje neke loše osobine i mnogo različitih unapređenih verzija je predloženo. Da bi se eliminisao uticaj pozitivnih i negativnih ocena, jer je potrebno da sve ocene budu apsolutne vrednosti uvedena je metrika ograničeni Pirsonov korelacioni koeficijent (eng. constrained Pearson correlation coefficient (CPCC)) dat sledećom formulom:

$$sim(u,v)^{CPCC} = \frac{\sum_{p\in I}(r_{u,p}-r_{med})(r_{u,p}-r_{med})}{\sqrt{\sum_{p\in I}(r_{u,p}-r_{med})^2}*\sqrt{\sum_{p\in I}(r_{u,p}-r_{med})^2}} \quad (3)$$

gde je $r_{med}$ srednja vrednost na skali ocena. Npr. ukoliko je skala od 1 do 5 rmed je 3, ukoliko je skala od 1 do 7 $r_{med}$ je 4.

Ukoliko su oba korisnika ocenila materijal metrika je imati više kredibiliteta. Zbog toga su predložene sledeće dve verzije PCC metrike, weighted Pearson correlation coefficient (WPCC) i sigmoid Pearson correlation coefficient (SPCC), date sledećim formulama:

$$sim(u,v)^{WPCC} = \begin{cases} sim(u,v)^{PCC} * \frac{|I|}{H}, |I| \le H \\ sim(u,v)^{PCC}, otherwise \end{cases} \quad (4)$$

$$sim(u,v)^{SPCC} = sim(u,v)^{PCC} * \frac{1}{1+\exp(-\frac{|I|}{2})} \quad (5)$$

gde je H eksperimentalna vrednost.

Da bi se uzela u obzir tendencija korisnika da ocenjuje čak i materijale koji su mu se dopali, nižim ocenama, ili obratno, sve materijale visokim ocenama uvedena je podešena kosinusna metrika (eng. adjusted cosine measure (ACOS)), data sledećom formulom:

$$sim(u,v)^{ACOS} = \frac{\sum_{p\in I}(r_{u,p}-\bar{r}_u)(r_{u,p}-\bar{r}_v)}{\sqrt{\sum_{p\in I}(r_{u,p}-\bar{r}_u)^2}*\sqrt{\sum_{p\in I}(r_{u,p}-\bar{r}_v)^2}} \quad (6)$$

gde je P skup svih materijala.

Dve takođe često korišćene metrike su Jaccard i mean squared difference (MSD), date sledećim formulama, respektivno:

$$sim(u,v)^{Jaccard} = \frac{|I_u| \cap |I_v|}{|I_u| \cup |I_v|} \quad (7)$$

$$sim(u,v)^{MSD} = 1 - \frac{\sum_{p \in I}(r_{u,p} - r_{v,p})^2}{|I|} \quad (8)$$

gde je I skup svih materijala.

Jaccard uzima u obzir samo broj ocena materijala koje su oba korisnika ocenila, ali ne i apsolutne vrednosti ocena. Sa druge strane MSD uzima u obzir samo apsolutne vrednosti ocena. Ove dve metrike mogu biti kombinovane u jednu novu metriku datu formulom:

$$sim(u,v)^{JMSD} = sim(u,v)^{Jaccard} * sim(u,v)^{MSD} \quad (9)$$

Loše strane navedenih metrika [2] su sledeće:

- Niska vrednost koeficijenta sličnosti uprkos sličnim ocenama. SPCC i ACOS i dalje pate od ovog problema, dok je kod CPCC izražen nešto manje.
- Visoki stepen sličnosti uprkos razlikama u ocenama korisnika. Ovaj problem je izražen u SPCC, dok je rešen u ACOS.
- Proporcije zajedničkih ocena vode do malog stepena tačnosti predviđanja. MSD pati od ovog problema jer izračunava srednju vrednost sličnosti dva korisnika a ne uzima odnos broja sličnih ocena i ukupnog broja ocena.
- Odbacivanje apsolutnih vrednosti ocena vodi do poteškoća u razlikovanju preroznavanju korisnika. Jaccard pati od ovog problema, uzima u obzir broj ocena istih materijala i računa odnos sa ukupnim brojem ocena.
- Kombinacija MSD-a i Jaccard-a samo parcijalno rešava probleme, ali ne povećava u mnogome tačnost preporuke. Štaviše, različiti korisniki sa različitim stepenom sličnosti mogu korišćenjem ove metrike dobiti relativno jednake koeficijente sličnosti sa datim korisnikom.

## IV. NOVEL HEURISTIC METRIKA

Relativno nova heuristička metrika sličnosti (eng. novel heuristic similarity measure (NHSM)) predložena u radu [2] nastoji da prevaziđe navedene probleme i kako eksperimentalni rezultati pokazuju, prikazani u istom radu, daje veoma dobre rezultate. Kako je ova metrika iskorišćena u praktičnom delu ovog rada posebno poglavlje posvećeno je njoj.

Na početku potrebno je definisati tri faktora koja utiču na izračunavanje koeficijenta sličnosti dva korisnika. To su Proximity, Significance i Singularity. Takođe, potrebno je naglasiti da ova metoda kažnjava „loše“ ili „lažne“ sličnosti, dok sa druge strane favorizuje prave sličnosti. Prvi korak u procesu izračunavanja koeficijenta sličnosti jeste izračunavanje vrednosti navedena tri faktora, što se čini sledećim formulama:

$$Proximity(r_{u,p}, r_{v,p}) = 1 - \frac{1}{1+\exp(-|r_{u,p} - r_{v,p}|)} \quad (10)$$

$$Significance(r_{u,p}, r_{v,p}) = \frac{1}{1+\exp(-|r_{u,p} - r_{med}| * |r_{u,p} - r_{med}|)} \quad (11)$$

$$Singularity(r_{u,p}, r_{v,p}) = 1 - \frac{1}{1+\exp\left(-\left|\frac{r_{u,p}+r_{v,p}}{2} - \mu_p\right|\right)} \quad (12)$$

gde je:

- rmed – srednja vrednost skale ocena,
- μp – srednja ocena materijala p korisnika u.

Nakon toga računa se njihov proizvod kako bi se dobio PSS koeficijent sličnosti korisnika u i v za materijal p.

$$PSS(r_{u,p}, r_{v,p}) = Proximity(r_{u,p}, r_{v,p}) * Significance(r_{u,p}, r_{v,p}) * Singularity(r_{u,p}, r_{v,p}) \quad (13)$$

Ukupni koeficijent sličnosti dva korisnika dobija se sumom pojedinačnih koeficijenata sličnosti korisnika za materijale pi iz skupa I.

$$sim(u,v)^{PSS} = \sum_{p \in I} PSS(r_{u,p}, r_{v,p}) \quad (14)$$

Odnos zajedničkih ocena i ukupnog broja ocena je vrlo bitan faktor u ovoj metrici i predstavlja sledeći korak u izračunavanju koeficijenta sličnosti. Za to se koristi modifikovana Jaccard-ova metrika, kako bi se kaznio mali broj zajedničkih ocena.

$$sim(u,v)^{Jaccard`} = \frac{|I_u \cap I_v|}{|I_u| \times |I_v|} \quad (15)$$

Kombinacijom dve prethodno dobijene vrednosti dobija se parcijalni rezultat koji će se koristiti u krajnjem izrazu.

$$sim(u,v)^{JPSS} = sim(u,v)^{PSS} * sim(u,v)^{Jaccard`} \quad (16)$$

Poslednji parcijalni rezultat koji moramo izračunati je tendencija samog korisnika, ka nižim ili višim ocenama. To se računa pomoću formule:

$$sim(u,v)^{URP} = 1 - \frac{1}{1+exp(-|\mu_u - \mu_v| * |\sigma_u - \sigma_v|)} \quad (17)$$

gde su:

- μu i μv – mean rating korisnika u i v, respektivno i dobija se formulom (18)
- σu i σv – standardne varijanse korisnika u и v, respektivno i dobijaju se formulom (19)

$$\mu_u = \sum_{p \in I_u} \frac{r_{u,p}}{|I_u|} \quad (18) \qquad \sigma_u = \sqrt{\sum_{p \in I_u} \frac{(r_{u,p} - \bar{r}_u)^2}{|I_u|}} \quad (19)$$

Na kraju kombinovanjem parcijalnih rezultata dobija se krajnja vrednost koeficijenta sličnosti korišćenjem NHSM metode, u segmentu od 0 do 1:

$$sim(u,v)^{NHSM} = sim(u,v)^{JPSS} * sim(u,v)^{URP} \quad (20)$$

## V. WEB APLIKACIJA ZA PRETRAGU UGOSTITELJSKIH OBJEKATA

### A. *Arhitektura aplikacije*

Slika 3 predstavlja implementaciju MVC (Model-View-Controller) sa komponentama navedenim u nastavku:

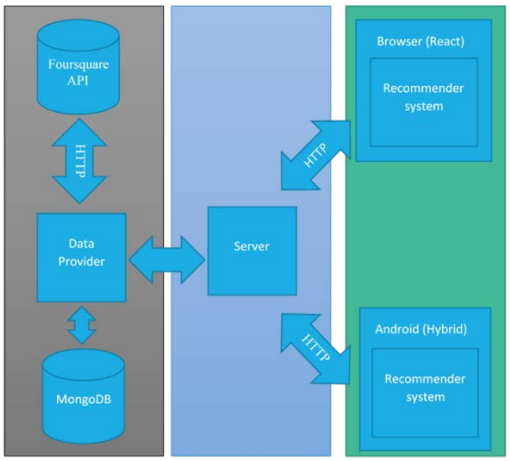
- Model:
  - NoSQL MongoDB baza podataka koja je u direktnoj komunikaciji sa Data Provider modulom
  - Data Provider modul koji komunicira sa Foursquare bazom podataka preko Foursquare API-ja i koji je u komunikaciji sa lokalnom MongoDB bazom podataka
- View
  - React.js Web aplikacija koja putem HTTP protokola komunicira sa serverom
  - Sistem za preporučivanje koji na osnovu profila korisnika i prosleđenih parametara predlaže restorane korisniku
- Controller
  - Node.js (Express.js) server koji pruža REST preko HTTP-a.

Server se nalazi na cloud-u i pruža REST servis, dok je odgovor servera u formatu:

```
{
    status_code,
    response_data
}
```

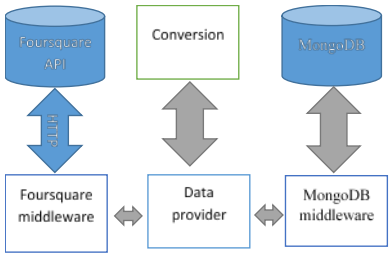
Interni statusni kodovi servera su sledeći:

- 0 – nema rezultata pretrage za navedene parametre, response_data je null
- 1 – pretraga je uspešna i rezultati se nalaze u response_data
- 2 – uspešna registracija/prijavljivanje, id korisnika se nalazi u response_data
- 3 – uspešno ažuriranje
- 4 – uspešno brisanje
- -1 – nevalidni parametri
- -2 – neuspešna registracija/prijavljivanje, poruka o grešci se nalazi u response_data
- -3 – neuspešno ažuriranje, poruka o grešci se nalazi u response_data
- -4 – neuspešno brisanje, poruka o grešci se nalazi u response_data




Slika 3. MVC arhitektura sistema.

Poruka o grešci je u obliku stringa, dok je sadržaj ostalih objekata drugačiji u zavisnosti od metode. Svi objekti parametara i odgovora su predstavljeni u JSON formatu. Data provider, slika 4, je modul koji je u direktnoj vezi sa serverom, bazom podataka i Foursquare API-jem. Funkcije definisane u ovom modulu pozivaju se u midleware funkcijama na serveru i vrše pripremu podataka u odgovarajuće formate pre poziva Foursquare API-ja ili čitaju iz MongoDB baze.




Slika 4. Data provider modul.

### B. *Modul za preporuke*

predstavlja implementaciju kolaborativne metode za preporučivanje zasnovane na korisnicima koripćenjem NHSM metrike za izračunavanje keoficijenata sličnosti korisnika. Za korišćenje modula definisana je jedna javna funkcija „recommend“, koja kao parametre prima listu korisnika i broj željenih preporuka. Prvi član liste je korisnik aplikacije. Ova funkcija se izvršava u tri koraka: inicijalizacija potrebnih struktura, izračunavanje koeficijenata sličnosti i preporuka sadržaja na osnovu najsličnijih korisnika.

```
var recommend =
function(users, numOfRecommendations) {
 //1. inicijalizacija
 var places = extractPlaces(users);
 initializeRatingsMatrix(users,
       places.length);
```

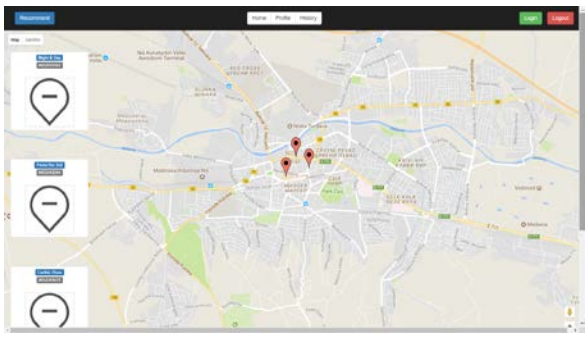
```
 calculateAvg(users, places);
 //2. izračunavanje koeficijenata sličnosti
 calculateSimilarities(users);
 //3. preporuka restorana
 var usersToIterate =
       getTheMostSimilarUsers(users);
 var recommendations = [];
 packPlaces(recommendations,
       usersToIterate,
numOfRecommendations);
 updateRatings(recommendations, places);
 return recommendations;
}
```

Prvi korak počinje kreiranjem matrice korisnik-restoran sa ocenama. To podrazumeva izvlačenje svih restorana iz „Liked“ sekcije proflila korisnika, a zatim i kreiranje matrice. Za potrebe izračunavanja koeficijenata sličnosti u ovom koraku se irazunavaju srednje ocene svih restorana i smeštaju u niz. Izračunavanje koeficijenata, tj. drugi korak, prati formulu za izračunavanje NHSM metrike definisanu u [2]. Za svaki parcijalni rezultat i za izračunavanje svakog faktora definisana je posebna funkcija. Koeficijenti se izračunavaju samo za tekućeg korisnika u poređenju sa ostalima. Nakon izračunavanja koeficijenata izvlače se dva najsličnija korisnika i uzimaju restorani iz njihovh „Liked“ sekcija. Na kraju, iz dobijene liste restorana uzimaju se oni sa najvišim ocenama i dodaju u listu preporuka sve dok se ne dobije željeni broj preporuka.

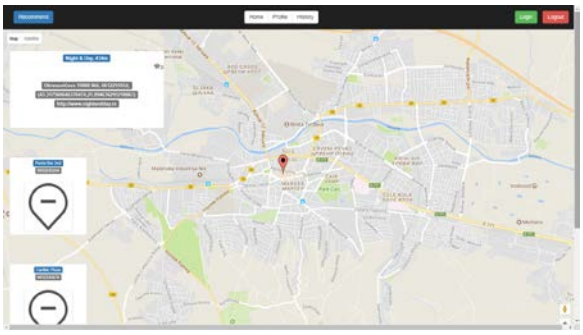
```
var calculateSimilarities = function(users)
{
    //kreiranje matrice
    for (var indexI=0;indexI<users.length-
1;
             indexI++){
        var row = [];
    for (var indexJ=0;
             indexJ<users.length-
      1;indexJ++)
        row.push(0);
        similarities.push(row);}
    //izračunavanje sličnosti
    for (var indexI=0;indexI<users.length-
1;
             indexI++){
        var userU = ratingsMatrix[indexI];
        for (var indexJ=indexI+1;
             indexJ<users.length;indexJ++){
            var userV =
ratingsMatrix[indexJ];
            var similarity =
                   simNHSM(userU, userV,
             avg);
            similarities[indexI][indexJ -
1] =
                    similarity;
}}};
var simNHSM = function (u, v, avg){
    var jpss = simJPSS(u, v, avg);
    var urp = simURP(u, v);
    return jpss * urp;
};
var simJPSS = function (u, v, avg) {
    var pss = simPSS(u, v, avg)
    var jacard = simJaccard(u, v);
    return pss * jacard;
};
var simURP = function (u,v ) {
    return 1-1/(1 + math.exp(
      -math.abs(meanRating(u) -
    meanRating(v))
      *math.abs(standardVariance(u)
      -standardVariance(v))
    ));
};
var simPSS = function (u, v, avg) {
    var sum = 0;
    for (var
index=0;index<avg.length;index++){
        var avgp = avg[index];
        var
pssValue=pss(u[index],v[index],avgp);
        sum = sum + pssValue;
    }
    return sum;
}
var pss = function (ru, rv, avgp) {
    var prox = proximity(ru, rv);
    var sign = significance(ru, rv);
    var sing = singularity(ru, rv, avgp);
    return prox * sign * sing;}
```

*C. Korisnički interfejs*

Na slikama 5 i 6 prikazani su rezultati preporuke i detalji o restoranu, respektivno. Zbog ograničenja u prostoru nisu prikazane stranice sa detaljima profila i istorija pretrage. Korisnički interfejs je dizajniran sa ciljim da bude što jednostavniji i intuitivniji.



Slika 5. Rezultati preporuke.



Slika 6. Detalji o restoranu.

*D. Iskorišćene tehnologije*

Node.JS [4] je asinhroni event-driven cross-platform JavaScript runtime, dizajniran za kreiranje skalabilnih mrežnih aplikacija. NodeJS je jezik za serversku stranu aplikacije.

Express.JS [5] je frejmvork na bazi Node.JS-a za pisanje servera, koji implementira osnovnu strukturu servera. Moguće, je kreiranje celokupne aplikacije, ali i API-ja.

React.JS [6] je biblioteka za kreiranje korisničkih interfejsa bazirana na NodeJS-u objavljena od strane Facebook-a. Trenutna stabilna verzija je 15.6.1.

MongoDB predstavlja NoSQL document orjentisana baza podataka pomenuta. Detaljniju dokumentaciju možete naći na [7].

„Foursquare“ [9] je društvena mreža, koja omogućava korisnicima da sa svojim prijateljima iz mreže podele svoje aktivnosti. Korisnici mogu da čekiraju svoju lokaciju (eng. "check in"), pretražuju objekte, ali i da koriste mapu za navigaciju.

## VI. ZAKLJUČAK

Cilj ovog projekta je pregled sistema za preporučivanje sadržaja i praktična implementacija Web aplikacije sa kolaborativnim sistemom za preporučivanje, koji koristi NHSM metriku (20) za određivanje koeficijenata sličnosti. NHSM metrika (20) prevazilazi pomenute probleme poznatih i popularnih metrika i predstavlja kombinaciju Jaccard-ove metode, uvedenih koeficijenata (proximity, significance i singularity), ali i tendencije samih korisnika da daju niže ili više ocene. Ova metoda daje preciznije rezultate od gore pomenutih metrika. Uvođenjem NHSM metrike u sistem povećana je preciznost preporuka, međutim postoji puno mesta za unapređenja aplikacije. Kako je u pitanju kolaborativni sistem za preporuku i zavisi od ocena ostalih sličnih korisnika implementirani sistem i dalje može imati problema prilikom određivanja „sličnih“ korinika. Takođe pati i od problema hladnog starta. Sistem može dalje biti unapređen nekom od metoda za preciznije određivanje „sličnih“ korisnika, ali i njegovom transformacijom u hibridnu varijantu, što će umnogome doprineti performansama.

## LITERATURA


[1] Recommendation systems: Principles, methods and evaluation, F.O.Isinkayea, Y.O.Folajimib, B.A.Ojokohc, Egyptian Informatics Journal, Volume 16, Issue 3, pg. 261-273, Novembar 2015.

[2] A new user similarity model to improve the accuracy of collaborative filtering, Haifeng Liu, Zheng Hu, Ahmad Mian, Hui Tian, Xuzhen Zhu, State Key, Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, China, Elsevier B.V., Novembar 2013.

[3] Beginners Guide to learn about Content Based Recommender Engines, Shuvayan Das, https://www.analyticsvidhya.com/blog/2015/08/beginners-guide-learn-content-based-recommender-systems, 11. Avgust 2015.

[4] Zvanična NodeJS dokumentacija, https://nodejs.org/en/, Septembar 2017.

[5] Zvanična ExpressJS dokumentacija, https://expressjs.com/, Septembar 2017.

[6] Zvanična React dokumentacija, https://facebook.github.io/react/, Septembar 2017.

[7] Zvanična Mongodb dokumentacija, https://docs.mongodb.com/, Septembar 2017.

[8] Zvanična Foursquare dokumentacija, https://developer.foursquare.com/, Septembar 2017.

# Uticaj geometrije krstastog silicijumskog senzora magnetnog polja na vrednosti Holovog napona

Milica Kostić

*Sadržaj* – U ovom radu akcenat je stavljen na značajnu ulogu magnetnih senzora i njihovih mogućih primena u zavisnosti od dimenzija i oblika pločice. Istraživanje je rađeno uz pomoć programa SILVACO, u simulatoru Atlas. Atlas je iskorišćen kao najpogodnija opcija za predstavljanje željenih struktura i karakteristika.

## I. UVOD

Merne informacije imaju veliki značaj u naučnim i tehničkim disciplinama, jer sadrže kvantitativne vrednosti pomoću kojih se procenjuju stanja objekta istraživanja. Moderna merenja baziraju se uglavnom na pretvaranju fizičkih u električne signale. Pojačanje i obrada električnih signala, daljinski prenos i očitavanje rezultata su usavršeni, a uz to se postiže i visok metrološki kvalitet u pogledu tačnosti, osetljivosti i pouzdanosti merenja.

U današnje moderno doba sa svih strana smo okruženi elektromagnetnim i magnetnim zračenjem iz najrazličitijih izvora (radarski sistemi, radio i TV predajnici, mobilni telefoni), pa se logično nameće pitanje kako i na koji način to zračenje deluje na ljude, žive organizme i njihovo zdravlje? U zemljama Evrope trenutno je u toku veliki broj istraživanja koja imaju za cilj da ispitaju i dokažu vezu između izloženosti elektromagnetnom zračenju i pojave bolesti u ljudskom organizmu. Elektromagnetno i magnetno polje (iz skoro svih delova spektra) mogu uticati na protok energije u ćelijama, promeniti frekventni opseg i izvršavanja transportnih procesa kroz ćelijsku membranu. Iz tog razloga se od sredine dvadesetog veka intenzivno radi na usavršavanju,u vidu pojednostavljenja i tehnoloških kompaktnosti, magnetnih senzora [1].

Rad se može podeliti na dva ispitivanja, prvi jeste kako se menja razlika Holovih napona u mernim tačkama senzora, pri konstantnoj jačini magnetnog polja od 1T, sa promenom anodnog napona, i drugi, kako se menja razlika Holovih napona u mernim tačkama senzora pri konstantnom anodnom naponu od 1V, sa promenom jačine magnetnog polja.

## II. PRINCIP RADA

Pretvaranje mehaničkih, hemijskih, bioloških i procesnih veličina u električni signal, vrši se pomoću različitih pretvarača, transdjusera, transmitera i sl. U novije vreme za sve ove uređaje ustalio se termin senzor, što je koincidiralo sa razvojem materijala i tehnologija, primenom mikromehanike i visokog stepena integracije komponenata, kao i velikim mogućnostima u obradi informacija pomoću sredstava mikroelektronike.

Milica Kostić, student master studija na modulu Elektronika i mikrosistemi, Katedra za mikroelektroniku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: milica.kostic@sicef.info

### *A. Senzori na bazi Holovog efekta, ulazni fajl i struktura*

Senzori na bazi Holovog efekta spadaju u red specijalnih elektromagnetnih senzora. Postoje različite vrste Holovih senzora, u zavisnosti od njihove geometrije, tj. oblika pločice. Neke od tih geometrija se razlikuju po karakteristikama, a neke i ne toliko značajno. Neki od oblika Holovih senzora imaju prednost u jednostavnijoj tehnologiji proizvodnje. U ovom radu obrađivani su magnetni senzori sa krstastom pločicom, kao najpogodniji za jednostavno postavljanje elektroda kod takvih struktura.

U programu SILVACO sledećim ulaznim fajlom, formirana je struktura pločice u obliku krsta, prikazana na slici 1.

```
#Simetrican Hallov senzor

go atlas
mesh

x.mesh l=0    s=0.05
x.mesh l=1.0  s=0.05
y.mesh l=0    s=0.05
y.mesh l=1.0  s=0.05

#Ceo region1
region num=1  material=silicon

#Region2, gore levo
region num=2 x.max=0.3 y.max=0.3
material=oxide

#Region3, gore desno
region num=3 x.min=0.7 y.max=0.3
material=oxide

#Region4, dole levo
region num=4 y.min=0.7 x.max=0.3
material=oxide

#Region5, dole desno
region num=5 y.min=0.7 x.min=0.7
material=oxide

electrode name=anode bottom
electrode name=cathode top

doping region=1 n.type uniform conc=1.0e18
method mag2dlsq
```

```
solve init
solve prev

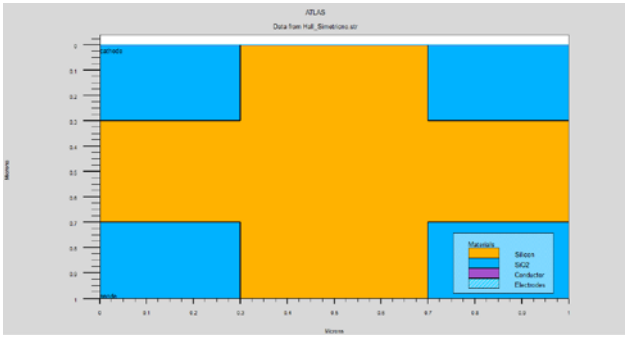
probe name=hall1 x=0.0 y=0.5 potential
probe name=hall2 x=1.0 y=0.5 potential
probe name=reference x=0.5 y=0.5 potential

log outf=Hall_Simetricno.log

solve name=anode vanode=0.0 vstep=0.05
vfinal=1.0

save outfile=Hall_Simetricno.str
tonyplot Hall_Simetricno.str
tonyplot Hall_Simetricno.log
log off

quit
```



Slika 1. Struktura Holovog magnetnog senzora sa simetričnim dimenzijama krsta.

*B. Promena jačine Holovog napona sa promenom anodnog napona*

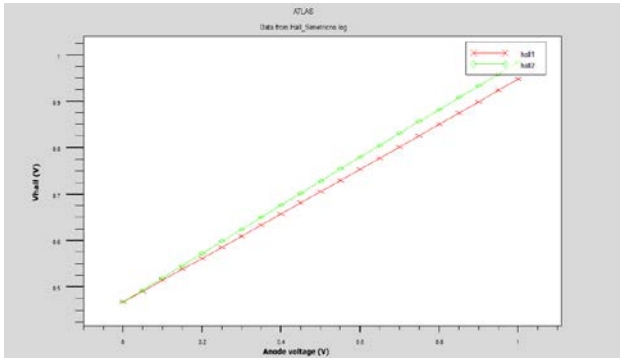
Ukoliko obratimo pažnju, posebno na dve linije koda, liniju u kojoj se definise vrednost jačine magnetnog polja, koje je u ovom primeru označeno sa bz:

```
models srh fldmob bz=1 rh.elec=1.2
rh.hole=0.8
```

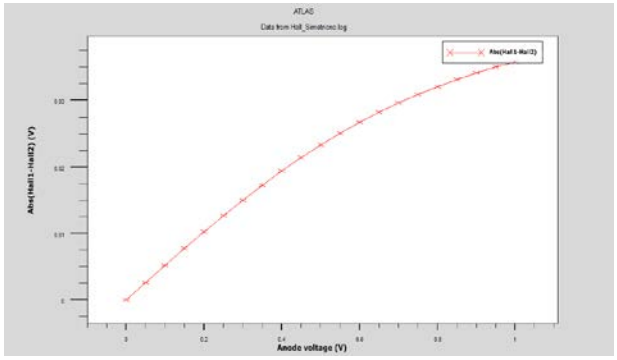
i liniju koda u kojoj se definiše vrednost anodnog napona, označenog sa vanode:

```
solve name=anode vanode=0.0 vstep=0.05
vfinal=1.0
```

jasno se vidi da je jačina magnetnog polja bz fiksirana na vrednost od 1T, a da se anodni napon u ovom slučaju menja od 0 do 1V sa korakom od 0.05. Takođe, bitno je napomenuti da su dimenzije krsta u ovom slučaju simetrične po X i Y osama, tj širina krsta kako po X tako i po Y osi iznosi od 0.3 do 0.7 mikrometara. Kao rezultat očitanih Holovih napona u dve merne tačke, čija je pozicija definisana u datom kodu, dobijeni su grafikoni na slici 2, a razlika dveju izmerenih Holovih napona, data je na slici 3.



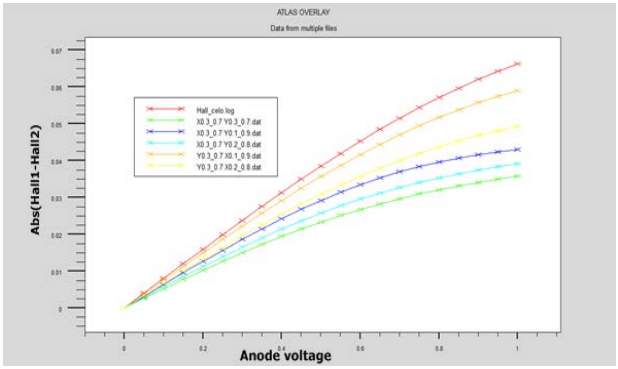
Slika 2. Grafički prikaz izmerenih vrednosti Holovih napona kod senzora sa simetričnim dimenzijama krsta.



Slika 3. Razlika dva izmerena Holova napona kod senzora sa simetričnim dimenzijama krsta.

Ukoliko u kodu promenimo dimenzije krsta, prvo po X osi (u jednom slučaju od 0.2 do 0.8 mikrometara, a zatim u drugom slučaju od 0.1 do 0.9), a zadržimo vrednost po Y osi (od 0.3 do 0.7 mikrometara), a nakon toga promenimo po Y osi (u jednom slučaju od 0.2 do 0.8 mikrometara, a zatim u drugom slučaju od 0.1 do 0.9), a zadržimo po X (od 0.3 do 0.7 mikrometara), i na kraju uzmemo slučaj u kome su dimenzije krsta i po X i po Y osi od 0 do 1 mikrometra tj. u kome krst dobija oblik kvadrata, dobijamo u svakom od tih slučajeva različite vrednosti izmerenih Holovih napona u dvema tačkama, tj. određenu razliku tih Holovih napona. Na slici 4 su upoređene karakteristike dobijene kao rezultat razlike Holovih napona u svim tim slučajevima.

Upoređivanjem rezultata može se doći do sledećeg zaključka: najveća razlika Holovih napona se dobija u slučaju gde je izgubljen oblik krsta, tj. gde nam je pločica kvadrat, a zatim kod onog senzora, kod koga je krst najvećih dimenzija po X osi (od 0.1 do 0.9 mikrometara), a po Y osi fiksirana (od 0.3 do 0.7 mikrometara). Kod senzora sa simetričnim dimenzijama krsta dobija se najmanja razlika Holovih napona sa promenom anodnog napona.




Slika 4. Upoređene karakteristike dobijene kao rezultat razlike Holovih napona pri različitim geometrijama krsta.

C. *Promena jačine Holovog napona sa promenom jačine magnetnog polja*

Ukoliko posmatramo sledeću liniju koda u našem ulaznom fajlu:

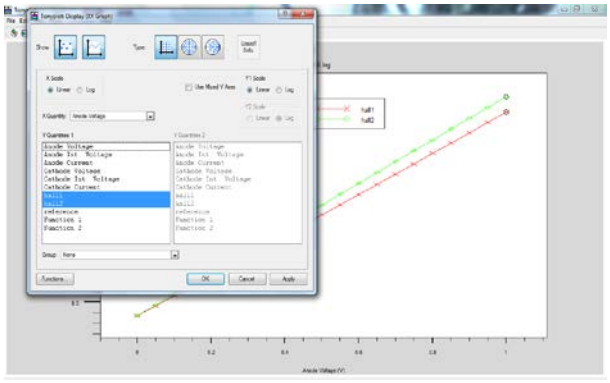
```
solve name=anode vanode=0.0 vstep=0.05
vfinal=1.0
```

vidimo da nam se anodni napon menja počev od 0V do 1V sa korakom od 0.05V. Ono što je bitno naglasiti jeste da nakon izvršenja ove naredbe anodni napon nakon svih promena je fiksiran na 1V za sledeće `solve` naredbe. Upravo to se može iskoristiti u sledećem ispitivanju, a to je, kako se Holovi naponi ponašaju pri promeni magnetnog polja B od 0 do 1T sa korakom od 0.1T pri fiksnom anodnom naponu. Zbog nemogućnosti da se magnetno polje menja automatski u program, ovaj proces je najsigurnije obaviti fizičkim putem, tj. menjanjem vrednosti magnetnog polja u modelu programa za svaku prethodno ispitanu i realizovanu geometriju krsta Holovog magnetnog senzora, kao:

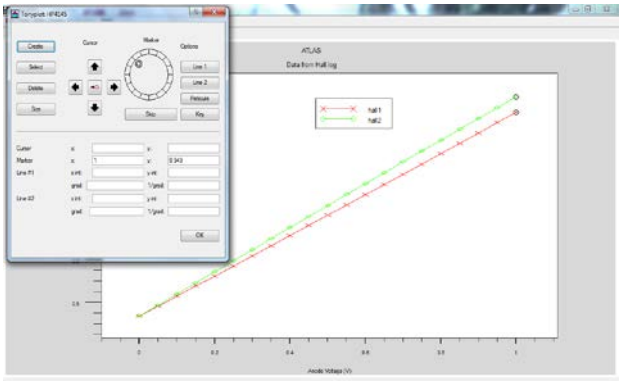
```
models srh fldmob bz=1 rh.elec=1.2
rh.hole=0.8
```

koraci koji su praćeni tokom provere vrednosti Holovih napona, prikazani su na slikama 5 i 6.

U tabeli 1 je prikazano kako se menja jačina magnetnog polja, zatim izmerene vrednosti Holovih napona u mernim tačkama pri određenoj jačini magnetnog polja, vrednost anodnog napona, kao i apsolutna vrednost razlike Holovih napona u tim tačkama. U tabeli 1 su date vrednosti koje se odnose na magnetni senzor sa simetričnim dimenzijama krsta i po X i po Y osi.



Slika 5. Na Y osi se nalaze vrednosti izmerenih Holovih napona, na X osi se nalazi anodni napon.

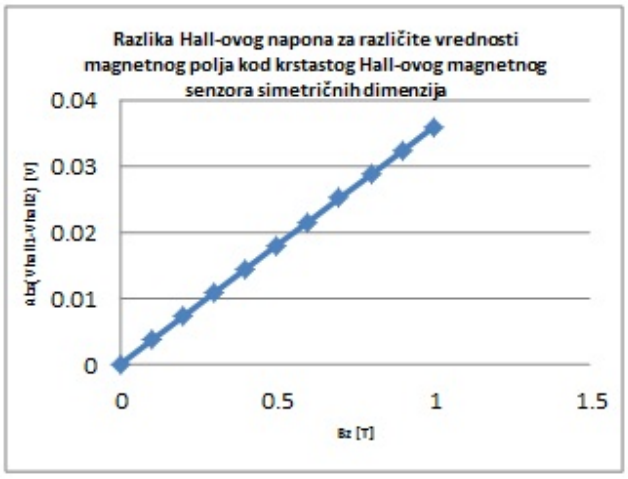


Slika 6. Očitavanje vrednosti Holovog napona, isključivo u tačkama gde je anodni napon jednak jedinici, za određenu vrednost magnetnog polja, definisanu u modelu programa.

TABELA I.
VREDNOSTI DOBIJENE ZA MAGNETNI SENZOR SA SIMETRIČNIM DIMENZIJAMA KRSTA (OD 0.3 DO 0.7 MIKROMETARA).

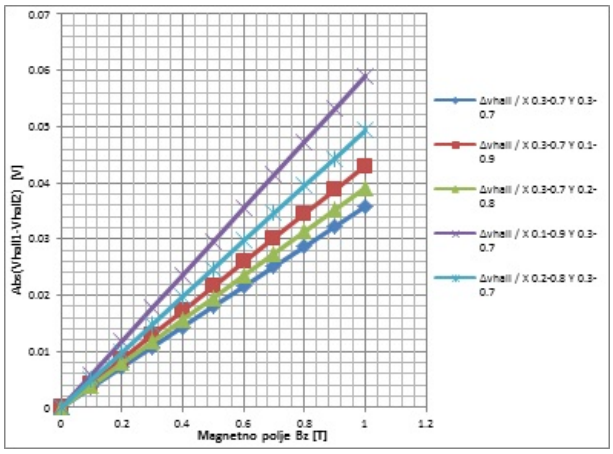
| **Vanode (V)** | **Bz (T)** | **Vhall1 (V)** | **Vhall2 (V)** | **$\Delta V_H$ (V)** |
|---|---|---|---|---|
| **1** | 0 | 0.9669 | 0.9669 | 0 |
| **1** | 0.1 | 0.9687 | 0.9651 | 0.0036 |
| **1** | 0.2 | 0.9705 | 0.9633 | 0.0072 |
| **1** | 0.3 | 0.9723 | 0.9615 | 0.0108 |
| **1** | 0.4 | 0.9741 | 0.9597 | 0.0144 |
| **1** | 0.5 | 0.9759 | 0.9579 | 0.018 |
| **1** | 0.6 | 0.9776 | 0.9561 | 0.0215 |
| **1** | 0.7 | 0.9794 | 0.9543 | 0.0251 |
| **1** | 0.8 | 0.9812 | 0.9526 | 0.0286 |
| **1** | 0.9 | 0.983 | 0.9508 | 0.0322 |
| **1** | 1 | 0.9848 | 0.949 | 0.0358 |

Ukoliko se ove vrednosti predstave grafički, promene razlike Holovih napona u odnosu na promene vrednosti magnetnog polja, u ovom slučaju dobijamo karakteristiku kao na slici 7.




Slika 7. Grafičko predstavljanje vrednosti za senzor sa simetričnim dimenzijama krsta.

Isti postupak i ispitivanje su ponovljeni za različite dimenzije krsta, pri različitim jačinama magnetnog polja, i dobijene vrednosti su predstavljene, takođe, grafički. Ukoliko se sumiraju svi dobijeni rezultati i vrednosti apsolutnih razlika Holovih napona predstave grafički, a zatim se ti grafici uporede, dobijene su karakteristike sa slike 8.




Slika 8. Poređenje dobijenih karakteristika za različite vrednosti magnetnog polja.

Upoređivanjem svih dobijenih karakteristika za različite vrednosti magnetnog polja Bz, pri konstantnom anodnom naponu senzora koji iznosi 1V, vidljivo je da su razlike Holovih napona Vhall1 i Vhall2 najveće, kada su u pitanju karakteristike Holovog senzora kod koga su dimenzije krsta po X osi, znatno veće u odnosu na dimenzije po Y osi.

## III. ZAKLJUČAK

Na osnovu svega priloženog i uzimajući u obzir, kako rezultate dobijene na grafiku sa slike 4, tako i rezultate sa slike 8, može se zaključiti da je najosetljiviji senzor sa dimenzijama krsta po X osi od 0.1 do 0.9 mikrometara, i po Y osi od 0.3 do 0.7 mikrometara, tj. da ukoliko dimenzije krsta nisu simetrične, raste i razlika potencijala u mernim tačkama. Takođe, kao zaključak sa grafika, možemo videti da je najmanja razlika napona Vhall1 i Vhall2 kod senzora sa simetričnim dimenzijama krsta od 0.3 do 0.7 mikrometara, tj. da su senzori sa takvim dimenzijama manje osetljivi. Te, ukoliko nam je neophodan senzor sa maksimalnom osetljivošću na promene magnetnog polja, uzećemo senzor sa većom površinom pločice, tj. onaj kome su dimenzije krsta po X osi znatno veće od dimenzija po Y osi.

## ZAHVALNICA

Ovaj rad je realizovan kao završni rad na smeru Elektronske komponente i mikrosistemi. Zahvaljujem se prof. dr Nebojši Jankoviću, kao svom mentoru, na izdvojenom vremenu, sugestijama i pomoći pri ispitivanju, realizaciji i pisanju rada.

## LITERATURA

[1] Dr Mladen Popović, Senzori i merenje-četvrto izdanje, Sarajevo, 2004.
[2] Prof. dr Dragan Pantić, Modeliranje i simulacija u mikroelektronici-drugo dopunjeno izdanje, Elektronski fakultet, Niš, 2006.
[3] Prof. dr Goran Ristić, Dozimetrija i dozimetri, Elektronski fakultet, Niš, 2013.
[4] Doc. dr Nenad Veljković, Kako rade senzori, PC PRESS, Beograd, 2017.
[5] Visoka tehnička škola, Magnetizam, Niš, 2012.
[6] https://www.automatika.rs/baza-znanja/senzori/holov-senzor.html
[7] https://sr.wikipedia.org/sr-el/Senzor

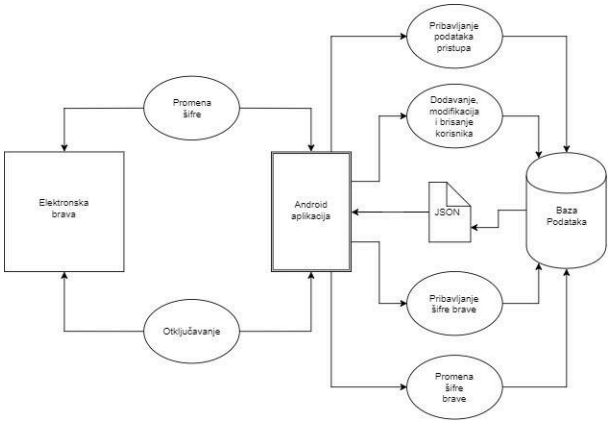
# Autentifikacioni sistem elektronske brave

Milan Ivković, Milan Milićević, Stefan Đokić, Nemanja Raković, Dragan Kostić, Lazar Bogdanović, Stefan Tomić, Marko Vučković

*Sadržaj* - Sve više je pametnih sistema koji se koriste kako u personalne, tako i u profesinonalne svrhe. U narednom tekstu biće izložen princip rada Android [2] aplikacije, i sve relevantne komunikacije između pojedinih modula realizovanog sistema.

## I. UVOD

Pristup prostoriji koja je obezbeđena elektronskom bravom je zamišljen preko aplikacije instalirane na mobilnom uređaju. Korisnik se logovanjem na aplikaciju identifikuje i time se šalje ID korisnika serveru, gde se na osnovu loga u bazi podataka utvrđuje da li ima dozvolu za pristup ili ne. Ukoliko ima pristup, server kao odgovor šalje kod aplikaciji koja se putem Bluetooth-a [3] šalje elektronskoj bravi. Algoritam brave proverava predefinisanu šifru i nakon poklapanja brava šalje povratnu informaciju aplikaciji da su vrata otključana. Nakon toga aplikacija šalje serveru da je korisnik sa datim ID-om ušao u prostoriju u određeno vreme. Dijagram brave je prikazan na slici 1. Zbog složenosti samog rada objašnjenje će biti podeljeno u više segmenata.




Slika 1. Blok šema celokupnog sistema elektronske brave.

Milan Ivković, Milan Milićević, Stefan Đokić, Nemanja Raković, Dragan Kostić, Lazar Bogdanović, Stefan Tomić i Marko Vučković su članovi Studentskog Inovacionog Centra Elektronskog Fakulteta (SICEF), Elektronski fakultet, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: milan.ivkovic@elfak.rs, milicevicm@elfak.rs, stefandjokic@elfak.rs, nemanja228@gmail.com, dragan.kostic@elfak.rs, invadersmustdie@elfak.rs, stefan.tomic@elfak.rs, marvuchko@gmail.com

## II. PRINCIP RADA

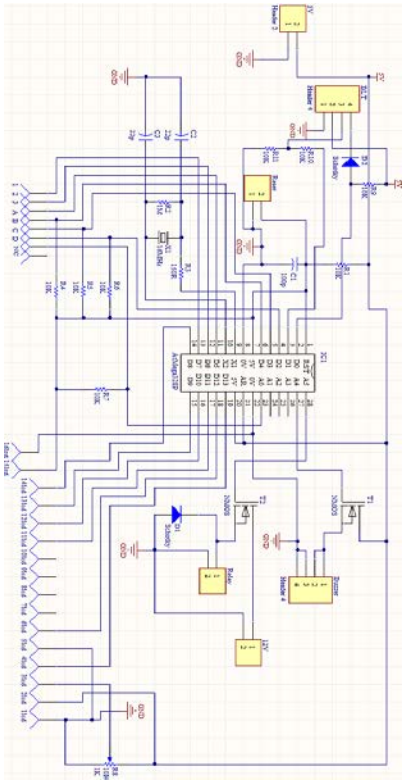
Projekat se sastoji iz četiri dela i to su: server sa bazom podataka, web aplikacija koja predstavlja administratorski panel celog sistema, android aplikacija i sama elektronska brava. Ceo sistem funkcioniše tako da android aplikacija komunicira sa bravom preko Bluetooth-a i poseduje funkcionalnosti poput otključavanja i promene šifre brave. Takođe, komunikacija sa bazom podataka omogućava funkcionalnosti poput registrovanja novih korisnika, modifikacije postojećih korisnika, brisanje korisnika, zahtevanje unosa šifre brave prilikom korišćenja i drugo. Za svaki od navedenih delova i specifične pojedinosti, biće reči u narednom delu.

### *A. Elektronska brava*

Hardverski deo elektronske brave se sastoji od više delova, a to su: Bluetooth modul, tastatura dimenzija 3x4 kao i LCD displej dimenzija 2x16 [4]. Putem Bluetooth modula elektronska brava komunicira sa android uređajem i ispisuje potrebne poruke na displeju. Tastatura se koristi da ukoliko korisnik ne poseduje mobilni telefon takođe može imati pristup. Ovakav pristup je obezbeđen tako što svaki korisnik ima svoju šifru i prilikom ulaska brava pamti njegov ID i vreme ulaska. Kada se neko od korisnika konektuje putem bluetootha na elektronsku bravu, ona šalje sve sačuvane podatke koje su do tog trenutka bile u memoriji brave.

### *B. Hardverska realizacija*

Za realizaciju hardvera iskorišćeno je okruženje Arduino Uno [5] sa mikrokontrolerom ATmega328 [6]. Da se ne bi koristilo celo okruženje u elektronskoj bravi, programiran je zasebni mikrokontroler preko pomenutog okruženja čiji će se način rada u daljem tekstu objasniti. Brava je realizovana prema električnoj šemi sa slike 2. Na slici 2 se mogu uočiti pojedinii delovi koji su bitni za funkcionalnost kako komunikacionog dela sa modulom tako i samog rada mikrokontrolera. Posebna pažnja je posvećena podešavanju oscilatora koji daje taktnu frekvenciju mikrokontrolera. U nekim slučajevima se dešava da oscilator neće zaoscilovati zbog lošeg lemljenja, lošije izrade štampane pločice i sve to zajedno utiče na to da mikrokontroler ne proradi. Da bi se to eliminisalo postavlja se otpornik od 1 do 2MΩ koji pomaže oscilatoru da zaosciluje, čime se obezbeđuje siguran rad mikrokontrolera.

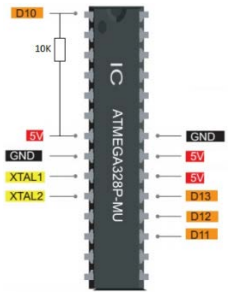


Slika 2. Električna šema elektronske brave.

Sledeća stvar oko koje treba obratiti pažnju jeste konekcija modula bluetooth-a i samog mikrokontrolera. Naponski nivoi logičkih jedinica i nula pomenuta dva dela nisu isti. Naponska jedinica Bluetooth-a je 3.3V dok je kod mikrokontrolera 5V. Zato se naponi moraju redukovati da bi se izbeglo pregorevanje modula. Kada mikrokontroler šalje podatak napon treba spustiti na 3.3V sa 5V i to se postiže naponskim razdelnikom korišćenjem otpornika $R_{10}$ i $R_{11}$, sa vrednošću od 10KΩ i 20KΩ respektivno. Time podatak koji je stigao na modul bluetooth-a ima 3.3V što ga čini bezbednim. Kada se podatak šalje iz modula u mikrokontroler treba ga podići ali nije neophodno. Podizanje napona se ostvaruje putem diode D2 i otpotnika $R_9$ na sledeći način: kada je na modulu naponska nula, dioda provodi i na mikrokontroleru se očitava 0.7V tj. logička nula, a kada se šalje jedinica, tada je dioda u blokadi i prosleđuje se napon od 4V (3.3V + 0.7V) što mikrokontroler očitava kao logičku jedinicu. Na taj način je postignut prenos podatakana različit nivo napona bez ikakvih problema.

*C. Programiranje ATmega328 mikrokontrolera*

Programiranje zasebnog mikrokontrolera se vrši putem postojećeg Arduino okruženja upotrebom ISP programa učitanog u arduino okruženje.

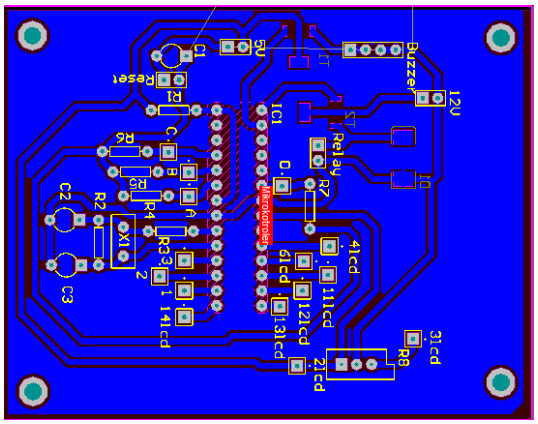



Slika 3. Povezivanje mikrokontrolera sa Arduinom.

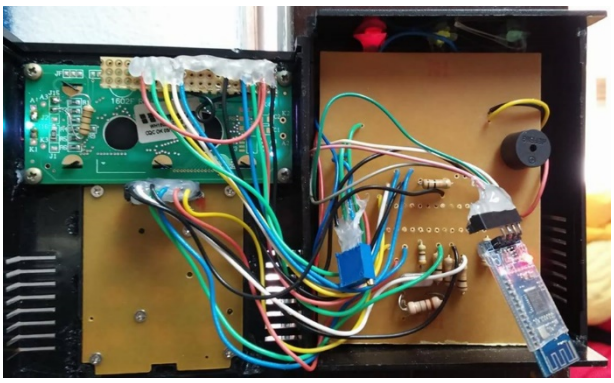
Nakon povezivanja mikrokontrolera sa Arduinom prema slici 3. potrebno je podesiti Arduino okruženje za upis programa. Prvo treba učitati program kojim će se Arduino ponašati kao programator. Putanja za to je: File → Examples → ArduinoISP. Nakon toga sledi upis datog programa. Zatim u Arduino okruženju treba odabrati programator "Arduino as ISP", putanja: Tools → Programme → Arduino as ISP. U istom padajućem meniju odabrati opciju „Burn Bootloader" kako bi se u novi mikrokontroler upisao *bootloader*. Učitavanje programa u novi mikrokontroler se vrši tako što se drži taster „Shift" i zatim pritisne dugme „Upload".

*D. Praktična realizacija*

Crtanje električne šeme kao i projektovanje pločice je realizovano u programu Altium designer[7]. Ovaj program daje mogućnost i prikaz 3D štampane pločice. Realizacija štampane pločice je rađena metodom prenosa tonera na bakarnu površinu. Data električna šema (slika 4) veze je preneta na bakarnu površinu putem termalnog postupka sa foto papira. Nakon prenošenja pločica je nagrizana u rastvoru hlorovodične kiseline i tridesetoprocentnog hidrogena u razmeri 2:1. Zatim sledi bušenje pločice i lemljenje komponenata. Slika 5 daje untrašnji prikaz sklopa, a slika 6 gotov izgled elektronske brave.




Slika 4. PCB izgled pločice.



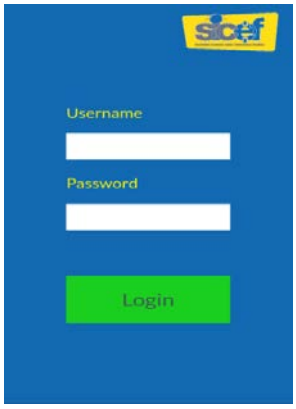
Slika 5. Untrašnji prikaz elektronske brave.




Slika 6. Gotov izgled elektronske brave.

*B. Android aplikacija*

Android aplikacija se sastoji od 3 segmenta (activity-ja) gde se nalaze sve funkcionalnosti. Na prvoj stranici se nalaze dva tekstualna polja gde korisnik unosi korisničko ime i šifru i dugme na koje se korisnik može ulogovati i time povezati sa serverom i bazom podataka (slika 7).



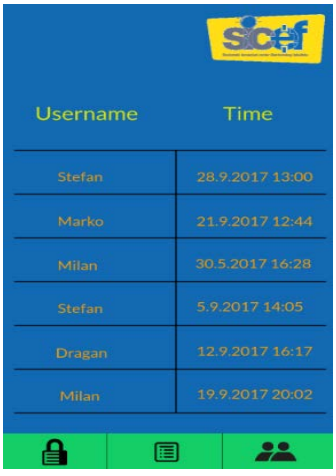

Slika 7. Login activity.

Nakon povezivanja sa serverom, ukoliko je izvršena autentifikacija korisnika, on u JSON (JavaScript Object Notation) [8] formatu dobija šifru brave i šifra se čuva u memoriji aplikacije. U slučaju da nije ispravno uneta šifra ili korisničko ime, korisnik dobija poruku o grešci. Nakon uspešnog logovanja, korisniku se nude sledeće mogućnosti. U slučaju da korisnik ima ulogu administratora, koja se određuje autorizacijom istog, dobija opcije da otključa bravu klikom na sliku i mogućnost da klikom na dugme pređe u administratorski panel kao i mogućnost menjanja sopstvene šifre pri logovanju (slika 8).



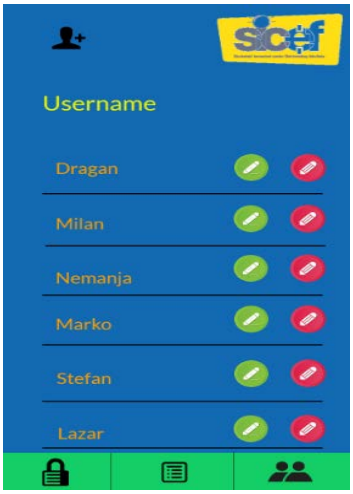
Slika 8. Main activity.

U slučaju da korisnik nema administratorsku ulogu imaće sve opcije bez administratorskog panela. Ukoliko korisnik želi da promeni sopstvenu šifru, otvara se dijalog gde korisnik treba uneti staru šifru, novu šifru i potvrdu nove šifre. Nakon toga aplikacija komunicira sa serverom i bazom i u zavisnosti od rezultata, vraća poruku korisniku. Na poslednjoj stranici, u slučaju odabira administratorskog panela, korisnik ima mogućnost da promeni šifru brave, vidi sve zapise o aktivnosti korisnika i brave, kada i ko je otključavao kao i spisak svih registrovanih korisnika (slika 9).




Slika 9. Admin activity.

Nakon prelaska u administratorski panel, aplikacija komunicira sa bazom i opet u JSON formatu dobija zapise o tome koji korisnik i kada je otključavao bravu i smešta u listu koja se nalazi na ekranu. Ako se korisnik opredeli za menjanje šifre brave, otvara se dijalog gde je potrebno uneti staru šifru brave, novu šifru i potvrdu za novu šifru.




Slika 10. Lista registrovanih korisnika.

Kao i u prethodnom slučaju u zavisnosti od rezultata komunikacije sa serverom, korisnik dobija povratnu poruku. Poslednja funkcionalnost je prikaz svih registrovanih korisnika (slika 10), klikom na dugme u dnu ekrana, gde zatim aplikacija pribavlja podatke iz baze podataka, u JSON formatu, i zatim ih smešta u listu u kojo se pored svakog korisnika nalaze dva dugmeta za brisanje i modifikaciju korisnika.

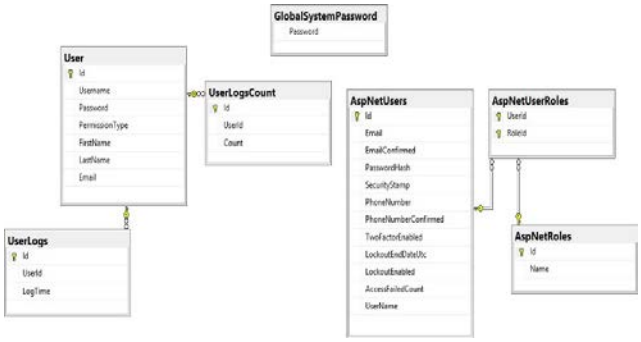
Takođe na vrhu liste se nalazi dugme gde korisnik klikom može dodati odnosno registrovati novog člana. Otvara se dijalog i u njemu je potrebno uneti korisničko ime novog korisnika, šifru i potvrdu šifre. Kao i u prethodnim slučajevima i ovde postoji povratna poruka o rezultatu. U slučaju modifikacije nekog od korisnika, otvara se dijalog sa već popunjenim podacima izabranog korisnika i nakon unete izmene vrši se korekcija u bazi podataka i korisnik dobija povratnu poruku o uspešnosti akcije. Sva komunikacija sa serverom i bazom podataka realizovana je preko API-a (Application Programming Interface) [10] koji je realizovan u .NET [11] okruženju i o kom će biti više reči u nastavku.

*C. Server*

Da bi sav sistem uspešno funkcionisao, naravno neophodna je i ova komponenta. Serversku stranu sistema čine baza podataka, realizovan API preko koga se vodi celokupna komunikacija sa svim delovima sistema, kao web aplikacija koja predstavlja admin panel.

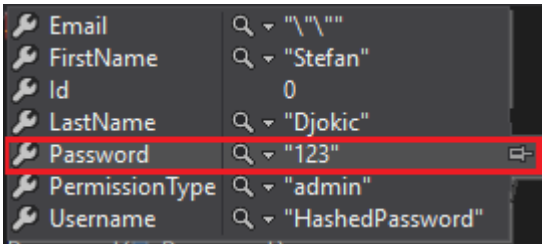
*C1. Baza podataka*

Baza podataka koja je korišćena u ovom sistemu je MSSQL (Microsoft SQL)[12]. Baza je hostovana na trial verziji servera http://my.gearhost.com sa ograničenim, ali za ovu svrhu dovoljnim prostorom. Bazi podataka se pristupa preko desktop Windows aplikacije Microsoft SQL Server Management Studio unošenjem imena hosta, korisničkog imena i šifre koje su generisane na samom serveru. Sadržaj baze podataka je prikazan na slici 11.

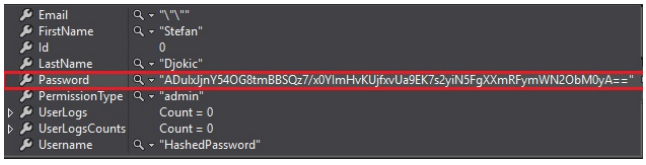



Slika 11. Sadržaj baze podataka.

Kao što vidimo na slici, u bazi podataka postoje nekoliko tabela. Najveći značaj za sistem imaju sledeće tabele: **User**, **UserLogs, UserLogsCount, AspNetUsers** kao i **GlobalSystemPassword**. **User** tabela predstavlja osnovni entitet celokupnog sistema, a to su sami korisnici koji ga koriste i koji interaguju sa elektronskom bravom. Čine je atributi: ID, korisničko ime, šifra, ime, prezime, email i permisija, Ovde se mora napomenuti da se sve šifre u bazi snimaju u enkriptovanom obliku, koji je kasnije nemoguće dekriptovati. Ovo čini osnovu bezbednosti ovog sistema. Ukoliko bude bilo napada čak i na bazu podataka, napadač neće moći da iskoristi šifru ni u kom obliku. Prikaz šifre nakon dodavanja korisnika pre i posle enkripcije prikazan je na slikama br. 12 i br. 13.




Slika 12. Šifra prilikom dodavanja korisnika, pre enkripcije.




Slika 13. Šifra tokom dodavanja korisnika, posle enkripcije.

| | | | |
|---|---|---|---|
| 3 | 17 | StDjokic | 3231 |
| 4 | 18 | StDjokic | 3231 |
| 5 | 19 | HashedPassword | ADuIxJjnY54OG8tmBBSQz7/x0YlmHvKUjfxvUa9EK7s2yiN5F... |

Slika 14. Šifra tokom dodavanja korisnika, posle enkripcije u bazi podataka.

**UserLogs** nam pokazuje sve aktivnosti korisnika, ko, kada, u koliko sati i kog datuma je neko otključao bravu. **UserLogsCount** vodi evidenciju o ukupnom broju otključavanja brave svih korisnika. **GlobalSystemPassword** je entitet u kome je smeštena šifra za otključavanje same brave. **AspNetUsers** je tabela koja ima najviše značaja što se tiče admin panela. Bazi podataka i svim tabelama pristupa se putem API-a. Pored klasičnih CRUD operacija (Create, Read, Update, Delete) bazi se pristupa korišćenjem i nekim dodatnim operacijama, kao što su upad na sistem, autentifikacija, autorizacija i slično.

*C2. Application Programming Inteface(API)*

API predstavlja programski interfejs koji služi za komunikaciju sa bazom podataka, tj. pribavljanje podataka. Suština API-a je korišćenje REST servisa radi pristupa resursima. U ovom projektu, API je realizovan kao WebApi[13] što u prevodu znači da se resursu pristupa putem linka. Razlog odabira ovog tipa su fleksibilnost, skalabilnost kao i pristupačnost. Resurse koje "dohvati" WebApi moguće je pristupiti na bilo kojoj platformi, što je i u samom sistemu pokazano – pristupaju i android aplikacija kao i web aplikacija. WebApi je implementiran u .NET API tehnologiji programskog jezika C#, korišćenjem integrisanog razvojnog okruženja Visual Studio 2015 Enterprise. Međuvezu baze podataka i API-a predstavlja ORM (Objektno Relacioni Maper)[14]. Kao ORM tehnologija je iskorišćen, jako popularan framework „Entity Framework" (EF) verzija 6. Funkcionalnost API-a se svodi na pozivanju metoda koje putem EF-a pristupaju bazi podataka i vraćanju rezultata tj. resurs u JSON formatu, kome se pristupa putem unapred definisanog linka. Jedan primer API poziva i rezultata, iz sistema koji se predstavlja u ovom dokumentu prikazan je na slici br. 14.




Slika 15. Poziv API-a putem linka i odgovor u JSON formatu.

Na slici br. 13 vidimo primer slanja zahteva API-u (API Request) za vraćanje svih korisnika u sistemu. Kao odgovor (API Response) vraća JSON objekat sa listom svih korisnika na sistemu, u ovom primeru prikazanom putem XML notacije.

*C3. Web aplikacija – Admin panel*

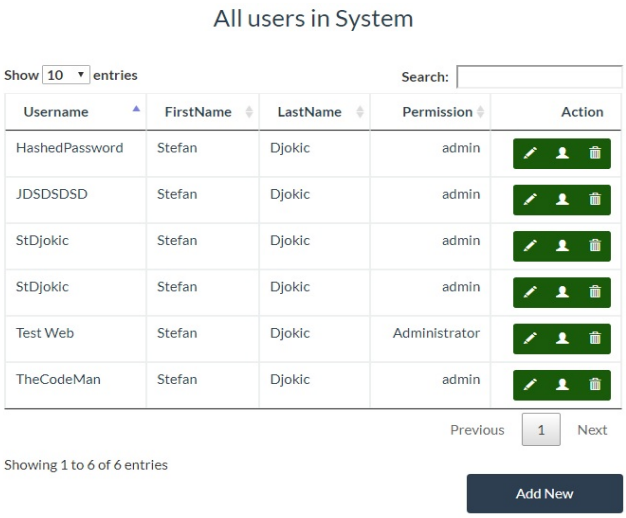
Admin panel je deo sistema koji koriste samo administratori ovog sistema. Ovde, administrator može videti apsolutno sve informacije o korisnicima, bravi, celokupnom sistemu. Kao npr. uvid u listu svih korisnika, svih njihovih informacija, uvid u sve njihove aktivnosti vezane za aplikaciju, kao i za elektronsku bravu. Pored toga, njemu su omogućene apsolutno sve moguće funkcije koje pruža sistem, poput svih operacija koje su vezane za korisnike, tipa dodavanje, brisanje, modifikovanje, menjanje šifre, kao i svih ostalih podataka. Admin panel je realizovan kao .NET MVC5 aplikacija u programskog jeziku C#. Kao ORM je takođe kao i kod API-ja korišćen EntityFramework[15]. Primer jednog poziva funkcije koja radi sa EF-om je prikazan na slici 15. Rezultat funkcije je prikazan na slici 16.

```
[HttpPost]
[AjaxErrorHandler]
0 references
public LargeJsonResult GetAllUsers(DataTableParamViewModel dtFilter)
 {
    var filter = ModelMappings.Map(dtFilter);

    var data = UserService.GetAllUsers(filter);

    return ResponseHelper.GetLargeJsonResultResponse(data);
}
```

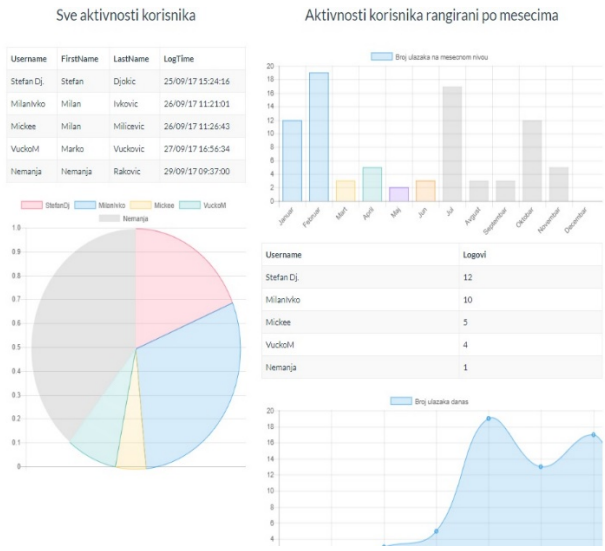
Slika 16. HttpPost zahtev za vraćanje svih korisnika.




Slika 17. Rezultat poslatog zahteva – lista korisnika.

Što se tiče klijentske strane i korisničkog interfejsa, korišćen je JavaScript skriptni jezik sa bibliotekama JQuery i Ajax za upućivanje poziva serveru. Ono što izdvaja admin panel od ostalog sistema je deo interfejsa

koji nam daje informacije o svim mogućim aktivnostima korisnika u sistemu. Za lep uvid u sve tipove aktivnosti korišćena je biblioteka Chart.js. Prikaz strane sa svim aktivnostima prikazan je na slici 17.




Slika 18. Sve aktivnosti koje može videti administrator.

Server predstavlja jedan od najznačajnih delova sistema ali i najranjivijih. Baš zbog toga treba obezbediti na razne načine mogućnost autentifikacije, autorizacije korisnika, hešovanje i enkriptovanje svih mogućih šifri u sistemu. Iz svega opisanog i objašnjenog, može se zaključiti da je ovom pametnom i upotrebljivom sistemu, pristup lak i najbitnije, zagarantovano bezbedan.

## III. Zaključak

U ovom radu je prezentovan jedan od načina za bezbedonosno osiguranog ulaska u prostoriju. Rešenje se bazira na korišćenju već postojećih i dostupnih tehnologija poput mobilnih uređaja, mikrokontrolera i Bluetooth modula. Ovakva kombinacija dovodi do novih mogućnosti osiguravanja prostorija kao i unapređivanja sistema bezbednosti. Sistem je planiran da se proširi tako što će se korisnik samo jednom logovati a zatim kada je u blizini, elektronska brava će automatski registrovati korisnika i otključati bravu. Osim već navedenih stvari svrha ovog rada je primena teorije u praksi koju smo savladali kroz predavanja, računskih i laboratorijskih vežbi na fakultetu.

## Zahvalnica

Autori rada žele da izraze posebnu zahvalnost organizaciji SICEF što je podržala obezbedivši materijalna sredstva neophodna za praktičnu realizaciju ovog projekta.

## Literatura

[1] http://www.itdesk.info/srp/edukacija/Osnovni%20pojmovi%20informacione%20i%20komunikacione%20tehnologije%20skripta.pdf
[2] https://www.android.com/
[3] https://sh.wikipedia.org/wiki/Bluetooth
[4] https://www.engineersgarage.com/electroniccomponents/16x2-lcd-module-datasheet
[5] https://store.arduino.cc/arduino-uno-rev3
[6] http://www.atmel.com/Images/Atmel-42735-8-bit-AVR Microcontroller-ATmega328-328P_Datasheet.pdf
[7] https://en.wikipedia.org/wiki/Altium_Designer
[8] https://en.wikipedia.org/wiki/JSON
[9] https://en.wikipedia.org/wiki/Application_programming_interface
[10] Jason Price, Visual C# .NET, Kompjuter biblioteka, 2003
[11] Dmitri Korotkevitch, Pro SQL Server Internals 2nd ed. Apress, 2017
[12] https://msdn.microsoft.com/enus/library/hh833994(v=vs.108).aspx
[13] https://en.wikipedia.org/wiki/Object-relational_mapping
[14] Julia Lerman, Programming Entity Framework 2nd Edition, O'Reilly Media, 2010

# Elektronski istorijski vodič kroz grad Niš

Nemanja Petrašinović, Marija Trajković, Marko Pešterac

*Sadržaj* – U ovom radu je prikazana realizacija mobilne Android aplikacije istorijskog vodiča kroz grad Niš. Aplikacija omogućava prikazivanje informacija o istorijskim spomenicima i lokalitetima koji se nalaze u Nišu, kao i putanju do istih.

## I. UVOD

Jedan od vodećih problema vezanih za istorijske spomenike i istorijske lokalitete koji se nalaze na teritoriji grada Niša jeste jako oskudno obeležavanje ovih objekata, kao i jako malo pruženih informacija građanima i turistima o datom istorijskom objektu ili lokalitetu. Jedan od načina na koji je prikaz informacija o ovim objektima u gradu rešen je postavljanje tabli sa infrmacijama o istorijskom objektu. Međutim table sa informacijama mogu biti fizički oštećene i informacije mogu postati nečitljive. Aplikacija elektronskog istorijskog vodiča kroz grad Niš omugućava brzo i jednostavno prikazivanje informacija o istorijskim objektima lokalizovanim u gradu Nišu. Aplikacija se sastoji iz dva dela, korisničkog i serverskog, koji će biti opisani kasnije u radu. Do sada za grad Niš slična rešenja nisu postojala. Jedno od rešenja za prikazivanje informacija o istorijskim objektima u gradu jeste mini muzej elektronskog fakulteta u kome su izložene slike i informacije o istorijskim lokalitetima. Motivacija za razvoj mobilne aplikacije istorijskog vodiča kroz grad Niš je ta što ne postoje slična rešenja u ovoj oblasti. Prednost ove aplikacije je ta što je svakom turisti koji dolazi u grad aplikacija dostupa kroz nekoliko klikova. Aplikacija je razvijena tako da poznaje princip gejmifikacije [1], što je velika prednost u odnosu na ostala rešenja, jer razvija takmičarski duh kod turista. Takođe zbog nagrada koje turisti dobijaju na osnovu osvojenih poena, koje dobijaju posetom određenog istorijskog objekta, razvija se i želja za obilaskom što više istorijskih objekata. Ovaj princip pokazao se kao jako koristan. Mnoge aplikacije koriste princip gejmifikacije kako bi svoj sadržaj korisniku predstavile na zanimljiviji način. Jedna od najpoznatijih aplikacija je Pokemon Go [2]. Zbog poznavanja gejmifikacije aplikacija zahteva da svaki korisnik aplikacije napravi svoj profil. Preko svog profila korisnici mogu videti svoje bodove, ali i bodove drugih korisnika što korisnike aplikacije motiviše da obiđu što više istorijskih objekata. Još jedna velika prednost aplikacije elektronskog vodiča kroz grad Niš je to što je vodič implementiran kao mobilna aplikacija, što omogućava da korisnicima aplikacija bude uvek dostupna na njihovom mobilnom telefonu.

Nemanja Petrašinovic, Marija Trajković, Marko Pešterac studenti na modulu Računarstvo i informatika, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija. Email: nemanjapet@elfak.rs, markopesterac@elfak.rs, marija.trajkovic@elfak.rs

## II. OPIS APLIKACIJE

Aplikacija elektronskog vodiča kroz grad Niš se sastoji iz dva dela. Prvi deo je klijentski deo aplikacije, odnosno deo aplikacije koji je instaliran na korisnikovom mobilnom uređaju sa Android operativnim sistemom. Drugi deo aplikacije je serverski deo. Zadatak serverskog dela je da čuva informacije o istorijskim lokalitetima kao i o korisnicima, kao i da na zahtev podatke iz baze prosledjuje do korisnika.

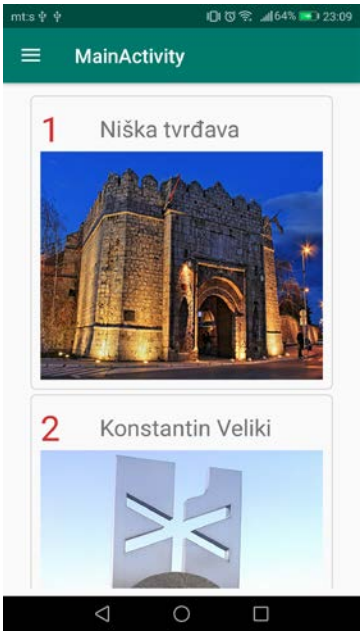
### *A. Klijentski deo aplikacije*

Klijentski deo elektronskog istorijskog vodiča kroz Niš realizovan je korišćenjem Android studia i programskog jezika Java. Klijentski deo predstavlja Android aplikaciju koja se može preuzeti na mobilni uređaj korisnika. Pokretanjem Android aplikacije elektronskog istorijskog vodiča kroz grad Niš korisniku se prikazuje forma za registraciju ili prijavljivanje na sistem (Slika 1). Registracija odnosno prijavljivanje na sistem je potrebana kako bi korisnik mogao da vidi broj obiđenih istorijskih lokaliteta, kao i broj osvojenih poena što će biti detaljnije objašnjeno kasnije u ovom radu. Nakon uspešnog prijavljivanja ili registracije korisniku se prikazuje lista istorijskih spomenika i lokaliteta na teritoriji grada Niša (Slika 2). Korisnik ima mogućnost pregleda liste, a klikom na neki objekat iz liste korisniku se prikazuju informacije o izabranm mestu. Pored pregleda informacija korisnik ima mogućnost pregledanja svoje lokacije, lokacije objekta kao i međusobnu udaljenost i putanju kojom se stiže do određenog istorijskog objekta. Mobilna aplikacija sadrži i servis koji se izvršava u pozadini čak i kada aplikacija nije aktivna. Pozadinski servis pristupa GPS senzoru mobilnog uređaja i na osnovu pozicije korisnika računa udaljenost od istorijskih objekata. Korisnici su obavešteni o postojanju pozadinskog servisa jer se prilikom pokretanja aplikacije od korisnika zahteva da odobri pristup GPS lokaciji. Korisnik takođe ima mogućnost isključivanja pozadinskog servisa preko dugmeta u aplikaciji koje je jasno obeleženo. Ukoliko se korisnik nađe na udaljenosti manjoj ili jednakoj pedeset metara od nekog istorijskog objekta, servis će obavestiti korisnika. Obaveštenje korisnika predstavlja notifikaciju sa zvukom i vibracijom koju korisnik može

pogledati u listi notifikacija, i saznati blizu kog istorijskog objekta ili lokaliteta se nalazi. Korisnik ima mogućnost jednostavnog uključivanja i isključivanja notifikacija.




Slika 1. Forma za prijavljivanje.




Slika 2. Prikaz liste spomenika i lokaliteta.

*B. Pojam gejmifikacije*

Gejmifikacija predstavlja primenu elemenata preuzetih iz dizajna igara i igračkih principa na sadržaj koji ne predstavlja video igru. Može se reći da je suština gejmifikacije da se prilikom formiranja aplikacije datoj oblasti podari igrački karakter dodavanjem određenih elemenata, a da se pritom sam sadržaj ne pretvara u igru. Način na koji su principi gejmifikacije primenjeni u aplikaciji elektronskog istorijskog vodiča kroz grad Niš objašnjen je u daljem tekstu.

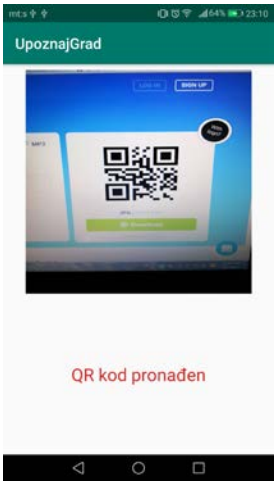
Svaki korisnik aplikacije ima svoj profil. U profilu korisnika čuvaju se osnovne informacije o korisnku. Na skup informacija o korisnku dodate su informacije o broju poena koje je korisnik prikupio kao i lista istorijskih mesta koje je korisnik obišao (Slika 6). Na svom android uređaju korisnik aplikacije može pristupitu kvizu i odgovarati na postavljena pitanja o istorijskom objektu ili lokalitetu do kog je došao. Poziciju korisnika aplikacija prepoznaje na osnovu prikupljenih informacija o geografskoj širini i geografskoj dužini sa GPS senzora mobilnog uređaja. Da bi korisnik startovao kviz vezan za istorijsko mesto pored kog se nalazi potrebno je da skenira QR kod postavljen pored datog mesta. Skeniranjem QR koda aplikacija prepoznaje mesto do kog je korisnik došao i otvara kviz sa pitanjima o datom istorijskom mestu. Odgovaranjem na pitanja korisnik osvaja poene. Kompletiranjem kviza broj poena se čuva u profilu korisnika u bazi podataka, a u listu obiđenih mesta se ubacuje mesto za koje je korisnik kompletirao kviz. Na osnovu osvojenih poena korisnici se rangiraju. Rang listi korisnik može pristupiti iz aplikacije koju ima instaliranu na svom mobilnom uređaju (Slika 5). Na opisan način korisnici su motivisani da istražuju istorijska mesta i lokalitete u gradu i pritom saznaju mnoštvo korisnih informacija. Korisnici koji od 47 ponuđenih lokaliteta obiđu više od 40 lokaliteta dobijaju zlatnu medalju poznavaoca grada Niša. Korisnici koji obiđu više od 30 lokaliteta osvajaju srebrnu, dok korisnici koji obiđu više od 15 mesta osvajaju bronzanu medalju.

Princip gejmifikacije jako je dobro prihvaćen od strane korisnika. Mnoge najpopularnije aplikacije koriste principe gejmifikacije. Jedna od njih je Pokemon Go. U svetu se takođe mogu naći mnoge aplikacije koje koriste ovaj princip. Teme aplikacija na koje se ovaj princip može primeniti jako je širok. Jedna od tema je ta da korisnici osvajaju bodove ukoliko dnevno pretrče određen broj kilometara, aplikacija Zombies, Run! [3] omogućava korisnicima da se na zanimljiv način posvete sportskim aktivnostima.

*C. Skeniranje i kompresija QR koda*

QR kod [4] (Quick Responsive) je matrični kod kreiran 1994. godine.QR je akronim za Quick Response (brz odziv) pošto je tvorac nameravao da se sadržaj koda jako brzo čita. Prvenstveno je namenjen za praćenje delova u proizvodnji vozila ali se sada koristi i u mnogo širem kontekstu, komercijalne aplikacije praćenja, kao i aplikacije koje su orijentisane na pogodnije korišćenje mobilnih telefona (poznato kao mobilno tagovanje). QR kodovi koji skladište adrese i URL-ove, mogu se pojavljivati u časopisima, na znacima, autobusima, vizit kartama, kao i na svakom objektu o kome bi korisniku mogle biti potrebne informacije.

QR kod sa informacijama o lokalitetu koje je korisnik posetio postavljen je pored lokaliteta. Skeniranjem QR koda (Slika 3) pokreće se kviz (Slika 4). Međutim QR kod korišćen u elektronskom istorijskom vodiču je kompresovan kako bi sadržao manje bitova podataka, na taj način QR kod će biti lakši za skeniranje na mobilnim uređajima sa lošijim kamerama [5]. Kako QR kod za svaki podatak koji se koristi u generisanju QR koda kodira sa 8 bitova, ukoliko utvrdimo da nam za ispisivanje informacija o lokalitetu nije potreban ceo skup slova, brojeva i karaktera možemo kompresovati QR kod. Odredili smo da nam je za prikaz informacija na QR kodu potrebno samo 30 slova, bez brojeva i sa jednim specijalnim karakterom. Na taj način svaki podatak u QR kodu možemo predstaviti korišćenjem 5 bitova umesto 8 i tako kompresovati količinu podataka sadržanih u kodu.



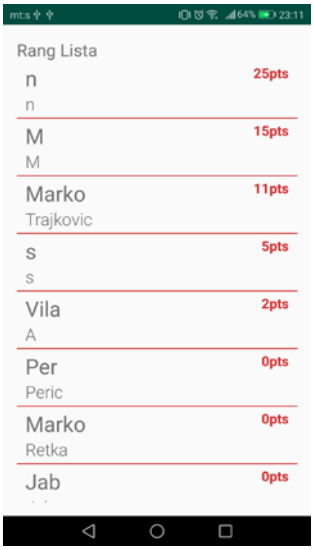

Slika 3. Skeniranje QR koda.




Slika 4. *Layout* kviza.

*D. Serverski deo*

Serverski deo elektronskog istorijskog vodiča kroz Niš implementiran je korišćenjem Firebase [6] servera. Za autentifikaciju korisnika koristi se Fireabse autentifikacija, dok se informacije o korisniku pamte u Firebase bazi podataka. U Firebase bazi podataka zapamćene su i informacije o istorijskim objektima i lokalitetima kao i pitanja za iste koja se koriste u kvizu. Prilikom pokretanja aplikacije na mobilnom uređaju aplikaicja se povezuje sa Firebase serverom. Klikom na neko od istorijskih mesta iz liste aplikacije, aplikacija šalje zahtev za pribavnjanje informacija Firebase serveru. Server klijentskom delu aplikacije prosleđuje zahtevane informacije koje se zatim prikazuju korisniku. Serverska komponenta takođe prosleđuje sortiranu listu korisnika prema osvojenim poenima na kvizu klijentskoj komponenti elektronskog istorijskog vodiča kroz Niš. Na taj način korisnik aplikacije ima uvid koji od korisnika ima najveći broj poena.



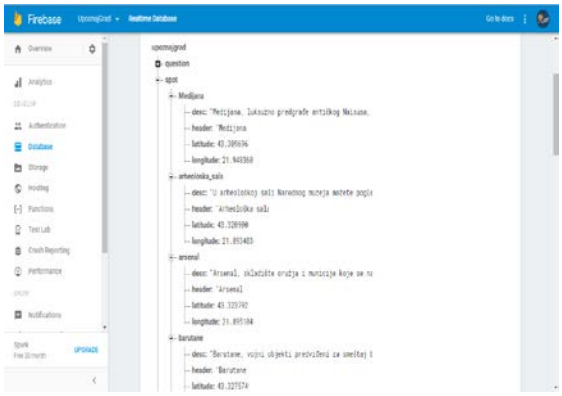

Slika 5. Rang lista.




Slika 6. Lista mesta koja je korisnik obišao.

Firebase baza podataka implementirana je kao key-value [7] baza podataka, što znači da se za određeni ključ u

bazi podataka pamte potrebne vrednosti. Ovaj tip baze podataka predstavlja NoSQL [8] bazu podataka. Za potrebe ove aplikacije baza podataka na Firebase serveru implementirana je tako da sadrži ključ za svaki istorijski objekat. Svakom od ključeva u bazi podataka dodate su odgovarajuće vrednosti koje opisuju objekat. U ovom slučaju to su desc vrednost koja predstavlja opis istorijskog objekta, header vrednost koja predstavlja ime istorijskog objekta zatim vrednosti longitude i latitude koje predstavljaju geografsku širinu i geografsku dužinu na kojoj se objekat nalazi. Prilikom skeniranja QR koda mobilni deo aplikacije pristupa Firebase serveru i na osnovu dekodiranog QR koda, koji predstavlja ključ u bazi podataka, zahteva informacije koje su pridružene tom ključu. Na slici 7 može se videti na koji način su ključevi u bazi podataka povezani sa vrednostima.




Slika 7. Način povezanih ključeva sa vrednostima.

## ZAKLJUČAK

Mobilni uređaji vremenom postaju sve moćniji i lako se mogu koristiti u različite svrhe. Jedna od primena jeste i prikazivanje informacija o gradu. Mobilni uređaji su svuda zastupljeni i korinici su dobro upoznati sa načinom korišćenja ovakvih urežaja. Za mobline uređaje korisnika mogu se praviti aplikacije koje mogu imati različitu primenu. Motivacija pravljenja aplikacije elektronskog vodiča kroz Niš bila je da se na jedan zanimljiv način turistima prikažu informacije o gradu. Takođe ova aplikacija može imati jako velikog značaja za decu jer korišćenjem aplikacije i međusobnim takmičenjem u odgovaranju na pitanja iz kviza deca mogu učiti istoriju grada. Jako značajno je to što se jedna ovakva mobilna aplikacija može lako prilagoditi za druge gradove svuda u svetu.

## LITERATURA

[1] https://homepage.rs/novosti/gejmifikacija/
[2] https://www.pokemongo.com/
[3] https://zombiesrungame.com/
[4] https://sr.wikipedia.org/sr-el/QR_kod
[5] http://www.ijcttjournal.org/Volume15/number-4/IJCTT-V15P134.pdf
[6] https://firebase.google.com/
[7] https://en.wikipedia.org/wiki/Key-value_database
[8] https://en.wikipedia.org/wiki/NoS

# Fooling a neural network with common adversarial noise

Nikola Popović, Marko Mihajlović

*Abstract* - these days deep Neural Networks (NN) show exceptional performance on speech and visual recognition tasks. These systems are still considered a black box without deep understanding why they perform in such a manner. This lack of understanding makes NNs vulnerable to specially crafted adversarial examples - inputs with small perturbations that make the model misclassify. In this paper, we generated adversarial examples that will fool a NN used for classifying handwritten digits. We start by generating additive adversarial noise for each image, then we craft a single adversarial noise that misclassifies different members of the same class.

## I. INTRODUCTION

Artificial intelligence (AI), especially its subfield of machine learning (ML) became very popular over the last few years. A lot of experts today are saying that the next big breakthrough in AI will mark the new industrial revolution. There is more and more application of AI in many different industries as time goes by, which indicates that we should be very concerned about safety.

In the 2014, Szegedy et al. [1] discovered that you could make Adversarial Examples (AE) by adding carefully crafted small perturbations to the input and fool several ML models, including NN that have high accuracy on previously unseen data. Goodfellow et al. [2] explained this as a result of the models being too linear. With that kind of attack you could make an autonomous vehicle misclassify a stop sign, fool an intrusion-detection system, fool an identification system and so on. Most popular approaches for crafting AE are: the fast gradient sign method [2, 3] and Papernot et al. method [6]. On the other side, best strategies that are currently known for making models more robust to adversarial examples are: adversarial training [1, 2] and defensive distillation [7]. It turns out that none of these are fully successful and that an attacker can always find a new way to synthesize AE to overcome these defenses. One interesting type of attack that overcomes these defenses without knowing the models internals is the black-box attack [4]. It is also shown that it is possible to make AE that are robust to rescaling, translation and rotation [9]. Generating AE and defending against them is still an open research topic.

Goals of attackers are diverse. Some simply want the model to make a mistake while others aim to achieve class-targeted misclassification. There are relevant scenarios where an attacker can get a time limited access to the model internals. In those scenarios, he can create a common adversarial noise that can later be used to make the model misclassify any member of a class. We explore that idea later in the paper.

In section 2, we describe the NN that we fool with AEs. In section 3, we present algorithms for generating adversarial noise and show their results. In the last section, we conclude the paper with a few of our thoughts.

The code, which was used for all experiments, can be found at: https://github.com/Maki94/cnn_adv_examples.

## II. NEURAL NETWORK

In recent years Neural Networks (NN) have become widely popular due to their universal character (used for supervised, unsupervised and reinforcement learning) and increasing affordability of powerful distributed systems capable of processing large amount of data. A NN is composed of layers which contain units called neurons. Each neuron is parametrized by its weight vector $w_{ij}$. The NN also has a nonnegative cost function $J(W)$ assigned to it. During the learning phase, a network is given training data, input-output pairs $(x, y)$, and the goal is to adjust the set of weights $W$ in a way that minimizes the cost function $J(W)$ by using some optimization algorithm.

First, we need a classifier that we want to fool. We use a specific type of NN - Convolutional Neural Network (CNN) to classify images of handwritten digits. We used the famous MNIST dataset [11] which contains 70,000 28x28 pixel images of digits. Each pixel has one of 256 intensity values evenly distributed between 0 and 1. We divide the data into the training set (55,000 images), validation set (5,000 images) and the test set (10,000 images).

We trained our CNN on the training set and evaluated it on the test set. The validation set will be used later for crafting the adversarial noise shared amongst the whole class. The network has the architecture described in Table I.

Nikola Popović is with the Department of Signals and Systems, Faculty of Electrical Engineering, University of Belgrade, Bulevar Kralja Aleksandara 74, 11000 Belgrade, Serbia, E-mail: pn173042m@student.etf.bg.ac.rs

Marko Mihajlović is with the Department of Computer Science, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: marko.mihajlovic@elfak.rs

The network was implemented in Python 3.5, using the *tensorflow* library. For the cost function we used cross-entropy. We minimized it using the Adam optimizer [5]. The training lasted 20,000 iterations and the weights were updated using a random batch of size 50 in each iteration. After the training, the NN had an accuracy of 99.19% on the test set.

TABLE I
ARCHITECTURE OF THE CNN CLASSIFIER

| Layer type | Layer parameters |
|---|---|
| Input | 28x28 image |
| Convolutional | 32 filters(5x5); ReLU activation |
| Max pooling | 2x2 |
| Convolutional | 64 filters(5x5); ReLU activation |
| Max pooling | 2x2 |
| Fully connected | 1204 neurons; ReLU activation |
| Dropout | $p = 0.5$ |
| Fully connected | 10 neurons; Softmax activation |

## III. ADVERSARIAL NOISE

In the first hidden layer each neuron computes the dot product between the input $x$ and its weight vector $w_{ij}$ and then applies an activation function to the product. If we take the input, add a small perturbation and align its signs with the signs of $w_{ij}$ along each dimension, we get an increase in the value of the dot product which is proportional to the dimensionality of the data. If the dimensionality is big enough, many small changes can add up to one big change. This can make the wrong neurons fire or stop firing, which can result in misclassification. The bigger the dimension of the data, the smaller the perturbation needed to make the model misclassify.

While generating adversarial examples we used the following iterative algorithm:

$$x_{adv}^{i+1} = Clip_{x,\varepsilon}\{x_{adv}^{i} + \alpha \nabla_x J(x_{adv}^{i}, y_{true})\}$$
$$x_{adv}^{0} = x \tag{1}$$

The algorithm stops when $x_{adv}^{i+1}$ achieves a misclassification, or when it reaches the maximal number of iterations. Function $Clip_{x,\varepsilon}\{\cdot\}$ is a pixel-wise clipping of the image that assures the AE stays in the $\varepsilon$-neighborhood of the original image:

$$Clip_{x_j,\varepsilon}\{x_j^i\} = \begin{cases} x_j + \varepsilon\,, & x_j^i > x_j + \varepsilon \\ x_j - \varepsilon\,, & x_j^i < x_j - \varepsilon \\ x_j^i\,, & otherwise \end{cases} \tag{2}$$

In other words, we want to maximize the cost function over the input image, but we don't want the adversarial image to differ too much from the original one.

On the other side, to achieve target-label misclassification, the following iterative algorithm can be used:

$$x_{adv}^{i+1} = Clip_{x,\varepsilon}\{x_{adv}^{i} - \alpha \nabla_x J(x_{adv}^{i}, y_{target})\}$$
$$x_{adv}^{0} = x \tag{3}$$

Now, we are minimizing the adversarial cost function over the input image, which pushes the model to classify the AE as $y_{target}$. Also, the algorithm could be modified to stop when a specified classification confidence is reached, instead of stopping as soon as the model outputs $y_{target}$.

We didn't use one of the popular gradient sign methods that are presented in [3], because besides the information about the direction in which to move that is contained in the gradients sign, its values inform us how relevant each pixel is for making the model misclassify. This means that the intensity values of pixels that are not too relevant won't change as much as the intensity of the relevant ones. We think that this will result in a less damaged image while paying the price of more iterations. Analyzing the difference in their performance can be a topic for further investigation.

To measure how strong the noise $e$ is, we will use the Root Mean Square (RMS) metric:

$$RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2} \tag{4}$$

$$e = x_{adv} - x \tag{5}$$

We made AE's for the whole test set using the algorithm described in (1). We used the clipping value of $\varepsilon = 0.15$ and the learning rate of $\frac{10}{255}\frac{1}{m}$, where $m$ is the maximum absolute value in the gradient vector. We generated the noise using 200 iterations for the maximum iteration parameter. In the 98.6% of the cases, the model makes a mistake, while the average RMS of the noise was 0.063. An example of a misclassified image is shown in Fig. 1.



Fig. 1. Left: original image. Right: AE.

As we can see, an average human would still label the image as a seven, but the NN, which has a very high accuracy on previously unseen data, made a mistake.

Let's now imagine an attacker gets a time limited access to a protected model. He can use that time to learn a common adversarial noise that can fool the model when applied to different members of the same class. We propose an algorithm for crafting such a noise on a set of N members from the same class:

$e = \vec{0}$
**for** $j = 1$ **to** *epochs*:
  **for** $i = 1$ **to** $N$:
    $e := Clip_{0,\varepsilon}\{e + \alpha \nabla_x J(x^{(i)} + e, y_{true})\}$
  **end**
**end**

Alg. 1: Algorithm for finding the common adversarial noise.

This is similar to stochastic gradient descent. When we craft an adversarial noise $e$ for a single image, we are updating the noise using only that image until we end up with a misclassification. Here, we update the noise by a little using each image and repeat that process until we find a good enough noise. By updating it a little on each image, we are forcing it to find a weak spot that is common for all the data points in that class.

We generate common adversarial noise for the class of 5's on the data in the validation set over 25 epochs with the learning rate of $\frac{1}{255}\frac{1}{m}$. We generate the noise for different values of the clipping value $\varepsilon$. For each generated noise, we apply it to all of the data points in the test set from the class of 5's and measure its success rate of fooling the network and the RMS of the noise signal. Results are shown in Figures 2 and 3.

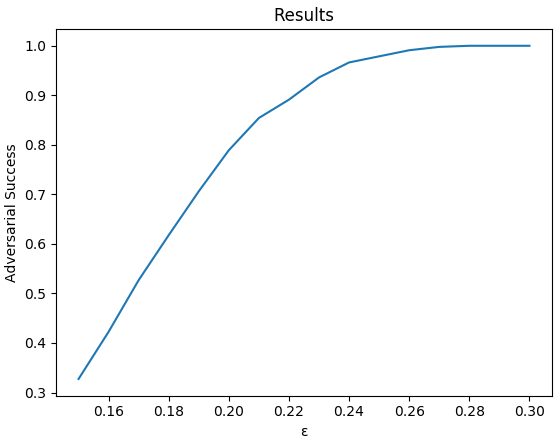



Fig. 2. Adversarial success for different clipping values.

Fig. 2 depicts that for $\varepsilon = 0.24$ and above the model is fooled most of the time. As we can see from Fig. 3. the RMS of the noise rises almost linearly with $\varepsilon$. This makes sense because as we loosen the box constraint, the perturbation get bigger in order to achieve a larger adversarial success.

We then generate common adversarial noise for all the classes, on the validation set, over 25 epochs with the clipping value of $\varepsilon = 0.24$, the learning rate of $\frac{1}{255}\frac{1}{m}$ and evaluate it on the test set. Results are shown in Fig. 4 and Fig. 5.

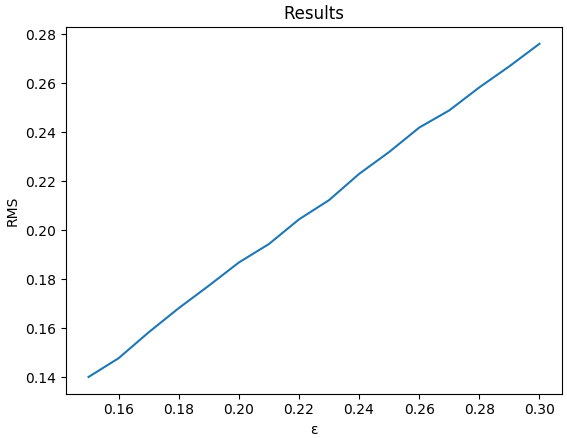



Fig. 3. RMS for different clipping values.

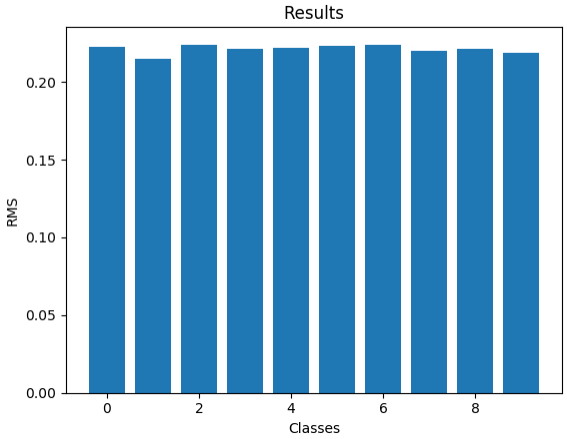



Fig. 4. RMS for different classes with the clipping value of $\varepsilon = 0.24$.

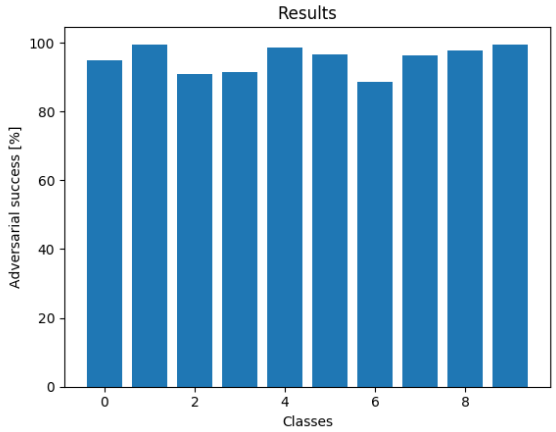



Fig. 5. Adversarial success for different classes with the clipping value of $\varepsilon = 0.24$.

We infer that the algorithm can make a single noise that we can apply to different inputs from the same class,

which were not involved in its making, and still fool the model most of the time. As Fig. 4. and Fig. 5. depict it is possible to create a successful common noise for each class. Also, adversarial success and RMS do not vary too much for all the classes. This tells us that making common noise for different classes is a problem of approximately the similar difficulty for digit classification.

This is a serious safety problem. The noise has a larger RMS than the noise crafted for one specific input, because crafting a more robust noise requires larger perturbations. A few AE with the generated common noise are shown in Fig. 6.

As we can see in Fig. 6, now AE are much more damaged than in the case of generating an AE on a single digit. However, a person would still classify them as their correct label or maybe discard them with an explanation that they are too damaged and do not resemble any digit. We think that this is because MNIST digits have a low resolution and that the perturbations would be much smaller on images with higher resolution. This could be a topic for future investigation.



Fig. 6. Above: original images. Below: AE.

## IV. Conclusion and Future Work

In this work, we have generated AE similar to the original data that are able to fool a CNN without a problem. Afterwards, we proved that it is possible to create a common adversarial noise that can be used to fool a CNN with great success. The purpose of this paper is to increase awareness about the drawbacks of modern ML algorithms that are used increasingly in all walks of life.

While working on this paper, we stumbled upon a few questions that we think are worthy of investigation. Maybe this can inspire some other authors to work on those problems, which are:

- Analyzing the difference in the performance of algorithms (1, 3) from the section 2 and the gradient sign algorithms.
- Crafting common adversarial noise on images that have a larger resolution and multiple color channels.
- Trying to craft common adversarial noise in the purpose of target-label misclassification.
- Trying to make common adversarial noise that is robust to rescaling, rotation and translation

## Acknowledgement

The authors would like to thank Goran Dubajić and Bruno Gavranović for introducing us to the topic of AE and providing advice throughout experiments.

## References

[1] Christian Szegedy, et al., Intriguing properties of neural networks. In Proc. Proceedings of the International Conference on Learning Representations, 2014.

[2] Ian J Goodfellow, et al. Explaining and harnessing adversarial examples. In Proceedings of the International Conference on Learning Representations, 2015.

[3] Alexey Kurakin, et al. Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533, 2016.

[4] N. Papernot, et al. Practical black-box attacks against deep learning systems using adversarial examples. arXiv preprint arXiv: 1602.02697, 2016.

[5] Diederik P. Kingma, et al. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference for Learning Representations, San Diego, 2015.

[6] N. Papernot, et al. The limitations of deep learning in adversarial settings. In Proceedings of the 1st IEEE European Symposium on Security and Privacy. IEEE, 2016.

[7] Nicolas Papernot, et al. Distillation as a defense to adversarial perturbations against deep neural networks. In Proceedings of the 37th IEEE Symposium on Security and Privacy.

[8] N. Papernot, et al. Towards the Science of Security and Privacy in Machine Learning. arXiv preprint arXiv: 1611.03814, 2016.

[9] A. Athalye, et al. Synthesizing Robust Adversarial Examples. arXiv preprint arXiv: 1707.07397, 2017.

[10] A. Krizhevsky, et al. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, 2012, pp. 1097–1105.

[11] Y. LeCun et al. The mnist database of handwritten digits. 1998.

# Dance style recognition using SVM Classificators

Fuad Hajdarević

***Abstract*— This paper deals with dance style pattern recognition using audio samples of songs . It serves to give the reader introduction into the field of music and dance style pattern recognition. Main method described in the paper is a technique in machine learning called Support Vector Machine, which is a supervised learning model that analyze data used for classification and regression analysis.**

## I. INTRODUCTION

Like many other social hobbies, dancing is an important part of social activity. Every dance type depends on rhythm and the rhythmic patterns, called claves. Every latin dance use 4/4 rhythm, but the difference between the styles depends on the tempo and how many beats are in one measure. Lot of people dont have an ear for rhythm or music, so sometimes it makes a problem to recognize which dance style to dance on which music. Searching this thematic, we found out that there exist similar applications, which deal with dance style recognition [2], but with an another approach. Many of these applications are dealing with recognizing dance styles like classic dances ( Viennese waltz, slow waltz, tango, foxtrot...) or latin competition dances (samba, rumba, cha cha,...). Methods used in recognition are PCA (Principal Component Analysis), Support Vector Machines, Matched Filters and others. Many authors consider PCA and Support Vector Machines as the most succesful methods to use for recognition, but the success depends from type to type of applications.

In preparing this paper, we have used PCA and Support Vector Machines for our application and the better results we matched with Support Vector Machines classificators, so we continued our work with them. In the next chapter of our paper we will explain SVM classification, also their using in other non similar applications.

## II. AUDIO RECOGNITION

Audio recognition is a field in digital signal processing, which is very popular for researching nowadays and every type of audio recognition use different methods for recognizing. For example, if we hear a song in the radio, but we do not know the name of the song or the singer, we use applications for recognizing the song. There are several methods used for an algorithm such like these. Usually, they use Matched filters for recognizing, because they only need a sample to compare with a database, and if the sample matched with a sample of the database, we get the searching information. In other cases, if we want to categorize samples, we must use classificators for traning. In this paper, we choosed Support Vector Machines classificators.

[1]Fuad Hajdarević is with Faculty of Electrical Engineering, Automatic Control and Electronics Department, University of Sarajevo, 71000 Sarajevo, Bosnia and Herzegowina, fh15279@etf.unsa.ba.

## III. SUPPORT VECTOR MACHINE

### A. Motivation

[1] Classifying data is a common task in machine learning. Suppose some given data points each belong to one of two classes, and the goal is to decide which class a new data point will be in. In the case of support vector machines, a data point is viewed as a *p-dimensional vector*, and we want to know whether we can separate such points with a $(p-1)$ *dimensional hyperplane*. This is called a linear classifier. There are many hyperplanes that might classify the data. One reasonable choice as the best hyperplane is the one that represents the largest separation, or margin, between the two classes. So we choose the hyperplane so that the distance from it to the nearest data point on each side is maximized. If such a hyperplane exists, it is known as the maximum-margin hyperplane and the linear classifier it defines is known as a maximum margin classifier, or equivalently, the perceptron of optimal stability. In Figure 1 is illustrated graphically how the hyperplanes make the seperation between the classes .

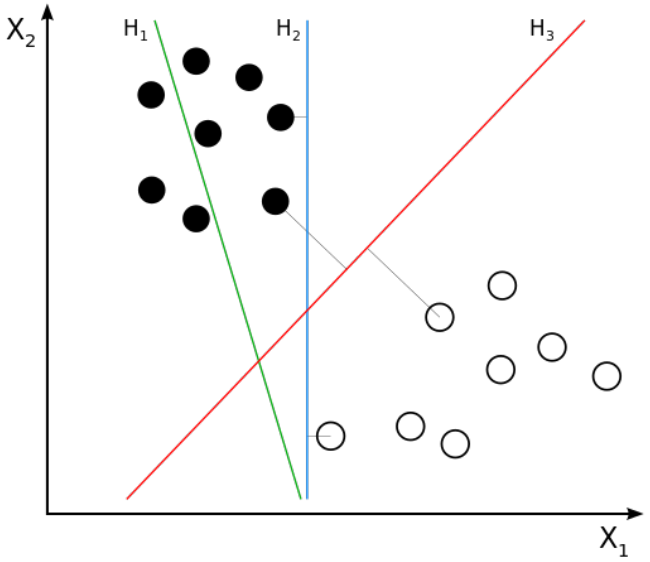



Figure 1. H1 does not separate the classes. H2 does, but only with a small margin. H3 separates them with the maximum margin.

### B. Definition

[2] More formally, a support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is

achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier. Whereas the original problem may be stated in a finite dimensional space, it often happens that the sets to discriminate are not linearly separable in that space. For this reason, it was proposed that the original finite-dimensional space be mapped into a much higher-dimensional space, presumably making the separation easier in that space. To keep the computational load reasonable, the mappings used by SVM schemes are designed to ensure that dot products may be computed easily in terms of the variables in the original space, by defining them in terms of a *kernel function k (x,y)* selected to suit the problem. The hyperplanes in the higher-dimensional space are defined as the set of points whose dot product with a vector in that space is constant. The vectors defining the hyperplanes can be chosen to be linear combinations with parameters $\alpha_i$ images of feature vectors $x_i$ that occur in the data base. With this choice of a hyperplane, the points $x$ in the feature space that are mapped into the hyperplane are defined by the relation: $\sum_i \alpha_i k(x_i, x) = constant$. Note that if $k(x, y)$ becomes small as $y$ grows further away from $x$, each term in the sum measures the degree of closeness of the test point $x$ to the corresponding data base point $x_i$. In this way, the sum of kernels above can be used to measure the relative nearness of each test point to the data points originating in one or the other of the sets to be discriminated. Note the fact that the set of points $x$ mapped into any hyperplane can be quite convoluted as a result, allowing much more complex discrimination between sets which are not convex at all in the original space.

### C. Mathematical formulations: Separable data

*1)* ***Primal****:* The data for training is a set of points (vectors) $x_i$ along with their categories $y_i$. For some dimension d, the $x_i \in R^d$, and the $y_i = \pm 1$. The equation of a hyperplane is

$$< w, x > +b = 0 \tag{1}$$

where $w \in R^d, < w, x >$ is the inner (dot) product of $w$, and $b$ is real.

The following problem defines the best separating hyperplane. Find $w$ and $b$ that minimize $||w||$ such that for all data points $(x_i, y_i)$.

$$y_i(< w, x > +b) \geq 1 \tag{2}$$

The support vectors are the $x_i$ on the boundary, those for which $y_i(< w, x > +b) = 1$. For mathematical convenience, the problem is usually given as the equivalent problem of minimizing $< w, w > /2$. This is a quadratic programming problem. The optimal solution $(\hat{w}, \hat{b})$ enables classification of a vector $z$ as follows:

$$class(z) = sign(< \hat{w}, x > +\hat{b}). \tag{3}$$

*2)* ***Dual****:* It is computationally simpler to solve the dual quadratic programming problem. To obtain the dual, take positive Lagrange multipliers i multiplied by each constraint, and subtract from the objective function:

$$L_p = \frac{1}{2} < w, w > - \sum_i \alpha_i (y_i(< w, x_i > +b) - 1) \tag{4}$$

where you look for a stationary point of $L_P$ over $w$ and $b$. Setting the gradient of $L_P$ to 0, you get

$$w = \sum_i \alpha_i y_i x_i \tag{5}$$

$$0 = \sum_i \alpha_i y_i \tag{6}$$

Substituting into $L_P$, you get the dual $L_D$:

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j < x_i, x_j > \tag{7}$$

which you maximize over $\alpha_i 0$. In general, many $\alpha_i$ are 0 at the maximum. The nonzero $\alpha_i$ in the solution to the dual problem define the hyperplane, as seen in Equation 7, which gives $w$ as the sum of $\alpha_i x_i y_i$ . The data points $x_i$ corresponding to nonzero i are the *support vectors*. The derivative of $L_D$ with respect to a nonzero $\alpha_i$ is 0 at an optimum. This gives

$$y_i(< w, x_i > +b) - 1 = 0. \tag{8}$$

In particular, this gives the value of $b$ at the solution, by taking any $i$ with nonzero $\alpha_i$.The dual is a standard quadratic programming problem.

### D. Mathematical formulations: Nonseparable Data

*1)* ***Primal****:* Your data might not allow for a separating hyperplane. In that case, SVM can use a *soft margin*, meaning a hyperplane that separates many, but not all data points.

There are two standard formulations of soft margins. Both involve adding slack variables $s_i$ and a penalty parameter $C$. The $L^1$-norm problem is:

$$\min_{w,b,s} (\frac{1}{2} < w, w > +C \sum_i s_i) \tag{9}$$

such that

$$y_i(< w, x_i > +b) \geq 1 - s_i \tag{10}$$

$$s_i \geq 0 \tag{11}$$

The $L^1$-norm refers to using $s_i$ as slack variables instead of their squares. The three solver options SMO, ISDA, and L1Qp of *fitcsvm* minimize the $L^1$-norm problem.
The $L^2$-norm problem is:

$$\min_{w,b,s} (\frac{1}{2} < w, w > +C \sum_i s_i^2) \tag{12}$$

subject to the same constraints.
In these formulations, you can see that increasing C places more weight on the slack variables $s_i$, meaning the optimization attempts to make a stricter separation between classes. Equivalently, reducing $C$ towards 0 makes misclassification less important.

*2) **Dual**:* For easier calculations, consider the $L^1$ dual problem to this soft-margin formulation. Using Lagrange multipliers $\mu_i$, the function to minimize for the $L^1$-norm problem is:

$$L_p = \frac{1}{2} < w, w > +C \sum_i s_i - \sum_i \alpha_i (y_i (< w, x_i > +b) - (1 - s_i)) - \sum_i \mu_i s_i \quad (13)$$

where you look for a stationary point of $L_P$ over $w, b$, and positive $s_i$. Setting the gradient of $L_P$ to 0, you get

$$b = \sum_i \alpha_i x_i y_i \quad (14)$$

$$\sum_i \alpha_i x_i y_i = 0 \quad (15)$$

$$\alpha_i = C - \mu_i \quad (16)$$

$$\alpha_i, \mu_i, s_i \geq 0. \quad (17)$$

These equations lead directly to the dual formulation:

$$\max_\alpha \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j < x_i, x_j > \quad (18)$$

subject to the constraints

$$\sum_i y_i \alpha_i = 0 \quad (19)$$

$$0 \leq \alpha_i \leq C \quad (20)$$

The final set of inequalities,$0 \leq \alpha_i \leq C$, shows why $C$ is sometimes called a *box constraint*. $C$ keeps the allowable values of the Lagrange multipliers $\alpha_i$ in a "box", a bounded region.

The gradient equation for $b$ gives the solution $b$ in terms of the set of nonzero $\alpha_i$, which correspond to the support vectors.

You can write and solve the dual of the $L^2$-norm problem in an analogous manner.

## IV. Experiment

For the purpose of the experiment, MATLAB will be used. The experiment has few phases, which will be now explained.

### *A. Creating audio database*

First, if the goal is dance style recognizing , then certain audio database should be created. In this paper, accent will be on recognizing one of the following categories of dance styles:

- Salsa
- Bachata
- Kizomba

### *B. Training and preparing*

For each dance style, we have made a database of 50 samples. Every sample used in the database has a duration of 20 seconds, what is enough space for analysing. For having better results, we calculated the mean value of every sample and reduced it from the original sample to denoise the signal. Because the SVM only seperates in two groups, we trained the first classificator to seperate bachata from the other two types. So the first classificator recognize if it is bachata or not. If not, we created a new classificator, who deals with the other two types. That was the main idea.

### *C. Testing*

For testing, we have put 18 samples of songs in a test algorithm. Every sample has the same beginning name, called "ples" (bosnian word for dancing), only the ending letter is a number from 1 to 18. The first 6 samples are samples of bachata music, the next 6 samples are samples of kizomba music and the last 6 samples are samples of salsa music. Because we know the type of every sample, we know exactly if the algoritm answers correctly. For every correct answer, the counter set up for one, so teorethically, if the counter counts 18, the algorithm has recognize every song correctly with a accuracy of 100 % . In our case, we matched a accuracy score of 50 %. That was the case when we had used normal signals. We had also made the same testing, but instead of signals (functions of time), we used the FFT spectrum amplitudes of our signals, and put them in our algorithm, and the results have been better. We matched an accuracy score of 60,6 %.

## V. CONCLUSION

Generally, audio recognition is very interesting and useful field of digital signal processing, and there are a lot of methods that can be used for that. One thing that is common for all methods in this field is that a perfect recognizer with a 100 percent reliability *cannot* be modelled. No audio recognizer gives 100% correct results. The recognizer modelled in this paper gives very good results and can be considered as successful, but for better results and applicability, the database should be bigger.

Usually, neuron networks and machine learning uses database of over 10 000 samples for training and testing the reliability of the algorithm. But, the focus in this paper is not on getting the highest reliability of the testing algorithm, it is the understanding of methods how to resolve a problem such as audio recognition or even recognition of music categories, which is a very hard task to handle with it, but it is not impossible.

## REFERENCES


[1] Boby John, U.H. Acharya, Ashis K. Chakrabotry,*"Quality and Reliability Engineering – Recent Trends and Future Directions"*, Allied Publishers Pvt. Ltd, 2013

[2] Guandong Xu, Yu Zong and Zhenglu Yang, *"Applied Data Mining"*, CRC Press,2013

[3] Matlab

[4] Adnan Tahirović, *Lectures in course "Digital Signal Processing"*, Faculty of Electrical Engineering Sarajevo, Sarajevo, 2016.

# Praktična realizacija robota za treniranje stonog tenisa

Dušan Dimitrijević, Anđela Kocić

*Sadržaj* - Rad prezentuje praktičnu realizaciju robota za ispaljivanje loptica za stoni tenis. Obuhvaćen je kompletan dizajn hardvera, realizovan u Altium Designer razvojnom alatu, firmware-a realizovanog u C programskom jeziku, kao i mogućnosti koje robot pruža tokom treninga. Robot je realizovan korišćenjem različitih vrsta motora i senzora za dopremanje i ispaljivanje loptica. Pametnim algoritmima za upravljanje motorima koji ispaljuju lopticu moguće je podesiti spin i polje na stolu gde će loptica biti plasirana. Time je korisniku omogućeno planiranje treninga, pa se zato realizovani robot izdvaja od jeftinih komercijalnih rešenja koja su prisutna na tržištu.

## I. UVOD

Svakodnevni napredak tehnologije doprineo je unapređenju kvaliteta života u svim sferama, čime sport, kao najčešća rekreativna aktivnost, nije izostavljen. Plan i program treninga osnovni su preduslov prema kome se realizuje proces sportske pripreme i kontrolišu efekti koji su postignuti njihovom primenom. Koordinacija je sposobnost usklađivanja pokreta, posebno naučenih, nepredvidljivih i zahtevnih motoričkih zadataka. Učinkovit program koordinacije, a sa time i vežbi za poboljšavanje tehnike, zahteva mnogo pomoćnih sredstava, među kojima se roboti kotiraju u samom vrhu. Za razliku od ostalih igara sa reketom, u stonom tenisu igrač mora sve vreme izvoditi kratke i brze koordinisane pokrete, kako bi adekvatno uzvratio napad protivnika. Za dovođenje koordinacije udarca do savršenstva, potrebno je da trener svoj napad ponavlja iznova i iznova na isti način, što vrlo često nije u potpunosti izvodljivo. Na tržištu postoji veliki broj robota koji u svrhu treninga, mogu plasirati lopticu i simulirati napad protivnika. Međutim, lako dostupni i jeftini roboti najčešće imaju oskudan set parametara kojim se udarci koje protivnik plasira ne mogu najefikasnije simulirati. S druge strane, kompleksniji, profesionalni roboti, svojim funkcijama ispunjavaju sve preduslove za kvalitetna trening, ali ih njihova cena na tržištu čini nepristupačnim prosečnim korisnicima. Robot, koji je tema ovog naučnog rada, predstavlja kompromis između dva najbitnija parametra, cene i performansi.

Realizovani robot je namenjen ljubiteljima stonog tenisa, koji svoju veštinu žele dovesti do savršenstva. Podešavanje spina i polja u koje će loptica sa podešenim spinom biti plasirana, preduslov su za efikasan trening. Ako tome dodamo mogućnost podešavanja do dvadeset loptica, ispaljivanje u nizu, jednostavan korisnički interfejs i nisku proizvodnu cenu, realizovani robot predstavlja ozbiljno pomoćno sredstvo za trening kako profesionalnih tako i rekreativnih igrača.

Dušan Dimitrijević, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: dimi.page@gmail.com

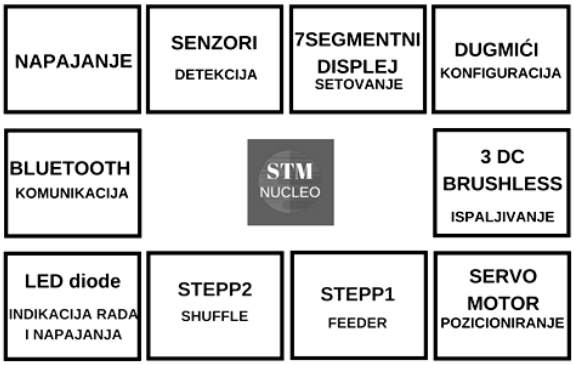
Anđela Kocić, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: andjela.kocic@elfak.com

## II. ELEKTRIČNI PODSISTEM

Hardverske komponente koje su upotrebljene prilikom realizacije su:

- NUCLEO-F411RE evaluaciona ploča na kojoj je montiran STM32F411RE mikrokontroler
- Dva koračna motora
- Tri DC motora bez četkica (Brushless motor)
- RC Servo motor
- Dva senzora za detekciju loptica
- 7-segmentni displej
- Četiri tastera
- 2 LE Diode, za indikaciju rada i indikaciju prisustva napajanja
- RN42 Bluetooth modul

Blok dijagram, koji prikazuje sve komponente sistema i njihovu uzajamnu povezanost, prikazan je na slici 1.




Slika 1. Blok dijagram realizovanog električnog podsistema.

Srce sistema predstavlja STM32F411RE mikrokontroler koji se nalazi na NUCLEO-F411RE evaluacionoj ploči. Ona dolazi sa interno realizovanim napajanjem pa predstavlja vrlo fleksibilno rešenje za „rapid prototyping“. In Circuit programator debager je prisutan na evaluacionoj ploči, čime je vreme potrebno za podešavanje sistema svedeno na minimum.

Da bi se komponente koje čine sistem upravljale, bilo je potrebno realizovati odgovarajući prilagodni interfejs i signale sa mikrokontrolera pretvoriti u upravljačke signale za upravljanje upotrebljenim električnim motorima.

U tu svrhu je razvijen tkz. „shield“, koji se povezuje sa mikrokontrolerskim evolucionim modulom. Prilagodna

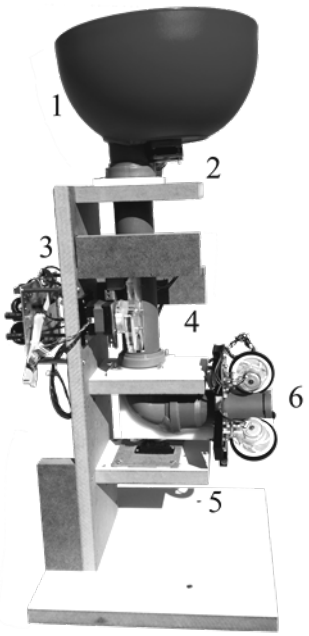
interfejs ploča je dizajnirana u Altium Designer razvojnom okruženju i sastoji se od integrisanih kola i diskretnih komponenti koje zajedno čine sistem kompletnim i omogućavaju upravljanje robotom. Shield se napaja sa +24V, što diktira nominalni napon napajanja upotrebljenih brushless motora. Međutim, za napajanje ostalih modula, bilo je potrebe za još +12V , +6V i +3.3V. Stabilizacija napona je realizovana uz pomoć DC/DC regulatora MC34603. Upotrebljeno SMPS regulator se može konfigurisati kao step-up, step-down ili inverteing konvertor. U konkretnom slučaju step-down konfiguracija je upotrebljena. Maksimalna izlazna struja ovako realizovanog napajanja je 1.5A u piku, dok je maksimalno kontinualno opterećenje strujom ograničeno na 750mA. Raspon ulaznog napona se kreće od +3V do +40V. Napon na izlazu se podešava pravilnim odabirom eksternih komponenti. Sa MC34063, ulazni napon od 24V je spušten na +12V i +6V. Napomenuto je da NUCLEO-F411RE razvojni modul dolazi sa realizovanim internim napajanjem, pa je +3.3V logičko napajanje dovedeno sa ove ploče.

Koračni motor je vrsta električnog motora bez četkica koji digitalne pulseve struje pretvara u fiksne inkremente ugaonog pomeranja koji se nazivaju koracima. Koračni motori obezbeđuju precizno pozicioniranje osovine, a kontrola motora se vrši direktno sa mikrokontrolera dovođenjem odgovarajuće sekvence na namotaje motora. Zbog svoje konstrukcije bez četkica, koračni motori su pouzdani, izdržljivi i ne zahtevaju nikakvo održavanje.

U konkretnom slučaju koračni motori su iskorišćeni za realizaciju robota zbog osobine održanja konstantnog momenta osovine kada se motor ne okreće. Ova osobina je iskorišćena kod realizacije fidera za dopremanje loptica iz rezervoara sa lopticama. Upotrebljeni motori se napajaju sa +12V i upravljaju sa četiri pravougaona signala koji se na namotaje motora dovode u četiri predefinisane sekvence. Signali dolaze do motora preko ULN2004 koji sadrži sedam integrisanih Darlington tranzistora i koji obezbeđuju strujno pojačanje. ULN je iskorišćen jer je struja koju namotaji motora konzumiraju daleko iznad maksimalne struje koju digitalni izlaz iz mikrokontrolera može da obezbedi i tipično iznosi 10mA.

DC brushless motori su upotrebljeni zbog male potrošnje struje i velike brzine okretanja koju osovina motora može da proizvede. U pitanju su inrunner brushless motori sa internim pojačavačima i brushless kontrolerima. Upravljaju se PWM signalom i promenom Duty cycle odnosa može se upravljati brzinom motora. Maksimalna brzina koju iskorišćeni motori mogu da razviju je 9000 rpm, a maksimalna struja pri kojoj se osovina motora ne okreće 470mA. Za realizaciju topa koji ispaljuje loptice različitom brzinom i različitim spinom upotrebljena su tri DC brushless motora u mehaničkoj konfiguraciji koja je prikazana na slici 2.

RC servo motor je električni motor koji se upravlja u zatvorenoj petlji. Osovina motora je spregnuta sa potenciometrom, čime je ostvarena povratna sprega po poziciji. Upravljački signal se dovodi u vidu kvadratnog signala, učestanosti 50Hz. Kvadratni signal koji se šalje servo motoru se pretvara u komandu za poziciju preko elektronike smeštene u kućištu motora. RC servo motor reaguje na zadatu pobudu okretanjem osovine motora u zadatom smeru, dok osovina motora ne dostigne zadatu vrednost. RC servo motor je izabran zbog mogućnosti upravljanja u zatvorenoj petlji i jednostavnoj kontroli pozicije. Koristi za podešavanje ugla evaluacije topa prilikom ispaljivanja loptica. Napon napajanja upotrebljenog servo motora je +6V. Mehanička sprega motora u sistemu prikazana je na slici 2.



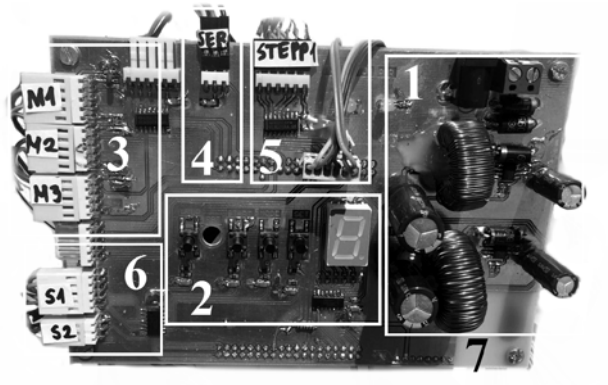

Slika 2. Mahanička konstrukcija robota; 1-činija za loptice, 2-servo motor za mešanje, 3-stm32f411re pločica, 4-stepper motor, 5–RC servo, 6–DC brushless motori.

7-segmentni displej je iskorišćen kao korisnički interfejs koji korisniku omogućava vizuelno podešavanje parametara robota. Upravlja se pomoću 8 bitnog šift registra, SN74HC595 koji se napaja sa +3.3V. Izabrani šift registar ima mogućnost serijskog upisa i paralelnog čitanja, zbog čega je i upotrebljen u konkretnoj realizaciji. Na ovaj način je broj pinova koji su potrebni za upravljanje displejom, redukovan sa sedam na tri.

Senzori za detektovanje loptice se takođe napajaju sa +3.3V i njihovi izlazi su dovedeni na pinove mikrokontrolera preko SN74HC14 invertujućeg Šmit trigera, koji ima svrhu komparatora. On služi da se analogni izlaz sa senzora prevede u digitalni oblik. Takođe, uvođenjem histerezisa se eliminiše treperenje oko graničnih nivoa. Osobina histerezis-a je da imamo dva granična nivoa kod prelaska sa logičke jedinice na logičku nulu i obrnuto, umesto jednog (jedan na ⅓, a drugi na ⅔ vrednosti logičke jedinice). Gornja granična vrednost se koristi kada signal na izlazu prelazi iz stanja logičke nule u stanje logičke

jedinice, dok se donja granična vrednost koristi kada signal na izlazu menja stanje sa logičke jedinice na logičku nulu.

Na shield-u se nalaze četiri tastera sa pull-up otpornicima i hardverski realizovanim kolom za eliminaciju treperenja kontakta, tkz. debouncing. Oni služe za setovanje loptica i podešavanje načina rada robota. Tasteri su aktivni sa logičkom nulom. Kada se taster pritisne, pre nego što se dobije stabilna logička nula, dolazi do treperenja naponskog nivoa dok se nakon nekog vremena signal ne smiri i ustali. Ovo treperenje može dovesti do lažnog očitavanja signala sa tastera. Eliminacija treperenja je realizovana uz pomoć kondenzatora i otpornika povezanih u low pass filter konfiguraciji. Ova konfiguracija dovodi do kašnjenja u sistemu i onemogućava detektovanje signala više frekvencije, što treperenje kontakta predstavlja. Punjenje kondenzatora koji je upotrebljen za realizaciju low pass filtera preko upotrebljenog otpornika, ublažava strmu ivicu signala sa tastera, pa je SN74HC14 ponovo upotrebljen kao generator čisto digitalnog signala koji se dovodi na ulaz mikrokontrolera.




Slika 3. Dizajnirana upravljačka elektronika realizovana u vidu shield-a koji se povezuje na NUCLEO-F411RE evaluacioni modul.

Za Bluetooth komunikaciju, iskorišćen je RN41 Click, čiji je proizvođač domaća kompanija Mikroelektronika. Osobine ovog modula su mala potrošnja, 3.3V napon napajanja i solidan domet od ~100m.

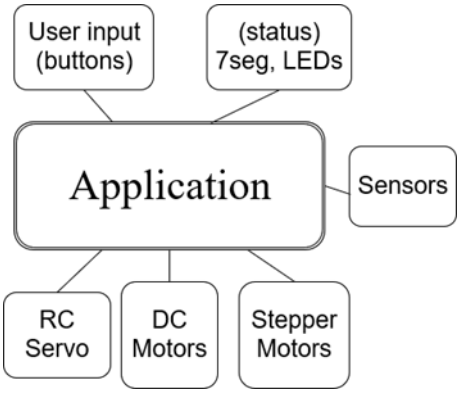
Slika 2 prikazuje izgled robota sa upotrebljenom numeracijom opisanih komponenti. Loptice su smeštene u rezervoaru (1) iz koga se ubacuju u cev. Veći broj loptica u rezervoaru dovodi do glavljena, pa je u tu svrhu ubačen jedan koračni motor (2) koji mešanjem obezbeđuje da loptice u cev ulaze jedna po jedna. Lopticu u cevi detektuje prvi senzor i šalje informaciju mikrokontroleru (3). Nakon detektovanja drugi koračni motor (4) prebacuje lopticu u fider. Drugi senzor detektuje da je loptica u fideru i da je spremna za ispaljivanje. Kada mikrokontroler dobije informaciju da je loptica u fideru, koračni motor je šalje dalje do tri DC brushless motora (6) koji je ubrzavaju, daju joj određeni spin i ispaljuju. DC brushless motori se nalaze na cevi čija se elevacija kontroliše pomoću RC servo motora (5) da bi mogla da bude ispaljena u određenom pravcu.

Na slici 3 je prikazana realizovana upravljačka elektronika sa označenim pod sistemima. Podsistemi koji čine ovaj upravljački interfejs su:

1. Napajanje
2. Human Machine Interface (HMI) sekcija
3. DC brushless motor kontroler
4. Servo motor kontroler
5. Kontroleri koračnih motora
6. Senzorski interfejs
7. Bluetooth Click (u bottom-u sloju)

## III. PROGRAMSKI PODSISTEM

Softverska realizacija obuhvata upravljanje komponentama koje su prethodno opisane. Program je zamišljen kao konačni automat koji komunicira sa različitim softverskim interfejsima koji su realizovani u različitim slojevima aplikacije. Arhitekturni dijagram aplikacije prikazan je na slici 4.




Slika 4. Arhitekturna organizacija aplikacije.

Program koji se izvršava na mikrokontroleru je razvijen korišćenjem programskog jezika C, u razvojnom okruženju Eclipse sa podrškom CDT-a i Standard peripheral library biblioteke. Za svaku komponentu sa slike 4 napisani su drajveri sa potrebnim funkcijama. Za RC Servo motor iskorišćena su prva tri kanala TIM3 interfejsa u PWM režimu rada. DC brushless motori su upravljani na sličan način kao servo motor, s tim da je korišćen TIM4 timer interfejs mikrokontrolera. Problem koji se javio kod nagle promene duty cycle PWM upravljačkog signala radi promene brzine okretanja osovine brushless motora, gde su motori zadržavali istu brzinu, rešen je de-inicijalizacijom PWM kanala i ponovnom inicijalizacijom istih. Robot ima mogućnost ispaljivanja loptica sa četiri različita spina, top spin, bottom spin, side left i side right spin. To je omogućeno upravljanjem motorima različitim brzinama. Brzine motora se menjaju u zavisnosti od željenog spina i prikazane su u tabeli 1. Vrednosti su predstavljene u procentima duty cycle-a za motor 1, 2 i 3 respektivno. Sto za stoni tenis je podeljen na 6 pravougaonih polja

rasporedjenim u gornjem delu suprotne strane stola na kojoj stoji robot.

TABELA I
KORELACIJA BRZINE MOTORA I OSTVARENOG SPINA LOPTICE

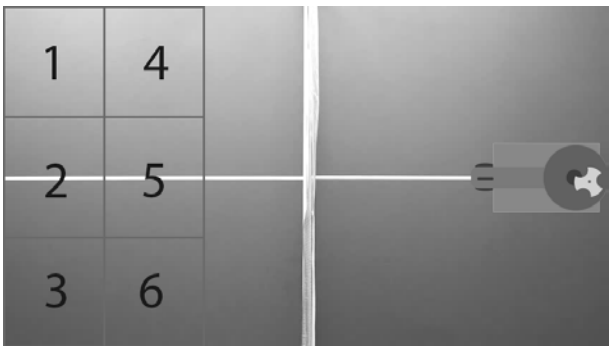
| *Spin tabela* | Polja 1,3 | Polje 2 | Polja 4,6 | Polje 5 |
|---|---|---|---|---|
| Side Left | **62,38,30** | **75,25,25** | **53,28,28** | **65,20,20** |
| Side Right | **30,38,62** | **25,25,75** | **28,28,53** | **20,20,65** |
| Top | **38,86,38** | **38,84,38** | **32,72,32** | **34,52,34** |
| Bottom | **43,28,43** | **43,36,43** | **38,23,38** | **37,23,37** |

Ponovljivost plasiranja loptice u određeno polje je 90% zbog raznih faktora, poput nesavršenosti loptica koje se koriste, njihove različite mase i početne brzine pri izlasku iz fidera, mehaničkog dizajna robota i sl.

Tokom razvoja aplikacije i testiranja bilo je dosta problema koji su rešavani u hodu. Jedan od problema koji se javljao je zaglavljivanje loptica prilikom prebacivanja iz rezervoara u fider. Ovaj problem je nastajao usled nemogućnosti optičkog senzora da očita prisustvo loptice u cevi, čime je dolazilo do glavljena u rezervoaru. Opisani problem je softverski rešen okretanjem smera rotacije koračnog motora koji kontroliše fider i vraćanjem loptica u rezervoar. Tako je ostavljen prostor da se pri povratku u rezervoar i cev loptica postavi u drugačiji položaj i bude detektovana optičkim senzorom. Displej je formiran kao struktura i iščitavanjem njegove trenutne vrednosti, korisnik je u mogućnosti da konfiguriše polja ili spin za svaku od maksimalno 20 loptica.

## IV. IGRA

Aplikacija je realizovana kao *Moore*-ov konačni automat koji obuhvata četiri stanja. Napajanjem električnog podsistema sa slike 3, program startuje u stanju inicijalizacije, gde se vrši inicijalizacija svih komponenti sistema. Nakon inicijalizacije program prelazi u stanje gde se vrši podešavanje loptica. Po ulasku u navedeno stanje displej pokazuje "0" i spreman je za podešavanje. Svaka loptica ima dva atributa koje je neophodno uneti. Prvi atribut je tip spina, dok je drugi polje na protivničkoj strani stola u koje treba plasirati lopticu sa izabranim spinom. Postoje četiri vrste spina koje se mogu selektovati, prikazani u tabeli 1. Spinovi su programski predstavljeni brojevima: 1-bottom, 2-left, 3-right, 4-top, 0-random. Pritiskanjem tastera *Mode* korisnik može izabrati jedan od četiri moguća spina ili random spin. Pritiskom na taster *Set*, prelazi se na setovanje polja u koje će loptica biti plasirana.



Slika 5. Sto za stoni tenis sa prikazanim virtuelnim poljima u koja loptica može da se plasira.

Protivnička polovina stola je podeljena na šest jednakih polja/pravougaonika. Polja su numerisana brojevima od 1 do 6 i korisnik je u mogućnosti da pritiskanjem tastera Mode izabere jedno od 6 polja ili 0-random polje. Nakon postavljanja polja sledeći pritisak tastera *Set* ubacuje tako podešenu lopticu u sekvencu za ispaljivanje. Korisnik je u mogućnosti da definiše do 20 loptica. Nakon podešavanja atributa kojim su definisane loptice, pritiskom na taster *Start* prelazi se u stanje ispaljivanja loptica. Ispaljivanje se vrši dokle god ima loptica u rezervoaru po predefinisanom rasporedu. Iz stanja ispaljivanja moguće je preći u stanje pauze pritiskom na taster *Start*. Na isti način je obezbeđen povratak u režim ispaljivanja. Po nestanku loptica program se vraća u stanje podešavanja ako je potrebno dodati još neku lopticu u sekvencu. Resetovanje sekvence i prelazak u stanje podešavanja se vrši pritiskom tastera *Start* duže od 3 sekunde. Taster *Power* prebacuje pločicu u režim niske potrošnje i vrši de-inicijalizaciju svih komponenti.

## IV. ZAKLJUČAK

Robot koji je realizovan predstavlja korisno sredstvo kojim se tehnika izvođenja udarca može unaprediti. Iako jednostavan, korisnički interfejs omogućava podešavanje većeg broja loptica, a time i planiranje igre, čime realizovani robot može stati rame uz rame sa daleko skupljim profesionalnim robotima. U planu je mehaničko unapređenje prototipa čime će nezgrapan rapid prototype dobiti kako na izgledu tako i na performansama i mobilnosti. Kako bi se povećao procenat ponovljivosti ispaljivanja loptica i omogućio jako precizan plan treninga u planu je uvođenje PID kontrolera za upravljanje brzinom brushless motora u zatvorenoj petlji. U radu je napomenuto da električni podsistem ima mogućnost proširenja povezivanjem bluetooth modula. Bluetooth modulom bi se ostvarila konekcija i upravljanje robota sa mobilnog telefona, korišćenjem Android aplikacije. Na taj način će oskudan, ali pre svega jednostavan, korisnički interfejs biti dodatno unapređen sa novim opcijama i daleko lepšim dizajnom.

## ZAHVALNICA

Autori se zahvaljuju firmi HTEC za ustupanje prostora i opreme tokom realizacije projekta.

## LITERATURA

[1] Behzard Razavi, “Fundamentals of Microelectronics”.
[2] Behzard Razavi, “Design of Integrated Circuit”

# Optoelektronske komponente i njihova primena

Aleksandar Jevtić, Emilija Živanović

*Sadržaj* – U ovom radu je obrađena tema funkcionisanja i primene optoelektronskih komponenata, čiji se rad bazira na fotoelektričnim svojstvima poluprovodničkih materijala. Pored toga, dato je i detaljno objašnjenje funkcionisanja i praktične realizacije kola za testiranje ovih komponenata.

## I. FOTOELEKTRIČNI EFEKAT

Optoelektronske komponente su komponente koje emituju svetlost ili reaguju na nju. Princip rada ovih komponenata bazira se na pojavi *unutrašnjeg fotoelektričnog efekta*. Unutrašnji fotoelektrični efekat predstavlja pojavu da pod dejstvom elektromagnetnog zračenja dolazi do prelaska elektrona unutar poluprovodnika ili dielektrika, iz vezanih stanja u slobodna bez njihovog napuštanja ovih materijala. Kao rezultat toga, koncentracija nosilaca struje, elektrona, unutar materijala raste, što dovodi do pojave fotoprovodljivosti, povećanja elektroprovodljivosti poluprovodnika ili dielektrika pri njihovom osvetljavanju, tj. do pojave elektromotorne sile. Ovaj efekat se javlja kada usled energije fotona elektroni prelaze iz valentne zone, ili sa primesnih nivoa, u provodnu zonu, kao i iz valentne zone na primesne nivoe. Pri tome, energija fotona mora biti veća ili jednaka *aktivacionoj energiji Ea* odgovarajućeg prelaza, po formuli:

$$h\nu \geq Ea \tag{1}$$

gde je $h = 6.67 \cdot 10^{-34}$ Js, Plankova konstanta, a $\nu$ frekvencija elektromagnetnog zračenja (svetlosti).

Iz prethodne jednačine i izraza $c = \lambda\nu$, gde je $c$ brzina svetlosti, a $\lambda$ talasna dužina upadne svetlosti, dobija se maksimalna talasna dužina elektromagnetnog talasa koji može izazvati fotoelektrični efekat, tzv. „crvena“ granica fotoefekta. „Crvena“ granica unutrašnjeg fotoefekta u poluprovodnicima često se nalazi u oblasti infracrvenog zračenja ($\lambda_{max}$ reda nekoliko μm), tako da se neke poluprovodničke komponente mogu koristiti u prijemnicima infracrvenog zračenja. Osvetljavanje poluprovodnika i poluprovodničkih komponenata ima sledeće posledice:

1. Povećava se provodnost poluprovodnika; ovaj fenomen je iskorišćen kod fotootpornika.
2. Ukoliko je fotoefekat u blizini p-n spoja, nastaje povećanje inverzne struje p-n spoja.
3. Usled difuzionog kretanja nosilaca, na p-n spoju će se promeniti visina barijere, te osvetljen p-n spoj može služiti kao izvor električne energije. Obrnuto, prilikom proticanja struje kroz p-n spoj, usled rekombinacije nosilaca može biti emitovana svetlost, te takav elemenat može služiti kao izvor svetlosti.

Aleksandar Jevtić je student osnovnih akademskih studija na modulu Elektronika i mikrosistemi, Katedra za mikroelektroniku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: aleksandar.s.jevtic@gmail.com

Emilija Živanović je docent na Katedri za mikroelektroniku, Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: emilija.zivanovic@elfak.ni.ac.rs

## II. KOMPONENTE

U tabeli su navedene optoelektronske komponente koje su korišćene u praktičnoj realizaciji kola. Osnovne karkteristike svake od korišćenih komponenata biće dat u daljem tekstu.

TABELA 1
KORIŠĆENE KOMPONENTE

| Oznaka | Opis |
|---|---|
| BPW 34 | Fotodioda |
| LDR07 | Fotootpornik |
| ELPT908-7C | Fototranzistor |
| LED dioda | Fotoluminiscetna dioda |
| PC817 | Optokapler |
| TSOP1138 | IR prijemnik |

### *A. Fotodioda*

Osnovna karakteristika fotodioda je da im se inverzna struja zasićenja menja sa promenom intenziteta upadne svetlosti. Fotodioda ima dva režima rada:

1. Fotonaponski režim;
2. Fotoprovodni režim.

### *B. Fotootpornik*

Fotootpornici su poluprovodnički otpornici kod kojih se otpornost smanjuje pod uticajem svetlosti. Rad poluprovodničkih fotootpornika zasnovan je na efektu fotoprovodnosti (unutrašnjem fotoelektričnom efektu).

U najvećem broju slučajeva otporni materijal se nanosi na izolacionu podlogu, a preko toga se prekriva providnim materijalom. Obično nosi oznaku LDR (eng. light-dependent resistor).

### *C. Fototranzistor*

Za razliku od fotodiode i fotootpornika, fototranzistor je optoelektronska komponenta koja može da vrši pojačanje ulaznog signala. Time se postiže veća osetljivost u odnosu na ostale optoelektronske komponente. Fototranzistor nastaje tako što se fotodiodi doda emitor i

samim tim nastaje tranzistor, koji ima veliku površinu kolektorskog spoja.

*D. Fotoluminescentna dioda*

LED, odnosno fotoluminescentna dioda (engl. Light-emiting diode) je posebna vrsta poluprovodničke diode koja emituje svetlost kada je propusno polarisana, tj. kada kroz nju teče struja. U daljem tekstu ovu diodu ćemo nazivati LE dioda.

*E. Optokapler*

Optokapler je optoelektronska komponenta sa svetlosnom spregom koja se sastoji od dve nezavisne optoelektronske komponente koje su u nekom kućištu, obično plastičnom, zatopljene. Prva komponenta je LE dioda, a druga je neka od optoelektronskih komponenata: fotootpornik, fotodioda ili, najčešće fototranzistor.

*F. IR prijemnik*

Infracrveni prijemnici su najširu upotrebu našli kod uređaja koji se kontrolišu infracrvenim zracima (daljinsko upravljanje).

## III. PRAKTIČNA REALIZACIJA KOLA ZA TESTIRANJE

*A. Komponente*

U tabeli 2 su navedene iskorišćene komponente za praktičnu realizaciju kola za testiranje.

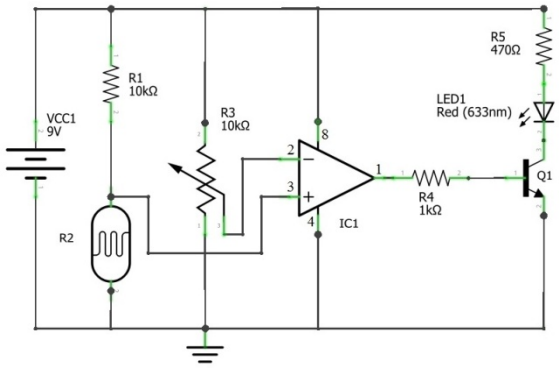
TABELA 2
KORIŠĆENE KOMPONENTE

| Oznaka | Opis | Količina |
|---|---|---|
| VCC1 | Baterija od 9V | 1 |
| R1,R11 | 10kΩ | 2 |
| R2,R12 | LDR07 | 2 |
| R3,R13 | 10kΩ trim. | 2 |
| R4,R9,R14 | 1kΩ | 3 |
| R5,R6,R10, R16,R20 | 470Ω | 5 |
| R7 | 15kΩ | 1 |
| R8 | 50kΩ trim. | 1 |
| R15,R17,R18 | 680Ω | 2 |
| IC1,IC2,IC3 | LM358 | 3 |
| Q1,Q2,Q4 | 2N3904 | 3 |
| Q5 | 2N3906 | 1 |
| Q3 | ELPT908-7C | 1 |
| P1 | BPW 34 | 1 |
| S1 | Prekidač dvopolni | 1 |
| U1 | PC817 | 1 |
| IR1 | TSOP1138 | 1 |
| LED1,2,4,5 | Crvena LE dioda | 4 |
| LED3 | IR LE dioda | 1 |

*B. Funkcionisanje kola*

Funkcionisanje kola se bazira na dualnom operacionom pojačavaču LM358, IC1 koji je za date potrebe povezan kao naponski komparator u kućištu DIP8. Na datoj električnoj šemi uočavaju se razdelnici napona „A" i „B". Naponski razdelnik „A" sačinjen od redne veze otpornika R1 i fotootpornika R2 koji je povezan na neinvertujući ulaz koji se nalazi na pinu broj 3. Razdelnik „B" koje je sačinjen od trimer potenciometra R3 koji je povezan na invertujući ulaz pin broj 2. Pomoću ovog trimer potenciometra podešavamo referentni napon za koji će se na izlaznom pinu 1 stvoriti pozitivan izlazni napon. Posmatramo razdelnik „A".

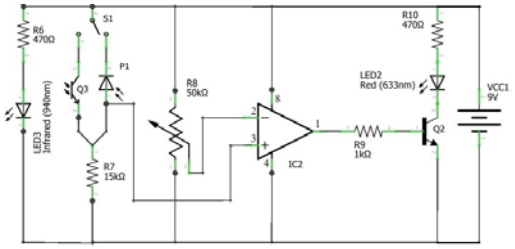
Kada je fotootpornik R2 u tami, njegova otpornost logaritamski raste i pad napona se povećava na njemu. U trenutku kada taj napon na pinu 3 postaje veći od napona na pinu 2, ta razlika potencijala izaziva da pin 1 bude u stanju logičke jedinice ili u idealnom slučaju ima vrednost napona napajanja. Kada je napon na pinu 3 manji od napona na pinu 2, pin 1 se u idealnom slučaju vezuje za masu. Da bi se smanjila disipacija snage na operacionom pojačavaču, iskorišćen je NPN tranzistor Q1, 2N3904 koji je u stanju zasićenja i ponaša se kao prekidač koji zatvara kolo otpornika R5 i LE diode na masu. Otpornik R4 osigurava da struja baze tranzistora Q1 ne pređe dozvoljenu granicu datu u kataloškim podacima tranzistora. Rad svih ostalih šematskih dijagrama se bazira na istom principu, razlika je u upotrebi ostalih optoelektronskih komponenata iz tabele 1 i one će biti naknadno objašnjene. Šema naponskog komparatora sa LDR otpornikom prikazana je na slici 1.




Slika 1. Naponski komparator sa LDR otpornikom.

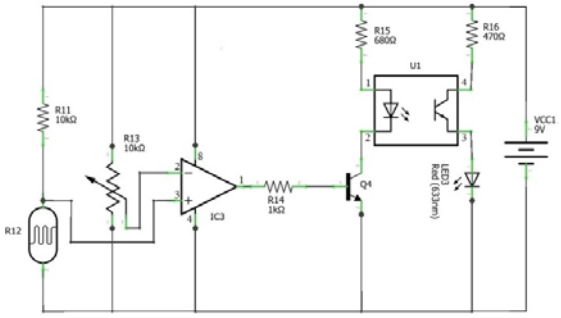
U narednon primeru, naponski razdelnik „A" je sačinjen od dvostranog prekidača S1, fototranzistora Q3, fotodiode P1 i otpornika R7. Razdelnik „B" je principijalno isti samo ima povećanu otpornost na 50kΩ. Princip rada na samom operacionom pojačavaču je isti. Razlika u odnosu na prethodno kolo je da su optoelektronske komponente vezane za VCC potencijal preko prekidača S pomoću kojeg biramo koju komponentu želimo da koristimo. Fototranzistor i fotodioda rade u fotoprovodnom režimu i ponašaju se kao otvoreni prekidači dok nisu izloženi

odgovarajućem vidu svetlosnog zračenja. Usled ove pojave napon na pinu 3 ima vrednost logičke nule. Ako se izabere fototranzistor da čini razdelnik napona „A“, dovoljno je da bude izložen relativno maloj osvetljenosti bele svetlosti i on će početi da provodi struju pojačanu $\beta$ puta. Samim tim se javlja pad napona na otporniku R7 koji se upoređuje sa naponom na pinu broj 2. U slučaju da je taj napon veći u odnosu na pin broj 2, crvena LE dioda će zasvetleti, dok ukoliko je napon manji LE dioda ostaje isključena. Fotodioda ima izraženija prekidačka dejstva, konkretan model diode se uključuje pomoću IR zraka (npr. iz bilo kog IR daljinskog upravljača) ili infracrvenom LE diodom označenom LED3. Šema ovog kola data je na slici 2.



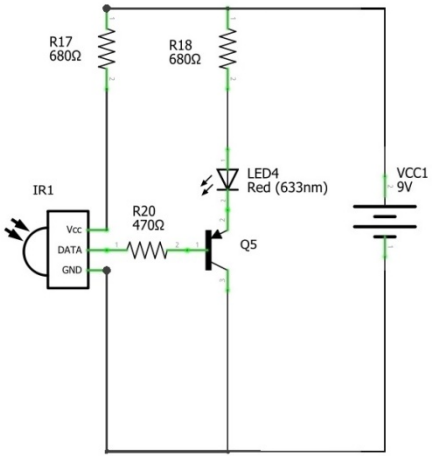

Slika 2. Naponski komparator sa fototranzistorom ili fotodiodom.

Sledeći primer je kolo naponskog razdelnika sa optokaplerom. Ovo kolo je identično kolu sa LDR otpornikom R12, uz dodatak da se sada LE dioda aktivira preko optokaplera U1. Između pinova 1 i 2 je smeštena LE dioda koja je direkto usmerena u fototranzistor između pinova 3 i 4. Prilikom direktne polarizacije LE diode dolazi do emisije svetlosti na glavanski odvojen fototranzistor koji samim tim postaje provodan, usled čega dolazi do zatvaranja diodnog kola i uključivanja crvene LE diode. Kolo LE diode je moglo da bude povezano na neki drugi izvor napajanja, zaseban u odnosu na pobudno kolo. Na slici 3 prikazana je šema naponskog komparatora sa optokaplerom.




Slika 3. Naponski komparator sa optokaplerom.

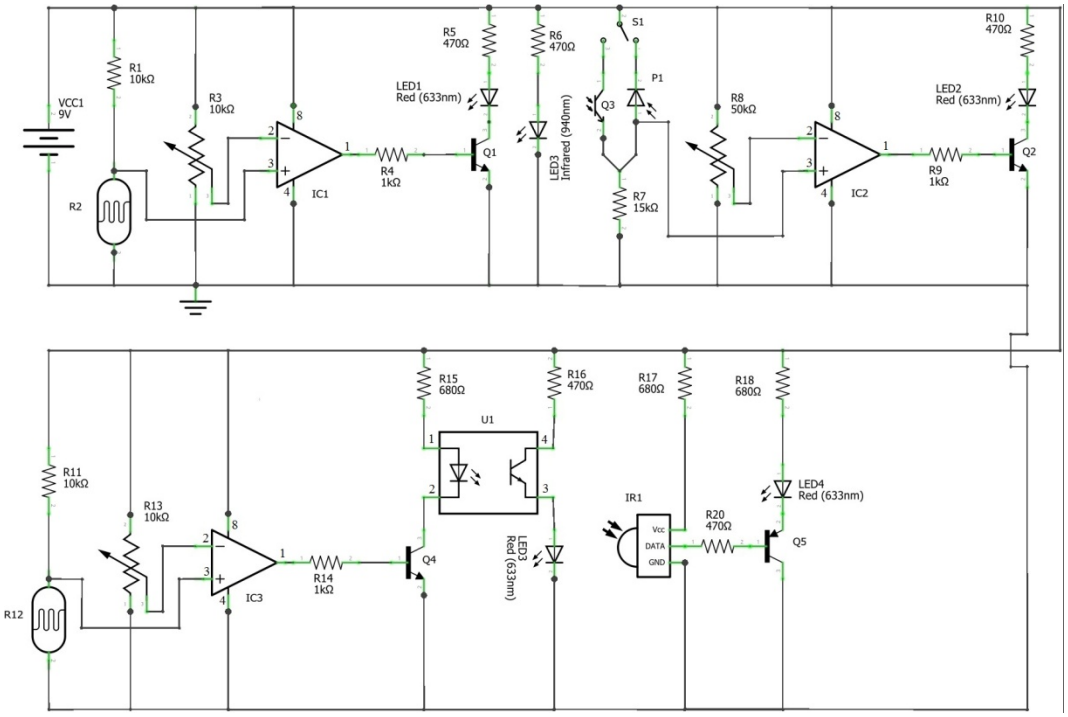
Poslednje kolo u ovom radu se ne bazira na operacionom pojačavaču, već na IR prijemniku čija je glavna namena prijem infracrvenih zraka. Šema je krajne jednostavna, IR prijemnik ima tri pina, a to su VCC koji se priključuje na izvor napajanja, DATA priključak koji je normalno spojen sa VCC izvorom ili sa masom i GND ili masa IR prijemnika. Za dat korišćen IR prijemnik prilikom izlaganja IR zracima, DATA priključak se vezuje za masu. Da bi kolo bilo najjednostavnije moguće, koristimo PNP tranzistor 2N3906, Q5 koji je bazom vezan za DATA priključak preko otpornika R20. Prilikom izlaganja infracrvenoj svetlosti, baza tranzistora se vezuje za masu i tranzistor spaja rednu vezu otpornika R18 i LE diodu LED 4 na masu. Otpornik R17 ograničava struju IR prijemnika. Šema ovog kola je data na slici 4.



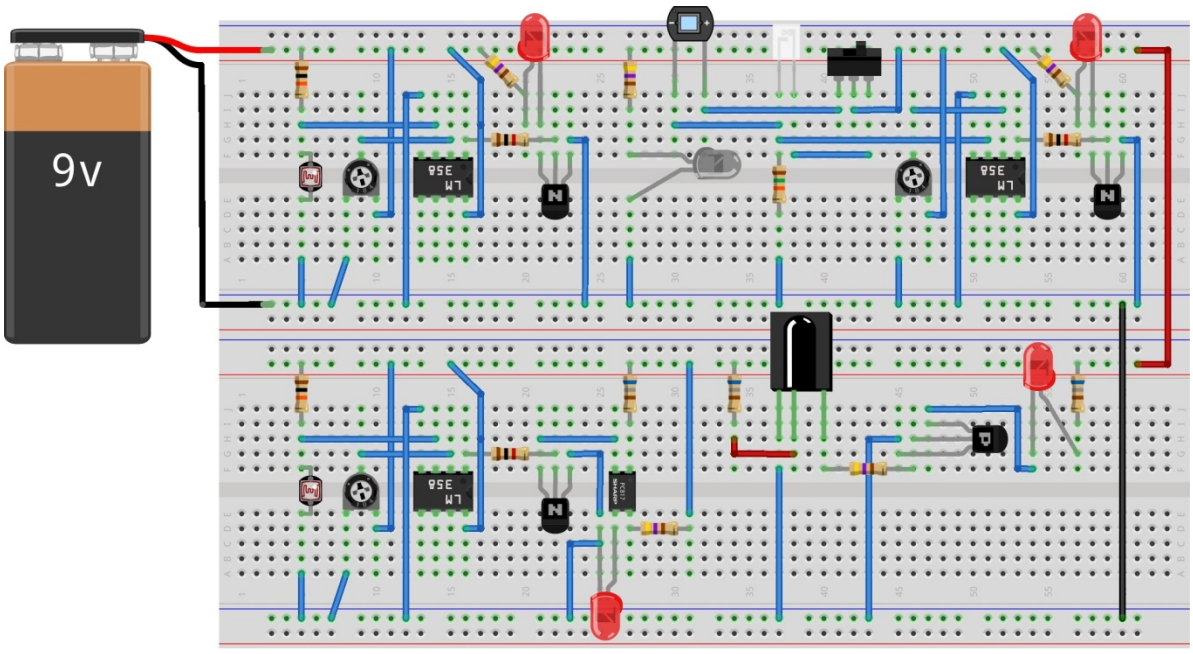

Slika 4.IR prijemnik (tester daljinskih upravljača).

## IV. PROJEKTOVANJE KOLA U PROGRAMU FRITZING

U ovom radu za iscrtavanje šematskih dijagrama i praktične realizacije kola, korišćena je aplikacija Fritzing, koja nam omogućava brzo i jednostavno realizovanje projekta. Na slici 5 predstavljena je celokupna šema svih elemenata, dok je na slici 6 dato samo test kolo sa raspoređenim elementima na protoploči. Napomena: U programu Fritzing su za fototranzistor i fotodiodu upotrebljene komponente opštih karakteristika iz već postojećih biblioteka i raspored njihovih izvoda na protoploči može se razlikovati od onih koje su navedene u tabeli 1.




Slika 5. Kompletna šema kola za testiranje.




Slika 6. Izgled realizovane protoploče u Fritzing-u.

## V. ZAKLJUČAK

Prilikom projektovanja ovakvog kola, uviđamo važnost i široku primenu kako optoelektronskih komponenata, tako i integrisanih operacionih pojačavača koji omogućavaju jednostavnu adaptaciju kola za zadatu primenu.

## LITERATURA

[1] M. Platiša, „Kurs opšte fizike, Elektromagnetizam i elementi atomske fizike“, Naučna knjiga, Beograd, 1997.
[2] Z. Prijić i A. Prijić, „Uvod u poluprovodničke komponente i njihovu primenu“, Univerzitet u Nišu, Elektronski fakultet, 2014.
[3] http://fritzing.org/home/.

# Vizuelizacija podataka korišćenjem R jezika

Miljan Jeremić

*Sadržaj* - Ovaj rad predstavlja upotrebu novih paketa kao vizuelni metod za predstavljanje grafičkih podataka u R programskom jeziku. Koristeći R Data Science, ggplot paket je adekvatan za vizuelno predstavljanje grafičkih podataka. Paket climatol predstavlja veliku pomoć za obradu podataka o količini padavina po mesecima kao i vrednostima temperatura po mesecima u godišnjem periodu. Rezultati pokazuju važan doprinos i značajna poboljšanja u odnosu na prethodne napore drugih programa i programskih jezika koji mogu statistički obrađivati podatke. Rad predstavljen ovde ima duboke implikacije za buduće studije R programskog jezika i može jednog dana pomoći u rešavanju problema obrade statističkih podataka za proveru parametara koji karakterišu slivove reka, nivoe padavina i promene temperatura u Srbiji.

## I. Uvod

Analiza istraživačkih podataka se bavi procedurama i tehnikama za analizu podataka kao i interpretacijom dobijenih rezultata, prikazujući ih najčešće vizuelno.

Obrada i analiza podataka se poboljšavaju vizuelno kodiranjem informacija kroz aktivnost vizualizacije [1]. Danas su tehnologije za web i mobilno mapiranje omogućile generisanje velikog broja prostornih podataka širom sveta [2-4]. Vizuelizacija prostornih podataka mogla bi se postići korišćenjem web-baziranih GIS aplikacija kao što su MapD, HiDRA, ArcGIS, Azavea, GISCloud, GeoIK. Informacione i komunikacione tehnologije igraju važnu ulogu u razvoju različitih geoprostornih pristupa, alata i aplikacija koje u celini utiču na sve sfere iz industrije, društvene zajednice i ljudskog života [5].

Postoje dve glavne svetske poznate organizacije koje koordiniraju razvoj otvorenih specifikacija, tj. World Wide Web Consortium (https://www.w3.org/) i OpenGIS Consortium Inc. (http://www.opengeospatial.org/ ). Open Geospatial Consortium (OGC) je uspostavio standarde Web Map Services (http://www.opengeospatial.org/standards/vms) kroz formiranje javnih standarda koji podržavaju interoperabilna rešenja za geo-web. U Evropi je uveden INSPIRE (Infrastruktura za prostorne informacije u Evropskoj zajednici) za standardizaciju podataka [6].

R sa svojim paketima može se koristiti ne samo za statistiku, već i za GIS analizu i rad sa prostornim podacima.

Ovaj rad predstavlja GIS funkcionalnost u programskom jeziku R i daje jedan primer kako koristiti pakete za vizuelizaciju i doći do grafičkih prikaza sliva reke Timok u Srbiji, kao i prikaza klima dijagrama na osnovu podataka sa meteo stanice.

Miljan Jeremić, Knjaževačka gimnazija, Karadjordjeva 16, 19350 Knjaževac, Serbia, E-mail: miljan.jeremic@gmail.com

## II. Podaci za studiju i kolekcije podataka

Reka Timok, prekogranična reka i desna pritoka Dunava, nalazi se u istočnom delu Srbije dužine 88 km, a površina sliva je 4.607 km² (98%), a manji deo u Bugarskoj (2% ili 93 km²). Dug je 201.7 km zajedno sa Belim i Trgovškim Timokom (slika 1). Timok je širok 35 m, a dubok 0,5 - 1,5 m. Na samom nizvodnom delu Timok čini granicu između Srbije i Bugarske 17,5 km. Sliv reke Timok je deo slivova Crnog mora.




Slika 1. Sliv reke Timok.

Prema Republičkoj hidrometeorološkoj službi Srbije, postoje deset hidroloških stanica: Rgoste za Svrljiški Timok, Knjaževac, Vratarnicu i Zaječar za Beli Timok, Bogovinu i Gamzigrad za Crni Timok, Cokonjar za Veliki Timok, Zlot za Zlotsku i Grlište za Grliško.

Najtopliji mesec je jul, sa prosečnom temperaturom od 21.3°C, a najhladniji mesec je januar sa prosečnom temperaturom od -0.8°C. Prosečna godišnja količina padavina je 590.5 mm sa različitim suvim i vlažnim sezonama.

## III. R jezik i njegovi paketi

R jezik uglavnom koristi podatke sa web stranica koji se uvoze, a zatim se procesiraju i na kraju se vrši analiza obrađenih podataka. Podaci obrađeni korišćenjem R jezika najčešće se izvoze u datoteku Excel datoteke ili tekstualne tabele. Takvi podaci se kasnije lako mogu preneti na web aplikaciju koju korisnici mogu videti kroz pregledače nakon obrade.

Da bi prikazali podatke sa jezika R i direktno poslali web aplikaciju, neophodno je ovde obezbediti web servis za ubrzavanje obrade podataka na jeziku R.

Ovo istraživanje je učinjeno kako bi se izabrao odgovarajući pogled na reku Timok i kako bi se dobili novi geospatijalni podaci koristeći jezik R sa ggmap paketom

Vizualizacija daje priliku da uporede različite odlike sliva. Novi osmar paket integriše se u R, čime otvara mogućnost za planiranje geografskih objekata na vrhu mapa ili satelitskih slika u okviru R pomoću ggmap-a (Schlesinger and Eugster, 2013) [9]. Alternativno, integracija sa drugim prostornim paketima u R može da obezbedi nekoliko jako korisnih praktičnih alata za analizu prostornih podataka.

## IV. VIZUELIZACIJA PODATAKA KORIŠĆENJEM R JEZIKA

R programski jezik je dizajniran za obavljanje statističkih analiza. Istraživanja programskih jezika pokazuju da se upotreba R brzo povećava, očigledno u vezi sa sve većim prikupljanjem podataka. R se može koristiti na dva različita načina. Najčešće, R se koristi kao interaktivni alat. Na primer, u ljusci (shell) R konzole, korisnik bi mogao upisati komande. R ima na stotine ugrađenih funkcija koje mogu obavljati hiljade statističkih zadataka.

Paket ggmap je novi alat u jeziku R koji omogućava takvu vizualizaciju kombinovanjem prostornih informacija statičkih mapa. Koristeći nekoliko metoda, moguće je napraviti osnovne geografske informacije zapisane u datoteci koja sadrži podatke o poligonima iz određenog područja ili tačke za referentne podatke. U poređenju sa specijalizovanim geografskim informacionim sistemima (GIS) kao što je ESRI ArcGIS, prikazane vizualizacije mogu biti prilično nepotpune i nejasne.

Ovaj rad opisuje metode za vizualizaciju prostornih podataka u R koristeći slojevitu gramatiku grafičke implementacije paketa ggplot2 u vezi sa kontekstualnim informacijama statičkih mapa sa Google Maps, OpenStreetMap, Stamen Maps ili Cloudmade Maps. Rezultat se dobija korištenjem jezika R i ggmap paketa.

Jedan od načina vizuelnog predstavljanja podataka jeste da se zapišu na geografskoj mapi. Osnovna ideja za pokretanje ggmap-a je da preuzme preuzetu sliku mape, zapiše je kao kontekstni sloj koristeći ggplot2, a zatim prikaže dodatne slojeve podataka sadržaja koji prikazuje, statističke ili modele na vrhu mape.

R je računarski jezik za statistiku, uključujući geostatistiku. Moguće je koristiti R kao jednostavan geografski informacioni sistem (GIS) kako bi se podaci zapisivali, ili prikazivali, na mapi sveta. Mogu se napraviti karte za štampanje direktno u R, tako što će se memorisati mape u vektorskom grafičkom formatu (npr. pdf ili eps), a kasnije se mogu dalje urediti koristeći neki od programa vektorske grafike.

Koristićemo mape R paketa koji imaju podatke za mapu za vizualizaciju, u našem slučaju za sliv reke Timok, a primenom ggplot2 paketa preuzimaju se podaci i prikazuju u obliku slike. Sledeći deo primera pokazuje kako započeti korišćenje paketa ggplot.

```
library(maps)
library(ggplot2)
timok = get_map(location = "timok", zoom=10)
ggmap(timok)
```

Ovom naredbom možemo da odredimo tačne geografske koordinate objekta, a rezultat je prikazan odmah ispod naredbe.

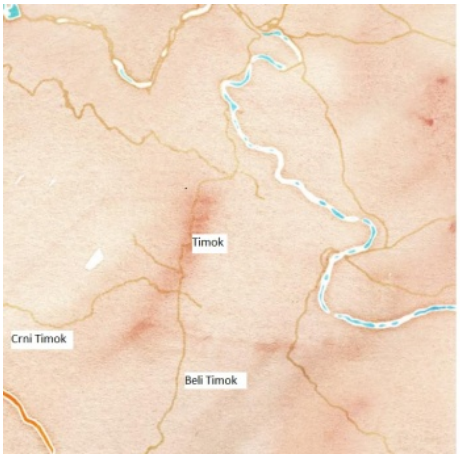
```
geocode("Timok", source="google")
```

| | lon | lat |
|---|---|---|
| 1 | 22.48707 | 44.06911 |

Primenom stila prikaza koji koriste mape Stamen Maps i Cloudmade Maps dobili bi prikaz reke Timok sa svojim pritokama u obliku na slici 2 i slici 3.




Slika 2. Pogled na mapu reke Timok korišćenjme ggplot paketa.




Slika 3. Pogled na mapu reke Timok korišćenjem stila prikaza vodenih tokova.

## V. KLIMA DIJAGRAMI

Radi prikaza i proučavanja klimatskih prilika u toku jedne godine koristi se klima dijagram (grafički prikaz prosečnih količina padavina i temperatura za određeno područje u toku 12 meseci. Naučnici klimatski dijagram koriste za analizu klimatskih uslova određenog područja (Borisavljević i Kostadinov (2012) [7].

Klimatski dijagram Valtera se veoma često koristi u bio-klimatologiji kako bi pokazao klimu datog područja. Razlog za to je da klimatski dijagram Valtera daje izvanredan vizuelni uvid u vodni balans eko sistema radi preglednog prikaza odnosa temperatura vode [7].

Klimatski dijagram Valtera je kreiran tako da padavine budu na primarnoj osi (primarni red) i temperatura na sekundarnoj osi (sekundarni red). Vrednost temperature i brzina promena na skali od 1: 2, tj. vrednost temperature od 10 ° C na ordinati odgovara padavinama od 20 mm, pa do vrednosti padavina od 100 mm. Padavine iznad 100 mm unose se u skali 1:10, a na dijagramu su obeležene crne i predstavljaju vlažni period. Ako kriva padavina pada ispod krive temperature na dijagramu, ona se tumači kao suv ili sušni period (na dijagramu označenom crvenom bojom). Ako se sušni period pojavljuje na dijagramu, linija padavina se takođe unosi još jednom, ali na skali 1: 3 i predstavlja period suše (dijagram je žut). Period suše se izdvaja na tačku i daje dobar uvid u mesecima koji su u nepovoljnom položaju zbog nedostatka vode.

## VI. VIZUELIZACIJA PODATAKA KORIŠĆENJEM PAKETA CLIMATOL

Ovde želimo da pokažemo kako se pomoću jezika R i paketa u okviru ovog jezika, climatol, može generisati klima dijagram na osnovu prikupljenih podataka dobijenih sa meteorološke stanice u Knjaževcu (http://www.knjazevac.rs/Meteo/wxindex.php), i koji su zapisani u Excel fajlu koji nam kasnije služi za obradu u R jeziku.

Za generisanje klima dijagrama korišćen je paket climatol koji se naknadno instalira u okviru R jezika kako bi mogao da se koristi u daljem radu.

Klimatski dijagrami Valter & Leith ilustruju padavine i promene temperature tokom cele godine u jednom standardizovanom grafikonu. Posebno su korisni za određivanje vodenih promena ili drugih značajnih klimatskih faktora izmedju ostalog i na biljkama. Ovde želimo da opišemo kako da se generiše sopstveni klimatski dijagram Valter & Leith pomoću softvera koji se zove R. Ovde su uzorci koji su napravljeni na osnovu podataka sa meteo stanice u Knjaževcu u period od 3 godine. Treba naglasiti da su podaci smešteni u tri odvojena Excel fajla kako bi mogli da se generišu klima dijagrami na period od jedne godine, tako da će ovde biti predstavljena tri dijagrama koja su generisana u programskom jeziku R.

Ovde će biti predstavljen i deo koda u R konzoli koji je korišćen prilikom generisanja klima dijagrama za jednu kalendarsku godinu..

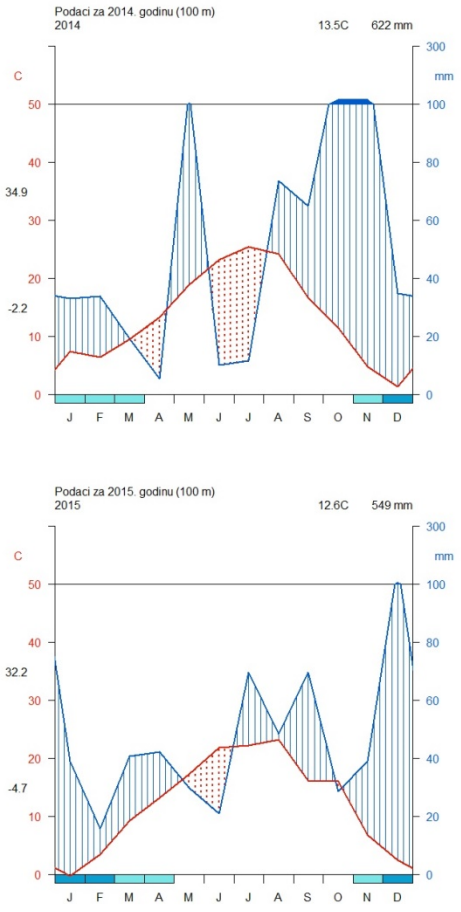
```
name1=read.csv("klima2014.csv")
diagwl(name1,est="Podaci za 2014. godinu",alt=100,per="2014",mlab="en")
```

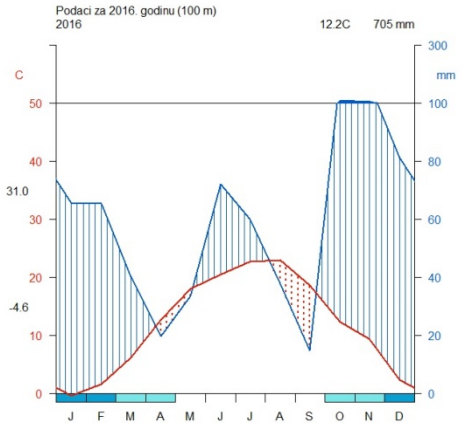
Dokument u Excel-u mora biti predstavljen na sledeći način.

TABELA I
KOLEKCIJA PODATAKA U EXCEL-U ZA GENERISANJE KLIMA DIJAGRAMA

| GODINA | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MESEC | Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec |
| Količna padavina (mm) | 38.8 | 15,6 | 40,6 | 42,2 | 29,8 | 20,9 | 69,6 | 48,4 | 69,4 | 28,6 | 38,8 | 106,4 |
| Srednja mesečna maksimalna temperatura | 4.0 | 10.5 | 15.4 | 19.2 | 25.1 | 29.5 | 30.4 | 32.2 | 22.5 | 22.2 | 10.7 | 5.2 |
| Srednja mesečna minimalna temperatura | -4.7 | -3.6 | 2.9 | 7 | 9.4 | 14.1 | 14 | 14 | 9.7 | 9.8 | 2.8 | -0.4 |
| Apsolutna minimalna temperatura | -14.5 | -15.6 | -4.4 | -1 | 2.4 | 9.4 | 9.9 | 7.6 | 2.1 | 3.8 | -8.4 | -10.4 |

Dijagrami nakon pokretanja programa i korišćenjem climatol paketa izgledaju kao na slici 4.






Slika 4. Klima dijagram za 2014, 2015 i 2016. godinu.

## VI. ZAKLJUČAK

Podaci dobijeni iz jezika R i vizualizovani korišćenjem ggplot paketa pružaju dodatnu mogućnost za njihovo povezivanje sa web aplikacijom koja bi prikupljala podatke i prikazala za specifičnu mernu stanicu sve relevantne informacije o vodi na reci Timok. To bi omogućilo da još manja reka i pritoke Dunava budu potpunije i preciznije statistike o nivou reke, toka, padavina u gradovima kroz koje prolazi reka. U daljem razvoju web aplikacija, bilo bi dobra ideja da se dopuni web aplikacija koja bi takođe pokazala generisane linije trendova na osnovu prikupljenih podataka i izvršiti dalju analizu. Ovo bi bilo korisno jer je Timok 2010 katastrofalno ugrozio grad Zaječar i izazvao veliku štetu stanovništvu. Grafički podaci generisani korišćenjem paketa climatol mogu da pomognu za određivanje vodenih promena ili drugih značajnih klimatskih faktora izmedju ostalog i na biljkama.

## ZAHVALNICA

Na kraju ovoga rada autor se zahvaljuju Knjaževačkoj gimnaziji na razumevanju i pružanju potrebne podrške da istrajemo do kraja. Autor je zahvalan i profesorima Elektronskog fakulteta u Nišu koji su nas naučili da razmišljamo o problemima, i da ih rešavamo na jedan kreativni način, podarili nam dovoljno znanja o programiranju i primeni novih IT tehnologija.

## LITERATURA

[1] Hibbard W.L., Kao D.T., Wierse A. (1996) “Database issues for data visualization: Scientific data modelling”. In: Wierse A., Grinstein G.G., Lang U. (eds) “Database Issues for Data Visualization”. DBVIS 1995. Lecture Notes in Computer Science, vol 1183. Springer, Berlin, Heidelberg

[2] Yi Lin, Juha Hyyppä, Anttoni Jaakkola, Markus Holopainen. (2012) “Characterization of mobile LiDAR data collected with multiple echoes per pulse from crowns during foliation”. Scandinavian Journal of Forest Research 27:3, pages 298-311.

[3] Keith C. Clarke, 2004. “Mobile Mapping and Geographic Information Systems”, Cartography and Geographic Information Science 31(3), 131-136.

[4] C.E. Bond, Z.K. Shipton, R.R. Jones, R.W.H. Butler, A.D. Gibbs, 2007. “Knowledge transfer in a digital world: Field data acquisition, uncertainty, visualization, and data management”. Geosphere 3(6), 568-578.

[5] “A Web-Based Spatial Data Visualization Tool for Yantra River Basin Area Kjellström”, E.: Recent and future signatures of climate change in Europe, Ambio, 23, (2004), pp. 193–198, ISSN 0044-7447.

[6] “Directive 2007/2/EC Of The European Parliament And Of The Council”, Official Journal of the European Union, 2007, L108/1-L108/14. [Online]. February 2007. Dostupno na: http://eur-lex.europa.eu/legal content/EN/ALL/?uri=CELEX:32007L0002. [Preuzeto: 15.08.2017]

[7] Borisavljević A, Kostadinov S., 2012, “Integrated river basin management of Juzna Morava River”, Glasnik Srpskog geografskog društva

[8] SchoolKIT, Klimatski dijagrami, [Online]. Oktobar 2004. Dostupno na: http://pup.skole.hr/vodichtml.aspx?xml=/datoteke/hr/hr/student/sk/activities/climatograms/sk-climatograms%20tutorial.xml. [Preuzeto: 15.09.2017]

[9] Kahle D. and Wickham H. R Journal, ggmap: Spatial Visualization with ggplot2. [Online]. June 2013. Dostupno na: https://journal.r-project.org/archive/2013-1/kahle-wickham.pdf. [Preuzeto: 25.06.2017]

# Kolo za merenje kapacitivnosti diskretnih kondenzatora korišćenjem timer NE555 kola i Arduino okruženja

Miloš Ljubenović, Miljana Milić

*Sadržaj* – U ovom radu će biti predstavljeno jednostavno kolo za merenje kapacitivnosti diskretnih kondenzatora. Kolo je realizovano tako da bude malo, kompaktno i jednostavno za korišćenje. Faze projektovanja su rađenje u okviru Fritzing okruženja. Za realizaciju kola korišćen je Timer NE555 integrisano kolo i Arduino okruženje. Na kraju je izrađena pločica korišćenjem metode za izradu štampanih pločica sa fotoosetljivim materijalom.

## I. UVOD

Merenje kapacitivnosti predstavlja jedan od važnih zadataka u oblasti merenja i elektronike. Često je nakon proračuna kondenzatora za elektrosnko kolo potrebna procena njegovog odstupanja, radi preciznijeg rada samog kola [1], [2]. Sam način testiranja kondenzatora, odnosno njegovog odstupanja od fabrički zadate kapacitivnosti može predstavljati složen proces.

Korišćenjem jednostavnih i diskretnih komponenti, moguće je realizovati merač kapacitivnosti relativno preciznih karakteristika [3] . Opšti mehanizam merenja kapacitivnosti kondenzatora, zasniva se na merenju vremena punjenja i pražnjenja samog kondenzatora. Poznato je da je vreme punjenja kondenzatora $\tau = 5RC$. Poznavanjem otpornisti preko koje se kondenzator puni R, i vremena njegovog punjenja, može se izračunati kapacitivnost kondenzatora. Zbog neidealnosti samog otpornika, njegova otpornost se menja u opsegu navedenom na samoj komponenti. To odstupanje može biti ponekad presudno u izračunavanju kapacitivnosti kondenzatora. Zbog toga je potrebno kompenzovati sve neidealnosti koje se mogu javiti prilikom računanja kapacitivnosti, i svesti ih na minimum. Tada će greška pri izračunavanju kapacitivnosti biti jako mala [4].

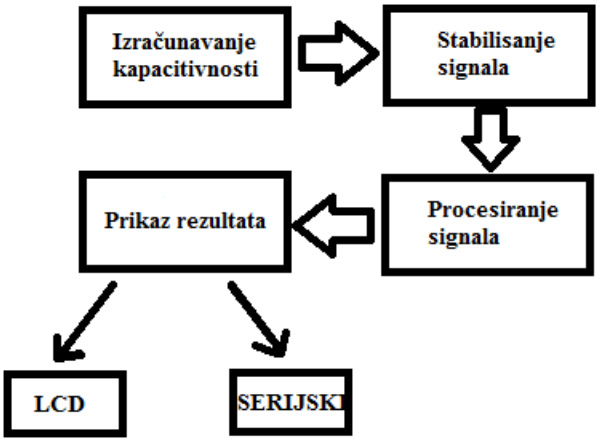
Modul za merenje kapacitivnosti se može koristiti na dva načina. Jedan način je da se za prikaz rezultata koristi računar ili laptop, i takvo rešenje je jako pogodno za kalibraciju, ispravljanje smetnji koje se javljaju ili za proračun komponenti koje bi realizovale najpogodnije okruženje za izračunavnaje kapacitivnost. Drugi način realizacije predstavlja modul kao nezavisan uređaj, sa ugrađenim LCD-om za prikaz rezultata merenja i baterijskim napajanjem, čime se obezbeđuje mobilnost modula. Naravno, uz adekvatnu korekciju napajanje šeme, što podrazumeva primenu naponskog stabilizatora, jer bi se usled pražnjenja baterije remetio rad tajmera.

Miloš Ljubenović je student na modulu Elektronika, podmodul Elektronska kola i sistemi, Univerzitet u Nišu, Elektronski fakultet Niš, Aleksandra Medvedeva 14, Niš, Srbija, E-mail: traputopia@elfak.rs.

Doc. dr Miljana Miljić, katedra za Elektroniku, Univerzitet u Nišu, Elektronski fakultet Niš, Aleksandra Medvedeva 14, Niš, Srbija, E-mail: miljana.milic@elfak.ni.ac.rs.

## II. OPIS SISTEMA

Rad celokupnog modula se može podeliti u četiri dela (Slika 1): izračunavanje kapacitivnosti, stabilizacija signala, procesiranje signala i prikaz rezultata. Prve dve faze, odvijaju se direktno na modulu i one zahtevaju dodatni hardver. Druge dve faze predstavljaju softverski deo ovog modula. Četvrta faza se može realizovati preko serijskog porta ka računaru ili u vidu prikaza na LCD-u.




Slika 1. Blok šema operacija koje vrši modul.

### *A. Izračunavanje kapacitivnosti*

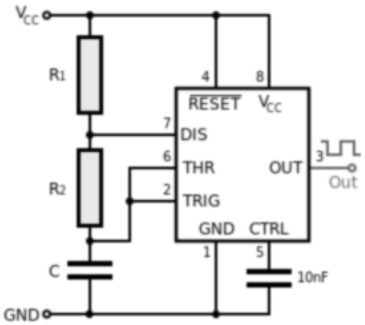
Prilikom projektovanja ovog dela sistema neophodno je da se izračuna i dovoljno dobro podesi okruženje NE555 tajmera koji služi za generisanje povorke četvrtastih impulsa. Tajmer je potrebno postaviti u astabian radni režim, što znači da je potrebno izračunati otpornike $R_1$ i $R_2$ sa slike 2. Kondenzator od 10nF između izvoda 5 i GND (pin 1) služi za filtriranje šuma koji može biti na izvoru napajanja +Vcc.

Dovoljna frekvencija povorke impulsa koja se javlja na izlazu je 700Hz. Ova frekvencija omogućava detekciju kapacitivnosti do reda pF. Nakon što nam je poznata

izlazna frekvencija, za određivanje vrednosti nepoznate kapacitivnosti koristimo obrazac 1:

$$f = \frac{1}{\left(ln(2)\cdot C\cdot(R_1+2R_2)\right)} \quad (1)$$

Kako bismo izračunali nepoznatu otpornost, jednu otpornost ($R_1$) postavimo na 1kΩ; kapacitivnost C postavimo na 0.1μF. Odatle se dobija da je $R_2$=10kΩ. Sada je već jasno da se promenom kapacitivnosti C, menja frekvencija izlaza (nožica 3 na slici 2), što nam omogućava dalje procesiranje signala [5].

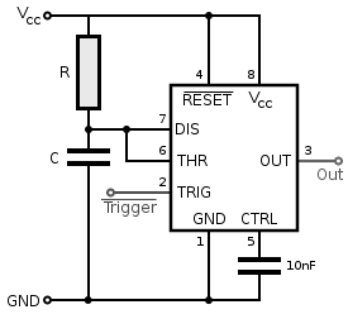



Slika 2. Astabilni režim 555 tajmera.

*B. Stabilisanje signala*

Moguće je signal sa izlaza NE555 tajmera dovesti direktno na jedan od digitalnih ulaza Arduino okruženja, ali se time ne bi mogla postići željena preciznost. Takođe, kod velikih kapacitivnosti, ili kondenzatora koji npr. rade na 16V, moguće je oštetiti ulaz Arduina. Zbog toga se signal prvo mora stabilisati za rad sa Arduino okruženjem.

Najjednostavniji način stabilisanja signala jeste pomoću NE555 tajmera u konfiguraciji Šmitovog okidnog kola (Slika 3.) [6], [7]. Signal sa izlaza NE555 tajmera iz astabilnog režima dovodi se na okidni ulaz Šmitovog okidnog kola. Signal koji se javlja na izlazu Šmitovog okidnog kola pogodan je za procesranje na Arduino okruženju i proporcionalan je kapacitivnosti koja se unosi u kolo tajmera (u astabilnom režimu).




Slika 3. 555 tajmer u konfiguraciji Šmitovog okidnog kola.

*C. Procesiranje signala*

Jasno je da vreme punjenja i pražnjenja kondenzatora, zavisi od njegove kapacitivnosti. Upravo te osobine kapacitivnosti se mogu iskoristiti u Arduino okruženju da bi se izmerila nepoznata kapacitivnost kondenzatora. Najjednostavnija varijanta za analizu impulsa već je implementirana u Arduino okruženju i naziva se PulseIn(); funkcija.

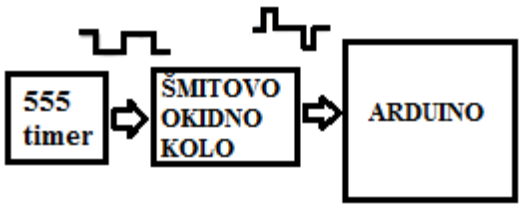
Funkcija PulseIn() u zavisnosti od parametara koji joj se proslede, ima mogućnost da učitava impuls sa digitalnog pina i meri vreme koje protekne od pojavljivanja visokog (ili niskog) naponskog nivoa na određenom digitalnom ulazu do pojavljivanja suprotnog naponskog nivoa na istom ulazu. Teoretski je moguće očitavati signal i sa analognih ulaznig pinova, jer, sa stanovišta mikrokontrolera, ne postoji drastična razlika kod razgraničavanja ovih pinova. Međutim, u slučaju analognog ulaza ostvaruje se znatno manja preciznost. U konkretnom slučaju, koristiće se obe merne mogućnosti, odnosno merenje visokog naponskog nivoa impulsa, i niskog naponskog nivoa impulsa. Pojavljvanje visokog nivoa na digitalnom ulazu Arduino okruženja označava zapravo punjenje kondenzatora. Pojavljivanje niskog naponskog nivoa, označava pražnjenje kondenzatora. Radi obezbeđivanja veće preciznosti, ovo merenje se izvodi pet puta, dok se na kraju izvršenja petlji kao konačan rezultat merenja uzima srednje vreme punjenja, odnosno pražnjenja kondenzatora. Šmitovo okidno kolo upravo ima ulogu da obezbedi pogodan, dovoljno dobar impuls za Arduino okruženje, kako ne bi zbog svoje osetljivosti, Arduino tretirao šum kao impuls koji treba izmeriti.

PulseIn() ima jasno ograničenje po pitanju trajanja impulsa koje treba meriti. Naime, specificirana dužina impulsa iznosi od 10μs do 3 minuta, što jasno ograničava mogućnost merenja velikih kapacitivnosti. Dodatno ograničenje predstavlja napon napajanja sa kojim funkcioniše sam Arduino, a koji iznosi 5V. Ovo predstavlja problem kod kondenzatora koji imaju veći nominalni radni napon od npr. 16V. Teoretski je moguće izmeriti i kapacitivnosti kondenzatora koji rade na 30V, ali se onda ne može garantovati željenu preciznost merenja.

Nakon nalaženja srednje vrednosti izmerenih impulsa, primenom formule 2, računa se kapacitivnost. Pošto se trenutno vrši izračunavanje u nF, potrebno je izvršiti konverziju u prilagođenu veličinu. To se radi jednostavnim množenjem sa 1000 ili deljenjem sa 1000. Ovim smo završili kalkulaciju kapacitivnosti.

$$C = \frac{1.44\times1000}{20800\times(Htime+Ltime)} \quad (2)$$

Na slici 4, prikazan je postupak procesiranja signala koje idu ka Arduino okruženju



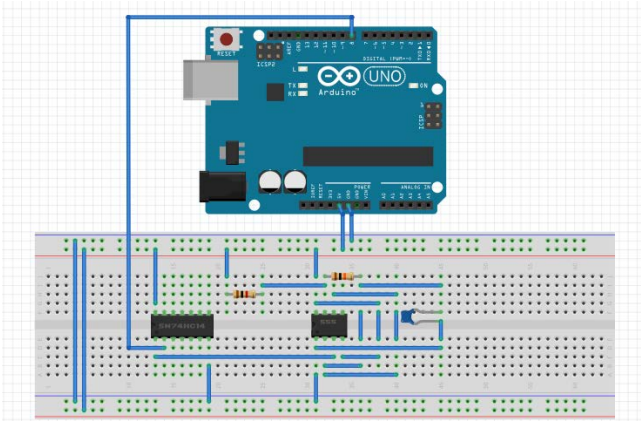

Slika 4. Blok šema procesiranja impulsa.

## III. FAZE RAZVOJA SISTEMA ZA MERENJE KAPACITIVNOSTI

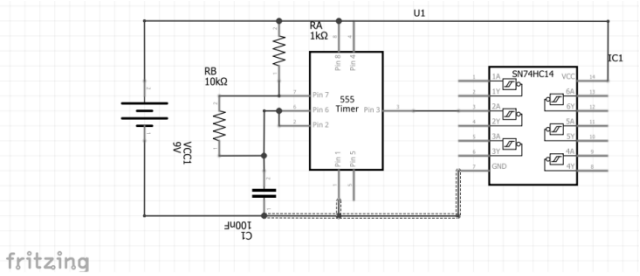
Za realizaciju ovog sistema korišćen je programski paket Fritzing [8].

Prvi korak u realizaciji jeste programiranje ATMEGA3285P-PU mikrokontrolera koji je sastavni deo Arduino razvojnog okruženja. Za to je korišćen Arduino "jezik", što zapravo predstavlja najjednostavniji nivo jezika C++. Svi potrebni kodovi za programiranje mikrokontrolera mogu se naći na [9].

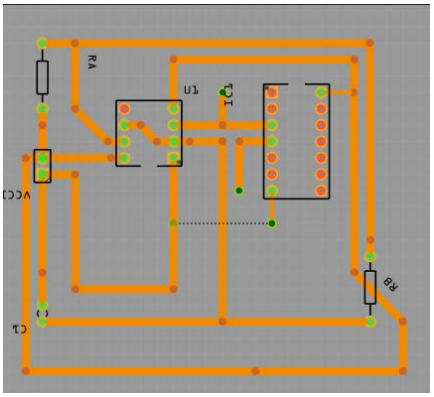
Potvrdu ispravnosti željenog kola, moguće je takođe dobiti u Fritzing programskom paketu. Fritzing paket je relativno lak za korišćenje i spada u programe koji se upotrebljavaju za amaterske projekte. Najjednostavniji način realizacije PCB-a u Fritzingu jeste da se potrebni elementi izređaju na protoploči, kao slici 5. Potom će Fritzing sam predložiti okviran izgled šeme, kao i PCB-a. Na slici 6 prikazan je izgled konačne šeme koju treba realizovati. Takođe, Fritzing može da odradi određene elementarne DRC (Design Rule Check) provere dobijenog izgleda štampane ploče (npr. da li se neke linije seku ili je jedna preblizu druge). Nakon uspešnog iscrtavanja linija, i povezivanja, potrebno je dobijeni fajl exportovati da bi se dobio format dokumenta koji je povoljan za štampu. Dobija se verzija crteža pod nazivom copper_top koja je prikazana na slici 7. Postupkom nagrizanja nezaštićenog bakra ukloni se sve osim provodnih linija i kontaktnih tačaka.



Slika 5. Prikaz Fritzing emulatora protoploče i Arduina.

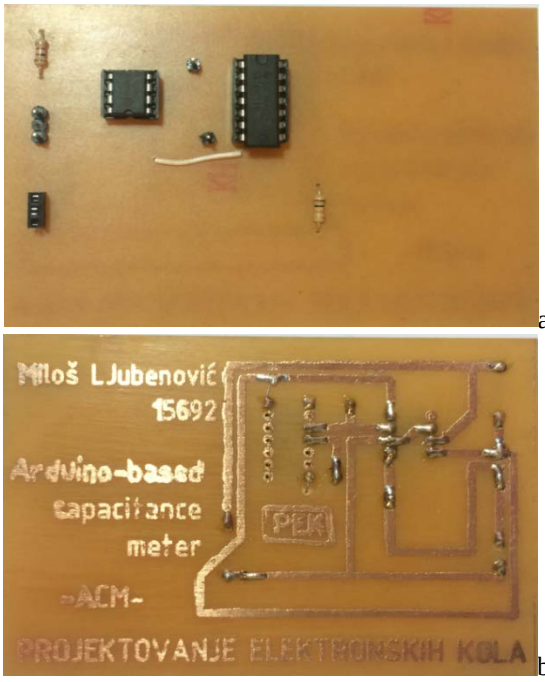



Slika 6. Šema realizovana u Fritzingu.



Slika 7. PCB meraca kapacitivnosti (Fritzing).

Poslednji korak je postavljanje komponenti u odgovarajućim vijama PCB-a i njihovo lemljenje. Rezultat toga prikazan je na slici 8.



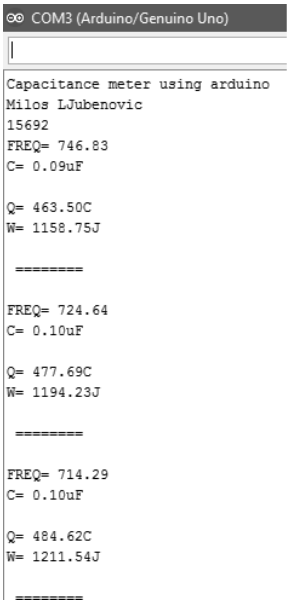

a)

b)

Slika 8. Izgled ploče nakon povezivanja komponenti: a) top-view i b) bottom-view.

## IV. VERIFIKACIJA

Ispravnost realizovanog kola utvrđena je mnogobrojnim merenjima. Rezultat merenja koji se može videti na monitoru odgovarajućeg računara prikazan je na slici 9. Na slici je dat prikaz serijskog porta Arduina za kapacitivnost od 0.1µF




Slika 9. Prikaz serijskog porta arduina za kapacitivnost 0.1µF.

### *A. Relativna greška*

Izračunavanje relativne greške rađeno je na uzorku od 10 kondenzatora, različitih nazivnih kapacitivnosti, i različitih napona. Izmerena kapacitivnost predstavlja srednju vrednost tri merenja kapacitivnosti kondenzatora. Apsolutna greška predstavlja razliku kapacitivnosti na samom kondenzatoru i one koja je izmerena, na prethodno opisanom načinu. Relativna greška predstavlja količnik apsolutne greške, sa nazivnom vrednošću kondenzatora, pomnožen sa 100 radi prikaza u procentima.

TABELA I
REZULTATI MERENJA

| Nazivna kapacitivnost | Izmerena kapacitvnost | Apsolutna greška | Relativna greška (%) |
|---|---|---|---|
| 1 pF | 0.9999pF | $-1*10^{-4}$ | -0.01 |
| 1.5 nF | 1.3349 nF | -0.1651 | -11.0067 |
| 4.7 nF | 4.5998 nF | -0.1002 | -2.13191 |
| 0.15 µF | 0.16 µF | 0.01 | 6.666667 |
| 1 µF 100V | 0.97 µF | -0.03 | -3 |
| 100µF 25V | 98 µF | -2 | -2 |
| 100µF 16V | 103 µF | 3 | 3 |
| 220 µF 25V | 224.6 µF | 4.6 | 2.090909 |
| 470 µF 16V | 483 µF | 13 | 2.765957 |

## V. ZAKLJUČAK

U radu je opisan postupak projektovanja kola za merenje kapacitivnosti diskretnih kondenzatora. Kolo je zasnovano na korišćenju tajmer NE555 integrisanog kola i Arduino okruženja. Kolo je realizovano u vidu štampane pločice, i merenjima je potvrđena njegova ispravnost. U budućnosti je potrebno implementirati LCD displej, smanjiti veličinu PCB ploče i upotrebiti manje verzije integrisanih kola.

## LITERATURA


[1] Predrag Petković, Projektovanje CMOS integrisanih kola sa mešovitim signalima, Univerzitet u Nišu, Elektronski fakultet, Niš, 2009
[2] Vančo B. Litovski, Osnovi testiranja elektronskih kola, Univerzitet u Nišu, Elektronski fakultet, Niš, 2006.
[3] Practical Oscillator Handbook, Irving M. Gottlieb PE
[4] Diskretne mreže i procesiranje signala, Vidosav S. Stojanović, Univerzitet u Nišu, Elektronski fakultet, Niš 2004.
[5] Measuring capacitance & ESR, januar 2014; https://meettechniek.info/passive/capacitance.html
[6] Schmit Trigger – wiki https://en.m.wikipedia.org/wiki/Schmitt_trigger
[7] https://meettechniek.info/passive/capacitance.html
[8] Programski paket Fritzing: http://fritzing.org/
[9] https://pastebin.com/xcPBSaCv

# Realizacija audio efekata za gitaru korišćenjem Raspberry Pi

Igor Mirčić

*Sadrzaj* – Elektronski uređaji koji menjaju način na koji muzički instrument daje zvuk koriste se već duže vreme. Savremeni uređaji za generisanje efekata koriste procesore. Najčešće se koriste sledeći efekti: fuzz/overdrive [1], phaser [2], tremolo, vibrato, wah, delay i reverb [3]. U ovom radu realizovano je sedam pomenutih efekata i štimer za gitaru.

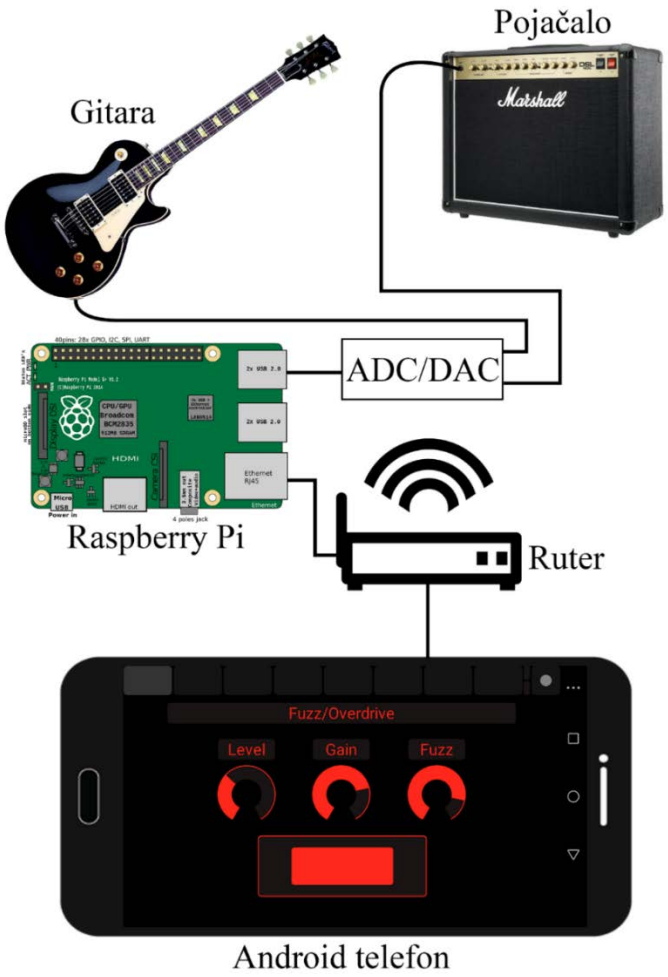
## I. UVOD

Prvi efekti za gitaru su se pojavili 1930. godine i bili su ugrađeni u pojačala za gitaru. Tek 10 godina kasnije su se pojavile prve analogne pedale za gitaru. U današnje vreme za obradu zvuka se koriste i procesori za gitaru. Oni imitiraju zvuk analognih pedala.

Naravno postoji dobar razlog zašto su procesori za gitaru popularni u današnje vreme. Cena analgnih pedela je znatno veća u odnosu na procesor za gitaru. Takođe pedale zauzimaju znatno više prostora. Još jedna dobra strana procesora je što podešavanja za efekte mogu da se snime, dok kod analognih pedala mora svaki put ručno da se podese.

U ovom radu realizovani su fuzz/overdrive, phaser, tremolo, vibrato, auto-wah, delay i reverb efekti kao i štimer za gitaru u Pure Data (PD) vizualnom programskom jeziku za multimediju [4], [5], [6]. Obradu zvuka vrši Raspberry Pi (RasPi) [7], [8] razvojno okruženje. RasPi je embeded računar opšte namene, pristupačan po ceni, malih dimenzija i pruža mogućnost priključivanja nestandardnih dodatnih komponenata i opreme. Za A/D i D/A konverziju koristi se eksterna muzička kartica koja se povezuje na USB port RasPi-a. Za kontrolisanje efekata koristi se Android telefon i TouchOSC [9] aplikacija. Android telefon i RasPi su povezani na isti ruter. Komunikacija se vrši preko Open Sound Control (OSC) protokola [10]. Open Sound Control (OSC) je protokol za komunikaciju između računara, sintisajzera zvuka i drugih multimedijalnih uređaja koji su prilagođeni za savremenu mrežnu tehnologiju. OSC donosi prednosti savremene tehnologije umrežavanja u svet elektronskih muzičkih instrumenata uključujući interoperabilnost, tačnost, fleksibilnost i unapređenu organizaciju i dokumentaciju. Na Slici 1. prikazana je opšta šema povezivanja opreme.

Igor Mirčić je student na smeru Elektronika, Elektronski fakultet, Univezitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: mircha90@gmail.com



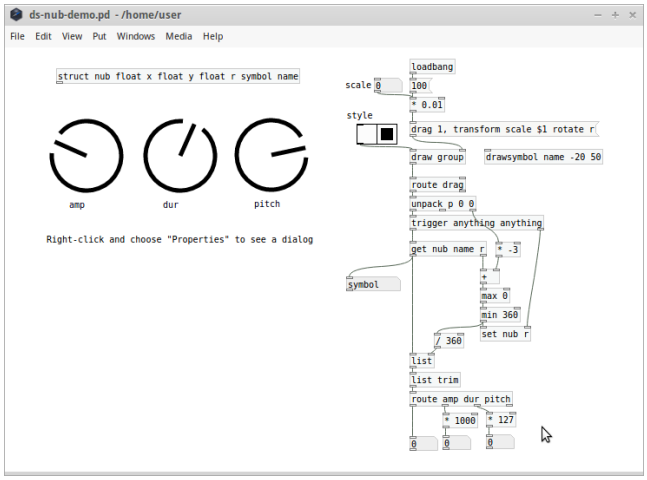

Slika 1. Opšta blok šema povezivanja opreme.

## II. TEHNIČKE KARAKTERISTIKE RASPBERRY PI

RasPi je embeded računar koji je smešten na jednoj PCB ploči. Raspberry Pi ima na sebi sistem na čipu (system on chip) BCM283x (x=5, 6 ili 7 zavisno od verzije RasPi) proizvođača Broadcom. On je vrlo sličan čipovima koji su se nalazili na starijim pametnim telefonima. Na tom čipu je procesor ARM (ili sličan zavisno od verzije RasPi), grafički porocesor i RAM memorija. Na Slici 2 prikazan je RasPi.



Slika 2. Raspberry Pi 3 Model B.




Slika 3. Pure Data.

RasPi 3 Model B koristi ARMv8 64-bitni procesor sa četiri jezgra sa osnovnim taktom od 1.2GHz. Kao što se vidi na Slici 2. RasPi ima 4 USB porta, LAN port, Full HDMI port, 40 GPIO pina, 3.5mm audio ulaz i Micro USB ulaz za napajanje. RasPi 3 je kompletan embeded računar koji se dobija dodavanjem tastature, miša i monitora i priključka za internet. Njegove procesorske mogućnosti su velike.

RasPi koristi napajanje od 5V. U zavisnosti koliko periferija je priključeno na RasPi, napajnje može da bude od oko 700mA do 1000mA. Ako ništa nije priključeno, RasPi može da koristi i napajanje od 500mA [8].

RasPi pruža veliki izbor operativnih sistema. Najčešće su to specijalizovane Linux distribucije, a kao mogućnost imamo i Windows 10 IOT Core. U ovom radu korišćena je standardna Linux distribucija koja se zove Raspbian. Raspbian je otvoreni operativni sistem baziran na Debian Linux distribuciji.

## III. TEHNOLOGIJE ZA RAZVOJ AUDIO EFEKATA

Postoje dve popularne metode za razvoj audio efekata. Jedna od tih metoda je korišćenje specijalizovanih vizuelnih programskih jezika za razvoj audio efekata. Dva najpopularnija takva jezika su MaxMSP i Pure Data [4]. Druga metoda bi bila korišćenje klasičnih programskih jezika sa dodacima za obradu signala. Za ovu metodu mogu da se koriste cSound, SuperCollider, ChucK itd.

Vizuelni programski jezici imaju prednost u brzini razvoja audio efekata. MaxMSP je najpopularniji komercijalni jezik za pravljenje audio efekata. Dok Pure Data predstavlja otvorenu alternativu koja je dostupna pod licencom sličnoj BSD licenci. Za razvoj audio efekta u ovom radu korišćen je Pure Data (Slika 3).

PD je vizuelni programski jezik koji je razvio Miller Puckette 1990-ih godina za stvaranje interaktivnih kompjuterskih muzičkih i multimedijskih radova. Iako je Puckette glavni autor programa, PD je otvoreni projekat sa velikom brojem ljudi koji rade na razvoju programa. PD radi na GNU/Linux, Mac OS, iOS, Android i Windows. Portovi postoje za FreeBSD i IRIKS.

PD program je napravljen u C programskom jeziku. Obzirom da je PD otvoren projekat, mogućnost pristupa source kodu i dodavanje novog koda u C jeziku ne predstavlja veliki problem. PD takođe omogućava dodavanje koda i u drugim programskim jezicima kao što su Python, Scheme, Lua, Tcl i drugi.

Programi napisani u PD vizualnom jeziku se zovu pečevi (patches). PD jezik je objektno orijentisan i interpretiran jezik. Iz razloga što je reč o interpretiranom jeziku ne moramo da rekompajliramo program svaki put kad promenimo kod. Pravljenje pečeva je dosta zanimljiv, interaktivan i brz proces. Naravno program napisan u C jeziku bi trebao da bude brži, ali razvoj programa bi bio znatno sporiji. PD nam omogućava brz razvoj i mogućnost da, ako je potrebno, uvek možemo da prebacimo PD peč u C kod kako bismo izvukli performanse.

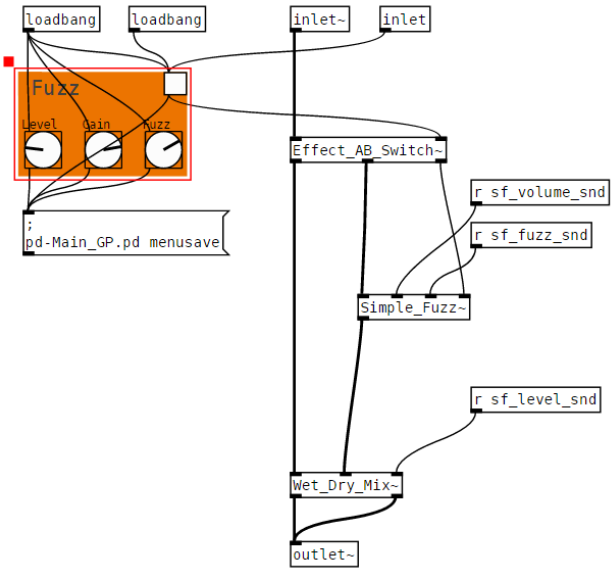
## IV. FUZZ EFEKAT ZA GITARU

Kao primer realizovanog efekta u PD vizuelnom programskom jeziku biće predstavljen fuzz efekat.

*Distortion*, *fuzz* i *overdrive* su oblici obrade audio signala koji se koriste za promenu zvuka električnih muzičkih instrumenata, obično povećavajući njihov signal, stvarajući "*fuzzy*", "*growling*", i "*gritty*" ton. Distorzija se najčešće koristi sa električnom gitarom, ali se može koristiti i sa drugim električnim instrumentima kao što su bas gitara ili električni klavir. Gitaristi koji sviraju električni bluz prvobitno su dobili overdrive zvuk pojačavajći *gain* (potenciometar) na svojim pojačalima za

gitaru sa vakumskim cevima, što je uzrokovalo da se signal izobliči. Iako se *overdrive* i danas dobija korišćem pojačala sa vakumskim cevima, pogotovo u žanrima kao što su *blues* i *rockabilly*, još od 1960-ih godina razvijeni su brojni drugi načini stvaranja distorzije, kao što su pedale za efekt distorzije. Overdrive ton električne gitare je ključni deo mnogih žanrova, uključujući blues i mnoge rok muzičke žanrove, naročito *hard rock, punk rock, hardcore punk i heavy metal* muziku.

Svi efekti za gitaru koriste sub peč šablon (Slika 4) za kontrolu efekta. Ono što se menja je broj potenciometara i njihove minimalne i maksimalne vrednosti.

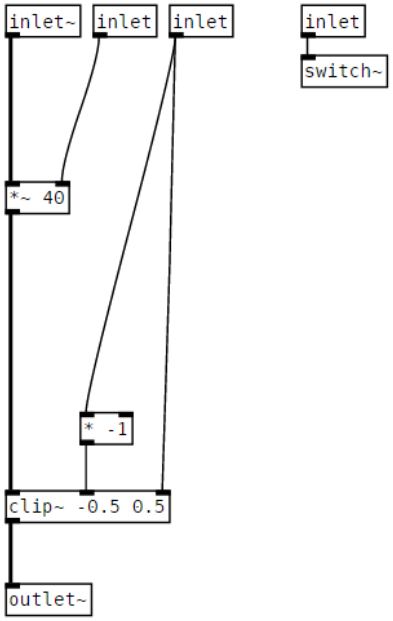



Slika 4. Fuzz sub peč.

Na Slici 4 je dat sub peč za *fuzz* efekt. Ovaj sub peč sadrži tri apstrakcije. Simple_Fuzz~ [6] apstrakciju koja vrši Obrada audio signala. Zatim Effect_AB_Switch~ [6] koja sadrži ON/OFF kontrolnu logiku. I konačno Wet_Dry_Mix~ koja kontroliše odnos obrađenog i ne obrađenog signala.

Potenciometri imaju različite minimalne i maksimalne vrednosti. Level potenciometar kod svih efekata može da ima vrednosti između 0 i 1. *Gain* potenciometar ima vrednost od 1 do 100. Dok Fuzz može da ima vrednost od 1 do 0.2. Kod *fuzz* potencimetra je praktično minimalna vrednost veća od maksimalne, odnostno obrnute su.

Na Slici 5 data je Simple_Fuzz [6] apstrakcija koja radi obradu audio signala. Fuzz efekt je izuzetno jednostavan. Ono što radi je da pojača signal od 1 do 100 puta a zatim ga klipuje. Ovo klipovanje proizvodi distorziju zvuka, a to je u ovom slučaju efekt koji želimo.



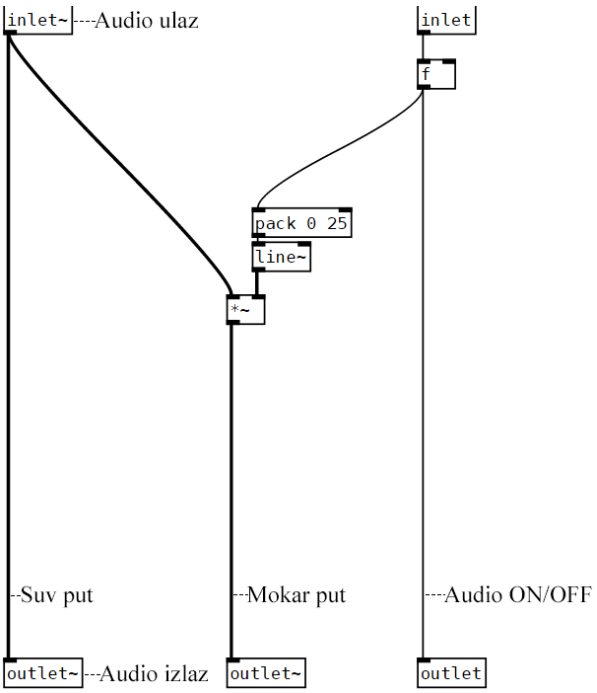

Slika 5. Simple_Fuzz apstrakcija.

Kao što se vidi na Slika 5 imamo četiri inleta (ulaza), od toga jedan je za audio i tri kontrolna. Prvi inlet sa leva je audio signal, zatim je pored njega inlet za Gain signal, a pored njega se nalazi inlet za Fuzz signal. Četvrti inlet se koristi za uključivanje/isključivanje izračunavanja u ovoj apstrakciji. Ovde je jako bitno napomenuti da kada je isključeno izračunavanje ne trosi se CPU vreme na tu apstrakciji. Dakle što manje efekta imamo uključeno manje CPU vremena se troši na izračunavanje. Ovo je jako bitno kod štimera jer on troši znatno više CPU vremena u odnosu na druge efekte.

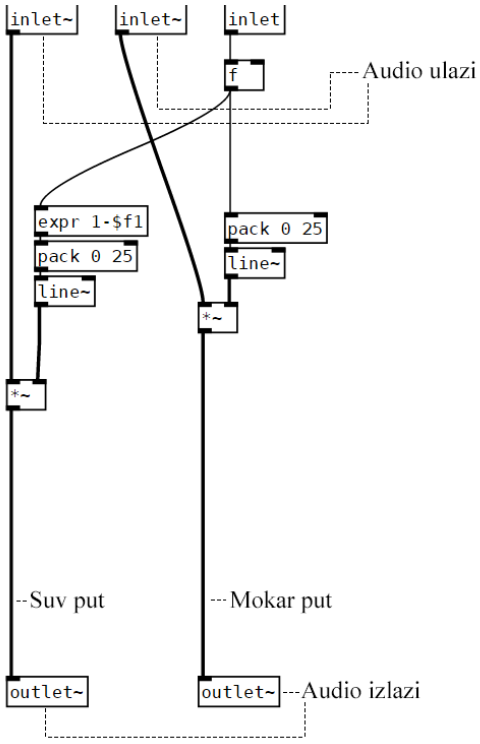
Na Slici 6 prikazana je Effect_AB_Switch apstrakcija. Ona ima za cilj samo jednu stvar, a to je da utiša polako zvuk kada se isključuje izračunavanje za neki od efekta. Objekat line~ generiše brojeve od 1 do 0 u vremenskom prediodu od 25ms. Kada pomnožimo audio ulaz sa ovim brojevima dobijamo utišavanje zvuka. Ovo je neophodno kako se ne bi čulo pucketanje prilikom isključivanja nekog od efekata.

Na Slici 7 data je apstrakcija Wet_Dry_Mix. Kao što je već rečeno ova apstrakcija kontroliše odnos obrađenog t.j. mokrog signala i ne obrađenog t.j. suvog signala. I kod ove apstrakcije koristi se line~ objekat kako se menjanje vrednosti ne bi desilo odmah već u toku malog vremenskog perioda i kako se ne bi čulo pucketanje zvuka.

Treći inlet je kontrolni i dobija *level* vrednosti koje su između 0 i 1. Recimo da na ulazu dobijemo vrednost 0.4, mokri put će biti pomnožen sa 0.4 i dobićemo 40% mokrog signala. Suv signal se množi sa 0.6 t.j dobijamo 60% suvog signala. Ovo se vidi iz expr 1-$f1 gde je $f1 ulaz koji je 0.4 tako da dobijamo 0.6 na izlazu koji se dalje prosleđuje objektu koji množi audio signal.




Slika 6. Effect_AB_Switch apstrakcija.




Slika 7. Wet_Dry_Mix apstrakcija.

## VI. ZAKLJUČAK

U ovom radu realizovani su efekti za gitaru koji rade na RasPi razvojnom okruženju. RasPi koristi ARM procesor opšte namene. Procesori opšte namene odavno mogu da obrađuju digitalne signale u realnom vremenu. Iako je ovakva obrada signala odavno moguća, tek u skorije vreme možemo da kažemo da se isplati. Novi procesori opšte namene, kao što je procesor koji koristi RasPi, su relativno jeftini i dovoljno moćni da rade obradu signala u realnom vremenu. Moje lično mišljenje je da će se procesori opšte namene sve više koristiti kada se prave manje serije uređaja.

## LITERATURA

[1] Fuzz/Overdrive/Distortion wiki: https://en.wikipedia.org/wiki/Distortion_(music)
[2] Phaser wiki: https://en.wikipedia.org/wiki/Phaser_(effect)
[3] "Physical Audio Signal Processing", by Julius O. Smith III, W3K Publishing, 2010, ISBN 978-0-9745607-2-4.
[4] Pure Data wiki: https://en.wikipedia.org/wiki/Pure_Data
[5] Pure Data tutorijali od prof. dr. Rafaela Hernandeza: https://youtu.be/rtgGol-I4gA
[6] Guitar extended Pure Data, RasPi blog od Pierra Massata: https://guitarextended.wordpress.com/
[7] Raspberry Pi wiki: https://en.wikipedia.org/wiki/Raspberry_Pi
[8] Raspberry Pi specifikacija za napajanje: https://www.raspberrypi.org/documentation/hardware/raspberrypi/power/README.md
[9] TouchOSC: https://hexler.net/software/touchosc
[10] Open Sound Control: http://opensoundcontrol.org/introduction-osc

# Ultrazvučni senzor parkiranja

Stefan Bibulović

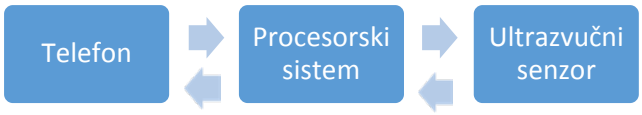
*Sadržaj* – Cilj ovog rada je da pokaže kako uz jednostavni sistem svaki automobil nižeg budžeta može imati senzor za parkiranje realizovano Atmega328 [1], Bluetooth [2] i Ultrazvučnim senzorim HC-SR04 [3]

## I. Uvod

Ideja ovog rada je da pokaže kako na krajnje jednostavan način postoji mogućnost realizacije senzora za parkiranje čija bi ugradnja bila krajnje jednostavna bez prethodnog znanja automehanike. Kao glavni deo korišćen je Atmega328 mikrokontroler, bluetooth za komunikaciju sa Android [4] telefonom i ultrazvučni senzor za prikupljanje podataka iz okoline.

## II. Ideja

Cilj projekta je da se putem telefona, i okruženja razmenjuju podaci o udaljenosti objekta iza automobila, bez specijalne ugradnje senzora za prepoznavanje kada vozač ubaci u rikverc. Sve je to zamenjeno aplikacijom u kojoj vozač sam unosi da li je u režimu rikverca ili ne.




Slika 1. Blok šema sistema.

Na slici 1. je prikazan način funkcionisanja uređaja. Preko aplikacije telefona se šalju komande putem bluetooth-a procesorskom sistemu koji uključuje sistem. Da bi se izbeglo otvaranje menjača i ugradnja senzora za prepoznavanje rikverca ili prve brzine, sam korisnik to jednostavnim komandama u aplikaciji na svom telefonu objašnjava uredjaju u kom je položaju menjač.

## III. Razvoj uređaja

Uređaj je sačinjen iz dva dela. Prvi deo sadrži aplikaciju kojom korisnik direktno upravlja i drugi deo koji sadrži bluetooth modul, ultrazvučni senzor otstojanja i kontrolnu jedinicu.

Stefan Bibulović, student četvrte godine na modulu Elektronske komponente i mikrosistemi Elektronskog fakulteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: stefan.bibulovic@elfak.rs

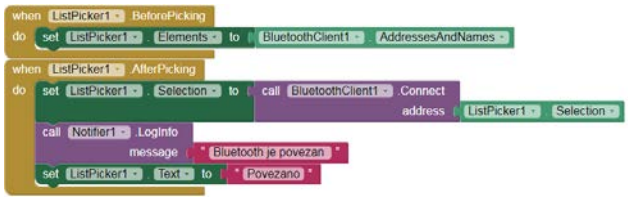
### *A. Razvoj Android aplikacije*

Za realizaciju Android aplikacije korišćena je web online MIT App Inventor [5]. MIT App Inventor omogućava da se kompleksan programski jezik za programiranje android aplikacija svede na nivo blokova. U samoj aplikaciji sve se svodi na prevlačenje određenih blokova koji uz sebe nose opis komande koju sa sobom nose. Prvi deo u izradi aplikacije jeste osmisliti visuelno okruženje koje će kasnije korisnik upotrebljavati.




Slika 2. Dizajn aplikacije.

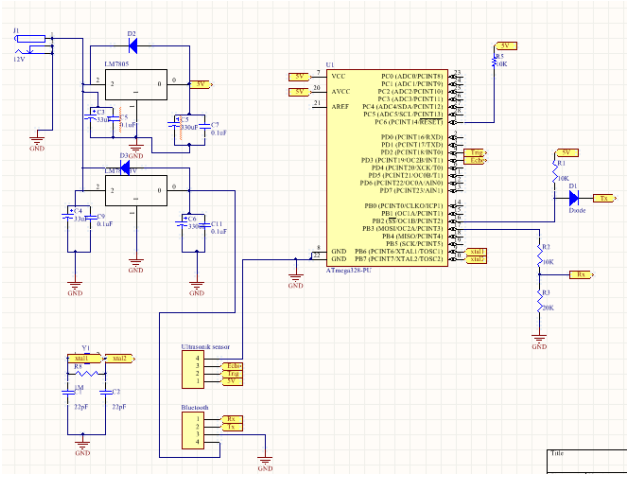
U pozadini vizuelnog dela kriju se pomenuti blokovi prikazani na slici 3. Ovaj deo bloka ima ulogu konekcije uređaja sa aplikacijom.




Slika 3. Blok šema za konekciju Bluetooth-a.

### *B. Razvoj i šema uređaja*

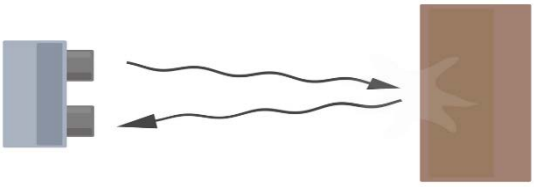
Kod drugog dela sistema gde se nalazi Bluetooth modul, ultrazvučni senzor i kontroler korisnik nema pristupa.



Slika 4. Šema uredjaja za izračunavanje odstojanja i komunikacija sa aplikacijom.

Glavna stavka sa šeme Slika 4. jeste da se kompletno napajanje ovog uredjaja vrši se direktno sa električne instalacije automobila. Na samom ulazu uredjaja poznato je da dolazi 12V, na samom ulazu u kolo projektovana su dva stabilizatora napona. Prvi stabilizator LM7805 [6] ima ulogu da ulazni napon od 12V stabilize na 5V koji je potreban za napajanje mikrokontrolera Atmega328 i ultrazvučnog senzora. Stabilizator 78M33 [7] ima ulogu stabilizacije ulaznog napona od 12V da stabiliše na 3.3V koji je potreban bluetooth modulu.

Modul koji prikuplja informacije iz okoline jeste ultrazvučni senzor HC-SR04 koji je korišćen u ovom uređju. Princip rada ovog senzora zasniva se na emitovanju zvučnih impulsa visoke frekvence. Ilustrovano na slici 5.



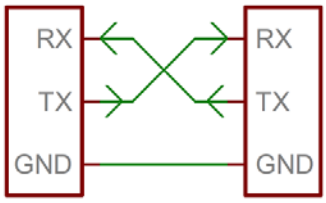
Slika 5. Ilustracija ultrazvučnog senzora.

Ako se zna da je prenos zvuka u vazduhu 340 m/s, distanca udaljenosti dobija se kada vreme koje je potremno da signal predje do objekta i vrati se do senzora pomnoži sa 340m/s i podeli sa 2. Senzor kotroleru šalje distance u vidu signala sa odredjenim trajanjem.

cm = pulseIn(EchoPin, HIGH) / 58.0; //Echo vreme prebačeno u cm

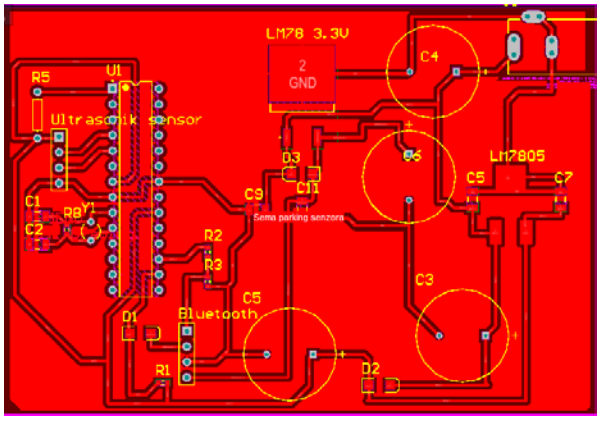
cm = (int(cm * 100.0)) / 100.0; // podešavanje odstojanja na dve decimale

U delu programa iz mikrokontrolera prikazano je da signal dolazi na EchoPin i deli se brojem koji se iščitava iz dokumentacije senzora. Promenjiva cm usvakom trenutku ima vrednost distance senzora od objekta. Informacija se dalje prenosi preko Rx i Tx komunikacije na bluethoot modul sistem vezivanja prikazano na slici 6. U praktičnoj realizaciji postavljena je diode direktno polarisana od mikrokontrolera kako bi se izbegao prestanak rada modula.




Slika 6. Sistem vezivanja Rx i Tx komunikacije.

Kompletno iscrtavanje šeme izvršeno je u programu Altium Designer [8], kao i PCB proektovanje pločice prikazano na slici 7.

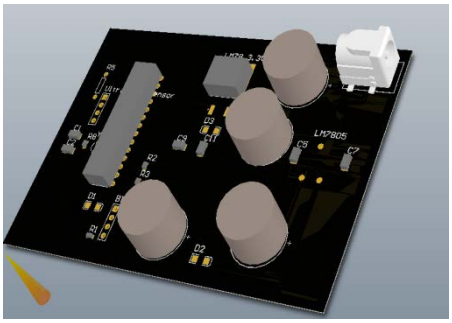



Slika 7. Projektovanje PCB-a za uredjaj.

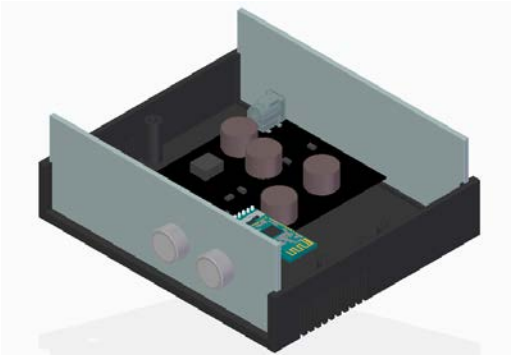
Sam program ima veliku prednost, konstruisana šema se automacki prebacuje na pcb projekat a površina na kojoj se rasporedjuju komponente se ručno unose u zavisnosti od raspoloživih pločica u praksi. Pre same izrade i odabira kućišta izvršava se vertuelna provera. Iz programa Altium Designer izvozi se STEP model prikazano na slici 8.

3D model izvezen u step formatu uvozi se u program Solid Edge[9] koji ima za cilj u ovom slučaju da konstruisan uredjaj u 3D-u spoji sa kućištem koji poseduje svoj 3D model i softverski proveri da li su ugradjene komponente kompatibilne sa kućištem. Kako ne bi došlo do problema nakon izrade uredjaja. kao na slici 9.

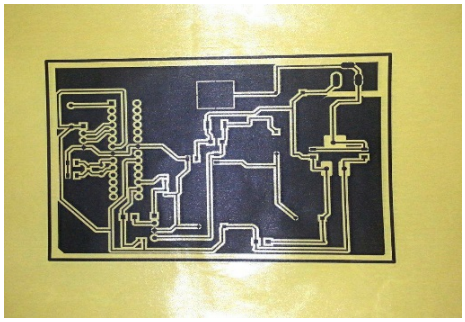
Nakon svih softverskih provera izradjuje se pločica sa izvezenim maskama iz Altium Designer-a, kao na slici 10. Nakon čega se pristupa ugradnji komponenata po izrađenoj šemi slika 11. Kompletan sistem pakuje se u kućiste i dobija finalni proizvod slika 12.



Slika 8. STEP model 3D prikaz.



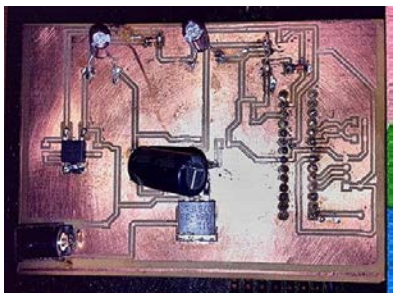
Slika 9. Provera kompatibilnosti sklopa.



Slika 10. Maska PCB-a.



Slika 11. Ugradnja komponenata na izrađenoj pločici.



Slika12. Izgled krajnjeg produkta.

## IV. ZAKLJUČAK

Ovim projektom je pokazano kako uz minimalno ulaganje postoji način izrade pomoći pri parkiranju u rikverc. Mana ovog sisteme jeste korišćenje jednog ultrazvučnog senzora čime se ne dobija dovoljna preciznost kod automobila ovalnog oblika. Problem se može rešiti ubacivanjem sistema od više senzora koji će praviti aproksimaciju daljine u odnosu na sve ivice automobile.

## ZAHVALNICA

Autor se zahvaljuje kolegama Aleksandri Divović i Milanu Ivkoviću na pruženoj nesebičnoj pomoći pri izradi rada .

## LITERATURA

[1]http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf
[2]Bluetooth modul (https://Learn.sparkfun.com/tutorials/bluetooth-basics)
[3] HC-SR04 (https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf)
[4] Android (https://www.android.com/)
[5] MIT App Inventor (http://appinventor.mit.edu/explore/about-us.html)
[6] LM7805 (http://www.ti.com/lit/ds/symlink/lm3805.pdf)
[7] LM78M33 (http://www.ti.com/lit/ds/symlink/lm78m33.pdf)
[8] Altium Designer (By Ben Jordan Editors: Max Clemons, Ben Jordan)
[9]SolidEdge (https://www.plm.automation.siemens.com/en/products/solid-edge/)

# Integracija sofisticirane real time web aplikacije uz pomoć soketa i kompaktne custom dizajnirane igrice korišćenjem RPC-a

Jovan Pešić, Petar Maksimović

*Sadržaj* – U ovom radu je dat primer kombinacije web aplikacije, koja je bazirana na real time-u, i MOBA (Massive Online Battle Arena) online igrice "War of Mages", u kojoj je akcenat dat na sukobu heroja između više igrača. Svi objekti su custom dizajna (crtani ručno).

## I. TEHNOLOGIJE

Web aplikacija je rađena u ASP.NET-u, MVC (Model – View – Controller ) arhitekturi. Model predstavlja bazu koja je rađena u Microsoft SQL-u (SQL Server 2016 Microsoft), a za integraciju sa sajtom je korišćen Entity Framework kao objektno-relacioni maper. Za View (Frontend) je iskorišćen Razor engine u kombinaciji sa Javascript-om i JQuery-em. Controller, odnosno aplikaciona logika (Backend) je napisana u C#-u. Igrica je rađena korišćenjem Unity Game Engine-a i dizajnirana je kao Desktop aplikacija. Za povezivanje sa serverom koristi se Restful Service. Napravljena je tako da se igra online, sa mogućnošću višeplejerskog igranja, uz pomoć Unity Master Server-a koji povezuje igrače u jednoj partiji. Uparivanje igrača se vrši na osnovu poena(iskustva) korišćenjem Machmaker-a.

## II. APLIKACIJA

### *A. Funkcionalnost web aplikacije*

Danas se masovno koristi internet i pretraživanje preko web sajtova. Zbog toga je veoma korisna stvar da svaka osoba koja je zainteresovana za igricu može da vidi pregled ključnih podataka igrice, povezivanje i komunikaciju sa ostalim igračima, kao i samo rangiranje igrice. Web aplikacija je napravljena da upotpuni igricu.

Korisnik ukoliko želi da se registruje, dobija meil sa linkom koji vodi ka krajnjoj registaciji gde bira svoje korisničko ime i šifru.

Nakon toga ima pristup sajtu i mogućnost skidanja (download-ovanja) igrice. Postoji mogućnost pregleda svog profila, sa listom trenutnih item-a u posedu, korišćenih heroja, statistiku same igre, kao i listu prijatelja.

U prodavnici korisnik može da vidi tabelu svih item-a, njihove osobine i mogućnost kupovine uz pomoć zlatnika.

Jovan Pešić je student Elektronskog fakulteta, Univerziteta u Nišu, smer Računarstvo i informatika, Aleksandra Medvedeva 14, 18000 Niš, Srbija. E-mail: enimen@elfak.rs.

Petar Maksimović je student Elektronskog fakulteta, Univerziteta u Nišu, smer Računarstvo i informatika, Aleksandra Medvedeva 14, 18000 Niš, Srbija. E-mail: petar21121995@gmail.com




Slika 1. Registrovanje.



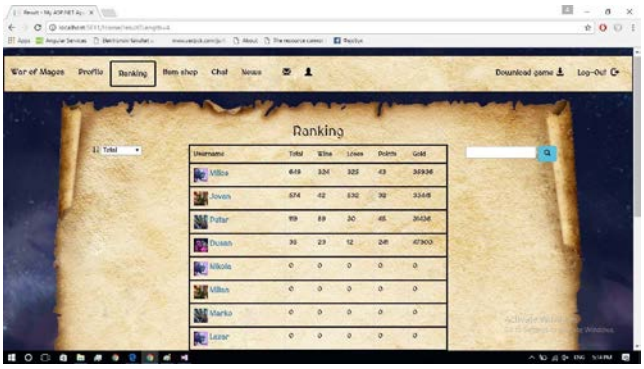

Slika 2. Korisnički profil.




Slika 3. Kupovina item-a.

Osim toga može se pregledati statistika svih igrača, njihovo rangiranje, kao i njihovi profili. Korisnici su u mogućnosti da prate najnovije informacije o aktuelnim

dešavanjima i promenama u igrici. Može se poslati zahtev za prijateljstvo nekom igraču, a on ima mogućnost prihavatanja zahteva ili odbijanja. Za pregled zahteva za prijateljstvo i poruka od prijatelja je iskoriscen Ajax (asinhroni Javascript i XML/JSON).




Slika 4. Rangiranje.

Sajt ima podršku da omogući korisnicima međusobno dopisivanje (chat-ovanje), tj. pregled korisnika sa kojima se logovani korisnik najskorije dopisivao (njihov status, da li su online/offline, da li trenutno igraju igricu i broj nepročitanih poruka). Klikom na određenog korisnika otvara se chat sa njim (Slika 5.) [6], [8]. Na chat-u se mogu skrolovati poruke, videti kada korisnik kuca (typing), kao i kada je video primljenu poruku. Takođe, postoji opcija da se vide sve nepročitane poruke od svih prijatelja klikom na ✉. Dizajniran je tako da se na poslednje poslatoj poruci vidi slika korisnika.




Slika 5. Chat i lista prijatelja.

*B. Realizacija real time-a u web aplikaciji*

Za real time je iskoriscen SignalR [6] koji se sastoji od web soketa i RPC-a (Remote Procedure Call), a za indentifikovanje trenutnih korisnika se koristi Singlton koji u sebi sadrži hash tabelu, gde se na osnovu korisničkog imena korisnika (ključa) dobijaju sve otvorene konekcije klijenata (ConnectionId).

Korisnici uspostavljaju perzistentnu vezu sa serverom koriscenjem web soketa, a za međusobnu komunikaciju se koristi RPC, odnosno mogucnost da klijent poziva udaljene funkcije na serveru i obrnuto. Za identifikovanje korisnika koji trenutno igraju igricu se koristi poseban parametar koji se šalje prilikom uspostavljanja konekcije iz igrice ka web sajtu. Server ima posebnu klasu, odnosno serverski Hub koji vodi računa o svim konektovanim korisnicima i implementira funkcije za obavestenje klijenata o dogadjajima.

C. *Funkcionalnost igrice*

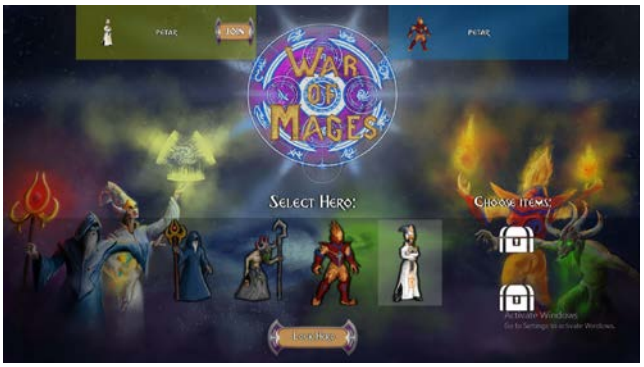
Igrica je zasnovana na sukobu heroja. Jedna partija se sastoji iz dva tima od više igrača (broj igrača mora biti isti u oba tima), a cilj igre je pobediti protivnički tim. Igrač ima pravo na jednog heroja koji ima svoje specijalne sposobnosti. Herojeve sposobnosti se mogu poboljšati unapređenjem njegovih mogućnosti, tj. item-a (ima pravo na dva item-a po partiji). Pobedom/porazom, ubistvom, asistencijom i smrću se dobijaju/oduzimaju zlatnici uz pomoć kojih se mogu kupiti Item-i u prodavnici na sajtu.

Pri pokretanju igrice, svaki igrač mora da se loguje na nalog. Nakon toga, igrač ima mogućnost da pogleda svoj profil, listu item-a sa njihovim specifikacijama, link ka sajtu, kao i da odabere tip igre (1 vs 1, 2 vs 2, 3 vs 3, 4 vs 4 i custom).



Slika 6. Glavni meni.

Odabirom tipa, šalje se zahtev Master serveru koji preko Machmaker-a vrati traženu sobu (Slika 7). Ulaskom u određenu sobu, igrać ima mogućnost odabira heroja, kao i item-a.




Slika 7. Soba.

Partija počinje kada su svi igrači spremni. Igrica je napravljena kao platformska 2D grafika. Timovi se sponuju na dva različita dela mape (okruženje gde se igračev heroj nalazi). Svaki heroj ima dve ofanzivne, dve defanzivne, jednu ultimate i jednu heal magiju. Njihovim korišćenjem se smanjuje količina mana koje heroj poseduje. (Slika 9.)



Slika 8. Game play 1.

Periodično se na mapi prikazuje paket koji povećava jačinu neke osobine. Heroji imaju odredjenu vrednost za svaku sposobnost (healt, mana, damage, diffence, healtregen, manaregen).



Slika 9. Game play 2.

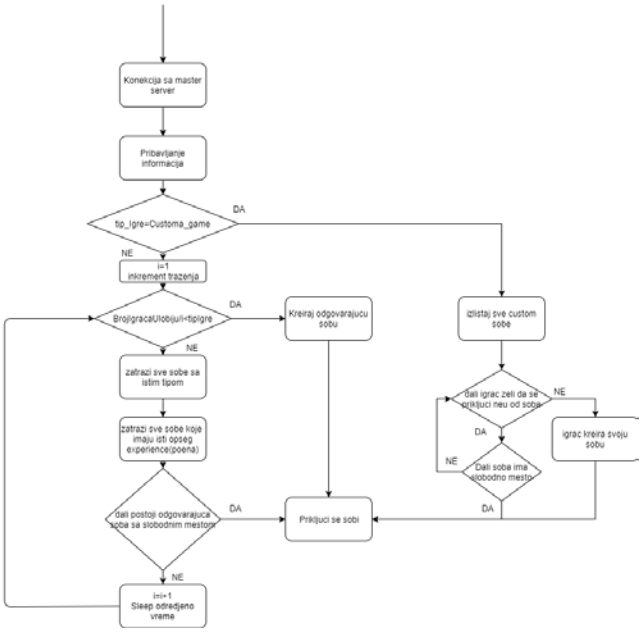
Na kraju partije se pamti postignuti experience i osvojeni zlatnici svakog igrača (Slika 10.).




Slika 10. Partija završena.

*D. Realizacija igrice*

Za implementaciju Matchmaking-a [1], [2], [3], [4], [5] se koristi servis Unity-a pod nazivom Master Server. Svi igrači kada iz glavnog menija odu na start game, aplikacija se povezuje sa Master serverom i u zavisnosi od tipa i experience-a, traži se lista soba. U slučaju da nije custom game, u pozadini se vrši biranje sobe sa slobodnim mestom, a ukoliko ne postoji, čeka se određeno vreme. To se ponavlja nekoliko puta u zavisnosti od broja korisnika koji su trenutno online. Ako se ni tada ne nađe, kreira se nova soba. U slučaju da je tip igre custom game tada se od Master servera traže sve sobe koje su tipa custom. Igrač, ukoliko ima slobodnih mesta, može da uđe u sobu, ili kreira svoju. Ulaskom u sobu, igrač se smešta u odgovarajući tim (Slika 11.).




Slika 11. Algoritam za Matchmaking.

Za implementaciju multiplayer sistema se koristi Unity Server koji služi da spoji igrače koji su u istoj partiji. Prilikom ulaska u igricu, bira se igrač koji će biti server i on kontroliše i sinhroniše ostale klijente (igrače u istoj partiji). U slučaju da taj igrač izgubi konekciju, vrši se migracija ka nekom od klijenata i tada on postaje server partije. Multiplayer koristi RPC protokol (funkcije koje se nalaze na serveru su cmd, a funkcije koje su na klijentu su rpc). Te funkcije se koriste za sinhronizaciju igrača (njihovih atributa, položaja, animacija, magija, itd..). Prilikom smrti heroja, igrač ima mogucnost da posmatra nekog iz svog tima.

## III. ZAKLJUČAK

Smatramo da smo realizacijom ovog projekta, naučili da programiramo u gore navedenim tehnologijama i

programskim jezicima (c#,asp .NET MVC ,SignalR, Javascript, Unity Game Engine). A kao rezultat toga smo dobili adekvatan spoj jedinstvene i interesantne igrice za mlade, i sajta sa neverovatnom funkcionalnošću online chat-a i konekcije sa korisnicima.

## ZAHVALNICA

Želimo da se zahvalimo našim prijateljima koji su nam pomogli u realizaciji ovog projekta. Aleksandar Jakovljević je učestvovao u nekim delovima pravljenja igrice, a Pavle Golubović je ručno crtao sve heroje i item-e.

## LITERATURA

[1] Joshua GLAZER, Sanjay MADHAV, "Multiplayer game programming - Architecting Networked Games", November 2015.
[2] Timo JETSONEN, "Development of online game prototype with Unity engine Multiplayer solutions", Bachelor's thesis, May 2016.
[3] Igor GALOCHKIN, "Implementation of a cross-platform strategy multiplayer game based on Unity3D", January 2015.
[4] Will GOLDSTONE, "Unity Game Development Essentials", October 2009.
[5] https://docs.unity3d.com/Manual/index.html
[6] https://docs.microsoft.com/en-us/aspnet/signalr/
[7] https://stackoverflow.com/
[8] https://plugins.jquery.com/

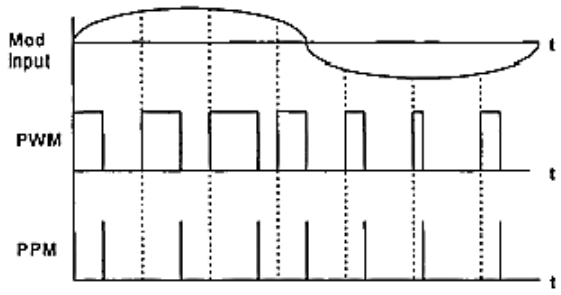
# Optički predajnik audio signala sa impulsnom širinskom modulacijom

Anđela Jovanović, Milan Ilić

*Sadržaj* - U ovom radu predstavljen je postupak projektovanja i realizacija optičkog modulatora koji ulazni audio signal pretvara u impulsno-širinski modulisani signal kojim se zatim moduliše laser kako bi se postigla impulsna modulacija optičkog zračenja lasera. Fizički je realizovan pototip predajnika, a zatim je potvrđeno njegovo ispravno funkcionisanje i date su merene karakteristike.

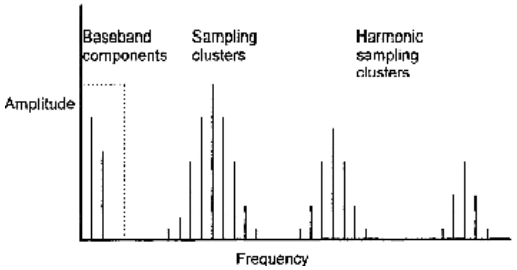
## I. UVOD

Analogna amplitudska modulacija lasera se izbegava kad god je to moguće, zbog različitih uticaja koji mogu dovesti do varijacije amplitude. Ovi uticaji se direktno reflektuju na signal detektovan od strane prijemnika i pretvaraju u neželjene smetnje. Ovi nedostaci se mogu predupredi ako se koristi impulsna (ili čak digitalna) modulacija, kod koje nivoi srednje i vršne snage ne igraju presudnu ulogu u prenosu informacija. S obzirom da je direktna analogna modulacija intenziteta zračenja lasera veoma jednostavna, a da digitalni prenos zahteva sinhronizaciju i kodovanje signala, impulsne modulacije mogu biti zanimljiva alternativa koja se nalazi negde na pola puta u smislu složenosti. Jedna od najjednostavnijih impulsnih modulacija jeste impulsna širinska modulacija ili PWM (*pulse-width modulation)*.




Slika 1. Vrste impulsnih modulacija.

Impulsna vremenska modulacij (PTM) se deli na impulsnu pozicionu modulacija (PPM) i impulsnu širinsku modulaciju (PWM) [1]. Obe imaju svoje prednosti i mane, ali su njihove karakteristike u osnovi slične. Na slici 1 prikazan je oblik impulsno širinsko i impulsno položajno modulisanog signala ili PPM signala, ukoliko je modulišući signal sinusoidalnog oblika. Na slici 2 prikazan je spektar PTM modulisanog signala [2].




Slika 2. Spektar PTM modulacije.

## II. ZAHTEVANE KARAKTERISTIKE PREDAJNIKA

Ulazni audio signal treba da bude u opsegu 20 Hz-20 kHz, a frekvencija ponavljanja impulsa treba da bude znatno veća. Prototip treba da ima mogućnost podešavanja ove frekvencije, kao i podešavanje pojačanja ulaznog signala kako bi na optimalan način opterećivao modulator. Potrebno je predvideti da se srednja (ili vršna, ako je pogodnije) snaga lasera može podesiti na željenu vrednost.

Potrebno je obezbediti napajanje modulatora iz jednostavnog ispravljača koji daje 12 V. Ukoliko to nije moguće, specificirati izvor napajanja koji je potreban za ispravno funkcionisanje modulatora. Preporuka je da najveći deo elektronike koristi napajanje od 5 V koje bi se dobio stabilizacijom napona iz ispravljača.

Projektovanje i realizacija prototipa sistema izvedeni su u više faza, pri čemu su se neke od njih delimično preklapale:

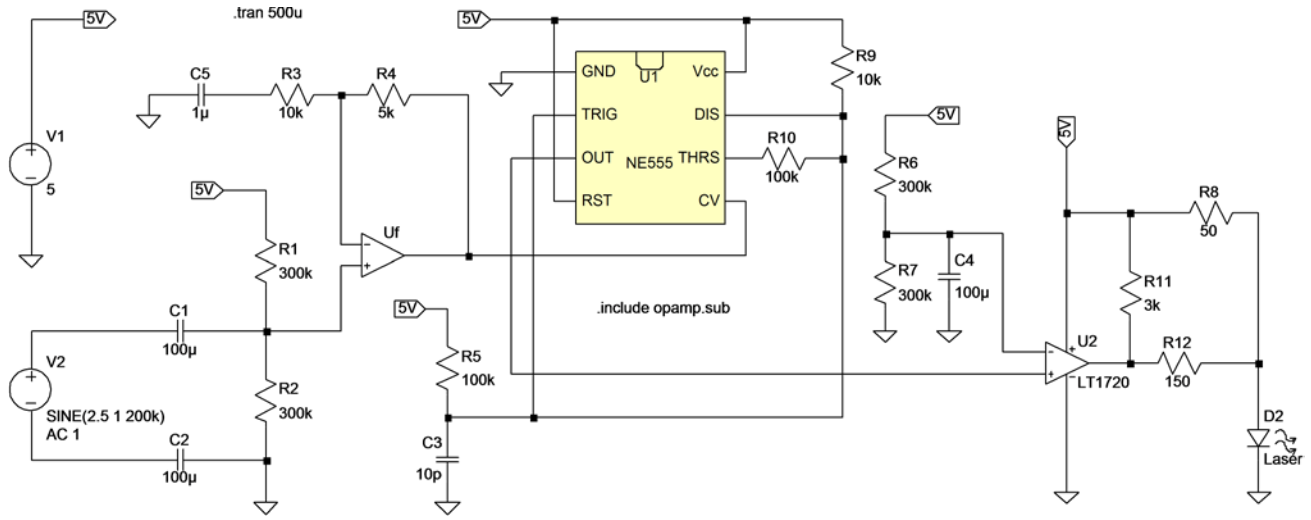
- Prikupljanje literature
- Izrada koncepta sistema
- Simulacije u SPICE-u i podešavanje parametara
- Izrada prototipa
- Testiranje prototipa
- Merenje karakteristika

## III. PREDLOG KONCEPTA PREDAJNIKA

Najjednostavniji koncept predajnika sastoji se iz 3 dela:

- ulazni stepen
- PWM modulator
- izlazni stepen, kako je prikazano na slici 3.

Anđela Jovanović i Milan Ilić su studenti treće godine osnovnih akademskih studija na Elektronskom fakultetu Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000Niš, Srbija, E-mail: andjela95.aj@gmail.com, ilicmilannis@gmail.com.



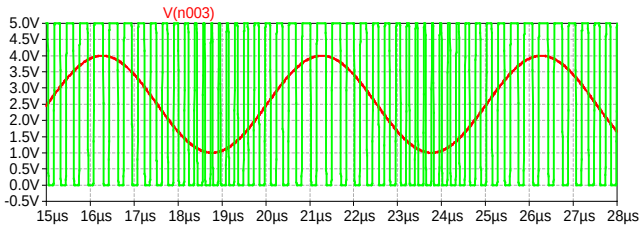

Slika 3. Koncept predajnika.

Ulazni stepen služi za pojačanje signala, pri čemu se koriste dva kondenzatora od C = 100µF i otpornika od R = 300kΩ (formula 1) koji služe za eliminisanje jednosmerne komponente ulaznog signala i prilagođenje jednosmernog nivoa na polovinu napona napajanja. Vrednost kondenzatora se dobija iz jednostavne formule:

$$C = \frac{2}{\pi R f_0} \qquad (1)$$

gde je: $f_0 = 20Hz$ .

Kondenzator od 1µF se vezuje za potenciometar predstavljen otpornicima $R_3$ i $R_4$, koji kontroliše nivo pojačanja, i služi za eliminisanje jednosmerne komponente prilikom pojačanja. Drugi deo kola ima zadatak da moduliše dolazni signal. Sastoji se iz komponente LM 555 CN koja ima karakteristike koje su potrebne za realizaciju modulatora, uzimajući u obzir kataloške podatke komponente [5].

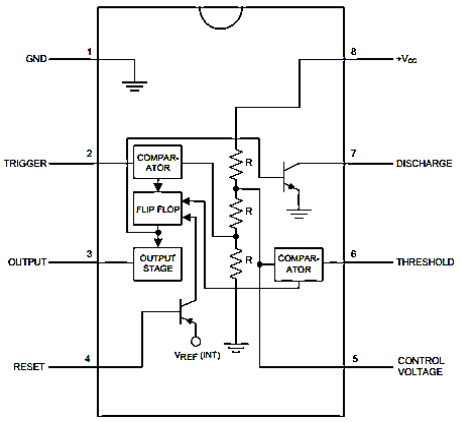



Slika 4. Talasni oblici na ulazu i izlazu kola u simulaciji [3].

Izlazni stepen se sastoji iz komparatora koji kontroliše nivo izlaznog signala gde se na izlazu dobija signal koji sevodi na laser i šalje kroz slobodan prostor do prijemnika koji predstavlja foto dioda. Izlazni stepen funkcioniše tako što otpornik od 50Ω obezbeđuje vršnu struju lasera od 50mA kada je komparator na visokom nivou. Kada je komparator na niskom nivou, on odvodi od lasera deo struje koji komparator može da provede, čime struja kroz laser pada ispod praga laserskog zračenja i optička snaga se mnogostruko smanjuje.

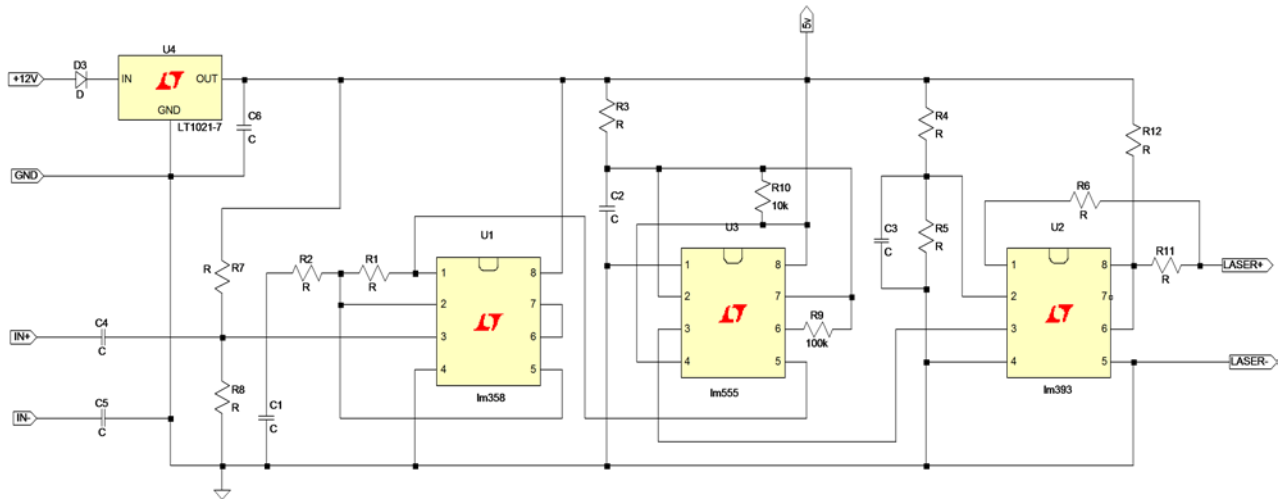
## IV. KARAKTERISTIKE IZABRANIH KOMPONENTI

Za izradu našeg prototipa smo predvideli integrisano kolo LM358 N, ova komponenta se sastoji od 2 visokofrekventna operaciona pojačavača, napravljena da funkcionišu od 1 ili 2 izvora napajanja u širokom opsegu vrednosti, ona takođe sadrži transduktor pojačavača jednosmerne struje i sve komponente koje sadrže operacione pojačavače koje mogu lakše da se implementuju sa jednim izvorom napajaja. Ova komponenta može da se napaja direktno iz izvora od 5V, i može da omogući potreban interfejs, bez dodatnog napajanja od -5V. Pojačavač smo povezali uz pomoć potenciometra u grani povratne sprege, i pomoću njega možemo da kontrolišemo kolko želimo da pojačamo signal.




Slika 5. Integrisano kolo LM 555 CN.

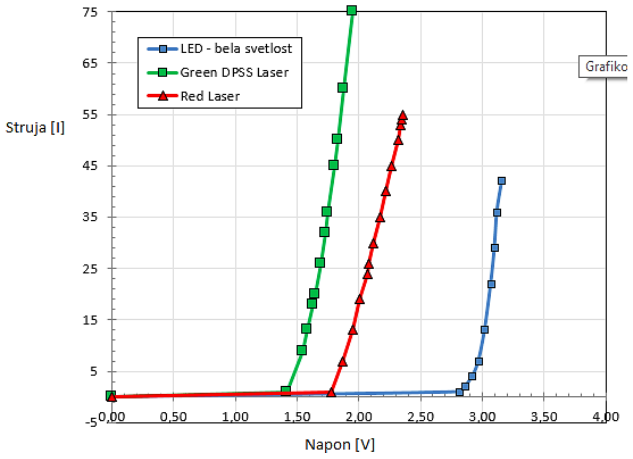
Takođe nam je potrebano integrisano kolo LM 555 CN čija je principijelna šema prikazana na slici 5. Ova komponenta je visoko stabilna komponenta koja generiše




Slika 6. Kompletna šema impulsno širinski kodovanog audio signala.

vremensko kašnjenje ili oscilacije. Dodatni terminali su tu za resetovanje i trigerovanje po potrebi u modulu vremenskog kašnjenja. Vreme je precizno kontrolisano od strane otpornika i kondenzatora, u modulu oscilatora, a radna frekvencija i radni ciklus su kontrolisani od strane dva otpornika i jednog kondenzatora. Ovu komponentu koristimo za impulsnu širinsku modulaciju. Zavisno od dolaznog signala širina impulsa modulisanog signala će se razlikovati.

Dalje nam je potreban komparator da uporedimo nivo modulisanog signala pragom koji smo postavili na polovinu napona napajanja,čime dobijamo impulsni signal koji prenosimo putem svetlosti koju zrači laser. Kao komparator koristimo integrisano kolo M393N [6].




Slika 7. Strujno naponska karakteristika nekoliko izvora svetlosti.

Ova komponenta se sastoji od dva niskonaponska komparatora napravljena da funkcionišu sa jednim izvorom napajanja. Ono što je specifično za ovu komponentu jeste da su izlazi komparatora ostavljeni sa otvorenim kolektorom, odnosno zahtevaju spoljašnji pull-up otpornik na izlazu. Ovaj otpornik je na Slici 3. predstavljen kao $R_{11}$.

Kako su laseri suštinski nelinearne komponente, potrebno je da znamo strujno-naponsku karakteristiku kako bismo odredili vrednosti otpornika u izlaznom delu predajnika. Zbog toga smo testirali tri izvora svetlosti: zeleni laser, crveni laser i belu LE diodu, a njihove strujno-naponske karakteristike su prikazane na slici 7. Zeleni laser je imao struju praga od 180mA, što je previše za korišćene komponente, dok je svetlost LE diode bila nedovoljno usmerena. Zbog ovoga smo se odlučili da koristimo crveni laser, koji pri struji od 50 mA ima napon između anode i katode od oko 2.3V. Prema tome, za polarizaciju laser je potreban otpornik $R_6$ vrednosti:

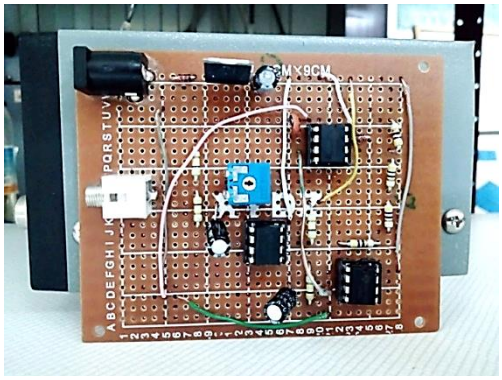
$$R_6 = \frac{(5-2.3)V}{50mA} = 54\Omega \qquad (2)$$

## V. PRAKTIČNA REALIZACIJA I TESTIRANJE PROTOTIPA

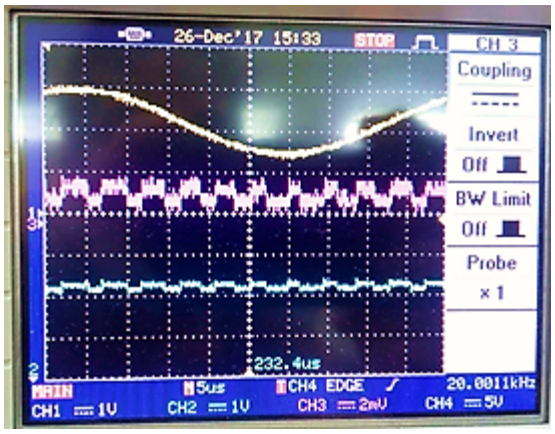
Za realizaciju našeg projekta iskoristili smo raster-ploču veličine 7×9cm, koja je prikazana na Slici 8. Predajnik ima dvostruku ulogu. U sebi mora da sadrži svetlosni izvor, koji će zračiti modulisani signal u slobodan prostor, kao i modulator koji treba da moduliše tu svetlost, tako da ona reprezentuje binarni ili analogni ulazni signal. Predajnici se mogu upoređivati po dva osnova. Jedan je posmatranje karakteristika optičkog dela, koji predstavlja izvor svetlosti, a drugi je način poređenje modulacije svetlosnog signala. Statičke karakteristike vezane za optički deo predajnika, prikazane su na Slici 7.

Audio signal dolazi preko kondenzatora na naponski razdelnik koji postavlja srednju vrednost signala na polovinu napona napajanja. Nakon ovoga sledi pojačavač gde se pomoću potenciometra kontroliše naizmenično pojačanje signala kako bi se on prilagodio narednom stepenu. Pojačani signal dalje vodimo u tajmer 555 koji radi u režimu impulsne širinske modulacije (PWM),

Glavna prednosti korišćenja PWM signala je to što signal ostaje u impulsnom obliku celim putem od procesora do primaoca. PWM signal se smatra digitalnim jer je u svakom trenutku vremena nivo signala ili na nuli, ili na maksimalnoj vrednosti koja predstavlja loiču jedinicu. Korišćenjem ovakvog digitalnog signala se efekat šuma minimalizuje, što je velika prednost u odnosu na analogni prenos. Sa izlaza modulatora, signal dovodimo na ulaz komparatora, i u zavisnosti od nivoa signala će na izlazu komparatora biti signal nule ili jedinice.



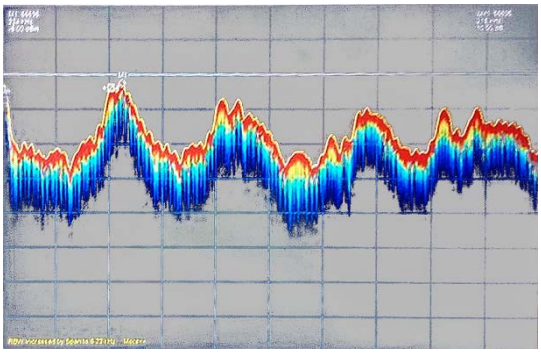
Slika 8. Izgled realizovanog protoripa na raster ploči.




Slika 9 Talasi oblici: ulazni signal (na vrhu, CH1), signal na fotodiodi (sredina, CH3) i signal na izlazu neopterećenog komparatora (na dnu, CH2).

Na Slici 9 je prikazan oblik signala na izlazu PWM modulatora (na dnu), kao i na fotodiodi koja nam služi za praćenje optičke snage (talasni oblik u sredini), kada se na ulaz kola dovodi sinusoida. Očigledno je da su impusli značajno uži kada je nivo ulaznog signala manji.

Spektar izlaznog signala, kada se na ulaz dovodi muzički signal iz mobilnog telefona, prikazan je na Slici 10. Ovaj spektar dobijen je realnim snimanjem uz pomoć analizatora spektra, koji je dostupan u laboratoriji. Poređenjem sa teorijskim spektrom koji je prikazan na Slici 2, može se uočiti da postoji gotovo potpuno slaganje, čime smo potvrdili da je signal modulisan onako kako je i bilo predviđeno.



Slika 10. Spektar PWM signala na izlazu realizovanog kola.

## VI. Zaključak

Projektovani modulator koji ulazni audio signal pretvara u impulsno-širinski modulisani signal je uspešno realizovan i testiran. Merenje karakteristika predajnika je potvrdilo da su rezultati simulacije ispravni i da prototip ima predviđene karakteristike. Dalje testiranje sa realizovanim kompatibilnim prijemnikom potvrdilo je da se emitovani optički signal može uspešno demodulisati i iz njega dobiti kvalitetna reprezentacija originalnog audio signala.

## Zahvalnica

Ovaj rad je napisan kao deo studentskog projekta na ispitu Optičke komunikacije, na smeru Telekomunikacije, Elektronskog fakulteta Univerziteta u Nišu. Kompletna realizacija projekta izvedena je pri laboratoriji za telekomunikacione sisteme, uz mentorstvo prof. dr Dejana Milića i pomoć članova laboratorije 304.

## Literatura

[1] Aleksandar Marinčić „Osnovi optoelektronskih komunikacija“, Gradjevinska knjiga, Beograd, 1986.
[2] Dejana Milić „Optičke telekomunikacije“, Niš 2017.
[3] LTSpice IV, softver za simulaciju sa dokumentacijom http://www.linear.com/designtools/software/
[4] Texas instruments, „LMx58-N Low-Power, Dual-Operational Amplifiers“, kataloški podaci za integrisano kolo LM358, dostupno putem interneta: http://www.ti.com/lit/ds/symlink/lm158-n.pdf
[5] Texas instruments, "LM555 Timer", kataloški podaci za integrisano kolo LM555, dostupno putem interneta: http://www.ti.com/lit/ds/symlink/lm555.pdf
[6] Texas instruments, "LMx93-N, LM2903-N Low-Power, Low-Offset Voltage, Dual Comparators", kataloški podaci za integrisano kolo LM293 dostupno putem interneta: http://www.ti.com/lit/ds/symlink/lm393-n.pdf

# Optički prijemnik impulsno-širinski modulisanih audio signala

Ilija Đorđević, Marija Gojković

*Sadržaj* - U ovom radu prikazano je projektovanje i realizacija optičkog prijemnika na bazi PIN fotodiode koji služi za prijem i demodulaciju impulsno-širinski modulisanih audio signala. Koncept prijemnika je najpre potvrđen korektnim simulacionim rezultatima, a realizovani prototip je testiran uz pomoć muzičkog signala i odgovarajućeg predajnika. Prototip je pokazao očekivane rezultate i dobru pouzdanost.

## I. UVOD

Impulsna širinska modulacija (PWM) je tehnika kojom se korisna analogna informacija utiskuje u impulsni signal [1]. Napon, odnosno struja, se dovode do prijemnika kao serija impulsa, a informacija o amplitudi analognog signala se predstavlja širinom, odnosno trajanjem impulsa PWM signala. PWM ima vrlo široku primenu, koristi se pri merenjima, u komunikaciji, kontroli napajanja, kontroli motora, kao i u raznim vrstama pretvarača [2], [3]. U savremenim optičkim telekomunikacionim sistemima PWM se razmatra kao jedna od mogućnosti za optičke sisteme koji funkcionišu u slobodnom prostoru, kao i za podmorske komunikacione sisteme. Postoje predlozi da se M-arna PWM modulacija koristi za optičke pristupne bežične mreže na bazi CDMA multipleksa [2]. U ovom radu smo razmatrali najjednostavniju varijantu binarne PWM modulacije za analogne signale u audio opsegu.

Prijemnik, u našem slučaju, treba da sadrži optički podsklop, kao i demodulator i izlazni pojačavač koji funkcionišu u električnom domenu. S obzirom da je detaljnija analiza spektra PWM signala dosta složena, naš demodulator se zasniva na činjenici da u osnovnom opsegu postoje komponente signala koje su jednake skaliranim modulišućim komponentama, tako da se demodulacija najjednostavnije može izvršiti filterom propusnikom niskih frekvencija. Izlazni pojačavač treba imati funkcionalnost razdvajanja demodulatora od uređaja koji su priključeni na izlaz, kao i da omogući promenjivo pojačanje.

## II. ZAHTEVI I KONCEPT PRIJEMNIKA

Prilikom projektovanja optičkog prijemnika, treba obratiti pažnju na to kakvu konfiguraciju treba odabrati. Najpovoljnije je projektovanje prijemnika sa velikom impedansom, jer on u odnosu na prijemnike sa malom ili transimpedansom ima veoma dobru osetljivost, jer čak i mala fotostruja na velikoj ulaznoj otpornosti stvara veliki napon. Potrebno je da se na samom ulazu u prijemnik postavi i komparator koji će vršiti odlučivanje u zavisnosti od praga odlučivanja koji se može menjati. Na izlazu iz komparatora imamo impulsno širinski modulisan signal, koji je potrebno demodulisati i vratiti u prvobitni analogni oblik. Impulsno širinska modulacija na predajniku se koristi jer bi analogna amplitudska modulacija dovela do uticaja koji bi se direktno reflektovali na signal koji dolazi na ulaz prijemnika i stvorili neželjene smetnje.

Demodulisani signal je potrebno pojačati, pa se posle demodulatora postavlja pojačavač sa mogućnošću promene pojačanja izlaznog stepena. Signal na izlazu pojačavača treba biti sposoban da pobudi standardne audio uređaje, a da ne bi napravio neko oštećenje, na samom izlazu ovog prototipa postavlja se i atenuator koji ima ulogu da po potrebi oslabi napon.

Poželjno je da se komponente u prototipu napajaju naponom od 5 V, što se može dobiti stabilizacijom napona iz ispravljača koji generiše napon od 12V. Projektovanje i realizacija prototipa sistema izvedeni su u više faza, pri čemu se neke od njih delimično preklapaju:

- Prikupljanje literature
- Izrada koncepta sistema
- Simulacije u SPICE-u i podešavanje parametara
- Izrada prototipa
- Testiranje prototipa
- Merenje karakteristika

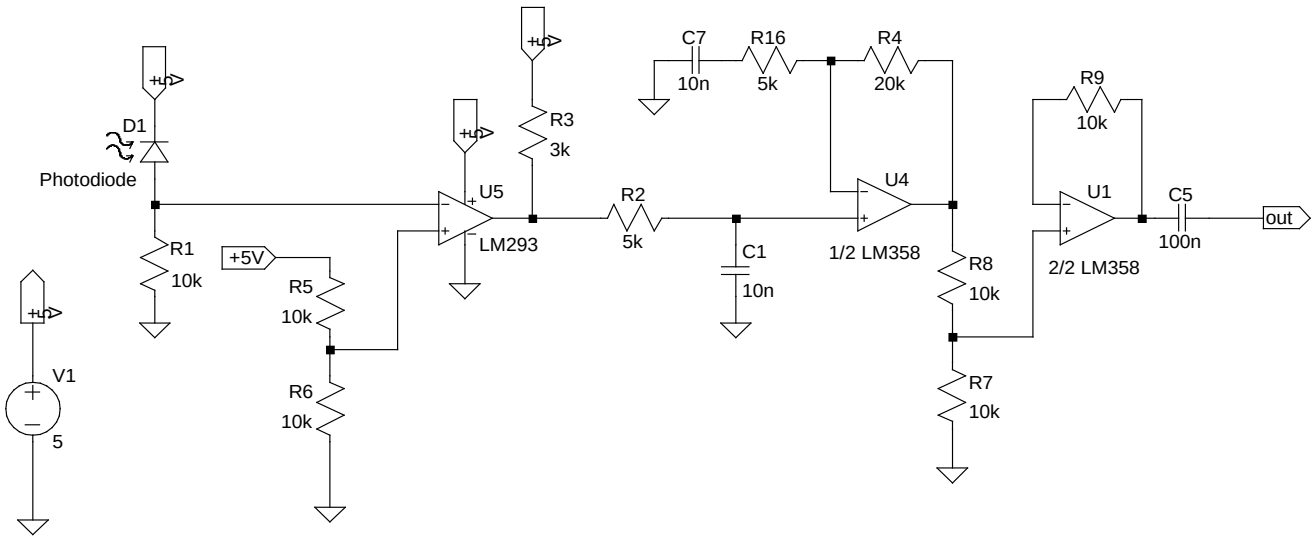
## III. IZABRANE KOMPONENTE I NJIHOVE KARAKTERISTIKE

Na ulazu se nalazi fotodioda koja pod uticajem svetlosti stvara malu fotostruju, koja narednoj impedansi od 10 kΩ stvara srazmerno veliki napon pa se na taj način mogu detektovati veoma niski nivoi svetlosti.

Kao komparator koristi se integrisano kolo LM 293 (slika 1) koja se sastoji od dva nezavisna komparatora napona koja se napajaju preko jednog pina. Za komparator se povezuje potenciometar pomoću koga možemo da odaberemo prag odlučivanja. Na slici 2 je prikazano integrisano kolo LM 293 [4]-[7].

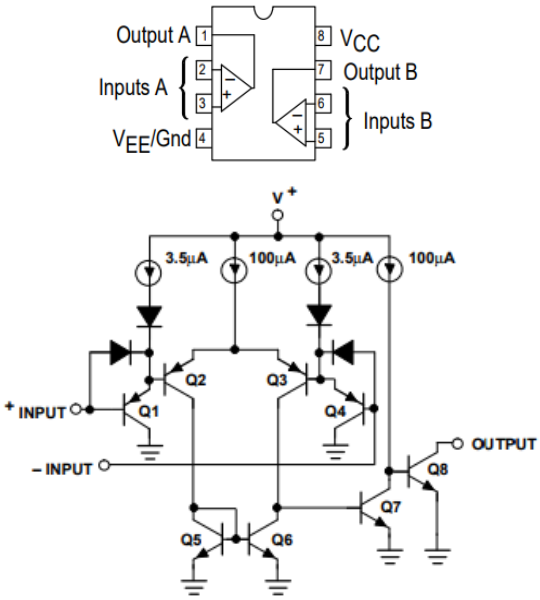
Kao pojačavač, zbog najboljeg odnosa cene i kavaliteta, koristi se integrisano kolo LM358N (slika3) koje se sastoji od dva visokofrekventna operaciona pojačavača, koja se mogu napajati sa jednim ili dva izvora napajanja. Za izradu ovog prototipa se jedan deo

Ilija Đorđević i Marija Gojković, su studenti treće godine Osnovnih akademskih studija na Elektronskom fakultetu Univerziteta u Nišu, A. Medvedeva 14, 18000 Niš, Srbija (e-mailovi: ilijaelfak@outlook.com, m.gojkovic217@gmail.com)

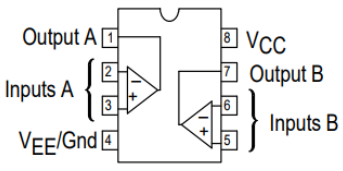



Slika 1. Simulaciona šema prijemnika nacrtana u LT-SPICE-u.

integrisanog kola može koristiti kao operacioni pojačavač, drugi kao atenuator koji treba, po potrebi, smanjiti napon koji se dovodi na audio uređaj povezan na izlaz prototipa.




Slika 2. Raspored pinova i funkcionalni blok dijagram integrisanog kola LM 293.




Slika 3. Raspored pinova integrisanog kola LM 358.

## IV. PRAKTIČNA REALIZACIJA I TESTIRANJE PROTOTIPA

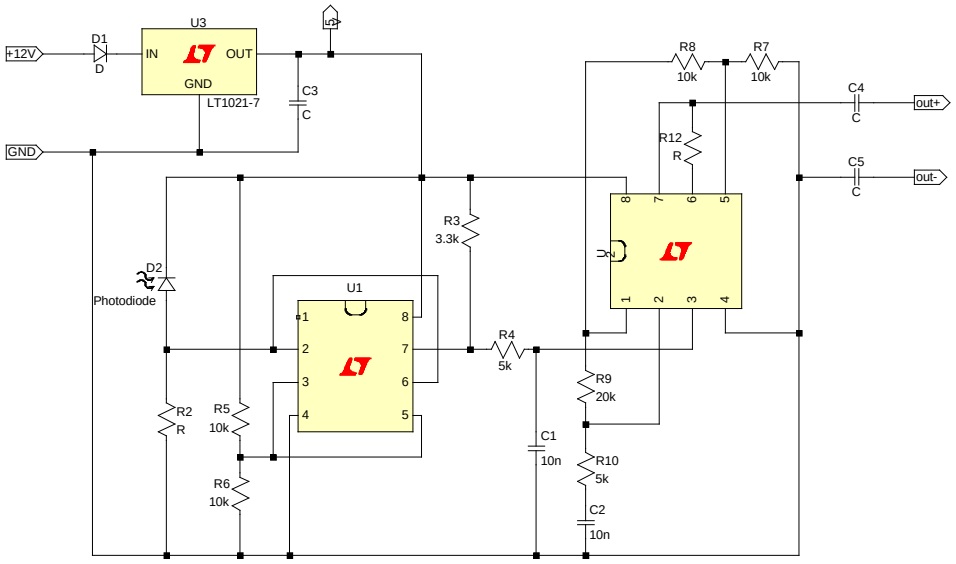
Šema sa slike 4. se može realizovati na raster-ploči dimenzija 7x9 cm. Preko ispravljača dovodimo napon od 12V, koji se stabilizuje pomoću stabilizatora napona. Na izlazu iz stabilizatora imamo napon od 5V koji napaja ostatak kola. Na ulazu u kolo postavlja se fotodioda koja treba da detektuje impulsno širinski modulisan audio signal i u zavisnosti od toga u komparatoru koji je povezan sa diodom se vrši odlučivanje. Pomoću potenciometra se bira prag odlučivanja u komparatoru. Dioda se povezuje za komparator određene parazitne kapacitivnosti, zbog koje je neophodno diodu vezati za masu preko otpornika. Ako uzmemo u obzir da kapacitivnost otprilike iznosi 20pF, na osnovu radne frekvencije koja iznosi

$$f = \frac{1}{2\pi RC} \tag{1}$$

određuje se da je potrebna otpornost 30kΩ, ali za realizaciju ovog prototipa korišćen je otpornik od 10kΩ, jer sa manjom vrednošću te otpornosti dobijamo manji propusni opseg, što ustvari znači da će dioda biti pobuđena i na manjoj osvetljenosti. Nakon što je detektovan signal, komparator će signal dalje proslediti do filtra propusnika opsega od 0 do 20kHz. Filter ovog prototipa je realizovan vezom otpornika i kondenzatora. S obzirom da je granična frekvencija poznata, mogu se odrediti vrednosti otpornika i kondenzatora.

Prenosna funkcija napona na izlazu filtra je

$$H_{in}(j\omega) = \frac{1}{1 + j\omega R_L C_p}. \tag{2}$$




Slika 4. Blok šema prototipa.

Očigledno je da je propusni opseg ovog kola ograničen zbog postojanja ulazne kapacitivnosti i iznosi

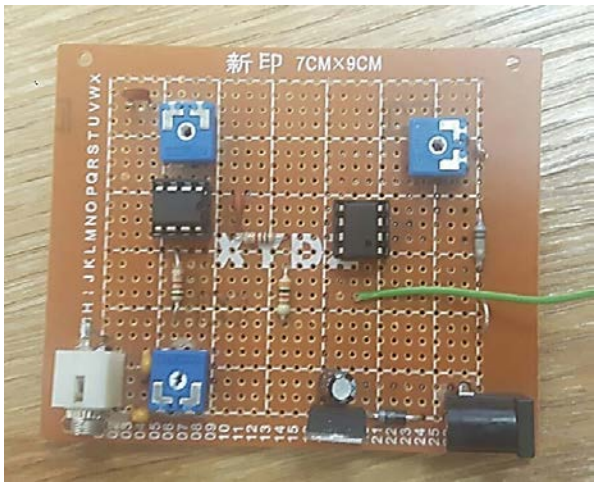
$$\omega_g = \frac{1}{1 + R_L C_p}. \qquad (3)$$

Pošto se radi o prijemniku sa velikom impedansom, vrednost otpora RL može biti vrlo velika, tako da upotrebljivi propusni opseg u suštini može biti nedovoljan za mnoge primene jer se sa porastom vrednosti otpornosti, smanjuje propusni opseg filtra. Zbog toga, pri njegovoj realizaciji, na to treba posebno obratiti pažnju.

Dobijeni signal treba po potrebi pojačati ili oslabiti, što se može postići rednom vezom pojačavača i atenuatora. U ovom projektu korišćeno je LM358 integrisano kolo jer se sastoji iz dva nezavisna operaciona pojačavača koji se pojedinačno snabdevaju naponom ili distribuitivno. Opseg napona se može kretati od 3V do 32V što je za izradu ovog projekta vrlo zadovoljavajuće.

Jedan pojačavač ovog integrisanog kola ima ulogu da pojača signal nakon demodulacije. Kao pojačavač pogodno je koristiti neinvertujući operacioni pojačavač, jer na taj način regulišemo negativnu povratnu spregu. Kako prenosna funkcija pojačavača zavisi od vrednosti dva otpornika i ne može biti manja od jedinice, umesto dva otpornika korišćen je potenciometar kojim se može podešavati odnos dva zamenjena otpornika, odnosno, može se menjati pojačanje izlaznog signala.

Treba, takođe, uzeti u obzir slučaj da nam se na izlazu iz demodulatra pojavi vrlo jak signal koji potencijalno može oštetiti audio uređaj kome je signal namenjen. Zato se drugi pojačavač u integrisanom kolu koristi kao atenuator radi što optimalnije izrade prototipa. Atenuator posle pojačavača ima ulogu da smanji napon na izlazu iz prototipa. Za atenuator se, takođe, vezuje potenciometar kojim se može podešavati nivo slabljenja izlaznog signala.



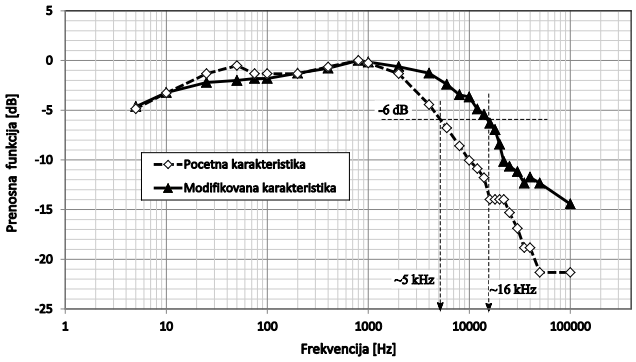

Slika 5. Realizovani prototip na raster ploči.

## V. MERENJE KARAKTERISTIKA PROTOTIPA

Posle završetka izrade prototipa na raster-ploči (slika 5), izvršena je provera ispravnosti kola pomoću osciloskopa. Upoređivan je signala na ulazu i izlazu iz prijemnika. Za određene frekvencije sa proizvoljnim korakom očitavana je vrednost napona koji se u tom slučaju nalazio na izlazu. Dobijeni rezultati su normalizovani odgovarajućim vrednostima dobijenim za najviši merni napon, takođe, vrednosti napona su prikazane u dB Normalizovane prenosne funkcije prikazane su na slici 6.

Sa grafika se može uočiti da se za normalizovanu vrednost napona od npr. -6 dB maksimalne izlazne frekvencije veoma razlikuju. Ovde se nailazi na problem jer je prijemnik projektovan da propusti frekvencije do 20kHz. Međutim, na osciloskopu je očitana skoro duplo manja vrednost. Uzrok ovog problema leži u projektovanju filtra. Za realizaciju filtra je proračunom utvrđeno da je

neophodan otpornik od 5kΩ. Međutim, otpornik koji se nalazi na izlazu iz komparatora značajno utiče na promenu otpornosti filtra. Ovo je dovelo do neželjenog povećanja otpornosti koja je promenila propusni opseg filtra. S obzirom da je izlazna impedansa komparatora nepoznata, pri izradi ovog projekta korišćen je otpornik od 3 kΩ na komparatoru što je prouzrokovalo manji propusni opseg filtra (8kHz umesto 20kHz) Kako bi se rešio ovaj problem za otpornik u filtru paralelno je vezan još jedan otpornik, približno iste otpornosti kao otpornik na izlazu iz komparatora, koji je omogućio smanjenje celokupne otpornosti filtra i povećanje propusnog opsega. Plava kriva predstavlja signal dobijen korišćenjem filtra realizovanog pomoću kondenzatora od 10nF i otpornika od 5kΩ. Zelena kriva predstavlja signal koji je propušten kroz filtar kome je dodat otpornik. Razlika propuštenih frekvencija u predelu oko -6dB je očigledna.




Slika 6. Prenosna karakteristika.

## VI. ZAKLJUČAK

Zadatak rada je bio projektovanje i realizacija optičkog prijemnika za impulsno širinske modulisane signale. Izvršena je analiza zadatka i obavljeno je projektovanje prototipa čija je validnost proverena uz pomoć simulacije. Prototip je zatim praktično realizovan korišćenjem raster ploče i diskretnih komponenti. Njegovo funkcionisanje je potvrđeno izvršenim merenjima. Nakon merenja prenosne karakteristike utvrđeno je da propusni opseg nije bio zadovoljavajući što je ispravljeno potrebnim modifikacijama. Nakon modifikacije prijemnik se može efikasno upotrebiti za prijem optičkih impulsno širinski modulisanih signala.

## ZAHVALNICA

Ovaj rad napisan je kao deo studentskog projekta na ispitu Optičke komunikacije, na smeru Telekomunikacije, Elektronskog fakulteta Univerziteta u Nišu. Kompletna realizacija projekta izvedena je pri laboratoriji za komunikacione sisteme. Izrada ovog projekta ne bi bila moguća bez pomoći prof. dr. Dejana Milića i članova laboratorije 304.

## LITERATURA

[1] Dejana Milić „Optičke telekomunikacije“, Niš 2017.
[2] W. Wu, C. Hsu, "Performance analysis of optical CDMA communication systems with M-ary PWM signalling", Journal Transactions on Emerging Telecommunications Technologies, vol. 25 no. 6, June 2014, pp. 648-659, John Wiley & Sons, Inc. New York, NY, USA doi: 10.1002/ett.2793
[3] Aleksandar Marinčić „Osnovi optoelektronskih komunikacija“, Gradjevinska knjiga, Beograd, 1986.
[4] LTSpice IV, software za simulaciju sa dokumentacijom http://www.linear.com/designtools/software/
[5] Wikipedia, „Amplifier“, dostupno putem interneta: https://en.wikipedia.org/wiki/Amplifier
[6] Kataloški podaci za integrisano kolo LM358 dostupni putem interneta: http://www.alldatasheet.com/datasheet-pdf/pdf/3067/MOTOROLA/LM358.html
[7] Kataloški podaci za integrisano kolo LM293 dostupni putem interneta: http://pdf1.alldatasheet.com/datasheet-pdf/view/17872/PHILIPS/LM293.html

# Design, Simulation and Fabrication of a Microstrip Low-Pass Filter with Open-Stubs

Zornitsa Krasteva, Ivaylo Vladimirov

*Abstract* - This paper describes the design procedure of a microstrip low-pass filter with open-stubs and makes a comparison between the results of fabricated filter and the virtually designed one. The used software and hardware in this paper is: ADS (Advanced Design System) for the design, Board Master and ProMat S100 for fabrication and Hewlett Packard 8753C Network Analyzer and Hewlett Packard 85047A S Parameter Test Set for measurements.

## I. INTRODUCTION

Microstrip is a type of electrical transmission line which can be fabricated using printed circuit board technology, and is used to convey microwave-frequency signals. It consists of a conducting strip separated from a ground plane by a dielectric layer known as the substrate. In such a technology reciprocal and nonreciprocal passive components are obtained by varying the configuration of the printed metallic strips [1].

A low-pass filter is a filter that passes signals with a frequency lower than a certain cutoff frequency and attenuates signals with frequencies higher than the cutoff frequency. This filter is commonly used in microwave systems such as radars, measurement and test systems, satellite communication [2].

## II. SPECIFICATIONS

The goal is to design a microstrip Chebyshev low-pass filter with open-stubs. This filter has to fulfill the following requirements: a maximum attenuation in pass band ($\alpha_{\max}$) of 0.5 dB; a minimum attenuation in stop band ($\alpha_{\min}$) of 30 dB; a cut-off pass-band frequency ($f_p$) of 1GHz; a stop-band frequency ($f_s$) of 1.65GHz.

For lowest cost, microstrip devices may be built on an ordinary FR-4 (standard PCB) substrate. However it is often found that the dielectric losses in FR-4 are too high at microwave frequencies, and that the dielectric constant not sufficiently tightly controlled. For these reasons, an alumina substrate is commonly used [1].

The FR-4 substrate has a relative dielectric constant of $\varepsilon_r = 4.2$, thickness of $h = 1.5$ mm, loss tangent of $\tan\delta = 0.02$ and a metallization thickness of $t = 35$ μm [3].

Zornitsa Krasteva and Ivaylo Vladimirov are with the Faculty of Telecommunications, Technical University of Sofia, St. Kliment Ohridski 8, 1756 Sofia, Bulgaria, E-mails: zornica95@abv.bg, i_vladimirov@mail.bg

## III. DESIGN PROCEDURE

The design procedure given in this paper consists of seven design steps. The basic idea is to calculate and generate a low-pass prototype, then remodels it into a model entirely built up by opened-circuited microstrip stubs.

### *A. Calculation of Lumped Element Low-pass Prototype*

Microstrip filter is design by transformation of network of prototype low-pass (PLP) filter into filter network with stubs. The first step is to choose the most suitable electrical length $\theta_p$. In order to make a substitution between the lumped components and the microstrip stubs later on the electrical length must satisfied relation $0 < \theta_p < 90^\circ$. Therefore, here is chosen to be $\theta_p = 45^\circ$.

After that, the scaling parameter $a$ and the repetition frequency of attenuation characteristic can be calculated:

$$a = 1/\tan(\theta_p) = 1, \quad (1)$$

$$F_o = \pi \cdot f_p / \theta_p = 4GHz. \quad (2)$$

With knowing the frequency of repeatance and the stop-band frequency, the normalized cut-off stop-band frequency and the selective factor can be found as follows:

$$\Omega_s = \tan(\pi \cdot f_s / F_o) = 3.5457, \quad (3)$$

$$\gamma = \Omega_s / \Omega_p = 3.5457, \quad (4)$$

where $\Omega_p = 1$ for PLP filter.

The next stage of the step consists of finding the values of the parameter ε and the order of the filter:

$$\varepsilon = \sqrt{10^{0.1\cdot\alpha_{\max}} - 1} = 0.3493, \quad (5)$$

$$n \geq \frac{\operatorname{arccosh}\left(\frac{\sqrt{10^{0.1\cdot\alpha_{\min}} - 1}}{\varepsilon}\right)}{\operatorname{arccosh}(\gamma)} \geq 2.6818. \quad (6)$$

The parameter *n* (the filter order) should be an integer number equal or greater than the result of Eq. (6). Therefore the order of filter is 3.

The Chebyshev response is a mathematical strategy for achieving a faster roll-off by allowing ripple in the

frequency response. Analog and digital filters that use this approach are called Chebyshev filters. For instance, analog Chebyshev filters are used for analog-to-digital and digital-to-analog conversion [4].

In order to make our device a Chebyshev filter the values of the lumped elements $g_i$ in the prototype for case $R_{pn}=1$ must be calculated by Eqs. (7)-(11). The results from Eqs. (8)-(11) used to design filter are given in Table I.

$$u=\sinh\left(\frac{1}{n}\cdot\operatorname{arcsinh}\left(\frac{1}{\varepsilon}\right)\right)=2.6818\,, \tag{7}$$

$$\alpha_i=2\cdot\sin\left(\frac{i\pi}{2n}\right)\,,\ i=1,2,\cdots,2n-1\,, \tag{8}$$

$$f_i=u^2+\frac{1}{4}\cdot\alpha_{2i}^2\,,\ i=1,2,\cdots,n-1\,, \tag{9}$$

$$g_1=\frac{\alpha_1}{u}\,, \tag{10}$$

$$g_i=\frac{1}{g_{i-1}}.\frac{\alpha_{2i-3}\cdot\alpha_{2i-1}}{f_{i-1}},\ i=2,3,...,n\,. \tag{11}$$

TABLE I
PROTOTYPE DESIGN PARAMETERS

| $i$ | $\alpha_i$ | $f_i$ | $g_i$ |
|---|---|---|---|
| 1 | 1 | 1.1425 | 1.5962 |
| 2 | 1.732 | 1.1425 | 1.0967 |
| 3 | 2 | - | 1.5962 |
| 4 | 1.732 | - | - |
| 5 | 1 | - | - |

*B. The Structure of the Prototype Low-pass (PLP) Filter*

The configuration of the prototype will be Type B [5] (The organization of the part where the reactive lumped elements are located has a repeating formation of a series inductor followed by a shunt capacitor). This device has a filter order of 3, which means that it has 2 series inductors and one shunt capacitor. The capacitor is located between the two inductors. The stricter of the PLP filter is shown of Fig. 1, where $R_{gn}$ is the normalized resistance of the generator and $R_{pn}$ is the normalized load resistance. The configuration of the prototype could be also Type A [5]. In that case, more steps during application of Kuroda's identities are needed.

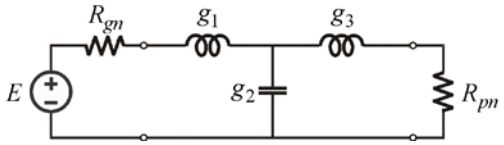



Fig. 1. The structure of PLP filter.

*C. Replacement of Series Inductors and Shunt Capacitors*

A stub or resonant stub is a length of transmission line or waveguide that is connected at one end only. The free end of the stub is either left open-circuit or (always in the case of waveguides) short-circuited. Neglecting transmission line losses, the input impedance of the stub is purely reactive; either capacitive or inductive, depending on the electrical length of the stub, and on whether it is open or short circuit [6].

At high frequencies transmission line stubs in parallel can be used instead of lumped elements to deliver compatible performance. Very short line stubs terminated in low impedances behave like inductors, while very short segments terminated with high impedances behave like capacitors. Therefore a short-circuited one behaves like an inductor (Fig. 2) and an open-circuited stub behaves like a capacitor (Fig. 3) [7]. Fig. 4 depicts the substitution of the lumped elements with open-circuited or short-circuited stubs.

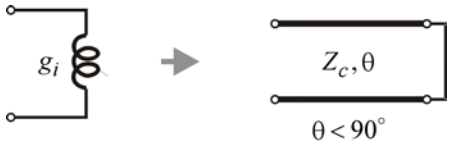



Fig. 2. Transformation of an inductor into a short-circuited stub.

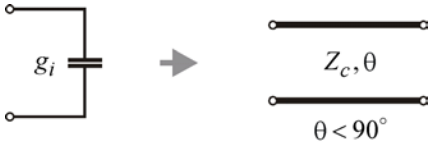



Fig. 3. Transformation of a capacitor into an open-circuited stub.

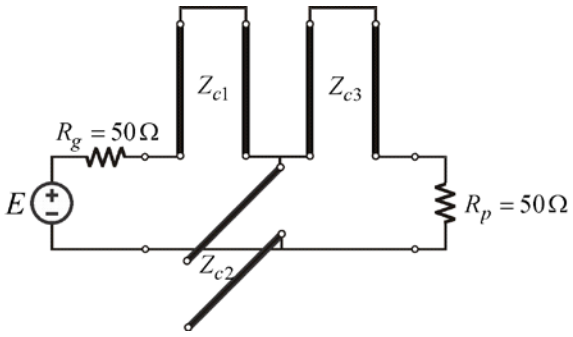



Fig. 4. The scheme of the PLP after the substitution.

*D. Calculation of the Characteristic Impedances*

In Figs. 2 and 3, $Z_c$ is the characteristic impedance of the transmission line, $a$ is a scalar parameter, $g_i$ is the normalized value of the reactive lumped element that we are transforming and $R_0$ is the normalization resistance. To calculate the characteristic impedances of our device we are going to use the Eq. (12) for the inductors or Eq. (13) for the capacitors:

$$Z_{ci} = a \cdot g_i \cdot R_0 , \quad (12)$$

$$Z_{ci} = \frac{R_0}{a \cdot g_i} , \quad (13)$$

and results of this Richard's transformation [5] are given in Table II.

TABLE II
CHARACTERISTIC IMPEDANCES OF THE STUBS

| $i$ | 1 | 2 | 3 |
|---|---|---|---|
| $Z_{ci}$ [Ω] | 79.8150 | 45.5913 | 79.8150 |

*E. Kuroda Identity*

Kuroda's identities essentially provide a list of equivalent two port networks. By equivalent, it is meant that they have precisely the same scattering/impedance /admittance/transmission matrices. We can replace one two-port network with its equivalent shown in a circuit (Fig. 5), and the behaviour and characteristics of the circuit will not change [8].

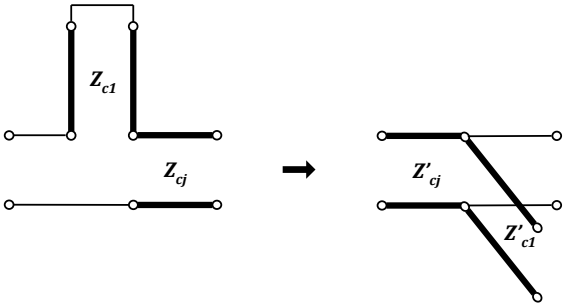



Fig. 5. The "Kuroda's identities" transformation that will be used.

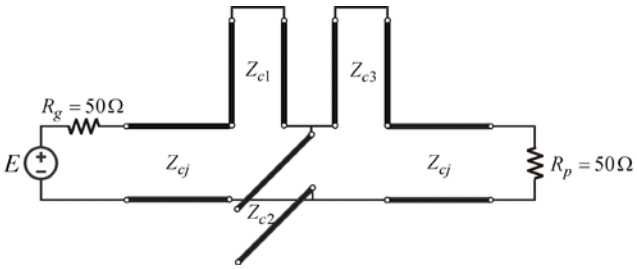



Fig. 6. Scheme of the filter after adding the unit transmission lines.

In order to use Kuroda's identities, a unit transmission line must be added before or after the open-circuited stub. In our case, for PLP filter shown in Fig. 4, two λ/8 transmission lines with a characteristic impedance of 50 Ω ($Z_{cj}$) will be added at the beginning (before the first inductor) and at the end (after the second inductor) of the filter network, i.e. on the source and load sides. By adding them at these specific points the characteristics of the filter will not change. Fig. 6 shows the scheme of the filter after the two transmission lines are added.

After using the Kuroda's identities the design of the filter (Fig. 7) will only consist of open-circuited stubs and 50Ω's transmission lines. These changes will make the fabrication easier and cheaper. By using Kuroda's identities some of the characteristic impedances change. Their new values are calculated by equations (14), (15) and (16), as follows

$$N = 1 + Z_{cj} / Z_{c1} = 1.6265 , \quad (14)$$

$$Z'_{c1} = Z'_{c3} = N \cdot Z_{cj} = 81.32\,\Omega , \quad (15)$$

$$Z'_{cj} = N \cdot Z_{c1} = 129.81\,\Omega . \quad (16)$$

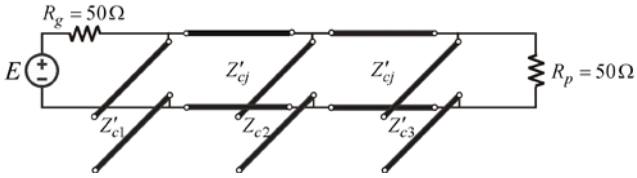



Fig. 7. Scheme of the filter after using Kuroda's identities.

*F. Calculation of Filter Physical Dimensions*

ADS (Advanced Design System) have a function called "LineCalc" which allow us to easily obtain the physical dimensions of the microstrip filter. For using "LineCalc" it is required that substrate parameters, characteristic impedances and the cut-off pass-band frequency are known. All these parameters have either been evaluated by us or given as specifications for the filter. Table III contains physical dimensions of microstrip lines calculated by "LineCalc".

TABLE III
PHYSICAL DIMENSIONS OF MICROSTRIP LINES

| | $Z$ [Ω] | $w$ [mm] | $l$ [mm] |
|---|---|---|---|
| $R_g$ , $R_p$ | 50 | 3.0233 | 10 |
| $Z'_{c1}$ , $Z'_{c3}$ | 81.32 | 1.1615 | 21.7957 |
| $Z'_{c2}$ | 45.59 | 3.5161 | 20.8331 |
| $Z'_{cj}$ | 129.81 | 0.2795 | 22.5976 |

*G. Schematic of Designed Filter*

Microstrip filter schematic given in Fig. 8 is used to create layout of the filter.

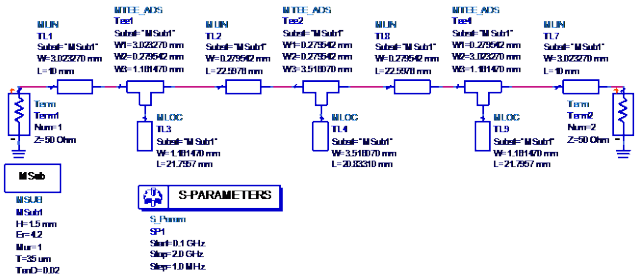



Fig. 8. Schematic of designed microstrip filter with stubs.

## IV. Simulation and Fabrication

ADS is a design software package which gives us an opportunity to both virtually test the filter before fabrication and make a filter layout in Gerber format used later by ProMat S100 (Fig. 8) in fabrication process.

The test set up, i.e. frequency response of fabricated component on Vector Network Analyzer (VNA) is depicted in Fig. 9.

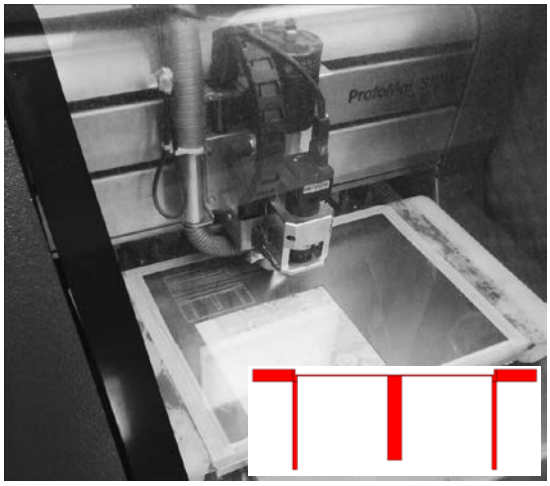


Fig. 8. The process of fabrication (ProMat S100 and filter layout).

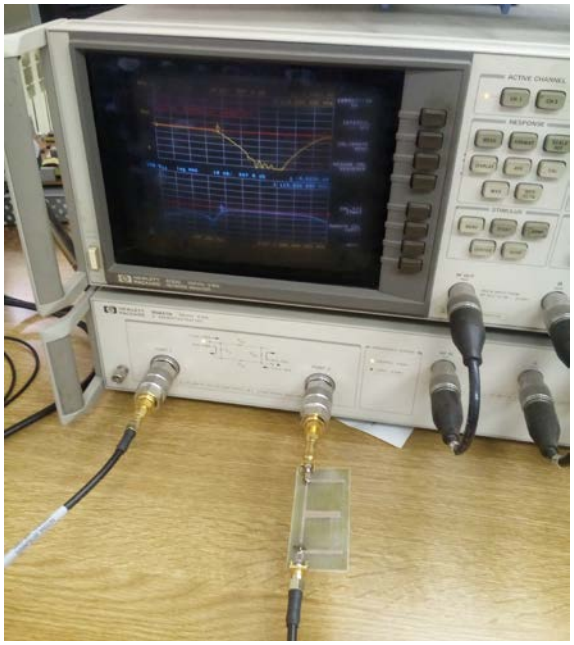


Fig. 9. The test set up (VNA with fabricated filter).

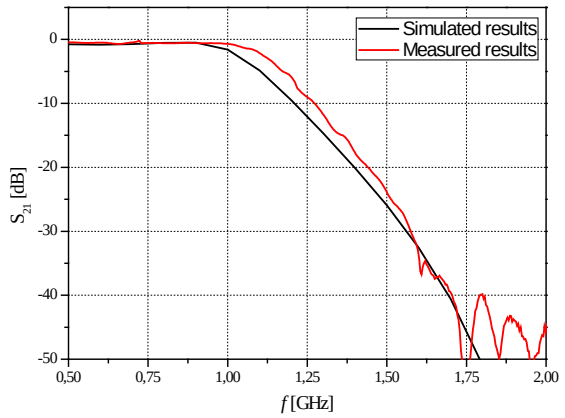



Fig. 10. $S_{21}$ responses.

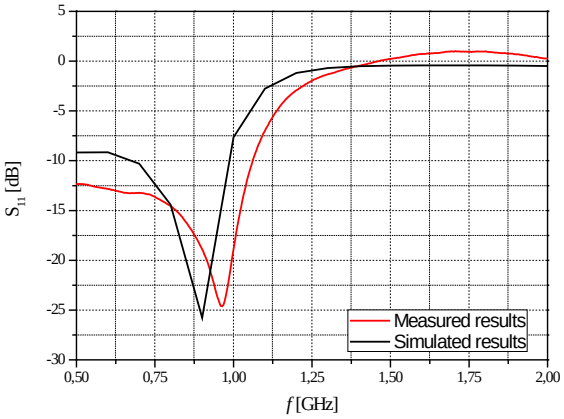



Fig. 11. $S_{11}$ responses.

## V. Comparison

Comparison of measured and simulated results for the transmission $S_{21}$ and reflection $S_{11}$ parameters of the microstrip structure is shown in Figs. 10 and 11. The filter responses were measured by a VNA at the frequencies from 300 KHz to 3 GHz. The measurement was performed using SMA connectors. Simulated results are obtained by Momentum simulation in ADS. The measured result shows a small difference in comparison with the simulated result.


## Acknowledgement

This scientific paper is wrote and prepared during our visit in the Department of Telecommunication, Faculty of Electronic Engineering, University of Niš on an "ERASMUS+" program. This document is a part of our team project.

We would like to thank our professors and mentors Biljana Stošić, Aleksandar Atanasković and Zlatica Marinković for their help and guidance.

## REFERENCES

[1] S. Halder, S. Sarkar, “A Low-Pass Filter Design using Microstrip”, India, Haldia Institute of Technology, 2012.
[2] Deep K. Chauhan, Falguni Raval, “Design of Microstrip Low Pass Filter”, *International Journal of Innovative and Emerging Research in Engineering*, vol. 3, no. 4, 2016.
[3] ProtoMat S100 Product Catalog, Part# 115970, http://www.lpkf.com/index.htm
[4] S. W. Smith, “Chapter 20: Chebyshev Filters”, *The Scientist and Engineer's Guide to Digital Signal Processing*, 1997.
[5] Miodrag V. Gmitrović, *Microwave and Wave Digital Filters*, Faculty of Electronic Engineering, Niš, Serbia, 2007.
[6] Ganesh Prasad Srivastava, Vijay Laxmi Gupta, *Microwave Devices and Circuit Design*, PHI Learning, 2006.
[7] “Transmission Line Stubs and the Design of Microwave Filters”, Signal Processing Group Inc., Technical memorandum, October 5, 2011.
[8] Jim Styles, ”Kuroda's Identities”, University of Kansas, 2007.

# Obezbeđivanje konstantnosti napajanja u postrojenjima gde je moguć paralelan rad transformatora

Uroš Ilić

*Sadržaj* - U ovom radu razmatrano je tehničko rešenje koje bi obezbedilo konstantnost napajanja postrojenja u kojima postoji mogućnost za paralelan rad transformatora. Takođe, data su opšta razmatranja o paralelnom radu transformatora. Primer na kojem je razmatrano tehničko rešenje je visokonaponsko postrojenje 400/110 kV, iako je moguća upotreba i u drugim postrojenjima. Simulacija mogućih stanja urađena je u CX-programmer-u.

## I. UVOD

Polazeći od činjenice da je za sveukupni razvoj čovečanstva električna energija bila jedan od najbitnijih resursa, shvatamo da je elektroenergetski sistem jedan od najvažnijih sistema. Pouzdanost, efikasnost i održivost su tri ključne osobine koje mora da ispuni svaki Elektroenergetiski sistem. Izuzetno mnogo resursa se ulaže kako bi se ove tri osobine dovele na nivo koji može kvalitetno da odgovori na sve zahtevnije potrebe stanovništva.

Učestvovanje u razmatranju mogućih rešenja koja bi dodatno poboljšala kvalitet rada EES-a, predstavlja izuzetnu čast i zahteva maksimalnu posvećenost. Ovaj rad daje određeni doprinos temi koja uvek zaokupljuje pažnju struke.

## II. VAŽNOST ODRŽANJA NAPAJANJA

Pri projektovanju postrojenja koja predstavljaju važna čvorišta u prenosu električne energije vodi se računa o tome da svako postrojenje ima više pravaca iz kojih se napaja. To automatski znači da postrojenja koja se nalaze na nižem naponskom nivou, tj. potrošači koji se nalaze na samom kraju prenosa električne energije imaju sigurniji pristup samoj električnoj energiji. U havarijskom režimu rada, koji može trajati neko kratko vreme (reda sekundi ili minuta), do onog koji može trajati do nekoliko dana, od suštinske je važnosti da krajnji potrošači ne osete ove poremećaje i da kvalitet električne energije ostane na visokom nivou.

Pojam kvaliteta isporuke električne energije sadrži više različitih aspekata koji se mogu grupisati u tri osnovne kategorije: kvalitet napona (Voltage Quality), komercijalni kvalitet (Commercial Quality), pouzdanost (neprekidnost) napajanja (Reliability (Continuity) of Supply). [2]

Prethodna podela definiše tri značajne osobine kvaliteta električne energije, dok je za ovaj rad najbitnija poslednja tj. pouzdanost (neprekidnost) napajanja.

Kvalitet napona je teško kvantifikovati, a praćenje performansi zahteva kontinuirana merenja u priključnim tačkama u dužem vremenskom intervalu, za šta je neophodna posebna merna oprema. Takođe, uzroci poremećaja napona nekada mogu biti sa strane isporučioca, a nekada sa strane potrošača ili distribuirane proizvodnje, što dodatno otežava merenje i regulaciju kvaliteta napona. Postoji više važećih međunarodnih tehničkih standarda u skladu s kojima se kontroliše kvalitet napona, kao što su EN 50160, IEC 61000-4-30, IEC 61000-4-15 i drugi.

Pouzdanost, odnosno neprekidnost napajanja predstavlja najvažniju osobinu kvaliteta isporuke električne energije i karakteriše se učestanošću i vremenom trajanja prekida napajanja potrošača. S obzirom na različite vrste i dužine trajanja prekida, moguće je primeniti nekoliko načina za merenje neprekidnosti napajanja. U skladu s tim, za ocenu pouzdanosti prenosne, odnosno distributivne mreže se koristi više pokazatelja, koji imaju različite definicije i načine primene u različitim zemljama, što često otežava njihovo poređenje. Najčešće korišćeni pokazatelji su:

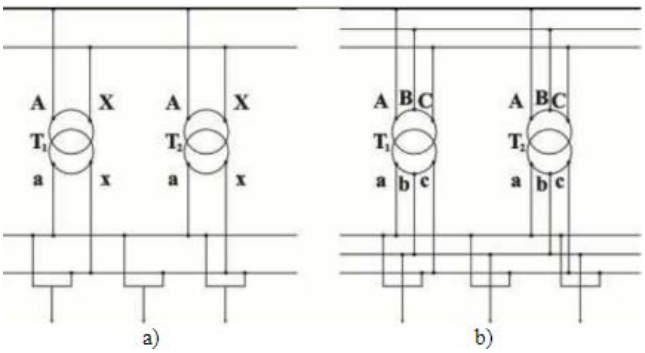
- SAIFI (System Average Interruption Frequency Index) predstavlja prosečnu učestanost prekida napajanja po potrošaču u određenom vremenskom periodu i dobija se kao količnik ukupnog broja prekida svih potrošača tokom tog vremenskog perioda i ukupnog broja potrošača,
- SAIDI (System Average Interruption Duration Index) predstavlja prosečno vreme trajanja prekida napajanja po potrošaču u određenom vremenskom periodu i dobija se kao količnik ukupnog trajanja prekida svih potrošača tokom tog vremenskog perioda i ukupnog broja potrošača,
- CAIDI (Customer Average Interruption Duration Index) predstavlja prosečno vreme trajanja jednog prekida prosečnog potrošača u određenom vremenskom periodu i dobija se kao količnik ukupnog trajanja prekida svih potrošača i ukupnog broja prekida svih potrošača tokom tog vremenskog perioda. CAIDI se računa kao količnik SAIDI i SAIFI indikatora.

Ovi pokazatelji predstavljaju meru pouzdanosti za određeni vremenski period, pri čemu se kao period posmatranja najčešće uzima jedna godina. [2]

Uroš Ilić, Katedra za Energetiku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija
E-mail: urosilic@elfak.rs

## III. PARALELNI RAD TRANSFORMATORA

Pod paralelnim radom dva ili više transformatora podrazumeva se režim u kome su njihovi primari priključeni na zajedničke sabirnice primarnog napona, a sekundari na zajedničku mrežu (Slika 1.).




Slika 1. Paralelan rad a) jednofaznih; b) trofaznih transformatora.

Posmatrano sa aspekta potrošnje materijala pri izradi transformatora opravdanije je korišćenje jednog većeg umesto nekoliko transformatora manjih snaga, ali se u nekim slučajevima, iz ekonomskih razloga, koristi paralelni rad. Na primer:

- Kada se za neku trafostanicu pojavi dodatno opterećenje usled planiranog ili neplaniranog rasta konzuma (izgradnja novih industrijskih pogona, porast široke potrošnje, itd.). U tom slučaju jeftinije je staviti još jedan transformator, nego kupovati novi, dva puta veće snage.
- Kada se želi veća sigurnost u snabdevanju potrošača energijom, jer pri ispadu jednog transformatora iz pogona ostali transformatori obezbeđuju delimično ili potpuno snabdevanje .
- Kada se želi ekonomičan pogon trafostanice. Naime, kako se opterećenje trafostanice menja u toku eksploatacije, to je nužno prilagoditi broj transformatora opterećenju. Sa povećanjem opterećenja treba povećati i broj transformatora u pogonu, a pri smanjenju opterećenja neke treba isključiti. Pri tome treba raditi sa onom kombinacijom transformatora koja daje maksimalan stepen iskorišćenja, tj. osnovni kriterijum je da gubici u aktivne i reaktivne snage budu što manji za dato opterećenje. [1]

### *A. Uslovi neophodni za pravilan paralelan rad transformatora*

Pravilan rad dva ili više transformatora u paraleli znači:

- da oni treba da dele opterećenje srazmerno svojim nominalnim snagama,
- da nemaju nepotrebnih gubitaka,
- da bude izbegnuto svako havarijsko stanje (pojava vrlo jakih struja).

Prethodni uslovi biće ispunjeni samo ako važi:

- jednakost nominalnih napona transformatora,
- ostvarena pravilna veza pojedinih napona (homologi naponi vezani na iste sabirnice),
- jednakost aktivnih i reaktivnih komponenti relativnih napona kratkih spojeva transformatora.

Pri paralelnom radu dva transformatora koji su izrađeni u istoj seriji, imaju istu snagu i isti su konstrukcionom pogledu, može se očekivati da ovi uslovi budu ispunjeni. Sa druge strane, veoma je čest slučaj kada dva ili više transformatora različitih snaga treba da rade paralelno. Iz tog razloga potrebno je analizirati slučajeve kada prethodno navedeni uslovi nisu u potpunosti ispunjeni i odrediti određena odstupanja koja je moguće tolerisati pri paralelnom radu. Ovi uslovi se neće razmatrati ovde, već će u primeru koji sledi biti prikazano tehničko rešenje gde pretpostavljamo da su svi uslovi ispunjeni.

## IV. PRIMER TEHNIČKOG REŠENJA ZA VISOKONAPONSKO POSTROJENJE 400/110 KV

Posmatra se visokonaponsko postrojenje 400/110 kV koje je projektovano tako da ima dva radna transformatora (TR1 i TR2) i jedan rezervni (TR3). Svaki od ovih transformatora je sposoban da preuzme kompletno opterecenje postrojenja. Optimum u pogledu rada sa kompletnim opterećenjem jednog transformatora je 15 minuta, zbog ubrzanog starenja izolacije nakon tog perioda. Svaki od dva radna transformatora napaja po dve transformatorske stanice 110/35 kV.

Transformator TR1 napaja TR4 i TR5, dok transformator TR2 napaja TR6 i TR7.

Cilj je obezbediti konstantno napajanje TS na nižem naponskom nivou i automatizovati paralelni rad transformatora pomoću PLC (Programmable logic controller).

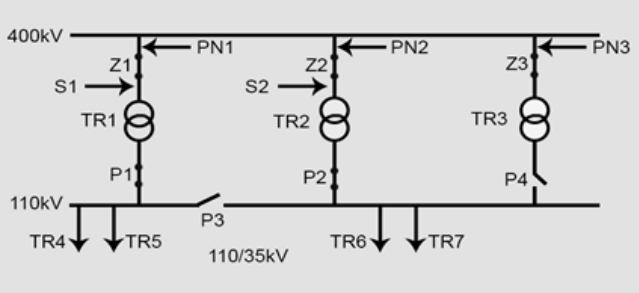
Transformatori su izabrani tako da u slučaju da jedan od njih ne radi, drugi može da preuzme potpuno opterećenje u trajanju od petnaest minuta. Razlog ovakve tehničke izvedbe je mogućnost prolaznog kvara. Očekuje se da ukoliko se kvar ne popravi unutar ovog perioda vremena da je kvar ozbiljnijeg karaktera. Zbog toga se nakon tog vremena uključuje rezervni transformator (TR3) koji na sebe preuzima opterećenje transformatora koji je ostao bez napajanja.

Nestanak napajanja detektujemo pomoću senzora S_1 za transformator TR1 i senzora S_2 za transformator TR2.

Koristimo prekidače P1 i P2 da bi odvojili transformatore TR1, odnosno TR2 od mreže, u zavisnosti od toga koji je ostao bez napajanja. Prekidač P3 služi za preuzimanje potpunog opterećenja od transformatora koji je izgubio napajanje. Prekidač P4 služi za povezivanje rezervnog transformatora na mrežu. U cilju dodatne zaštite transformatora u postrojenju postoje senzori koji detektuju prenapone (PN1, PN2, PN3). Kada senzor detektuje prenapon, ukoliko on traje duže od jedne sekunde,

transformator na kome je detektovan prenapon se automatski isključuje iz pogona.

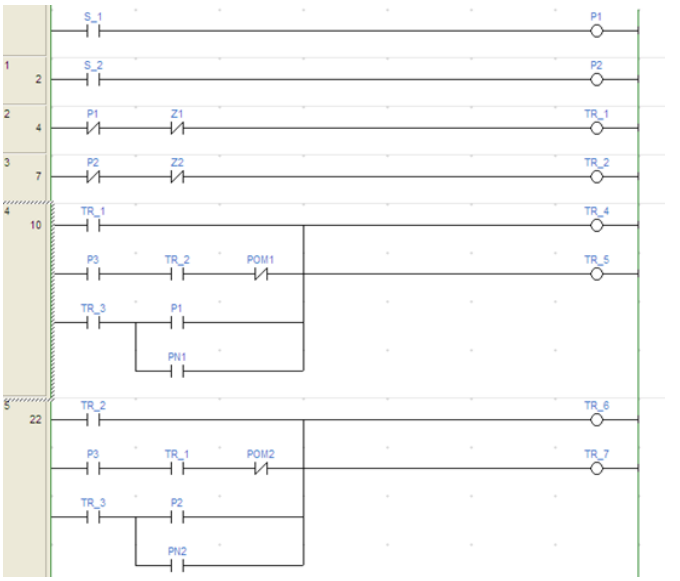
Ako je havarijski režim takav da oba radna transformatora gube napajanje celo postrojenje se takođe stavlja van pogona. Slika 2. prikazuje pogon koji se koristi za simulaciju.

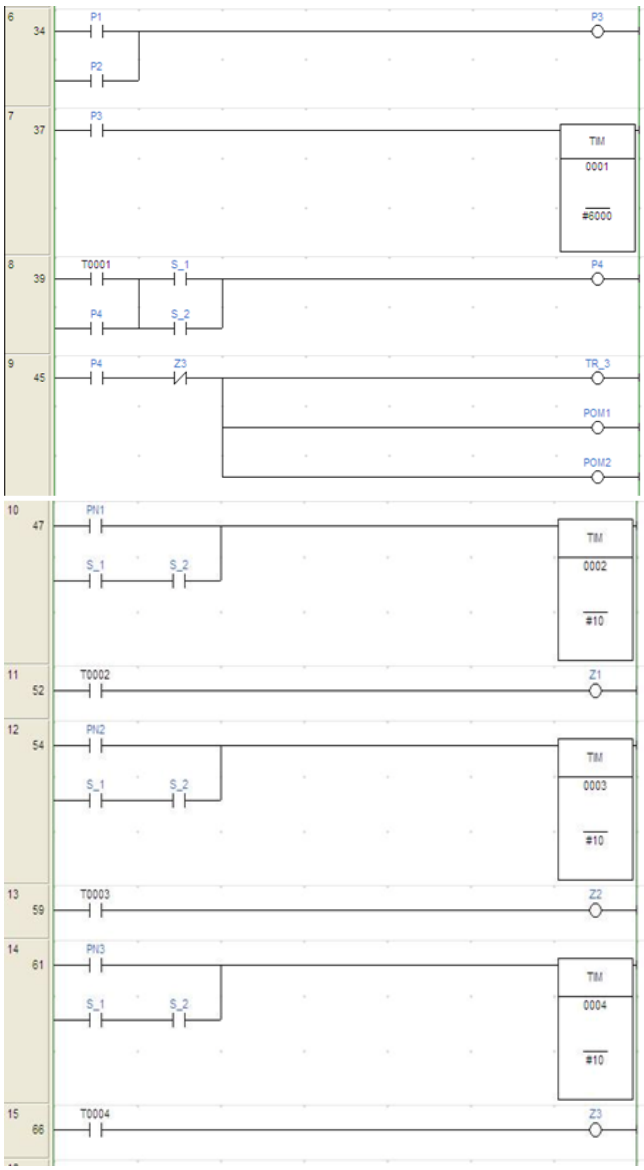



Slika 2. Visokonaponsko postrojenje 400/110 kV.

## *B. Ladder diagram predloženog tehničkog rešenja*

U ovom odeljku na početku je dat kompletan ladder diagram (Slika 3.), a zatim dato objašnjenje po strukturnim celinama.

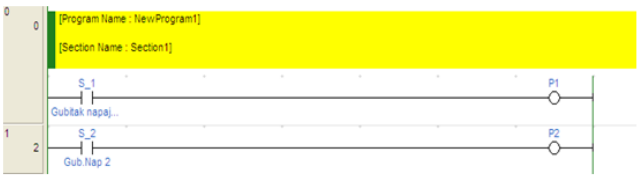




Slika 3. Ladder diagram.

## *C. Objašnjenje strukturalnih celina*

Setovanjem senzora S_1 ili S_2 uključuju se P1 ili P2, što za posledicu u realnom stanju ima otvaranje prekidača P1 ili P2 (Slika 4. Sekcija 1).



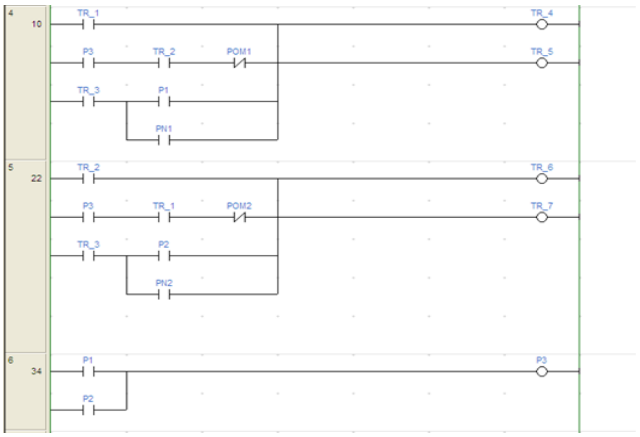

Slika 4. Sekcija 1.

Otvaranje prekidača P1 ili P2 ima za posledicu gubitak napajanja TR_1 ili TR_2 (Slika 5. Sekcija 2).




Slika 5. Sekcija 2.

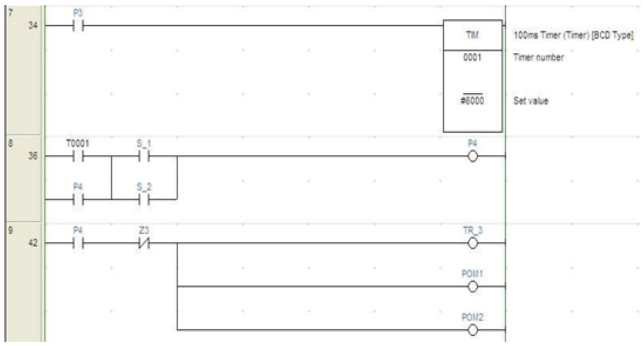
U normalnom pogonu TR_1 napaja TR_4 i TR_5, a TR_2 napaja TR_6 i TR_7. U havarijskom režimu, u kojem je jedan od transformatora van pogona, potrebno je omogućiti transformatoru koji je u pogonu preuzimanje celokupnog opterećenja. To se omogućava prekidačem P3, koji se uključuje odgovarajućom granom u zavisnosti od toga da li je setovan P1 ili P2 (Slika 6. Sekcija 3).



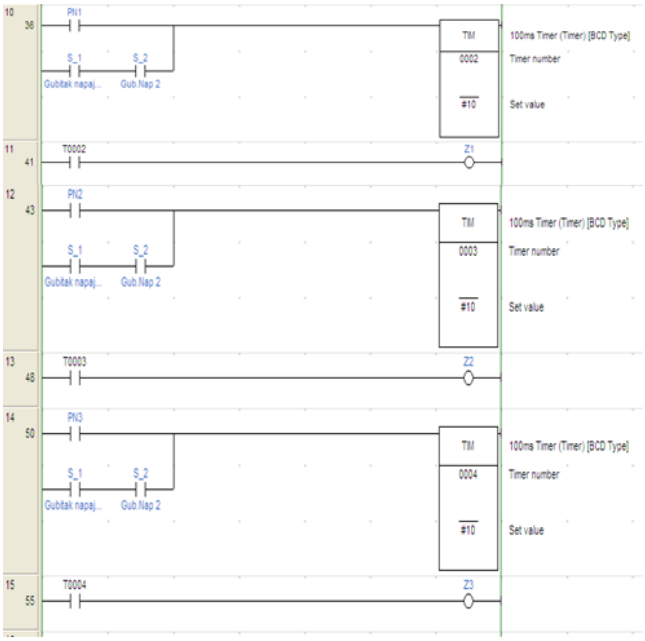

Slika 6. Sekcija 3.

Zatvaranjem prekidača P3 se uključuje tajmer TIM0001 koji odbrojava 15 minuta, odnosno odbrojava vreme za koje jedan transformator može da podnese opterećenje čitavog postrojenja. Nakon tog vremena uključuje se P4, i zatvara se prekidač P4 (samodržanje). U tom trenutku signal dobiju POM1 i POM2 i otvori se POM1 i POM2 koji su vezani na red sa P3 i TR_1, tj. Sa P3 i TR_2 (Slika 6.) . Tada radni transformator koji je preuzeo opterećenje vise ne drži to celokupno opterećenje, nego deo opterećenja koji je držao radni transformator, koji je van funkcije, sada preuzima rezervni transformator. Takođe, ako su oba senzora, S_1 i S_2 ne detektuju kvar, onda rezervni transformator neće dobiti signal za uključenje (Slika 7. Sekcija 4).

U cilju dodatne zaštite postoje senzori koji detektuju prenapone (PN1, PN2, PN3) koji ukoliko su setovani i ukoliko prenapon traje duže od jedne sekunde, transformator kod koga se desio prenapon se stavlja van upotrebe. Takodje dodatna zaštita je ta da ukoliko se desi gubitak napajanja kod oba radna transformatora, i ukoliko to traje duže od jedne sekunde, onda se svi transformatori stavljaju van pogona. Transformatori se stavljaju van pogona pomoću prekidača Z1, Z2 i Z3. Ako se kod jednog transformatora detektuje nestanak napajanja, a kod drugog se javi prenapon, potpuno opterecenje preuzima treci transformator (Slika 8. Sekcija 5).




Slika 7. Sekcija 4.




Slika 8. Sekcija 5.

## V. ZAKLJUČAK

U radu je dato tehničko rešenje za automatizaciju paralelnog rada transformatora koje je moguće primeniti u postrojenjima u cilju uštede novca ili smanjivanju ljudstva. Neprekidnost napajanja tj. kontinuirano snabdevanje potrošača električnom energijom je ključni zadatak EES-a. U pojedinim situacijama, odnosno u postrojenjima od izuzetog značaja ponekad je neophodno zanemariti ekonomski aspekt ugradnje opreme ukoliko to ide na stranu kvaliteta isporuke električne energije.

## LITERATURA

[1] B. Mitraković, "Transformatori", Naučna knjiga, Beograd 1987.

[2] J. Trhulj, N. Stefanović, Lj. Hadžibabić, REGULACIJA KVALITETA ISPORUKE ELEKTRIČNE ENERGIJE, Agencija za energetiku Republike Srbije, Beograd

# Veštačka inteligencija na primeru LEGO robota

Nevena Milenković, Teodora Stojanović

*Sadržaj* – U raznim oblastima nauke sve se više primenjuje veštačka inteligencija. U ovom radu istražene su mogućnosti veštačke inteligencije ograničenih mogućnosti (*Weak* AI) na primeru LEGO robota koji rešava Rubikovu kocku. Algoritam rešavanja kocke napisan je u softverskom paketu LEGO *Mindstorms* EV3.

## I. UVOD

Veštačka inteligencija ima sve veću primenu u svim oblastima naprednih tehnologija. Da bi veštačka inteligencija imala što veću primenu i zamenila čoveka u svakodnevnim poslovima, moramo poboljšati mogućnosti da budu slične mogućnostima ljudskog uma. Veštačka inteligencija ograničenih mogućnosti (*Weak* AI) [1] koristi se za izradu robota i modula za upravljanje, za programiranje video igara i programa koji rade kao inteligentni lični pomoćnici/asistenti. Primer ovakve veštačke inteligencije se može prikazati pomoću LEGO robota.

## II. VEŠTAČKA INTELIGENCIJA

Veštačka inteligencija (ili VI) [2] je grana informatike koja za cilj ima da računarima omogući da se ponašaju na način koji bi se okarakterisao kao inteligentan. U širem smislu, veštačka inteligencija je sposobnost računara ili robota da razmišlja na način sličan čoveku. Prvobitna ideja naučnika bila je stvaranje mašine sposobne za obavljanje različitih zadataka svojom "inteligencijom". Tada se "inteligentnom" smatrala svaka mašina koja može da obavlja složene operacije. Danas su istraživanja usmerena u pravcu stvaranja veštačke inteligencije koja je ograničena na samo jedan zadatak. Jedna od osnovnih podela veštačke inteligencije je na pravu (*strong/true*) i veštačku inteligenciju ograničenih mogućnosti (*weak*).

### *A. Prava veštačka inteligencija (strong AI)*

Prava veštačka inteligencija (*strong* ili *true* AI) je pojam koji opisuje određeni "način razmišljanja" veštačke inteligencije. Cilj je stvaranje veštačke inteligencije koja u potpunosti razmišlja sama, kao čovek. Postoje primetne razlike između mašina koje postoje danas i onoga što smatramo za pravu, potpunu veštačku inteligenciju. Mišljenja naučnika o tome kada će prava veštačka inteligencija biti moguća su podeljena. Neki veruju da će prototip prave VI biti gotov između 2030. i 2045. godine, neki da neće biti završena u ovom veku, a ostali da uopšte nije moguća.

Nevena Milenković i Teodora Stojanović su učenice trećeg razreda, smera: Elektrotehničar informacionih tehnologija – ogled u „Tehničkoj školi Zaječar", Knjeginje Ljubice 3-5, 19000 Zaječar. Email: nevena.sherly.chan@gmail.com, teateodora2000@gmail.com

Prava veštačka inteligencija bi podrazumevala da računar kreće sa radom bez ikakvih informacija i vremenom uči algoritme, kao što ljudi od ranog detinjstva uče i saznaju informacije. Drugi veliki problem sa realizacijom prave veštačke inteligencije je nemogućnost imitiranja emocionalnog dela ljudskog uma, koji je podjednako važan za donošenje odlika koliko i racionalni. Bez njega računar se oslanja samo na ono što je logično i to ga odvaja od čoveka kome teži.

### *B. Veštačka inteligencija ograničenih mogućnosti (weak AI)*

Veštačka inteligencija ograničenih mogućnosti je posebna vrsta veštačke inteligencije koja je programirana da se fokusira na jedan zadatak i izgleda "inteligentno" dok ga radi. Radi po principu veoma složenih algoritama. Naučnici su nesložni da li ovu vrstu programa još uvek treba nazivati veštačkom inteligencijom, jer ona to nije. [3] Jedan od najboljih primera ove vrste veštačke inteligencije je aplikacija Siri za IOS smart uređaje. Siri je lični asistent koji koristi jezički interfejs da odgovara na pitanja. Iako deluje inteligentno, ovaj softver to nije. Nije svestan, ne donosi sam odluke i ograničen je na pretraživanje interneta i interne memorije uređaja. [4]

## III. GLAVNE KOMPONENTE ROBOTA

Prilikom izrade naučno-istraživačkog rada korišćena su dosadašnja istraživanja u oblasti veštačke inteligencije ograničenih mogućnosti koja je primenjena na LEGO platformu na primeru rešavanja Rubikove kocke. LEGO komplet korišćen za izradu robota je LEGO *Mindstorms* EV3. Robot pomoću infracrvenog i senzora za boju „vidi" kocku i korišćenjem programa napisanog u LEGO MINDSTORMS EV3 Home Edition softveru određuje način kojim će rešiti kocku..

### *A. Kontrolna jedinica EV3 (Brick)*

Kontrolna jedinica EV3 služi kao „mozak" i izvor energije za robota. Operativni sistem kontrolne jedinice baziran je na Linux operativnom sistemu. Koristi TI Sitara AM1808 procesor. Ima 64MB RAM memorije, 16MB flash memorije i slot za microSDHC memorijske kartice. Sa računarom ili pametnim telefonom može se povezati pomoću USB-a, WiFi-a ili Bluetooth-a. Ulazni portovi 1, 2, 3, i 4 koriste se za povezivanje senzora sa jedinicom, a izlazni portovi A, B, C i D koriste se za povezivanje motora. EV3 *Brick* je prikazan na slici 1.

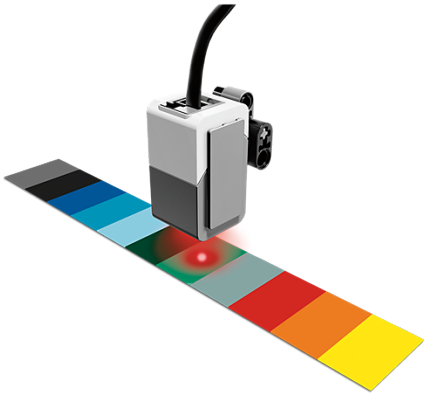



Slika 1. Kontrolna jedinica (Brick) EV3.

*B. Senzor za boju (color sensor)*

Senzor za boje (slika 2) je digitalni senzor koji detektuje boju ili intenzitet svetlosti pomoću fotootpornika i LED diode. Senzor obasjava objekat belom svetlošću i zabeležava reflektovanu boju ili intenzitet svetlosti u okolini. Pomoću crvene, zelene i plave LED diode intenzitet svetlosti se pretvara u vrednosti koje *brick* može da koristi.

Može se koristiti u tri režima*: Color Mode* (režim detektovanja boje), *Reflected Light Intensity Mode* (režim detektovanja intenziteta reflektovane svetlosti), i *Ambient Light Intensity Mode* (režim detektovanja intenziteta svetlosti u okolini).



Slika 2.Senzor za boju i intenzitet svetlosti (color sensor).

*C.. Infracrveni senzor (infrared sensor)*

Infracrveni senzor (slika 3) je digitalni senzor koji detektuje infracrvenu svetlost reflektovanu od čvrstih objekata ili detektuje signale poslate iz daljinskog infracrvenog predajnika (*Remote Infrared Beacon*). Svi predmeti sa temperaturom većom od apsolutne nule (0K) imaju toplotnu energiju i samim tim su izvor infracrvene radijacije. Za određivanje udaljenosti senzor emituje zrak infracrvene radijacije i beleži sve što sprečava zrak da putuje dalje.

Može se koristiti u tri režima: *Proximity Mode* (režim određivanja udaljenosti), *Beacon Mode* (režim praćenja), i *Remote Mode* (režim upravljanja daljinskim upravljačem).

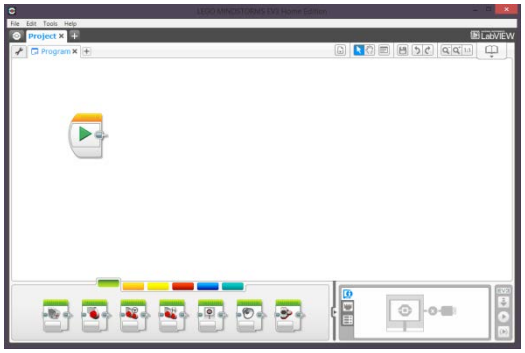


Slika 3. Infracrveni senzor i daljinski (infrared sensor).

## IV. APLIKACIJA ZA REŠAVANJE KOCKE

*A. LEGO Mindstorms EV3 softverski paket*

LEGO *Mindstorms* EV3 [5] je softverski paket za programiranje treće generacije LEGO robota (*EVolution* 3) Baziran je na operativnom sistemu Linux. Radi po principu blokova (*icons*) koji se prevlače preko ekrana na odgovarajuće mesto u programu. Postoje pet vrsta blokova: kretanje, petlje, senzori, matematičke operacije i napredni blokovi. Program je prikazan na slici 4.

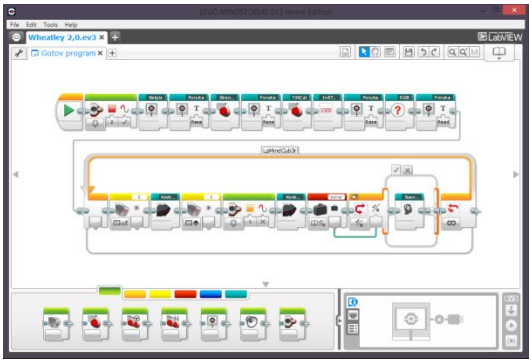


Slika 4. LEGO Mindstorms EV3.

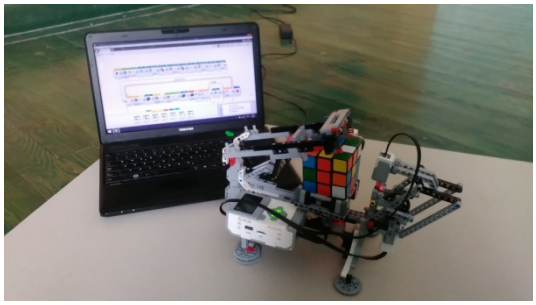
*B. Rad programa za rešavanje Rubikove kocke*

Robot na osnovu složenog algoritma rešava Rubikovu kocku. Na početku algoritma robot namešta mehaničku „ruku“ i pokretno postolje na pravo mesto, proverava da li su svi senzori priključeni i funkcionalni i da li je kocka na postolju. Ukoliko ima problema, na ekranu kontrolne jedinice ispisuje se poruka za grešku. Ako je sve na mestu i svi senzori funkcionalni i kalibrisani, prelazi se na sledeću fazu algoritma. Senzor za boju se pomera na svoje mesto, skenira svih šest strana kocke i beleži gde je koje obojeno polje. Nakon toga program po algoritmu računa najbrži

način (najmanje poteza) rešavanja kocke. Robot prelazi na poslednju fazu programa u kojoj mehaničkom „rukom“ i okretanjem pokretnog postolja rešava kocku na osnovu prethodno izračunatog šablona. Na osnovu dosadašnjih pokušaja utvrđeno je da je prosečno vreme rešavanja 74 sekunde, a prosečan broj poteza 23. Glavni deo programa prikazan je na slici 5, a robot na slici 6.



Slika 5. Glavni deo programa.



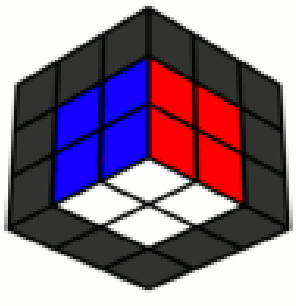
Slika 6. LEGO robot pored svog programa.

*C. Algoritam za rešavanje Rubikove kocke*

Metod koji robot koristi za rešavanje kocke je „Petrusov metod“ [6]. Lars Petrus ga je predstavio 1981, godinu ranije pre nego što je sa njim postao prvi svetski šampion u rešavanju Rubikove kocke.

Ovaj metod sastoji se iz nekoliko osnovnih koraka. Prvi korak, koji se najčešće završi pre petog poteza, je pravljenje 3 bloka veličine 2x2x2 iste boje. (prikazano na slici 7) Potom se ovi blokovi proširuju na veličinu 2x2x3 bez rasturanja prethodnih. Sledeći korak je nameštanje „loših“ ivica, odnosno stavljanje ivice jedne boje na stranicu kocke sa blokom te boje. Četvrti korak je rešavanje strana sa ivicama okretanjem. Ćoškovi se menjaju da bi bili na odgovarajućim mestima. Završava se zadnja stana kocke. Poslednji korak je rešavanje prednje strane bez remećenja ostalih.

Prednosti ovog metoda su: veća brzina rešavanja od nekih popularnijih metoda, mali broj poteza neophodnih za rešavanje, ima malo šema koje osoba, ili u ovom slučaju, robot, mora da nauči.



Slika 7. Prvi korak za rešavanje Rubikove kocke Petrusovim metodom.

## V. ZAKLJUČAK

Cilj našeg rada je prikaz veštačke inteligencije ograničenih mogućnosti na primeru robota koji rešava Rubikovu kocku uz pomoć LEGO konstrukcije, senzora i unapred zadatog algoritma. Robot sa svojim programom dokazuje da je i bez prave veštačke inteligencije ili mašinskog učenja moguće rešiti neke jednostavnije zadatke. Korišćenjem mašinskog učenja moguće je povećati brzinu i smanjiti broj neuspešnih pokušaja rešavanja kocke.

## ZAHVALNICA

Autori rada izražavaju duboku zahvalnost svima koji su pomogli realizaciju projekta i mentoru Dejanu Nikoliću.

## LITERATURA

[1] AI Horizon (http://aihorizon.com/essays/generalai/intro.html), 2016
[2] P. Domingos, The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World Penguin Books Limited, 2015
[3] Techopedia (https://www.techopedia.com/definition/31621/weak-artificial-ntelligence-weak-ai), Septembar, 2016
[4] Wikipedia (https://en.wikipedia.org/wiki/Weak_AI), Septembar, 2016
[5] LEGO Mindstorms' Official Website (https://www.lego.com/en-us/mindstorms/support), Septembar, 2016
[6] Petrus Method (https://www.speedsolving.com/wiki/index.php/Petrus_Method), Septembar, 2016

# VR vožnja

Kristijan Jović, David Milenković, Veljko Cvetković, Danko Danković, Igor Arsić

*Sadržaj* – U ovom radu opisan je projekat vožnje u VR okruženju. Ideja je proistekla kako bi korisnici stekli što realniji osećaj vožnje, samo bez pravih vozila i rizika koji ona donose.

## I. UVOD

Često maštamo o tome kako bi izgledalo voziti neki dobar i naperdan auto ili uopšte voziti, ali često je problem ka ostvarivanju tog cilja nedostatak finansija ili vozačke dozvole. Iz tog razloga smo došli na ideju da kreiramo projekat VR Vožnje.

## II. KARAKTERISTIKE I NAMENA

Projekat se realizuje uz pomoć Virtual Reality Headseta za računar, poznatiji pod nazivom "Oculus Rift" [1] (Slika 1.). Iako dosta zahtevan za programiranje i održavanje, kao i konfiguracijski, projekat VR vožnja pruža dosta realan osećaj prilikom korišćenja platforme. Uz mogućnosti koje pruža, ovaj VR uređaj ima i svoje konfiguracijske zahteve – jak računar. Potreban je računar koji poseduje dovoljno jaku konfiguraciju kako bi mogao izvršavati sinhronizaciju u realnom vremenu i obradu grafičkih podataka što zahteva jako velike performanse i mogućnost obrade komponenti koje ga sačinjavaju.




Slika 1. Oculus Rift Virtual Reality Headset.

## III. KONSTRUKCIJA/JEZGRO

Kristijan Jović, David Milenković i Veljko Cvetković su učenici Tehničke škole "Rade Metalac" iz Leskovca, Dimitrija Tucovića bb, 16000 Leskovac, Srbija, E-mail: it.kristijan.jovic@gmail.com, veljalig20@gmail.com, davideit172@gmail.com

Danko Danković i Igor Arsić su profesori elektrotehničke grupe predmeta u Tehničkoj školi "Rade Metalac" u Leskovcu, Dimitrija Tucovića bb, 16000 Leskovac, Srbija, E-mail: ddanko@live.com, igorarsic78@gmail.com

Osnova projekta je aplikacija koja je izrađena pomoću jednog od programa – Unreal Engine 4 ili Unity Designer-a. Generalna definicija aplikacije VR Vožnja se poistovećuje sa 3D igrama, što je osnova ovog Sistema tj. projekta.

Virtualna realnost je sama srž ovog projekta jer slama granice između Virtualnog i Realnog i pruža veliki užitak prilikom korišćenja aplikacije. U 3D okruženju aplikacije korisnik se može kretati na više načina, zavisno od tipa vozila kojim upravlja ( automobil, dvotočkaš i sl.):

1) Oculus Touch (Slika 2.). – Kontroleri koji se koriste da pokrete ruku prenesu u 3D okruženje aplikacije i omoguće interakciju korisnika sa okolinom. Na njima se nalaze kontrole za kretanje ( palice koje se pomeraju pomoću prsta) i programabilna dugmad koja predstavlja funkcije koje korisnik može da izvršava (koči, kreni i sl.). Oculus Touch se koristi kod vozila kao što su bicikl i motocikl jer je to najpogodnije rešenje s obzirom da se radi o Virtualnom okruženju bez fizičkih objekata.



Slika 2. Oculus Touch kontroleri.

2) Volan sa pedalama (Slika 3.) – najisplatljivije rešenje kod vozila poput automobila koja ne zahtevaju komplikovane pokrete. Volanom se upravlja vozilo, a pedale se koriste za funkcije koje izvršava korisnik, uobičajeno su to ubrzaj i uspori.



Slika 3. Volan sa pedalama.

## IV. PLATFORMA POKRETA

U sklopu projekta dolazi platforma koja imitira pokrete iz igre [2]. Primer takve platforme možete videti na slici 4.

Ova platforma drastično uvećava doživljaj realnosti i značajno smanjuje granicu između realnog i virtualnog sveta. Platforma se programira ručno da oponaša pokrete vozila iz igre u realnom svetu. Da bi se to postiglo platforma mora biti fiksirana za čvrstu i stabilnu površinu i osoba koja sedi u njoj treba biti pravilno vezana i obezbeđena da ne bi došlo do neželjenih povreda.

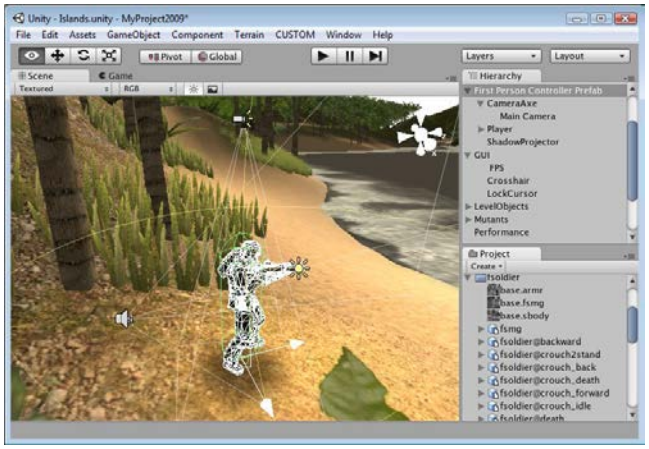


Slika 4. MotionSim platforma za imitiranje pokreta.

## V. APLIKACIJA

Kao što je već rečeno, sama srž ovog projekta je aplikacija tj. igra u kojoj se svi procesi odvijaju. Igra se izrađuje u nekom od razvojnih okruženja kompatibilnih sa Oculus Rift tehnologijom:

1) Unity 3D [3] (Slika 5.) – Jednostavnije razvojno okruženje koje ima višenamensku funkciju; U njemu se mogu kreirati mape po kojima će se vozila kretati, mogu se kreirati sama vozila (objekti) i fizika igre. Sama platforma nije dovoljna da se realizuje igra, tako da se mora koristiti i C# programski jezik (najpodržaniji jezik od strane razvojnog okruženja).



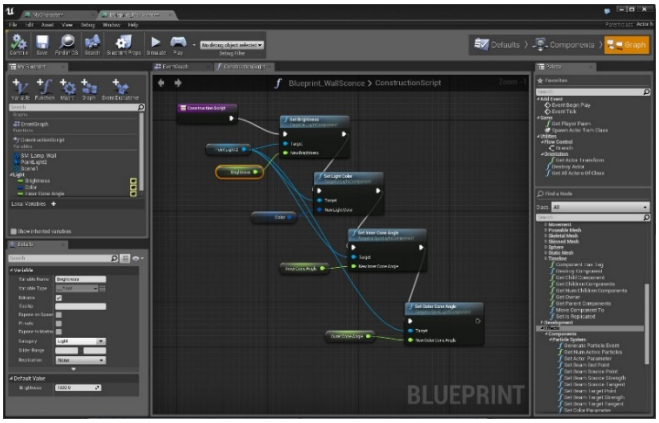
Slika 5. Unity 3D razvojno okruženje.

2) Unreal Engine 4 [4] (Slika 7) – Napredno razvojno okruženje koje ima sličnu funkcionalnost kao i Unity 3D, samo što pruža mnogo veći broj mogućnosti ( npr. dizajner može kreirati nove materijale i dodeljivati im svojstva ). UE4 razvojno okruženje čini odličnu kombinaciju sa najnovijom Vulkan API tehnologijom [5], sa kojom daje jako bogatu grafiku i fiziku, što treba biti glavna odlika kod virtuelne realnosti pa samim tim i ove aplikacije. Primer okruženja igre kreirane pomoću Unreal Engine 4 razvojnog okruženja prikazan je na slici 6.



Slika 6. Primer 3D okruženja kreiranog u UE 4 razvojnom okruženju.

Unreal Engine 4 je moguće koristiti bez dodatnog programskog jezika (što smanjuje funkcionalnost igre), jer u samom okruženju postoje integrisane metode za kreiranje događaja i fizike igre ( blueprints – nacrti, takođe prikazani na slici 7). Ukoliko se koristi dodatni jezik za programiranje, najpodržanije rešenje je C++.




Slika 7. UE4 okruženje sa Blueprints sistemom.

## VI. ZAKLJUČAK

Projekat VR Vožnja može biti jako koristan kako za zabavu, tako i za edukaciju, npr. Može se napraviti modifikacija projekta kako bi se on mogao koristiti u auto školama i kandidatima dati neki "Realni" osećaj vožnje, čemu će doprineti i platforma pokreta i virtualna realnost. Sam projekat je jako zahtevan te samim tim je skoro nemoguće realizovati isti sa nivoom znanja srednje škole. Iz tog razloga projekat je predstavljen samo kao ideja koju iskusan i obrazovan tim inžinjera, programera i dizajnera može sprovesti u delo.

## ZAHVALNICA

Zahvaljujemo se Tehničkoj školi "Rade Metalac" za kompletnu podršku prilikom pisanja ovog projekta.

## LITERATURA

[1] Oculus Rift Headset, uređaj kompanije Oculus, Dostupno na: https://www.oculus.com/rift/

[2] MotionSim simulator pokreta video primer, Dostupno na: https://www.youtube.com/watch?v=mrSEhzwGohU

[3] Unity 3D Designer, proizvod kompanije Unity, Dostupno na: https://unity3d.com/

[4] Unreal Engine 4, proizvod kompanije Epic, Dostupno na: https://www.unrealengine.com/en-US/what-is-unreal-engine-4/

[5] Unreal Engine 4 + Vulkan API video primer, Dostupno na: https://www.youtube.com/watch?v=zr6u5wJEs4w

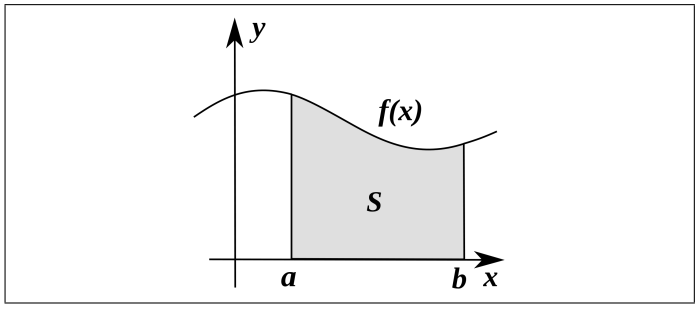
# Numerička integracija na neograničenom intervalu primenom ortogonalnih polinoma

Mihailo Krstić

*Sadržaj* - U ovom radu ćemo se baviti numeričkim metodima za izračunavanje integrala na intervalu $(0, +\infty)$. Osnova metoda koji će biti izložen je teorija ortogonalnih polinoma i Gaus-Kristofelove formule. Metod ima i jasnu algoritamsku formulaciju koja omogućuje jednostavnu implementaciju. Koristićemo programski paket `Mathematica` za implementaciju i numerička testiranja teorijskih razmatranja.

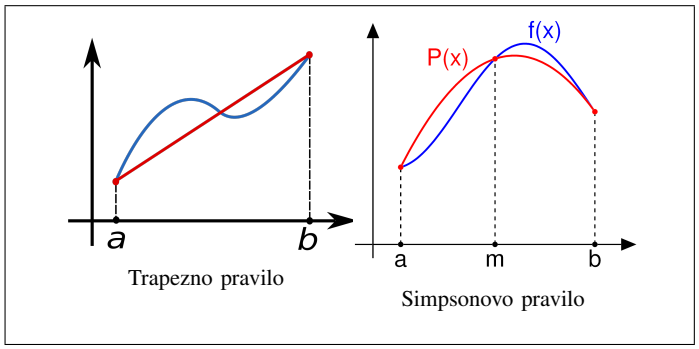
## I. Uvod

Potreba za numeričkom integracijom je u matematici i drugim srodnim naukama izuzetno velika. Rešiti numerički integral na konačnom segementu je moguće na više načina.



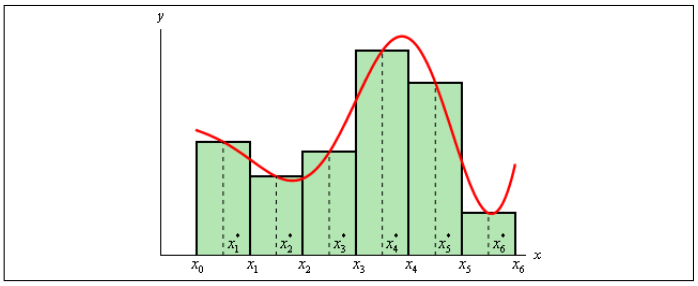

Slika 1: Integracija na konačnom segmentu

Mogu se koristiti intuitivni i dobro poznati geometrijski principi-Njutn Kotesove formule u kojima podintegralnu funkciju menjamo sa odgovarajućim interpolacionim polinomom dok su čvorovi integracije fiksirani. Kod ovog tipa integracije moguće je ne koristiti aparat ortogonalnosti.




Slika 2: Aproksimacija integrala linearnom i kvadratnom interpolacijom

Mihailo Krstić je student druge godine master studija Departmana za matematiku Prirodno-matematičkog fakulteta Univerziteta u Nišu, Višegradska 33, 18 000 Niš, e-mail: mihailo1994@yahoo.com.




Slika 3: Aproksimacija integrala Rimanovom sumom metodom srednje tačke

Kod beskonačnih intervala je problem komplikovaniji, jer se ne zna koliko daleko trebamo ići po realnoj osi u popunjavanju površine ispod grafika funkcije. Rešavaćemo opšti problem integracije

$$\int_0^{+\infty} f(x)dx. \tag{1}$$

Navodimo dobro poznata tvrđenja iz teorije ortogonalnih polinoma i Gaus-Kristofelovih formula, koji su deo standardnih udžbenika navedenih u literaturi. Na primer, videti [4], [5], [6] i [8]. Formule Gaus-Kristofelovog tipa imaju oblik:

$$\int_a^b p(x)f(x)dx = \sum_{k=1}^{n} A_k f(x_k) + E_n(f), \tag{2}$$

gde su koeficijenti i čvorovi kvadrature dati sledećom teoremom i gde je $p$ težinska funkcija na $(a, b)$.

**Teorema 1.** *Formula (2) ima maksimalni algebarski stepen tačnosti* $2n-1$ *ako se koeficijenti odaberu kao*

$$A_k = \frac{a_n}{a_{n-1}} \frac{||Q_{n-1}||^2}{Q_{n-1}(x_k)Q'_n(x_k)}, \quad k \in \{1, 2, ..., n\},$$

*pri čemu je* $a_n$ *je vodeći koeficijent polinoma* $Q_n$, $a_{n-1}$ *vodeći koeficijent polinoma* $Q_{n-1}$ *i ako se čvorovi kvadraturne formule (2) odaberu kao nule odgovarajućeg ortogonalnog polinoma* $Q_n$.

Poznata je i sledeća teorema koja daje izraz za ostatak u kvadraturnoj formuli (2).

**Teorema 2.** *Neka je funkcija* $f$ *iz klase* $C^{(2n)}[a, b]$. *Tada važi*

$$E_n(f) = \frac{||Q_n||^2}{(2n)!a_n^2} f^{(2n)}(\xi),$$

*za neko* $\xi \in (a, b)$.

Vidimo da su formule ovog tipa visokog reda tačnosti ako se čvorovi i koeficijenti u (2) odaberu po teoremi 1. Napomenimo još da svoju pravu vrednost ortogonalni polinomi pokazuju baš u numeričkoj integraciji.

## II. Konstrukcija formule za numeričku integraciju na beskonačnom intervalu

Transformišimo integral (1) na sledeći način:

$$\int_0^{+\infty} f(x)dx = \int_0^1 \frac{f(-\ln x)}{x}dx = \frac{1}{2}\int_{-1}^1 \frac{f(-\ln\frac{t+1}{2})}{\frac{t+1}{2}}dt = \int_{-1}^1 \frac{f(-\ln\frac{t+1}{2})}{t+1}dt. \quad (3)$$

Dakle, sada imamo integraciju na intervalu $(-1,1)$ te smo u mogućnosti da koristimo ortogonalnost na tom segmentu i Čebiševljeve polinome prve vrste. Primetimo da je najveći problem kod konstrukcije i primene opštih formula Gaus-Kristofelovog oblika nalaženje nula odgovarajućih ortogonalnih polinoma. Navedimo dobro poznatu teoremu koja daje karakterizaciju nula ortogonalnih polinoma.

**Teorema 3.** *Neka je $\{Q_n\}_n$ niz ortogonalnih polinoma na intervalu $(a,b)$ sa težinskom funkcijom p. Tada su sve nule polinoma $Q_n$ realne, proste i leže u intervalu ortogonalnosti $(a,b)$.*

Dakle, koristeći prethodnu teoremu i neki od metoda za numeričko izračunavanje nula polinoma koje su jednostruke i realne možemo odrediti čvorove integracije. Jedan takav metod za simultano nalaženje nula polinoma koje su jednostruke je metod Vajerštrasa i može se naći u [2]. U ovom radu će cilj biti opis metoda za integraciju na beskonačnom segmentu koji nema iterativnu pozadinu. Da bi ovaj problem bio prevaziđen, koristićemo Čebiševljeve ortogonalne polinome prve vrste, jer su oni definisani jednakošću

$$T_n(x) = \cos(n\arccos(x)), \quad n \in \mathbb{N}, \quad (4)$$

tako da smo u mogućnosti da eksplicitno odredimo nule ovih ortogonalnih polinoma. Iz (4) dobija se da su nule ovih ortogonalnih polinoma date sa

$$x_k^{(n)} = \cos\left(\frac{\pi}{2n}(2k-1)\right), \quad k = 1,...,n, \quad n \in \mathbb{N}. \quad (5)$$

Navedimo i tvrđenje koje se odnosi na norme Čebiševljevih ortogonalnih polinoma prve vrste.

**Teorema 4.** *Neka je $T_n$ $n-$ti član niza Čebiševljevih ortogonalnih polinoma na $(-1,1)$ sa težinskom funkcijom $p(x) = \frac{1}{\sqrt{1-x^2}}$. Tada važi:*

$$||T_n|| = \begin{cases} \pi, & \text{ako je } n = 0 \\ \frac{\pi}{2}, & \text{ako je n prirodan broj.} \end{cases} \quad (6)$$

Navedimo još par dobro poznatih teorema koja će nam pomoći da konstruišemo krajnji oblik Gaus-Čebiševljeve kvadraturne formule.

**Teorema 5.** *Neka je $a_n$ vodeći koeficijent $n-$tog člana niza Čebiševljevih ortogonalnih polinoma. Tada je*

$$a_n = 2^{n-1}. \quad (7)$$

**Teorema 6.** *Neka je $T_n$ $n-$ti član niza Čebiševevljevih ortogonalnih polinoma prve vrste. Tada važi*

$$T_{n-1}(x_k)T_n'(x_k) = n, \quad (8)$$

*gde je $x_k$ proizvoljna nula pomenutog polinoma.*

Ovu teoremu nije teško pokazati direktnom proverom. Navedena teorema je vrlo bitna jer otklanja potencijalno veliki računarski nedostatak u formulama Gaus-Kristofelovog tipa, zato što izraz u imeniocu iz teoreme 1, koji zavisi od polinoma, njegovog izvoda i njegovih nula prevodi u konstantu koja će biti ulazni parametar u algoritam.

Kod drugih formula Gaus-Kristofelovog tipa ovo nije slučaj te i u ovom delu postoji potencijalni problem. Imajući u vidu činjenicu da u tom slučaju nule odgovarajućeg ortogonalnog polinoma treba odrediti numerički problem bi se još dosta ukomplikovao.

Koristeći navedena tvrđenje i jednakost (5) imamo da Gaus-Čebiševljeva kvadraturna formula ima oblik

$$\int_0^{+\infty} \frac{f(x)}{\sqrt{1-x^2}}dx = \frac{\pi}{n}\sum_{k=1}^n f\left(\cos\left(\frac{\pi}{2n}(2k-1)\right)\right) + E_n(f), \quad (9)$$

gde je sa $p(x) = \frac{1}{\sqrt{1-x^2}}$ data težinska funkcija na intervalu $(-1,1)$.

Sada, primenimo formulu (9) na integral (1). Dobija se:

$$\int_0^{+\infty} f(x)dx = \int_{-1}^1 \frac{f\left(-\ln\frac{t+1}{2}\right)}{t+1}dt = \int_{-1}^1 \frac{1}{\sqrt{1-t^2}}\sqrt{\frac{1-t}{1+t}}f\left(-\ln\frac{t+1}{2}\right)dt = \frac{\pi}{n}\sum_{k=1}^n g\left(\cos\left(\frac{\pi}{2n}(2k-1)\right)\right), \quad (10)$$

gde smo sa $g$ označili funkciju

$$g(t) = \sqrt{\frac{1-t}{1+t}}f\left(-\ln\frac{t+1}{2}\right).$$

Sledeća teorema je posledica teoreme 2.

**Teorema 7.** *Neka je data funkcija $f$ iz klase $\mathcal{C}^{(2n)}[a,b]$. Ostatak u Gaus-Čebiševljevoj kvadraturnoj formuli (9) je dat sa*

$$E_n(f) = \frac{\pi}{2^{2n-1}(2n)!}f^{(2n)}(\xi), \quad (11)$$

*za neko $\xi \in (-1,1)$.*

Koriteći teoremu o ostatku u nekim slučajevima kada izvod funkcije možemo ograničiti, dobijamo procenu veličine greške, tj. možemo odrediti koliko je čvorova dovoljno uzeti da bi se postigla neka unapred zahtevana tačnost.

Primetimo da smo rešavali najpoštiji problem integracije dat sa (1). Ukoliko bismo činilac $p(x) = \frac{1}{\sqrt{1-x^2}}$ mogli da izdvojimo iz integrala tada bismo povećali tačnost izračunavanja. Navešćemo takav karakterističan primer a potom ćemo u posebnom odeljku numerički testirati ovaj algoritam na proizvoljnoj funkciji, kada nije moguće izdvojiti činilac $p$.

**Primer 1.** *Sa tačnošću $\epsilon = 10^{-5}$ izračunati integral*

$$I = \int_{-1}^1 \frac{\cos x}{\sqrt{1-x^2}}dx.$$

*Rešenje.* U ovom primeru se težinska funkcija $p(x) = \frac{1}{\sqrt{1-x^2}}$ pogodno izdvaja te je $f(x) = \cos x$. Koristeći (11) i nejednakost $|\cos x| \leq 1$ koja važi za svako $x \in \mathbb{R}$ možemo odrediti koliki je broj čvorova potreban za postizanje zahtevane tačnosti. Dobijamo da je $n > 3$ pa ćemo uzeti 4 čvora. Imamo da je

$$T_4(x) = \cos(4\arccos x) = 1 - 8x^2 + 8x^4,$$

te su čvorovi određenji sa

$$x_1 = -\frac{1}{2}\sqrt{2-\sqrt{2}} \approx -0.382683,$$

$$x_2 = \frac{1}{2}\sqrt{2-\sqrt{2}} \approx 0.382683,$$

$$x_3 = -\frac{1}{2}\sqrt{2+\sqrt{2}} \approx -0.923879,$$

$$x_4 = \frac{1}{2}\sqrt{2+\sqrt{2}} \approx 0.923879.$$

Sada primenom (9) dobijamo da je aproksimativna vrednost integrala $I$ sa tačnošću $\epsilon = 10^{-5}$ jednaka 2.40394.

Veliki broj rešenih primera iz numeričke integracije se može naći u [7], [9] i [10].

## III. Implementacija u programskom paketu Mathematica i numerička testiranja

U ovoj sekciji data je implementacija izloženog metoda u programskom paketu `Mathematica` i vršimo numerička testiranja na konkretnim primerima. Primetimo da je praktična upotreba teoreme o proceni greške vrlo ograničena, jer kod funkcija koje imaju komplikovan analitički izraz za $n-$ti izvod vrlo teško se vrši njegova majorizacija.

```
NesvojstvenIntegral[f_,n_] :=
  Module[{g,Int},
   g[x_] :=
   Sqrt[(1-x)/(1+x)]f[-Log[(x+1)/2]];
   Int = (Pi/(n))*
   Sum[g[Cos[(Pi/(2n))(2i-1)]],{i,1,n}];
   Return[Int];
   ];
```

U Tabeli I su dati integrali koji služe kao test primeri, njihove tačne vrednosti (vrednosti koje su tačne na svim datim decimalnim mestima) i vrednosti dobijene izloženom metodom. Boldirane su one cifre koje se poklapaju sa tačnim ciframa. U proseku, numerički rezultati se poklapaju sa tačnim vrednostima do trećeg decimalnog mesta a u nekim slučajevima rezultati su tačni i na četvrtom decimalnom mestu.

Analizirajući Tabelu I sa numeričkim rezultatima vidimo da rešavanjem opšteg problema integracije (1) metod gubi na tačnosti jer se podintegralna funkcija smenama koje prevode beskonačan interval u konačan znatno komplikuje. Sa druge strane pogodnost ovakvog pristupa je ta što na ovaj način imamo generalnu šemu, bez ikakvih pretpostavki o podintegralnoj funkciji (naravno osim pretpostavke da intergal za tu funkciju konvergira tj. da je data funkcija integrabilna). Takođe, posle primene transformacije (3) imamo integraciju na $(-1,1)$ ali nismo u mogućnosti da primenimo neku od tehnika za numeričku integraciju na konačnom segmentu jer transformacijom (3) dolazimo do singulariteta podintegralne funkcije na krajevima intervala integracije, te Njutn-Kotesove formule postaju praktično neupotrebljive jer su one zatvorenog tipa (prvi i poslednji čvor se poklapa sa krajevima intervala integracije).

Navedimo sada jedan karakterističan primer na kojem izložena metoda pokazuje loše rezultate.

**Primer 2.** *Izračunati integral Frenela*

$$\int_0^{+\infty} \sin(x^2)dx = \frac{1}{2}\sqrt{\frac{\pi}{2}} \approx 0.626657,$$

*koristeći izloženi pristup.*

*Rešenje.* Preglednosti radi, dati su rezultati dobijeni primenom Gaus-Čebiševljeve formule sa različitim brojem čvorova u obliku tabele (Tabela II).

Veliki broj računarskih kodova vezanih za ovu temu se mogu naći u [3], [7] i [10], dok je u [2] dat algoritamski osvrt na gotovo sve algoritme koji se obrađuju.

Iz Tabele II se vidi loše ponašanje ovog metoda na datoj funkciji. Razlog za takvo ponašanje metoda leži u tome što podintegralna funkcija beskonačno mnogo puta osciluje na intervalu integracije. Primenom izloženog metoda, zbog smena koje su korišćene u metodu, imamo da podintegralna funkcija beskonačno mnogo puta osciluje oko kraja intervala integracije te metod postaje neupotrebljiv. Dajemo grafik funkcije $x \mapsto \sin(x^2)$ da bi se video tok grafika i njegovo oscilovanje.

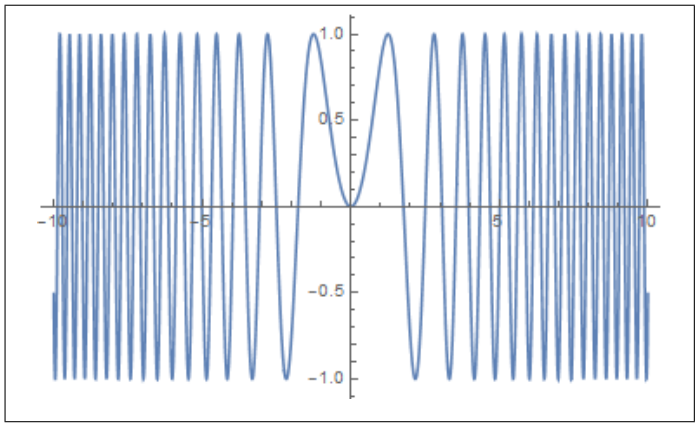
U literaturi [1], [6] posvećena je velika pažnja numeričkoj integraciji brzo oscilirajućih funkcija baš iz razloga koji su prikazani u primeru 2.

Tabela I: Rezultati testiranja na numeričkim primerima

| Test integral | Tačna vrednost | Aproksimacija | Broj čvorova |
|---|---|---|---|
| $\int_0^{+\infty} \frac{e^{-x}}{1+x^2}dx$ | **0.621**45 | **0.621**533 | 50 |
| $\int_0^{+\infty} e^{-x^2}dx$ | **0.886**227 | **0.886**741 | 20 |
| $\int_0^{+\infty} \frac{1}{(x^4+e^{-x^2})^2}dx$ | **1.260**49 | **1.260**91 | 22 |
| $\int_0^{+\infty} e^{-x}dx$ | **1.** | **1.00**645 | 8 |
| $\int_0^{+\infty} \frac{1-\cos 100x}{x^2}dx$ | **157.0**8 | **157.0**33 | 200 |
| $\int_0^{+\infty} \sqrt{x}e^{-x}dx$ | **0.886**227 | **0.886**902 | 30 |
| $\int_0^{+\infty} \cos(x)e^{-x}dx$ | **0**.5 | **0**.499927 | 30 |
| $\int_0^{+\infty} x^3e^{-2x^2}dx$ | **0.125** | **0.125003** | 12 |
| $\int_0^{+\infty} \frac{x}{e^x+1}dx$ | **0.822**955 | **0.822**467 | 65 |
| $\int_0^{+\infty} \frac{1}{x}(\frac{1}{1+x^{20}} - e^{-2x^2})dx$ | **0.635**181 | **0.635**152 | 40 |
| $\int_0^{+\infty} \sin\frac{x^2-2x+1}{x^6+22}dx$ | **0.037**2979 | **0.037**1531 | 50 |
| $\int_0^{+\infty} \frac{e^{-x^2}-e^{-x}}{x}dx$ | **0.288**608 | **0.288**682 | 50 |
| $\int_0^{+\infty} \frac{\ln(1+x^2)}{1+x^8}dx$ | **0.310606** | **0.310606** | 25 |
| $\int_0^{+\infty} \frac{x^3}{e^{2x}-1}dx$ | **0.405871** | **0.405854** | 30 |
| $\int_0^{+\infty} \frac{x^3+1}{e^{5x}-\sin x}dx$ | **0.220**361 | **0.220**59 | 30 |
| $\int_0^{+\infty} \frac{1}{12\sqrt{x^7}+10\ln(x^4+10)}dx$ | **0.058**9534 | **0.058**8921 | 40 |
| $\int_0^{+\infty} \frac{1}{1+2017x^{2017}}dx$ | **1.** | **1.00**406 | 29 |
| $\int_0^{+\infty} \frac{1}{10+\sqrt{x^{2017}}}dx$ | **1.** | **1.000**406 | 29 |
| $\int_0^{+\infty} \frac{1+\sqrt{x}}{1+x^4}dx$ | **1.960**83 | **1.960**58 | 15 |
| $\int_0^{+\infty} \frac{1}{1+x^4}dx$ | **1.11**072 | **1.11**268 | 10 |

Tabela II: Rezultati testiranja za Frenelov integral

| Broj čvorova | Aproksimacija |
|---|---|
| 10 | 4.34089 |
| 50 | 0.723158 |
| 100 | -1.31038 |
| 200 | -0.421788 |
| 300 | 0.592605 |
| 400 | -5.27946 |
| 500 | 0.321562 |



Slika 4: Grafik funkcije $f(x) = \sin(x^2)$

## IV. Zaključak

Iz svega navedenog zaključujemo da je numerička integracija na neograničenim intervalima vrlo aktuelna oblast puna mogućnosti za novim istraživanjima.

Izloženi metod je osnovni metod. Kako primena ovog metoda ne zahteva nikakve pretpostavke o funkciji $f$, osim pretpostavke o konvergenciji nesvojstvenog integrala, imamo da je oblast primene vrlo velika. Zbog velike opštosti ovakvog pristupa numerička tačnost je smanjena.

Iz Tabele I u kojoj su dati test primeri zaključujemo da je metod vrlo efikasan za funkcije koje brzo opadaju u beskonačnosti dok je vrlo nestabilan za brzo oscilirajuće funkcije što se vidi iz njegovog ponašanja primenjenog na integral Frenela (Tabela II).

## Zahvalnost

Dugujem veliku zahvalnost Prof. dr Nebojši Dinčiću sa Departmana za matematiku Prirodno-matematičkog fakulteta u Nišu, koji mi je pružio pomoć pri izradi ovog rada.

## Literatura

[1] Philip Davis, Philip Rabinowitz, *Methods of numerical integration*, Second edition, Academic press, England, 1983.

[2] Marko Petković, *Algoritmi numeričke analize*, Prirodno-matematički fakultet, Niš, 2013.

[3] Ljiljana Petković, *Numerička matematika sa programiranjem*, Niš, 1994.

[4] Ljiljana Petković, *Numerička analiza*, Mašinski fakultet, Niš, 2003.

[5] Gradimir Milovanović, *Numerička analiza, I deo*, Naučna knjiga, Beograd, 1988.

[6] Gradimir Milovanović, *Numerička analiza, II deo*, Naučna knjiga, Beograd, 1991.

[7] Desanka Radunović, Aleksandar Samardžić, Filip Marić *Numeričke metode, zbirka zadataka kroz C, Fortran i Matlab*, Akademska misao, Beograd, 2005.

[8] Desanka Radunović, *Numeričke metode*, Akademska misao, Beograd, 2003.

[9] Gradimir Milovanović, Milan Kovačević, Miodrag Splalević, *Numerička matematika, zbirka rešenih problema*, Elektronski fakultet, Niš, 2002.

[10] Ljiljana Petković, Slobodan Tričković, Predrag Rajković, *Zbirka zadataka iz numeričke matematike*, Mašinski fakultet, Niš, 1997.

# Automatizacija proizvodnje u staklenicima sa Arduino platformom

Veljko Tomašević, Ivan Golubović

*Sadržaj-* U ovom radu prikazana je realizacija automatizovane makete plastenika, sistem je zasnovan na Arduino razvojnoj platformi. U maketi plastenik vrši regulacija željene temperature, automatsko zalivanje biljaka sa sigurnosnim sistemima za pumpu i moguće kvarove u zalivanju i osvetljenja koje se menja u zavisnosti od spoljšnje svetlosti.

## I. UVOD

Celokupan projekat je osmišljen tako da omugući pojedincu ili grupi da na lak i efektivan način uzgajaju biljke, a da pri tome njihovo vreme i novac utrošen u to budu svedeni na minimum.[1]



Slika 1. Izgled završenog projekta.

Staklenik ima zadatak da uzgaja biljke u najboljim uslovima za njih, a da pri tome vlasnik staklenika bude u mogućnosti da menja postavke koje su u programu i time bude u mogućnosti da uzgaja različite biljke.

## II. PRINCIP RADA

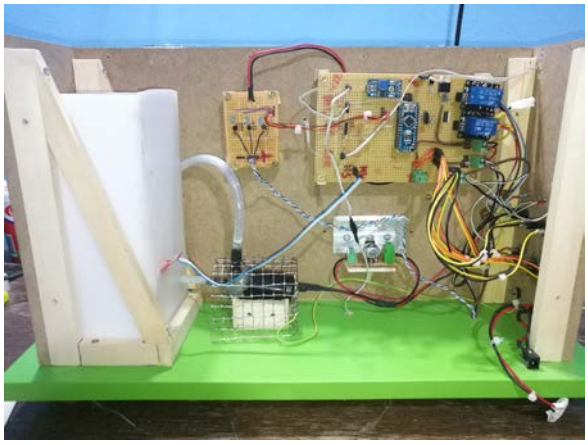
Koncept ovog projekta zasnovan je na monitoringu onih fizičkih i hemijskih veličina u plastenicima, koje su bitne za pravilan uzgoj biljaka, koji se vrši preko Arduino sa ostalim senzorima i potrošačima, kao što je na slici 2.

Podaci o ovim veličinama se prikupljaju putem različitih vrsta senzora uz pomoć kojih Arduino upravlja određenim aktivnostima u stakleniku, on vrši automatizovano navodnjavanje, regulaciju temperature i veštačko osvetljenje.[2]

Na stakleniku se nalazi displej i tri dugmeta se koriste za upravljanje njegovog programa preko kojeg možete da podesite vreme, datum, željenu temperaturu, vreme zalivanja, željenu vlažnost zemljišta tokom zalivanja i jačinu osvetljenja.

Uzgajanje biljki u plasteniku se vrši preko 3 procesa:

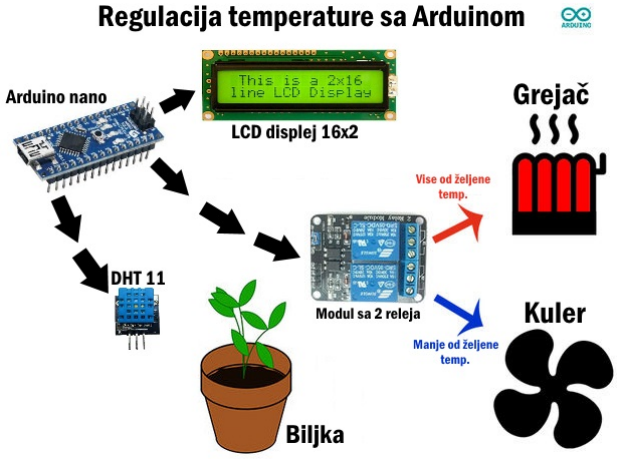
1. Regulacija temperature
2. Automatizovanog zalivanja
3. Veštačkog osvetljenja



Slika 2. Hardver od plastenika.

## III. REGULACIJA TEMPERATURE

Regulacija temperature u plasteniku se vrši pomoću senzora DHT11 koji meri temperaturu izolovane prostorije koji preko Arduina utiče na temperaturu, sto mozete videti na slici 3.




Slika 3. Proces regulisanja temperature u stakleniku.

Veljko Tomašević, student Visoke škole elektrotehnike i računarstva strukovnih studija, Vojvode Stepe 283 Beograd. E-mail: veljkot6@gmail.com.

Ivan Golubović, student fakulteta Singidunum, Danijelova 32, Beograd. E-mail: igolubovic5@gmail.com

Uz pomoć programa koji upravlja Arduino možemo uključiti ili isključiti regulaciju temperature i podesiti željenu temperaturu.[3]

U slučaju da je temperatura manja od odredjene, Arduino će preko releja da uključi grejač koji će postepeno povećavati temperaturu sve dok ne postigne željenu, gde ce se zaustaviti. Ako je temperatura veća od određene, onda će se upaliti kuler koji će izbacivanjem toplijeg vazduha smanjiti temperaturu prostorije.

## IV. AUTOMATIZOVANO ZALIVANJE

U plasteniku se nalazi rezervoar u kome je senzor nivoa vode. Uz pomoć tog senzora možemo zaštiti pumpu tako što ćemo je isključiti kada padne ispod određenog nivoa i indikacijom da rezervoar treba da se napuni.




Slika 4. Proces navodnjavanja u stakleniku.

Kada ima dovoljno vode u rezervoaru i kada je podešeno vreme za zalivanje isto kao vreme na satu onda pumpa kreće da radi i dovodi vodu do zemlje u kojoj se nalazi senzor vlažnosti zemlje, kao što možemo videti na slici 4. Njega koristimo tako sto poredimo odgovarajuću vrednost vlažnosti zemlje sa vrednosti kojom dobijamo od njega. Ako te vrednosti iste ili ako senzor pokazuje veću vrednost od zadate, tada staklenik prestaje sa zalivanjem.

U slučaju da zalivanje traje sat vremena bez prestanka, Arduino će prekinuti sa zalivanjem jer je to vrednost koja se smara mnogo većom od potrebne i dace nam indikaciju da nešto u stakleniku nije u redu.

U stakleniku se nalazi i regulator napona koji je direktno povezan sa pumpom za vodu i preko kojeg mozemo podesiti snagu pumpe i na taj način olakšati proces zalivanja.

## IV. VEŠTAČKO OSVETLJENJE

U stakleniku se nalazi fotootpornik koji koristimo kao senzor da bi znali koliko je osvetljena njegova spoljašnjost.

Tu vrednost možemo fino podesiti na programu preko menija kojim upravljamo preko dugmića kod displeja. Na taj način upravljamo jačinom svetla koje dobijamo preko led trake čiju boju možemo da biramo uz pomoć prekidača koji se nalaze u staleniku.

## V. I2C LIQUID CRYSTAL DISPLEJ

Da bi videli potrebne informacije koristimo 16x2 Liquid Crystal displej pored koga su tri dugmeta uz pomoć kojih upravljamo meni.

Kada se meni ne koristi automatski se prebacuje na početni ekran koji prikazuje sve bitne vrednosti koje dobijamo od senzora i koji se isključuje pritiskom bilo kojeg dugmeta.

Korisimo tri dugmeta a to su levo(smanjiti), desno(povećati) i izabrati(*select*), jedno dugme smanjuje selektovanu promenljivu, drugo povećava a treće menja selekciju.

## VI. ZAKLJUČAK

U ovom radu prikazan je uspešno realizovan projekat automatizovanog staklenika. Rad je koristan za vežbe i kao model za implementiranje sličnog sistema u pravi plastenik. Prilikom testiranja je utvrđeno da uspeva da održi izolovan prostor u odgovarajućim uslovima za uzgajanje biljki, i time smanji rizik od njihovog propadanja. Model ovog plastenika se u slučaju pravih, realnih plastenika može poboljšati preciznijim senzorima, jačim grejačem, adekvatnom pumpom i boljim sitemom za hlađenje, ali za potrebe ovog projekta on ispunjava sve sto je za njega prvobitno osmišljeno.

## ZAHVALNICA

Realizaciju ovog projekta je u finansijskom smislu doprinela i pomogla ,,Alatnica Barović" i JKP Toplovod iz Obrenovca. Zbog toga smo uspeli da od ovog projekta napravimo to što sada jeste i jako smo im zahvalni.

## LITERATURA

[1] Razvoj hardversko/softverske podrške za upravljanje plastenicima, Ivan Stefanović Fakultet tehničkih nauka, Čačak, 2016 http://www.ftn.kg.ac.rs/download/SIR/SIR%20Ivan%20Stefanovic%208242015.pdf

[2] Brian W. Evans, Arduino Programming Notebook, 2007, http://playground.arduino.cc/uploads/Main/arduino_notebook_v1-1.pdf

[3] Trajković, S., Milanovčić, M., Tuneli plastenici i staklenici, http://www.fb.org.rs/en/upload/content/Tuneli_plastenici_i_staklenici.pdf

# Elektronski „peščani“ sat

Đurđina Petković, Milica Bogićević, Tijana Petković

*Sadržaj* - U ovom radu prikazan je detaljan opis elektronskog „peščanog” sata, kao i njegov princip rada.

## I. UVOD

Elektronski „peščani“ sat je namenjen deci koja treba da peru zube sve dok se ne upali zujalica. Nakon uključenja zujalice, dete prestaje sa pranjem zuba.

### *A.Električna šema uređaja*

Šema prikazana na slici 2 i sastoji se od pet PNP bipolarnih tranzistora. Njihova svrha u šemi je da struja jednako protiče do dioda i time one svetle jednakim intenzitetom. Diode su postavljene u obliku peščanog sata, koristili smo jedanaest LE dioda u crvenoj, zelenoj i žutoj boji. Anode su povezane na napajanje od 5V, a katoda preko otpornika od 220Ω na kolektor tranzistora. Baza je povezana preko otpornika od 1kΩ na arduino. Emitori tranzistora su povezani na masu. Dva prekidača sa pull-down otpornikom povezana su na arduino na pin broj 10 i pin broj 11. Bazer je povezan na pin broj 9.

### *B.Izgled uređaja*

Uređaj se sastoji od pet segmenta. Prvi segment čine tri diode koje su zelene boje i one označavaju start. Sledeća tri segmenta sastoje se od žutih dioda koje označavaju da je u toku proces pranja zuba. Peti segment se sastoji od tri diode crvene boje koje označavaju zadnjih 30 sekundi i kraj odbrojavanja.

Prvi i peti segment se sastoje od tri diode preko otpornika povezanih na tranzistor, a zatim preko otpornika na arduino. Drugi i četvrti segment imaju dve diode isto povezane kao i u prethodnom redu, dok treći segment ima jednu diodu.

Pritiskom na taster počinje odbrojavanje. Prvi segment se uključuje samo prvih 30 sekundi, pri čemu poslednje tri sekunde samo treperi i gasi se. Zatim, svaki naredni segment isto funkcioniše, dok ne isteknu punih dva i po minuta. Nakon gašenja poslednjeg segmenta uključuje se bazer i on radi, dok se ne pritisne drugi taster čime se gasi sat. Ponovnim pritiskom na prvi prekidač se aktivira ceo sat i kreće ponovno odbrojavanje.

Đurđina Petković, Milica Bogićević i Tijana Petković su sa smera Elektronske komponente i mikrosistemi, Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: djurdjinapetkovic@gmail.com, milica.bogicevic@elfak.rs, tijanapetkovic58@gmail.com

## II. OPIS KORIŠĆENIH KOMPONENATA

U ovom radu koristili smo sledeće komponente:

**-LE diode**. U kolu ima 11 LE diode. Od toga su tri zelene, tri crvene i pet žute.

**-PNP bipolarni tranzistor.** U kolu se nalaze pet PNP bipolarnih tranzistora

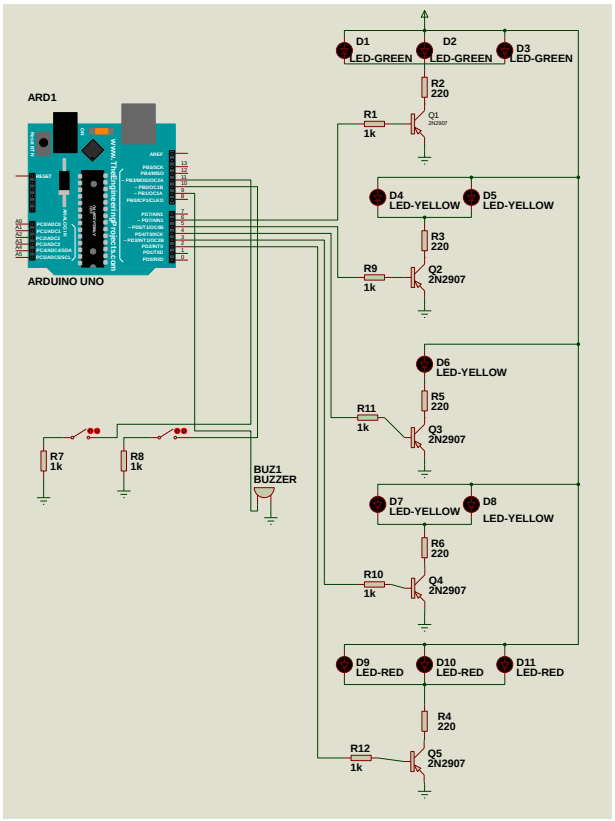
**-Otpornik**. U kolu se nalaze pet otpornika od 220Ω i sedam otpornika od 1kΩ

**-Bazer-zujalica**

**-Prekidači**

**-Arduino uno**

Na slici 2 je prikazan izgled elektronskog „peščanog“ sata u praktičnom obliku, povezan sa arduinom.

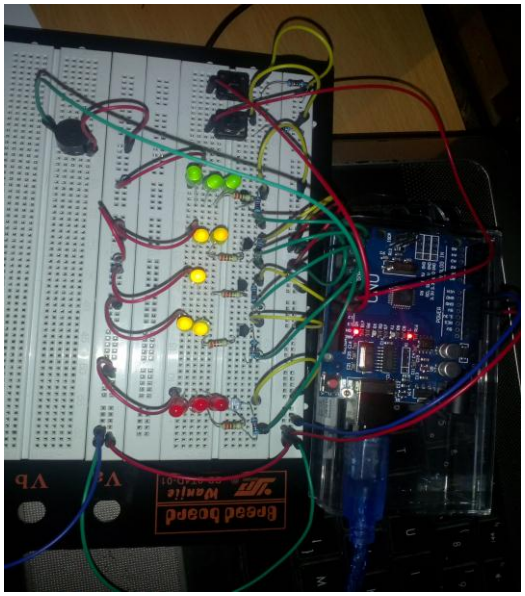



Slika 1. Šema izrađena u Proteus 8 Professional.

## III. ANALIZA DOBIJENIH SIGNALA

Na slici 3 prikazan je rezultat dobijen na osciloskopu koji predstavlja naponske nivoe signala koji se dobijaju na odgovarajućim izlazima. Na osnovu dobijenih signala utvrđuje se ispravnost rada opisanog uređaja. Kanal CH1

označava žute diode u drugom segmentu. Pikovi se javljaju kada dioda treperi. Kanal CH2 označava crvenu diodu u zadnjem segmentu. Pikovi se javljaju prilikom treperenja diode, a po završetku treperenja dioda se isključuje i tada se javljaju pikovi na CH4 koji predstavlja rad bazera. Kanal CH3 označava rad bipolarnog tranzistora, dok kanal CH4 označava rad bazera.



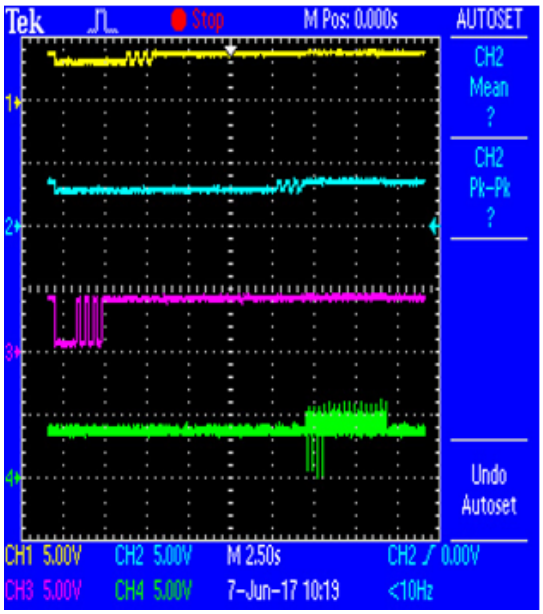
Slika.2. Praktični izgled „ peščanog“ sata.

## IV. KOD PROGRAMA

U ovom poglavlju se nalazi detaljni kod za efekat. uno kojim se pokreće kolo. Arduino hardver se programira koristeći programski jezik. Arduino ploču čine 8-bitni Atmel AVR mikrokontroler sa pripadajućim komponentama koje omogućavaju programiranje i povezivanje sa drugom elektronikom.

```
#define BR_RED      5     // 5 redova
#define KORAK       3000 // 30 sekundi
#define KORAK_KRAJ  2000  // Vreme za koje
treperi red
#define PODKORAK    250   // Korak treperenja
#define POCETAK_PIN 10     // Taster za pocetak
#define ZAVRSETAK_PIN 11     // Taster za
zavrsetak
#define BUZZ_PIN    9     // Pin za buzzer
#define BUZZ_PERISP 127   // Period ispunjenosti
pistanja
#define DEBOUNCE    20    // Vreme za cekanje
ocitavanja tastera 20ms
boolean pocetak=false;
unsigned char red=0,podkorak=0;
unsigned long vreme=0;
int redovi[BR_RED]={6,5,4,3,2};
void setup(){
  // put your setup code here, to run once:
  for(int i=0; i<BR_RED; i++)
    pinMode(redovi[i],OUTPUT);
  pinMode(POCETAK_PIN,INPUT);
  pinMode(ZAVRSETAK_PIN,INPUT);
  digitalWrite(LED_BUILTIN,HIGH);
  Serial.begin(9600);m
}
void loop(){
  // Treperenje zadnjih KORAK_KRAJ sekundi
  if(pocetak){
    if(millis()>vreme+(unsigned
long)(red+1)*KORAK-KORAK_KRAJ+(unsigned
long)(podkorak+1)*PODKORAK){

digitalWrite(redovi[red],(podkorak%2)?HIGH:LOW);
      //Serial.println(podkorak);
      podkorak++;
    }
    // Gasenje reda
    if(millis()>=vreme+(unsigned
long)(red+1)*KORAK){
      digitalWrite(redovi[red],LOW);
      red++;
      podkorak=0;
```




Slika.3. Očitani signali na osciloskopu.

## V. ZAKLJUČAK

Elektronski peščani sat je idealan zbog praćenja preporučenog minimalnog vremena pranja zuba. Naime, dok se zubi peru, peščani sat „ističe“ i na taj način se vreme pranja zuba ograničava. Kako je oralna higijena vrlo bitna treba na vreme učiti kako se pravilno peru zubi, a zatim i koliko dugo je potrebno prati zube za zadovoljavajuci efekat.

## LITERATURA

[1] http://pdf1.alldatasheet.com/datasheet-pdf/view/15072/PHILIPS/2N2907.html
[2] http://mikroelektronika.elfak.ni.ac.rs
[3] http://mikroelektronika.elfak.ni.ac.rs/lit_sr.php

# Drajver kostantne optičke snage za lasersku diodu

Nikola Stojković, Jovana Čukić

*Sadržaj* - U ovom radu prikazana je realizacija jedne moguće konfiguracije drajvera za održavanje konstantne optičke snage laserske diode, koja je otporna na promenu električne struje, kao i na termičke varijacije. Koncept drajvera je najpre potvrđen korektnim simulacionim rezultatima, a realizovani prototip je testiran najpre uz pomoć optokaplera koji je pokazao očekivane rezultate, a zatim i korišćenjem odgovarajuće laserske diode.

## I. UVOD

Drajver za održavanje konstantne optičke snage predstavlja upravljački modul, ili kolo, sa konstantnom optičkom snagom koji podržava modul laserske diode koji ima mogućnost merenja izlazne optičke snage. Ovakav uređaj obično ima ugrađeni potenciometar za kontinualno podešavanje izlazne snage lasera, pin sa povratnim signalom srazmernim optičkoj snazi, prekidački ulaz i monitorski strujni izlaz za posmatranje struje laserskog modula. Ovakvi uređaji su kompaktni i mogu biti ugrađeni i prilagođeni gotovo svakom dizajnu. Laserska dioda je veoma osetljiva komponenta koja se veoma lako može oštetiti ili uništiti. Jedna mogućnost oštećenja laserske diode je dovođenje prevelike struje kroz nju. Svaka specifikacija, za bilo koju vrstu laserske diode, sadrži podatak o maksimalnoj radnoj struji. Prekoračenje ove struje oštetiće lasersku diodu. Da bi se to izbeglo, u kolo uključujemo odgovarajući ograničavač struje. Korisnik određuje maksimalnu postavku i struja se ograničava tako da ne prelazi postavljenu maksimalnu vrednost. Zbog vrlo nelinearnih laserskih strujnih karakteristika koje zavise od temperature spoja, obično nije dovoljno da se primenjuje određeni konstantni napon, umesto toga električnu struju treba stabilizovati automatskim podešavanjem primenjenog napona. Prilikom povećanja temperature uređaja potreban je niži napon za istu struju. Ako bi se umesto toga držao napon konstantnim, struja bi mogla znatno porasti što bi dodatno povećalo temperaturu spoja.

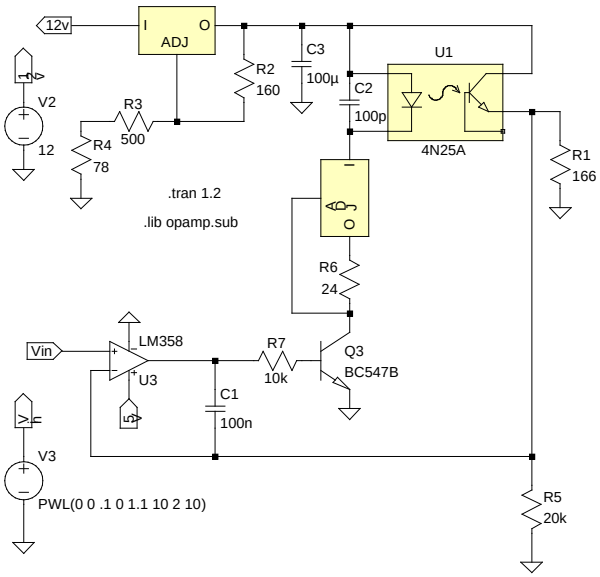
## II. ZAHTEVI I KONCEPT DRAJVERA

Prilikom projektovanja drajvera za održavanje konstantne optičke snage na leserskog diodi, treba obratiti pažnju na vrstu konfiguracije koja se može odabrati. Za realizacuju ovog kola koristili smo kao najvažniju komponentu kolo stabilizatora napona preko koga smo u posebnoj konfiguraciji zapravo ograničili struju koja dolazi u laser i time sprečili trajno oštećenje lasera. Osnovi deo kola predstavlja drajver konstantne struje. Ovaj koncept zahteva da se konstantnom održava ne struja koja protiče kroz lasersku diodu, već fotostruja na monitorskoj fotodiodi koja je sastavni deo laserskog modula. Kako je fotostruja srazmerna optičkoj snazi lasera, održavanje fotostruje konstantnom odgovara održavanju konstante optičke snage lasera. U kolu smo koristili operacioni pojačavač koji ima funkciju da omogući povratnu spregu tj. vraća deo signala sa izlaza monitorske fotodiode na invertujući ulaz kako bi se ova fotostruja održala konstantnom. U principu, ovakvo kolo predstavlja jedno jednostavno rešenje postavljenog problema.

Za napajanje lasera smo projektovali kolo za stabilizaciju napona koje se može podešavati. Za testiranje ovog kola koristili smo optokapler zbog karakteristika koje su slične karakteristikama laserskog modula, osim razlike u ceni. Pre korišćenja drajvera za održavanje snage na laserskoj diodi u nekom većem sistemu, treba proveriti da li mogu nastati problemi zbog nepravilnog uzemljenja. Čak i ako je isti pin priključen na uzemljenje na različitim lokacijama, može doći do problema zbog „uzemljene petlje".

Poželjno je da se komponente u prototipu napajaju naponom od 5 V, što se može dobiti stabilizacijom napona iz ispravljača koji generiše napon od 12V. Projektovanje i realizacija prototipa sistema izvedeni su u više faza, pri čemu se neke od njih delimično preklapaju: prikupljanje literature, izrada koncepta sistema, simulacije u SPICE-u i podešavanje parametara, izrada prototipa, testiranje prototipa, merenje karakteristika.

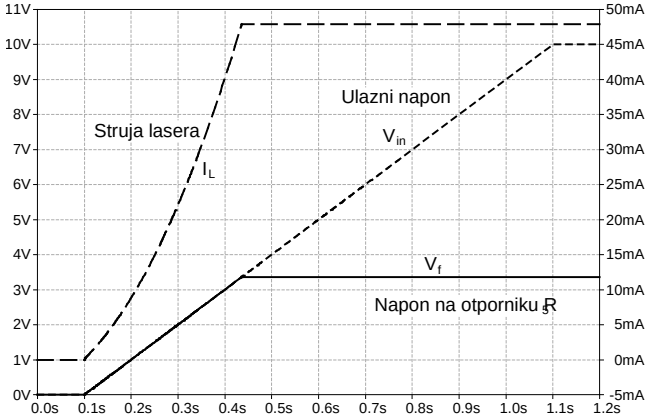
Koncept sistema i model koji smo koristili za simulaciju prikazan je na slici 1.




Slika 1. Simulaciona šema prijemnika nacrtana u LT-SPICE-u.

Nikola Stojković, Jovana Čukić, su student treće godine osnovnih akademskih studija na Elektronskom fakultetu Univerziteta u Nišu, A. Medvedeva 14, 18000 Niš, Srbija. Email: stojkovicnikola95@gmail.com, jovanacukic955@hotmail.com

Kolo povratne sprege deluje u smeru izjednačavanja ulaznog napona i napona na opterećenju fotodiode. Ukoliko se ulazni napon poveća, povećaće se i struja kroz laser, sve dok fotostruja na opterećenju ne stvori pad napona koji je jednak ulaznom naponu. Na ovaj način se ulaznim naponom efektivno kontroliše fotostruja. Rezultati simulacije kada se na ulaz dovede linearno rastući napon, prikazani su na slici 2. Vidi se da kolo za ograničavanje struje kroz laser ograničava ovu struju na maksimalno 50 mA. Kada struja lasera dostigne ovaj nivo, fotostruja više ne prati ulazni napon koji može da se i dalje povećava. U delu kada je povratna sprega funklcionalna, vidi se da postoji veoma dobra linearna korelacija između ulaznog napona i fotostruje. Sa druge strane, struja lasera je nelinearna funkcija ulaznog napona, što ovaj drajver za konstantnu optičku snagu lasera uspešno automatski kompenzuje.

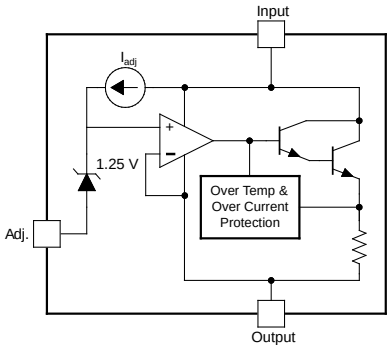



Slika 2. Karakteristične vrednosti u simulaciji.

## III. IZABRANE KOMPONENTE I NJIHOVE KARAKTERISTIKE

LM 317 je tro-terminalni regulator pozitivnog napona koji može snabdevati više od 1.5 A preko opsega izlaznog napona od 1.25 V do 37 V. Zahteva samo 2 spoljna otpornika za podešavanje izlaznog napona. Uređaj ima tipičnu liniju regulacija od 0.01% i tipičnu regulaciju opterećenja 0.1%. Uključuje ograničenje struje, toplotno preopterećenje, bezbednu zaštitu radnog područja. Zaštita od preopterećenja ostaje funkcionalna čak i ako je „Adj“ terminal isključen.

Kao pojačavač, zbog povoljnog odnosa cene i kavaliteta, koristi se integrisano kolo LM 358 N koje se sastoji od 2 visokofrekventna operaciona pojačavača, koja se mogu napajati sa jednim ili 2 izvora napajanja. Za izradu ovog prototipa se jedan deo integrisanog kola može koristiti kao operacioni pojačavač, a drugi ostaje neupotrebljen. Da ovaj neupotrebljni operacioni pojačavač ne bi pravio probleme u vidu haotičnog ponašanja, on je povezan kao jedinični bafer, a na pozitivni terminal je doveden neki fiksni naponski nivo.




Slika 3. Funkcionalni blok diagram integrisanog kola LM 317.

Optokapler je komponenta koja se sastoji od LE diode i fototranzistora integrisanih u jednom kućištu. Optokapler predstavlja komponentu sa svetlosnom spregom između ulaza i izlaza. Signal sa ulaza izaziva emisiju svetlosti LE diode. Ova svetlost predstavlja pobudu fototranzistora, tako da se na izlazu pojavljuje odgovarajući signal. Ulaz i izlaz su međusobno galvanski izolovani. Galvanska izolacija čini optokapler pogodnim za primenu u svim uređajima kod kojih korisnik može doći u dodir sa potencijalno opasnim nivoima signala. U našem projektu, optokapler ima istu funkcionalnost kao laserski modul, osim cene koja je znatno niža jer se optokapleri proizvode masovno. Zbog toga se u realizaciji najpre upotrebljava optokapler umesto lasera, kako bi se uverili da kolo ispravno funkcioniše i da je struja ograničena onako kako je i predviđeno.

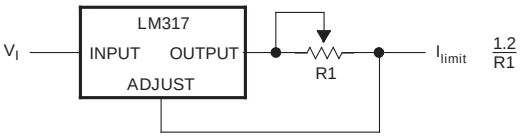
## IV. PRAKTIČNA REALIZACIJA I TESTIRANJE PROTOTIPA

Šema sa slike 2 se može realizovati na raster-ploči dimenzija 7x9cm. Preko ispravljača dovodimo napon napajanja od 12V, koji se stabilizuje pomoću stabilizatora napona. Na izlazu iz stabilizatora imamo napon od 5V koji napaja ostatak kola. Na ulazu u kolo se nalazi potenciometar preko koga postavljamo željenu vrednost struje u optokapleru, odnosno u laserskoj diodi. Kada se u kolu koristi optokapler koji treba da služi kao zamena laserskoj diodi zbog sličnih karakteristika i kako struja kroz nju ne bi smela da bude veća od maksimalne struje od 30mA uvodimo otpornik čija je otpornost određena Omovim zakonom tj. zavisi od odnosa ulaznog napona i maksimalne fotostruje struje u optokapleru:

$$R_1 = \frac{U}{I} = \frac{5V}{30mA} = 166\Omega \qquad (1)$$

Pri ovoj vrednosti opterećenja u kolu fotodiode, pri struji od 30 mA se dobija maksimalna vrednosti napona, tako da je ova struja ograničena upravo na maksimalnih 30 mA. Ovaj otpornik stoji u kolu kada se koristi optokapler za testiranje, međutim, potrebno ga je odspojiti kada se drajver koristi sa laserom. Razlog za ovo je što je fotostruja u laserskom modulu znatno manja, pa je vrednost otpora opterećenja znatno viša. U slučaju lasera smo predvideli otpornik vrednosti 20 kΩ.

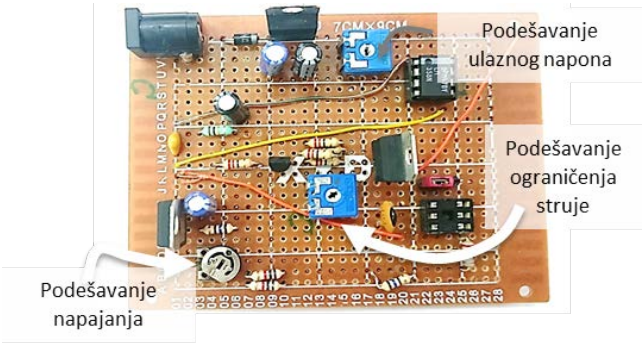
Pored ovog linearnog ograničenja opsega regulacije optičke snage lasera, dodali smo ograničavač struje koji ne dozvoljava da struja kroz laser (ili pobudni deo optokaplera) bude veća od 50 mA.




Slika 4. Prikaz ograničavača struje u kolu.

Vrednost otpora otpornika u kolu iznosi

$$R_6 = \frac{1.2V}{50mA} = 24\Omega \quad (3)$$

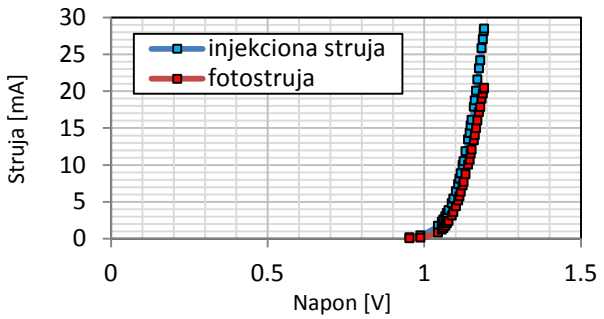



Slika 5. Realizovani prototip na raster ploči.

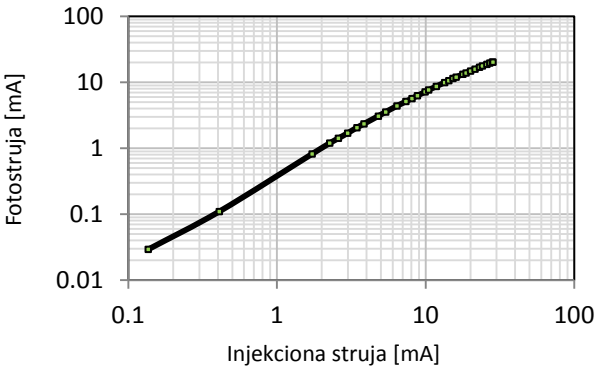
## V. MERENJE KARAKTERISTIKA PROTOTIPA

Nakon realizacije prototipa na raster-ploči, izvršena je provera ispravnog funkcionisanje kola pomoću osciloskopa. Zatim smo izvršili merenja karakterističnih velličina, i prikazali njihove međusobne zavisnosti, i time vizualizovali ispravnost drajvera sa optokaplerom, a zatim i sa laserskim modulom.

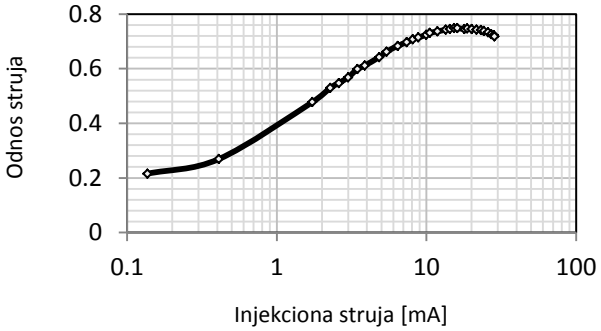
Sa ovih grafikona slici 6. možemo da uočimo kako pri naponu od oko 0.95V nema nikakvih strujnih porasta, a zatim sa naponom od 1.2V dobijamo nagli porast struje i maksimalnu vrednost injekcione struje od 30 mA i fotostuje od 20mA. Ove karakteristike odgovaraju datim specifikacijama optokaplera 4N25 koji smo koristili.




a) strujno-naponska karakteristika optokaplera

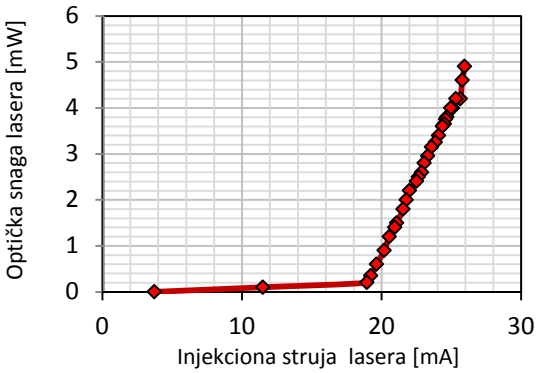



b) prenosna strujna karakteristika

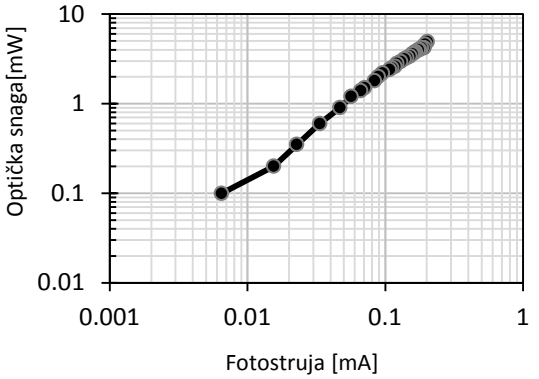



c) strujni odnos

Slika 6. Izmerene karakteristike optokaplera prilikom testiranja.




a) optička snaga u zavisnosti od injekcione struje




b) optička snaga u zavisnosti od fotostruje

Slika 7. Izmerene karakteristike optokaplera prilikom testiranja.

Kada smo se uverili da drajver funkcioniše ispravno sa optokaplerom, uklonili smo otrpornik $R_1$ i nastavili testiranje sa laserskim modulom ADL-65103TL. Strujno naponska karakteristika je veoma slična, s razlikom napona praga koji je iznosio oko 2.04 V. Značajna je i karakteristika optičke snage u zavisnosti od injekcione struje koja je prikazana na slici 6.a. Linearna zavisnost fotostruje i izlazne optičke snage očigledna je kada se pogleda slika 6.b. Nakon ovih merenja zaključili smo da laserski modul ispravno funkcioniše i da drajver može da precizno podesi izlaznu snagu oslanjajući se na informaciju o fotostruji samog modula.

## VI. ZAKLJUČAK

Zadatak projekta je bio projektovanje i realizacija drajvera za održavanje konstantne optičke snage na laseru. Izvršena je analiza zadatka i obavljeno je projektovanje prototipa čija je validnost proverena uz pomoć simulacije. Prototip je zatim praktično realizovan korišćenjem raster ploče i diskretnih komponenti. Njegovo funkcionisanje je potvrđeno izvršenim merenjima.

## ZAHVALNICA

Ovaj rad napisan je kao deo studentskog projekta na ispitu Optičke komunikacije, na smeru Telekomunikacije, Elektronskog fakulteta Univerziteta u Nišu. Kompletna realizacija projekta izvedena je pri laboratoriji za komunikacione sisteme. Izrada ovog projekta ne bi bila moguća bez pomoći prof. dr. Dejana Milića i članova laboratorije 304.

## LITERATURA

[1] Dejana Milić „Optičke telekomunikacije“, Niš 2017
[2] John M. Senior, "Optical Fiber Communications: Principles and Practice", Pearson Education, 2009.
[3] LTSpice IV, software za simulaciju sa dokumentacijom
[4] http://www.linear.com/designtools/software/
[5] Laser Components, "AlGaInP Visible Laser Diode ADL-65103TL", kataloški podaci, dostupno na:
[6] https://www.lasercomponents.com/fileadmin/user_upload/home/Datasheets/arima/655nm/adl-65103tl.pdf
[7] Kataloški podaci za integrisano kolo LM358 dostupno putem interneta: http://www.alldatasheet.com/datasheet-pdf/pdf/3067/MOTOROLA/LM358.html
[8] Kataloški podaci za integrisano kolo LM317 dostupno putem interneta: http://www.ti.com/lit/ds/symlink/lm317.pdf

# Alarmni sistem za sprečavanje krađe na poljoprivrednim parcelama

Nemanja Radić

*Sadržaj* – Cilj projekta je sprečavanje krađe i štete na poljoprivrenim parcelama. Predstavljeni alarmni sistem detektuje infracrveno zračenje od životanja ili ljudi na poljoprivrednim parcelama i SMS porukom obaveštava vlasnika parcele da je na istoj primećeno prisustvo ljudi ili životinja.

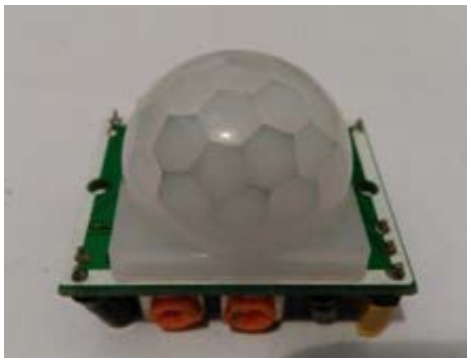
## I. UVOD

Alarmni sistem čini GSM modul, pasivni infracrveni senzor (PIR senzor) i Arduino Mega 2560. Radom alarmnog sistema upravlja Arduino Mega 2560. Princip rada se zasniva na detekciji infracrvene energije životinja ili ljudi na poljoprivrednim parcelama pomoću PIR senzora. Ako PIR senzor detektuje određen nivo infracrvenog zračenja uređaj vlasniku šalje SMS poruku da je primećeno prisustvo životinja ili ljudi na poljoprivrednoj parceli gde je alarmni sistem postavljen. Vlasnik može da upravlja radom alarmnog sistema pomoću SMS poruke. Primenom ovog uređaja, vlasnici parcela trenutno znaju da li postoje nedozvoljena kretanja ljudi ili životinja na svojim parcelama. Predstavljeni alarmni sistem može da se primeni kao zaštita od provala i na drugim mestima, ne samo na poljoprivrednim parcelama.

## II. ALATI I METODIKA RADA

### *A. Hardver*

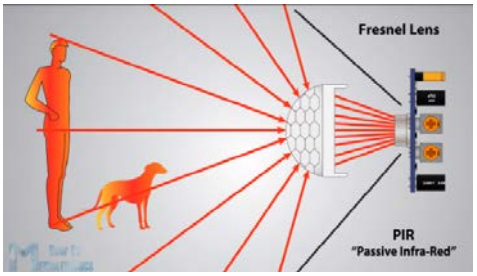
Fizički deo alarmnog sistema čini PIR sensor HC-SR501 [1] (Slika 1.), GSM modul SIM800L [2][3] (Slika 2.), Arduino Mega 2560 [4], baterijsko napajanje 2x4,5V (3R12) i prekidački izvor napajanja. Uloga GSM modula je komunikacija između alarmnog sistema i vlasnika putem SMS poruka; kontrola rada (uključenje, isključenje alarmnog sistema) i obaveštavanje vlasnika kada je PIR senzor detektovao infracrveno zračenje, odnosno kada je primećeno neželjeno prisustvo ljudi ili životinja na poljoprivrednim parcelama. Arduino Mega 2560 komunicira sa GSM modulom preko serijskog interfejsa. S obzirom na prirodu i mesto primene uređaja, napajanje celog uredjaja je baterijsko. GSM modul napaja se sa +5V. Napajanje GSM modula izvedeno je sa DC/DC prekidačkim konvertorom 9/5V. Za detekciju kretanja korišćen je PIR senzor HC-SR501 koji detektuje infracrveno zračenje. Upotrebljeni PIR senzor je malih gabarita, jeftin, ima malu potrošnju i jednostavan je za upotrebu. PIR senzor omogućava podešavanje nivoa detekcije infracrvenog zračenja i podesivo odlaganje reakcije. Senzor na sebi poseduje Frenelovo sočivo koje fokusira infracrveno zračenje (Slika 3) [5].



Slika 1. PIR senzor HC-SR501.




Slika 2. GSM Modul SIM800L.



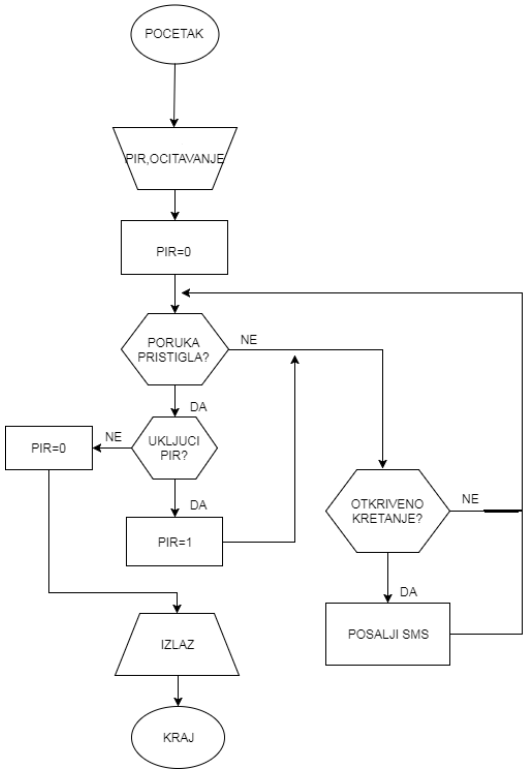

Slika 3. Frenelovo sočivo na PIR senzoru [5].

### *B. Softver*

Za programiranje mikrokontrolera korišćen je besplatan program Arduino a programski jezik je Arduino C/C++ [6], koji ima mogućnost serijske komunikacije sa kontrolerom bez drugih dodatnih programa. U programiranju rada GSM modula korišćena je biblioteka <Sim800l.h>. Softver u mikrokontroleru Arduino Mega

Nemanja Radić, obrazovni profil elektrotehničar računara, Tehnička škola Loznica, Trg Jovana Cvijića 3, 15300 Loznica, Srbija, E-mail: nemanjaradic1999@gmail.com

2560 vrši prijem SMS poruke i na osnovu dekodovanog sadržaja uključuje ili isključuje PIR senzor. Na osnovu stanja na pinu mikrokontrolera isti šalje poruku vlasniku da je detektovan određen nivo infracrvenog zračenja na PIR senzoru. Na slici 4. prikazan je algoritam rada sistema.




Slika 4. Algoritam.

## III. ZAKLJUČAK

S obzirom da je alarmni sistem je malih dimenzija lako je izvodljivo njegovo instaliranje i vizuelno maskiranje. U kombinaciji sa više PIR senzora koji zajedno pokrivaju širi prostorni ugao alarmni sistem mogao bi u potpunosti onemogućiti prilazak alarmnom sistemu i njegovo onesposobljavanje. Cena alarmnog sistema je niska, naročito u poređenju sa štetom koja može da nastane na poljoprivrednim dobrima usled uticaja neželjenih posetioca. Primena alarmnog sistema nije ograničena samo na upotrebu u poljoprivredi. Može se koristiti i na mestima kao što su kuće, vikendice, skladišta, radnje, itd.

## ZAHVALNICA

Posebnu zahvalnost pri izradi ovog projekta pridajem Marti Avramović, bivšoj učenici Tehničke škole u Loznici.

## LITERATURA

[1] https://www.mpja.com/download/31227sc.pdf
[2] https://cdnshop.adafruit.com/datasheets/sim800h_hardware_design_v1.00.pdf
[3] https://github.com/cristiansteib/Sim800l
[4] http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf
[5] http://howtomechatronics.com/tutorials/arduino/how-pir-sensor-works-and-how-to-use-it-with-arduino/
[6] https://www.arduino.cc/en/Main/Software

# Projektovanje i realizacija sistema za merenje napona

Marko Mijailović

*Sadržaj* - U ovom radu je projektovan i realizovan sistem za merenje napona baziran na mikrokontroleru PIC16F684 i korišćenje softverskih alata "Proteus 8" i "Proton IDE".

## I. UVOD

Napon ili elektrostatički potencijal je najčešća izmerena električna veličina u nekom elektronskom kolu. Napon se meri pomoću uređaja koji se zove voltmetar. Voltmetar je instrument za merenje napona između dve tačke u električnom kolu. Voltmetar se vezuje paralelno sa granom kola na kojoj se meri napon. Postoje dva načina obrade i prikazivanje rezultata merenja i na osnovu toga se voltmetri dele na analogne i digitalne voltmetre. Analogni voltmetar radi na prinicipu pomeranje kretne kazaljke na skali, dok digitalni voltmetar prikazuje vrednost izmerenog napoa na LCD displeju uređaja (slika 1).

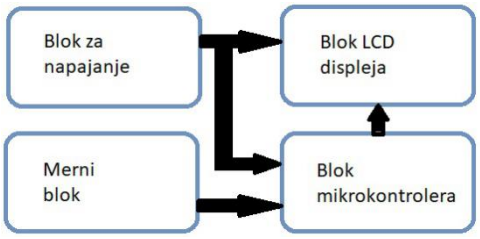



Slika 1.Digitalni voltmeta.r

## II. PRINCIP RADA

Sistem za merenje napona je podeljen u 4 funkcionalna logička bloka. Prvi blok je blok za napajanje. Realizovan je kao DC-DC konvertor pomoću linearnog stabilizator napona L7805CV [7] koji na izlazu ima 5V koliko je i potrebno za napajanje celog uređaja. Drugi blok je Merni blok koji je realizovan pomoću naponskog razdelnika, zener diode BZX55C5V1 koja služi kao zaštita kola od dolaska većeg napona i potenciometar za simulaciju merenja napona. Blok Mikrokontrolera je treći funkcionalni blok realizovan pomoću Mikrokontrolera PIC16F684, Pull-up otpornika i tastera koji sluzi za resetovanje mikrokontrolera. Četvrti blok je blok LCD displeja, potenciometra i otpornika za regulaciju kontrasta.

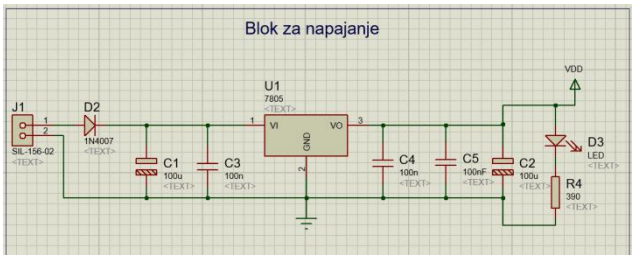
Marko Mijailović, student master akademskih studija na modulu Elektronske komponente i mikrosistemi, Elektronski fakultet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija. E-mail: markonemac@live.com




Slika 2. Blok šema digitalnog voltmetra.

### *A. Blok za napajanje*

U sistemu za merenje napona, napajanje je realizovano pomoću stabilizatora napona LM7805 spakovanog u kućište TO220 [7]. Ovaj stabilizator na svom izlazu daje 5V i struju do 1A, pri ulaznom naponu u opsegu od 7 do 24V. Na ulazu je dvopinski header konektor preko koga se dovodi DC napon. Za filtraciju kola upotrebljena su dva kondenzatora jedan keramički kondenzator od 100nF i jedan elektrolitski kondenzator od 100µF koji sprečavaju pojave oscilovanja stabilizatora. Silicijumska ispravljačka dioda 1N4007 služi za zaštitu kola od inverznog napona. Na izlazu stabilizatorskog kola stavljeno je nekoliko kondenzatora koji imaju funkciju filtera kao i bypass kondenzator koji je stavljen ispred integrisanog kola za dodatnu filtraciju i zaštitu od VF smetnji. Zelena LED dioda služi kao indikator priključenog napona a otpornik od 220Ω za ograničavanje struje kroz LED diodu. Na slici 3 je blok za napajanje.

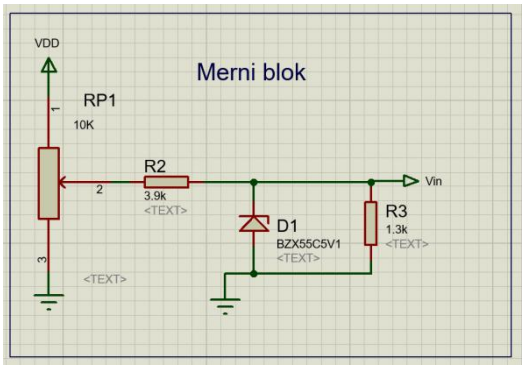



Slika 3. Blok za napajanje.

### *B. Merni blok*

Merni blok se sastoji od razdelnika napona koji čine otpornik od 3,9kΩ i 1,3kΩ. Zener dioda BZX55C5V1 za zaštitu kola od napona većeg od 5V i potenciometar od 10kΩ za simulaciju merenog napona. Kada se dovede napon na potenciometar i menjanjem otpornosti okretanjem osovine potenciometra menja se i napon koji je na izlazu

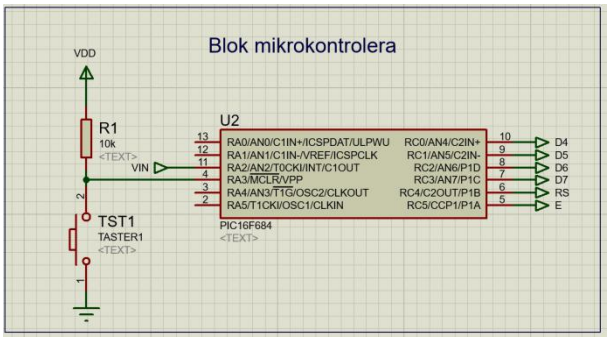
potenciometra. Time simuliramo različite vrednosti napona koje merimo. Na slici 4. prikazana je šema Mernog bloka.




Slika 4. Šema mernog bloka.

*C. Blok mikrokontrolera*

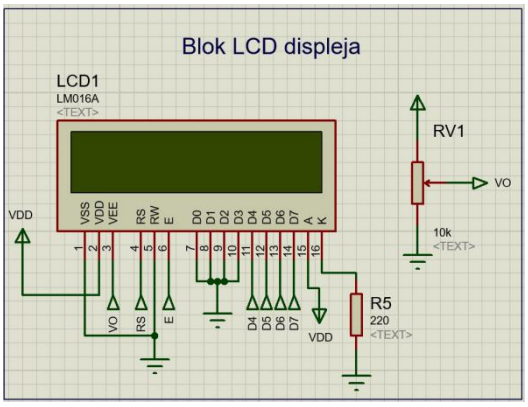
U ovom radu korišćen je mikrokontroler PIC16F684, kao što je prethodno navedeno. Za A/D konverziju iskorišćen je prvi pin Porta A (RA0). Na pinu RA3 (MCLR) je vezan pull-up otpornik od 10kΩ i taster koji služi za resetovanje mikrokontrolera. Port C mikrokontrolera je iskorišćen za komunikaciju sa LCD diplejem. Šema bloka mikrokontrolera je prikazan na slici 5.




Slika 5. Šema bloka mikrokontrolera.

*D. Blok LCD displeja*

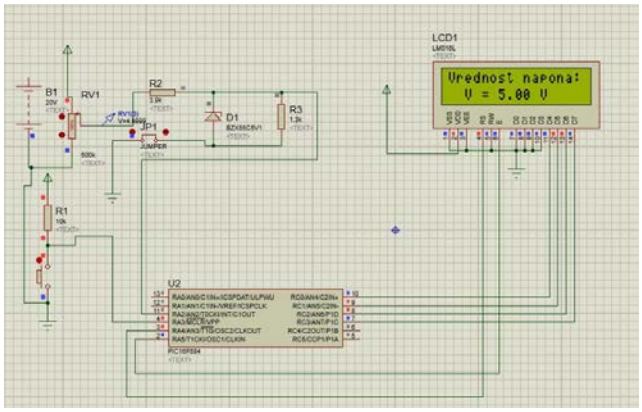
U ovom radu korišćen je monogromatski alfanumerički displej sa pozadinskim osvetljenjem. Ima mogućnost ispisivanja 16 karaktera u 2 reda. Skromne mogućnosti ovog displeja sasvim su dovoljne za prikazivanje rezultata merenje napona. Komunikacija sa displejom je uspostavljena preko 4 pina. Pinovi displeja D4 - D7 su povezani sa pinovima mikrokontrolera D0 - D3 i služe za slanje podataka. Pin RS i pin E su povezani sa pinovima RC4 i RC5. Pinovi D0-D3 se ne koriste za komunikaciju pa su kao i pin RW vezani za masu. Pinovi A (+) i K (-) služe za pozadinsko osvetljenje pa je shodno tome pin A povezan direktno na napajanje, dok je pin K vezan na masu. Pin Vee služi za podešavanje kontrasta displeja i on je povezan sa potenciometrom vrednosti 10kΩ. Električna šema Bloka LCD displeja prikazan je na slici 6.




Slika 6. Blok LCD displeja.

## III. SIMULACIJA KOLA

Kada je završeno kompletno projektovanje, odnosno izrada električne šeme sistema, za simulaciju (slika 7) u Proteusu pokreće se alat ISIS kako bi se izvršila provera funkcionalnostielektričnog kola [6].

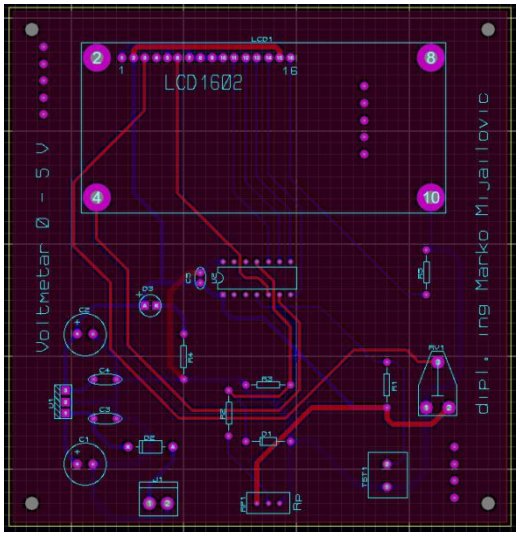



Slika 7. Simulacija voltmetra.

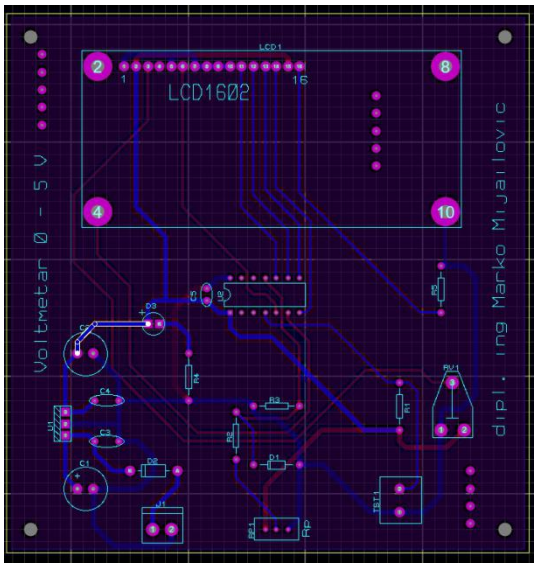
## IV. PROJEKTOVANJE ŠTAMPANE PLOČE

Za projektovanje štampane ploče je korišćen softverski alat ARES iz softverskog paketa Proteus.[6] Zbog jednostavnosti ploče, štampana ploča je dvoslojna, gde se vodovi kojima su spojeni kontakti nalaze na obe strane ploče. Najpre su prilikom projektovanja štampane ploče postavljene rupe pomoću kojih je moguće postaviti držače ili učvrstiti uređaj unutar neke kutije. Keep out odnosno deo gde ne prolaze štampani vodovi, postavljen je oko rupa. Displej, tj. konektor preko kog se displej povezuje sa štampanom pločom uređaja je na vrhu ploče, a sve ostale komponente nalaze se na donjoj polovini. Taster za resetovanje uređaja, potenciometar za simulaciju promene napona i konektor za povezivanje napona koji treba ispitati, nalaze se pri dnu ploče i lako su dostupni. Pri

projektovanju je vođeno računa da svi vodovi budu savijeni pod uglom od 45 stepeni, što je inače osnovno pravilo pri projektovanju štampanih ploča. Na obe površine ploče je razlivena masa, a samim tim je uređaj zaštićen od elektromagnetnog zračenja. Gornji i donji sloj razlivene mase je spojen metaliziranim rupama odnosno vijama.a slici 8 je prikazan gornji sloj, a na slici 9 donji sloj.

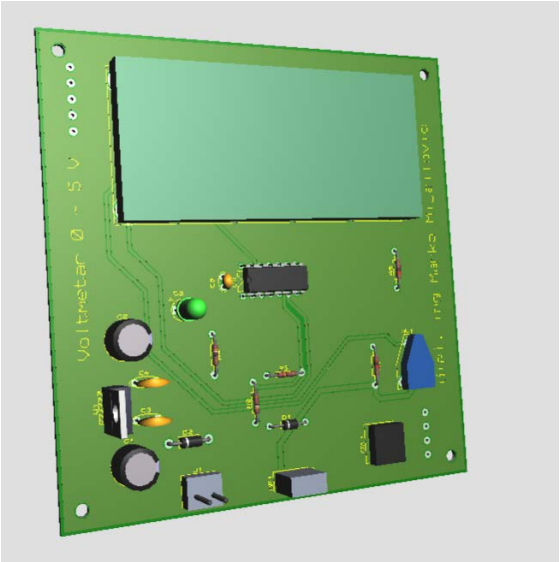



Slika 8 Gornji sloj štampane ploče.




Slika 9. Donji sloj štampane ploče.

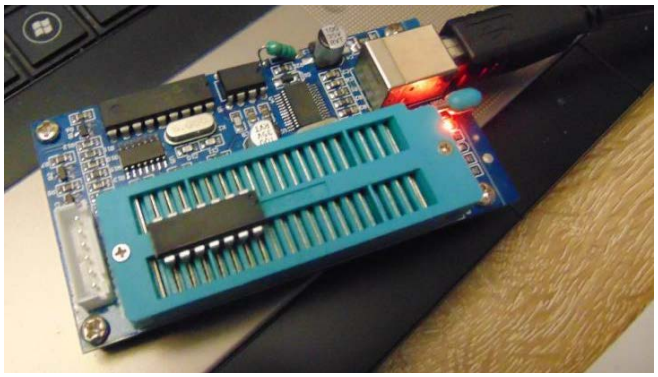
U okviru ARES alata postoji mogućnost vizuelnog prikaza štampane ploče u 3D. Na slici 10 je prikazan gornji deo štampane ploče u 3D.




Slika 10. Gornji sloj štampane ploče – 3D.

## V. REALIZACIJA

Kada je proverena ispravnost kola simulacijom, projekat je realizovan i na raster ploči. Na njoj je urađeno raspoređivanje elemenata, koji su međusobno povezani bakarnim vodovima. Nakon povezivanja uređaja, u mikrokontroleru je pomoću USB Programatora K150 (slika 11), isprogramiran mikrokontroler.



Slika 11. Programator K150 sa mikrokontrolerom PIC16F684.

## VI. ZAKLJUČAK

U ovom radu je prikazana projektovanje i realizacija jednog digitalnog voltmetra, koji može da bude jako koristan u srednjim školama i fakultetima za simulaciju merenja napona na laboratorijskim vežbama. Prikazan je najosnovniji model simulacije merenja napona, koji ima mogućnost proširenja i unapređivanja kao npr. USB port, memorijska kartica za smeštanje izmerenih vrednosti.

## ZAHVALNICA

Zahvaljujem se profesorki Doc. Dr Vesni Paunović i asistentu mast. inž. Milošu Đorđeviću na adekvatnoj stručnoj i nesebičnoj pomoći.

## LITERATURA

[1] Stojan Ristic: "Elektronske komponente" Niš, 2011 http://starisajt.elfak.ni.ac.rs/phptest/new/html/Studije/predavanja-literatura/Elektronske_komponente/ELEKTRONSKE%20KOMPONENTE-2011.pdf

[2] Đorđe Lazarević: "PROJEKTOVANJE I REALIZACIJA SISTEMA ZA MERENJE TEMPERATURE BAZIRANOG NA TEMPERATURNOM SENZORU LM35 I MIKROKONTROLERU PIC16F684" Niš, 2016

[3] https://en.wikipedia.org/wiki/Voltage_regulator

[4] Uglješa Jovanović i Marko Jovanović: "Digitalni voltmetar realizovan pomoću mikrokontrolera AT89S8253"

[5] Kenan Alajbegović: "MJERENJE ELEKTRIČNIH VELIČINA"

[6] http://geniusdevils.com/2013/03/what-is-proteus-software/

[7] https://www.sparkfun.com/datasheets/Components/LM7805.pdf

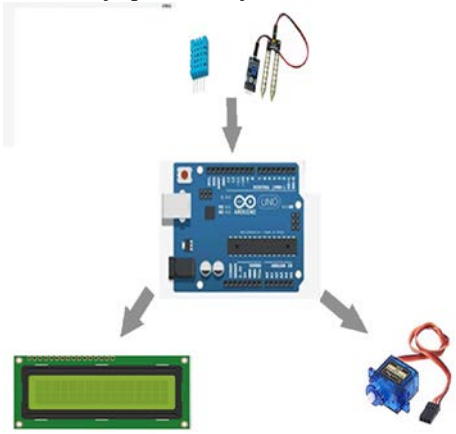
# Primena Arduino platforme u sistemu za navodnjavanje

Nikola Ristić

*Sadržaj* - Ovaj rad ima za cilj da predstavi sistem za navodnjavanje, pomoću Arduino platforme. Osnovu sistema čine: Arduino Uno, senzor vlažnosti zemljišta, senzor temperature i vlažnosti vazduha, motor (pumpa za vodu) i napajanje sa konektorima. Programiranje sistema vrši se u softverskom paketu Arduino IDE.

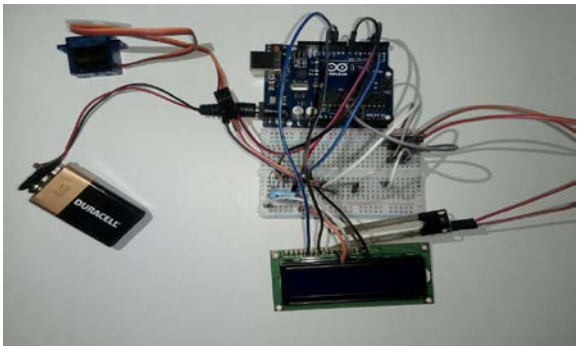
## I. UVOD

Zbog velike potrebe da se olakša proizvodnja u poljoprivrednoj proizvodnji i povećaju prinosi ovakvi sistemi su sve više zastupljeni u svetu i kod nas. Ovaj rad je jedno od mogućih rešenja. Osnovna šema sistema sistema za navodnavanje prikazana je na slici 1.



Slika 1. Osnovna šema.

Krajnji rezultat rada sistema za navodnjavanje prikazan je na slici 2 .



Slika 2. Izgled sistema.

Nikola Ristić učenik druge godine smera: Elektrotehničar informacionih tehnologija – ogled u „Tehničkoj školi Zaječar“, Knjeginje Ljubice 3-5, 19000 Zaječar. Email: risticza@gmail.com

## II. PRINCIP RADA SENZORA VLAŽNOSTI ZEMLJIŠTA

Sistem radi na principu prikupljanja informacija u razvojnom sistemu Arduino Uno [1] o vlažnosti zemljišta i temperaturi vazduha. Posle obrade prikupljenjenih informacija, rezultate prikazuje na LCD ekranu i po potrebi pokreće motor (pumpu za vodu).

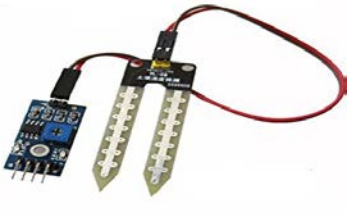
### *A. Predstavljanje komponenata i priprema za izradu*

Arduino Uno je platforma otvorenog koda. Hardver se sastoji od jednostavnog otvorenog hardverskog dizajna, Arduino ploče sa Atmel AVR procesorom i pratećim ulazno-izlaznim elementima, tačnije, na sebi poseduje mikrokontroler. Softver se sastoji od razvojnog okruženja koje čine standardni kompajler i bootloader koji se nalazi na samoj ploči. Arduino ploču čine 8-bitni Atmel AVR mikrokontroler sa pripadajućim komponentama koje omogućavaju programiranje i povezivanje sa drugom elektronikom. Na slici 3 prikazan je Arduino Uno koji je korišćen u razvoju ovog projekta.



Slika 3. Arduino Uno.

Senzor vlažnosti zemljista [2] je instrument koji se koristi za merenje vlažnosti u zemlji. Senzor radi tako sto njegove dve velike izložene podloge funkcionišu kao sonde za senzor, zajedno deluju kao promenljivi otpornik. Što je više vode u tlu znači da je bolja provodljivost izmedju podloga i rezultiraće se nižim otporom. Kalibrisanjem i izračunavanjem, ove izmerene količine mogu dovesti do merenja vlažnosti zemljišta. Senzor vlažnosti zemljišta prikazan je na slici 4.



Slika 4. Senzor vlažnosti zemljišta.

Senzor temperature i vlažnosti vazduha (DHT 11) [3] služi za dobijanje podataka o temperaturi koristi termistor (temperaturni otpornik), dok je za relativnu vlažnost vazduha zadužen je kapacitivni senzor, kome se rad zasniva na merenju električnog kapaciteta, koji se menja u zavisnosti od delovanja vlažnosti vazduha na hemijski supstrat. Senzor temperature i vlažnosti vazduha (DHT 11) prikazan je na slici 5.



Slika 5. Senzor temperature i vlažnosti vazduha.

Na slici 6. prikazan je LCD displej [4] sa sledećim karakteristikama: displej 2x16 karaktera, plavo pozadinsko osvetljenje i napajanje od 5V.



Slika 6. LCD displej.

Na slici 7. prikazan je motor [5] koji se pokreće kada vlažnost zemljišta opadne ispod zadatih granica.



Slika 7. Motor.

*B. Izrada sistema*

Povezivanje senzora vlažnosti zemljišta, temperature i LCD displeja sa Arduino Uno se vrši preko protoboard pločice. Senzor temperature i vlažnosti vazduha se povezuje tako što pozitivan kraj senzora se poveže na pozitivan kraj Arduinovog napajanja od 5V, a negativan na masu. Izlazni signal senzora se povezuje na Arduinov digitalni pin broj 7. Senzor vlažnosti zemljišta povezujemo tako što negativani kraj senzora povežemo na masu Arduina, a pozitivan kraj povežemo na pozitivan kraj Arduinovog 5V napajanja, izlazni signal senzora treba povezati na Arduinov analogni pin A0. Servo motor povezujemo tako što pozitivan kraj motora se poveže na Arduinovo 5V napajanje a negativan kraj motora se poveže na masu Arduina a digitalni pin motora se poveže na Arduinov digitalni pin broj 6. Posle povezivanja sistema, napisan je kod u softverskom paketu Arduino IDE [6]. Kod se preko USB kabla učitava u Arduino Uno. Nakon pokretanja Arduina senzori prikupljaju informacije sa senzora, prikazuju vrednosti temperature, vlažnosti vazduha i vlažnosti zemljišta na LCD monitoru. U slučaju da vlažnost zemljišta opadne ispod zadatih granica, aktivira se motor (pumpa za vodu). Motor radi dok vlažnost zemljišta ne dostigne zadate granice. Granica vlažnosti zemljišta se podešava prema potrebi.

Linija koda za očitavanje vlažnosti zemljišta na LCD ekranu i pokretanje servo motora.

```
#include<LiquidCrystal.h>
LiquidCrystal lcd(12,11,5,4,3,2)
int potPin1 = A0;    ///Analogni izlaz za senzor
vlažnosti zemljista
Int vlaznost = 0;
void setup() {
lcd.begin(16,  2);
lcd.print(“Vlaznost")
 Serial.begin(9600);
 pinMode(6,OUTPUT);    ////  Digitalni  izlaz  za
servo motor
}
void loop() {

 int vlaznost = analogRead(potPin1);

vlaznost = constrain (vlaznost, 485, 1023);
vlaznost = map(vlaznost, 485, 1023, 100, 0);
lcd.setCursor(0, 0);
lcd.print("Vlaznost");
lcd.setCursor(0,1);
lcd.print("Vlaznost :");
lcd.print(Vlaznost);
lcd.print("% ");
delay(60);

if(vlaznost<50){                ///Kada vlažnost
zemljišta opadne ispod zadatih granica  aktivira
se servo motor (u ovom slučaju servo motor se
aktivira kada je vlažnost zemljišta ispod 50%)
  digitalWrite(6,HIGH);
  delay(5*1000);
  digitalWrite(6,LOW);
}
else digitalWrite(6, LOW);    }
```

Linija koda za očitavanje temeprature i vlažnosti vazduha na LCD ekranu.

```
#include <dht.h>
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
dht DHT;

#define DHT11_PIN 7

void setup(){
  lcd.begin(16, 2);
}

void loop()
{
```

```
  int chk = DHT.read11(DHT11_PIN);
  lcd.setCursor(0,0);
  lcd.print("Temp: ");
  lcd.print(DHT.temperature);
  lcd.print("C");
  lcd.setCursor(0,1);
  lcd.print("Humidity: ");
  lcd.print(DHT.humidity);
  lcd.print("%");
  delay(1000);
}
```

## IV. ZAKLJUČAK

Celokupan projekat je osmišljen od strane učenika, a realizovan uz pomoć profesora-mentora. Bilo je potrebno primeniti znanje ne samo iz informatike već i iz elektronike, elektrotehnike i praktičnog rada sa elektro materijalima. Autor rada je zadovoljan sa krajnjim prozvodom.

Ovaj projekat pruža mogućnost za dalja istraživanja. Da bi dobili sistem koji je dobra osnova za prilagođavanje i proširivanje kapaciteta, koji uz odgovarajuća poboljšanja može služiti za regulisanje različitih parametara (npr. osvetljenja, brzine vetra i koncetracije ugljen dioksida u plasteniku). Od ovakve polazne tačke lako je napraviti sistem za zalivanje plastenika koji će biti mnogo efikasniji i praktičniji od sadašnjih sistema koji se ručno regulišu. Logičku osnovu za ovakvu primenu smo napravili, a kod realizacije bi samo trebalo prilagoditi dimenzije i rešenja dovoda vode.

## ZAHVALNICA

Autor rada zahvaljuje se Tehničkoj školi u Zaječaru radi pružanja tehničkih i materijalnih sredstava, takođe velika zahvalnost i profesorima elektroaktiva škole na pruženoj podršci, a posebno mentoru dipl. inž. el. Nikolić Dejanu.

## LITERATURA

[1] Arduino Uno (https://www.arduino.cc/en/Main/ArduinoBoardUno),2017
[2] Senzor vlažnosti zemljišta FC-28 (https://diyhacking.com/arduino-soil-moisture-sensor/),2017
[3] Svet kompjutera – DHT11 (www.sk.rs/),2017
[4] LCD displej https://www.openhacks.com/uploadsproductos/eone-1602a1.pdf, 2017
[5] Servo motor https://www.arduino.cc/en/reference/servo,2017
[6] Arduino IDE 1.6.12 (https://www.arduino.cc/en/Main/Software), 2017

# Sistem za testiranje vida i određivanje potrebne dioptrije korekcionog sredstva - OPTIS

Aleksandar Gavrić

*Sadržaj* – Od kratkovidosti i dalekovidosti tokom svog života pati više od polovine ljudi na svetu. Testiranjem koje se najčešće primenjuje zbog određenih nedostataka dodeli se pacijentu korekciono sredstvo vida pogrešne dioptrije kome se oko pacijenta prilagodi što ostavlja trajne posledice. Razvijena je platforma koja pruža rešenje ovog problema i testiranje oštrine vida, kao i umrežavanje medicinskog osoblja i pacijenta.

## I. UVOD

Poznato je da su još starogrčki mislioci pokušavali da objasne način na koji vidimo stvari oko nas. Demokrit i arapski mislilac Alhazen su među najstarijim poznatim izučavaocima oka. Već početkom 17. veka Šajner pravi prvi optički model oka nakon eksperimenta sa životinjskim okom uz zaključke Keplera koji je nešto pre toga otkrio prirodu uzroka kratkovidosti i dalekovidosti. Hajgens uspeva da napravi i prvi fizički model oka koji je kasnije u 19. i 20. veku zamenjen tačnijim modelima. Ljudsko oko, kao organ čula vida, ima funkciju stvaranja optičke slike na prijemnim ćelijama mrežnjače i njihovo pretvaranje u složeni električni signal koji registruje optički nerv povezan sa mozgom na osnovu koga se stvara konačna slika koju shvatamo kao vid. Osim biološkog načina, ljudsko oko se može opisati kao optički sklop i predstaviti optičkim nacrtom. Oko je složen i jak sabirni optički sistem, veoma sličan fotografskom aparatu. Neke aberacije oka, odnosno odstupanja od njegovog normalnog funkcionisanja, ljudi su još u 13. veku rešavali primenom sočiva čija je svojstva opisao kasnije Dekart [4].

## II. REFRAKCIJA OKA

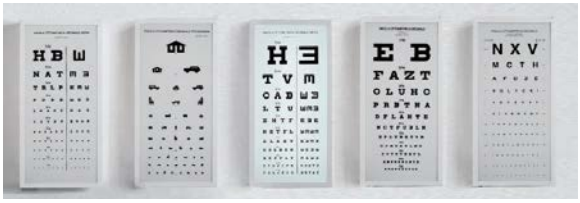
Primenom optičkih zakona o prelamanju svetlosnih zraka kroz oko govorimo o refrakciji oka. Objekat koji posmatraju oči reflektuje svetlosne zrake koji se prelamaju kroz providne delove oka koji ne sprečavaju prolazak svetlosti do mrežnjače. Optički sklop oba oka treba da omogući paralelnim svetlosnim zracima koji dolaze iz beskonačnosti da se seku na mrežnjači. Refrakcione anomalije nastaju kada je poremećen odnos optičke moći prelamanja sredine u oku kroz koje se zrak prelama i dužine oka, tako da optički sistem oka bez učešća akomodacije, odnosno prilagođavanja oka, ne može paralelne svetlosne zrake da fokusira na potrebnom mestu (žuta mrlja). U refrakcione anomalije spadaju: kratkovidost (myopia), dalekovidost (hypermetropia) i astigmatizam. (astigmatismus). Nastale probleme sa vidom na ovaj način lekari najčešće rešavaju primenom korektivnih sočiva odnosno naočara nakon podvrgavanja pacijenta na testiranje vida [1].

## III. TESTIRANJE VIDA

Pojam oštrine vida označava sposobnost oka da bistro vidi tj. da u određenom predmetu raspoznaje pojedinosti. Definicija oštrine vida je dospela u oftalmologiju iz astronomije, jer je Hoock 1705. godine utvrdio da među sto osoba nema ni jedna koja na vedrom noćnom nebu može da vidi dve zvezde kao odvojene ukoliko je rastojanje između njih manje od 1 minuta vidnog ugla. Osnovni fiziološki preduslov da se dve odvojene tačke percipiraju kao dve jeste da njihovi likovi padnu na dva fotoreceptora oka tako da između nadraženih ostane jedan nenadraženi fotoreceptor. To će se desiti u slučaju da zraci prilikom pada na mrežnjaču zaklapaju ugao od jednog minuta, što na samoj mrežnjači odgovara rastojanju od 4 mikrometara, koliko su široki sami fotoreceptori (štapići foveje na makuli oka) [2].

Oštrina vida se kvantitativno određuje merenjem sposobnosti oka da uoči sliku u fokusu na određenoj udaljenosti. Testovi koji mere vidnu oštrinu sadrže standardizovane simbole tj. optotipe koji se pokazuju pacijentu (slika 1), a prve optotipe kreirao je holandski oftalmolog Herman Snelen 1862. godine na principu minimalnog vidnog ugla od jedne minute. Ako ispitanik vidi slovo visine 9 mm sa udaljenosti od 6 metara (20 stopa) onda je njegova oštrina vida po Snelenovoj tablici 20/20. Ako slova koja bi trebalo da budu vidljiva sa udaljenosti od 6m vidi tek sa 12 metara (40 stopa), rezultat merenja vida je 20/10, i tako dalje. Razdaljina od 20 stopa odnosno 6 metara je važna jer oko praktično vrši akomodaciju samo na razdaljinama do 6 metara. Ako se neki predmet jasno vidi na toj razdaljini, on će se jasno videti i na većoj razdaljini. Postoje i drugi optotipi koji se primenjuju, npr. Golovin–Sivtseova tabla, Rosenbaumova tabla itd. Ukoliko se radi o predškolskoj deci ili nepismenim osobama za ispitivanje vidne oštrine koristi se Pfluegelova kukica „E” ili Landoltov prsten različitih veličina i za koje ispitanik dete pokazuje prstima gde je kukica okrenuta, odnosno gde je prekinut prsten [5].

Aleksandar Gavrić, student prve godine Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: agavric@elfak.rs



Slika 1. Različite table sa optotipovima.

Očni lekar određuje korekciju merenu dioptrijom koja je potrebna da bi pacijent imao stoprocentnu sposobnost vizuelnog razlučivanja sa udaljenosti od 6 metara (20 stopa). Određuje se, zapravo, koliku optičku moć treba da ima sočivo koje će da sadrže naočare. Uz takvo sočivo kao pomagalo oči kompenzuju sve greške u prelamanju svetlosti i daju jasan vid [2].

*A. Izvođenje testa za proveru oštrine vida*

Određivanje vidne oštrine vrši se sa korekcijom i bez korekcije. Čak iako pacijent ne navodi smanjenje vidne oštrine, važno je zabeležiti radi daljeg praćenja. Pri određivanju vidne oštrine korišćenjem Snelenove tablice što je najčešći metod, pacijent udobno sedi 6 metara udaljen od optotipa (direktna metoda provere vidne oštrine) ili je na udaljenosti od 3 metra, kada u ogledalu posmatra ponuđene simbole (indirektna metoda provere vidne oštrine). Ispituje se svako oko pojedinačno pa tek onda binokularno. Testira se prvo oko slabije vidne oštrine, odnosno ako pacijent ne navodi smanjenje vidne oštrine, testira se desno oko prvo. Pred okom na kojem se ne ispituje refrakcija postavlja se okluder čija je uloga prikrivanje oka tj. njegovo onemogućavanje da vidi. Ako ispitivana osoba zatvori drugo oko, dolazi do konsenzualnog širenja pupile na ispitivanom oku, što može da ima uticaj na kvalitet viđenja testa. Prvo se određuje oštrina vida bez korekcije, pa zatim sa korekcijom. Pacijent čita do najnižeg mogućeg reda i to predstavlja njegovu oštrinu vida koja se izražava kao decimalni broj. Ako je oštrina vida normalna, pacijent vidi deseti red na optotipu i to se izražava kao 1.0. Ako ne vidi deseti red znači da je vidna oštrina smanjena [5].

## IV. NEDOSTAK UTVRĐIVANJA I REŠENJE

Nekoliko optometrista je pisalo o posledicama koje nastaju zbog aproksimacije rezultata o potrebnoj korekciji ukoliko nije dovoljno testiranja sprovedeno ili ukoliko je zaključak donet na osnovu testiranja u neadekvatnom momentu. S. Galop je u Žurnalu bihejvijoralne optometrije napisao: *„Sočiva naočara mogu da stvore specifične probleme. Postoji obrazac adikcije na minusna sočiva koji se često ignoriše. Tipičnim propisivanjem dioptrije po pravilu se preopterećuje i zamara čulo vida tako da ono što je često samo prolazno stanje postaje doživotna situacija, koja se vremenom dalje pogoršava.“ [6]*

Normalno je da oči ostvaruju razliku u oštrini vida do čitave 2 dioptrije od jutra kada je čovek potpuno opušten i odmoran do večeri kada je umoran i spreman za spavanje. Dakle, kada bi se vid merio svakog sata, pojavljivale bi se razlike između svaka dva merenja. Na taj način se smanjuje verovatnoća teškog prihvatanja korekcionih sočiva koje je često praćeno bolom, umorom, nelagodnošću i, što je najbitnije, pogoršavanje stanja oka njegovim mogućim prlagođavanjem neadekvatnom korekcionom sredstvu.

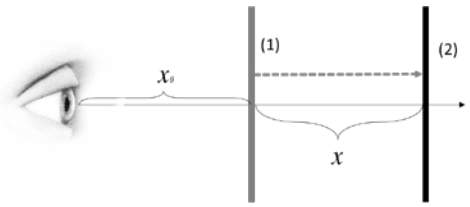
Rešenje bi svakako bilo omogućiti narodnoj masi lako dostupno testiranje koje će moći pružiti medicinskim stručnjacima dovoljno podataka tako da izbor terapije bude što uspešniji. Takvo rešenje je cilj ovog sprovedenog istraživanja. Budući da većina ljudi ima pristup pametnim telefonima ili kompjuteru, ideja je omogućiti dovoljno precizno testiranje vida baš na tim uređajima. Moguće bi bilo primeniti princip optotipa koji je i zvanično u upotrebi u svetu ukoliko bi se obezbedila tačna dimenzija prikazavinja optotipa i znalo rastojanje ispitanika od optotipa uz saradnju ispitanika koja će jednoznačno biti evidentirana.

Kreiran je program dostupan u vidu aplikacije za mobilne uređaje koji koriste Android ili Windows operativni sistem kao i web platforma kojoj se može koristiti sa bilo kog uređaja koji ima pristup internetu i pregledač web stranica.

## V. PRINCIP RADA, REALIZACIJA I RAZVOJ REŠENJA

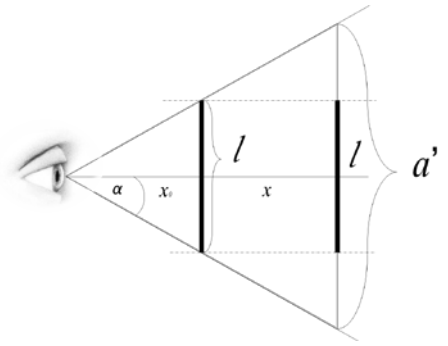
Ukoliko je poznato rastojanje od oka do ekrana koje obezbeđuje testiranje i dimenzije prikazanog polja može se proporcijom slučaj svesti na uslove koje zahteva testiranje vida optotipom. Ono šta nam se čini normalnim i na šta smo naviknuti je pojam perspektive odnosno, konkretno, predmet koji je dalje od posmatrača ukoliko bi doslovce njegove dimenzije posmatrali u odnosu na vidno polje se čini da ima manje dimenzije. Što je predmet dalje od posmatrača, čini mu se manje i obrnuto, ukoliko je rastojanje između njih manje utoliko mu se čini predmet većim. Proverimo zavisnost rastojanja između posmatrača i posmatranog predmeta i njegovih prividnih dimenzija sledećim postupkom.

Na slici 2 je ilustrovan slučaj kada se neki predmet na rastojanju $x_0$ od oka posmatrača pomeri translatorno duž pravca normale oka na predmet iz položaja (1) u položaj (2) pritom napravivši pomeraj dužine $x$.




Slika 2. Premeštanje predmeta iz položaja (1) u pololožaj (2) u odnosu na oko posmatrača predmeta.

Ukoliko primetimo karakteristične zrake svetlosti koji dopiru do oka posmatrača i to u ovom slučaju zrake ilustrovane na slici 3, gde normala oka na predmet deli predmet visine $l$ na polovine, možemo konstruisati slične trouglove i zaključiti zavisnost.




Slika 3. Premeštanje predmeta iz položaja (1) u pol. (2) ) u odnosu na oko posmatrača predmeta sa obeleženim karakterističnim zracima.

Jednačinama od (1) do (9) možemo pokazati da je prividna dimenzija predmeta koji se udaljio od posmatrača za rastojanje $x$ obrnuto proporcionalna tom rastojanju i to iskoristiti u primeni tako što umesto merenja rastojanja uređaja na kom se vrši testiranje od posmatrača možemo proporcionalno menjati dimenzije prikazanog optotipa u zamenu za njegovo pomeranje i obezbediti testiranje u mestu:

$$\tan\alpha = \frac{l}{2x_0} = \frac{a'}{2(x_0+x)} \tag{1}$$

$$a' = \frac{2l(x_0+x)}{2x_0} \tag{2}$$

$$a' = \frac{lx_0 + lx}{x_0} \tag{3}$$

$$a' = l(1+\frac{x}{x_0}) \tag{4}$$

$$\frac{a'}{l} = \frac{l}{l'} \tag{5}$$

$$l' = \frac{l^2}{a'} = \frac{l^2}{l(1+\frac{x}{x_0})} = \frac{l}{1+\frac{x}{x_0}} \tag{6}$$

$$\Delta l = l - l' = l(1-\frac{1}{1+\frac{x}{x_0}}) \tag{7}$$

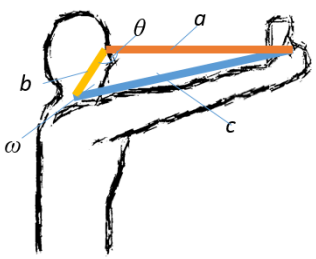
$$\Delta l = l\frac{x}{x_0+x} \tag{8}$$

Korisnik može biti udaljen od uređaja za konstantu i zadavanjem naredbi za uvećanje i smanjenje prikazanih optoptipa može testirati da li može da raspozna šta je prikazano.

Konstruisan je eksperiment kojim je dokazano da je dobijena zavisnost tačna. Eksperiment se sastoji od tri lenjira, žice izmerene dužine, i ostalih detalja. Za ram od naočara bez stakla koje nosi vršilac eksperimenta je zavezana tanka žica dužine 45 cm koja je drugim krajem povezan za konstrukciju od lenjira. Konstrukcija od lenjira se sastoji od 2 lenjira čije su kraće ivice sa strane gde počinju oznake normalne jedna na drugu i pričvršćeni su na postolje koje vršioc može lako da drži jednom rukom uz pomoć drške. Treći lenjir je pričvršćen za dršku koju korisnik drži drugom rukom i može da pomera taj lenjir paralelno jednom od lenjira sa konstrukcije duž površine onog drugog lenjira s konstrukcije. Pri vršenju eksperimenta vršioc drži konstrukciju normalno u odnosu na svoj pogled što je i osigurnano nesavitljivom žicom koja povezuje konstrukciju i ram oko njegovih očiju pod pravim uglom. Drugom rukom pomera dodatni lenjir duž površine jednog od lenjira konstrukcije stoga može očitati koliki je pomeraj izvršio u centimetrima. Oznake lenjira koji se pomera u odnosu na onaj koji je statičan (referentni lenjir) se poklapaju samo kad je rastojanje između njih 0 cm. Ukoliko vršioc odabere neku dužinu na lenjiru koji pomera može ispratiti njenu dužinu u odnosu na referentni lenjir spram pomeraja. Za proizvoljne vrednosti: dužina $l$=5 cm i pomeraj $x$=5 cm, očitana vrednost je: promena dužine $\Delta l$=0.5 cm, budući da znamo da je dužina žice odnosno rastojanje od oka do predmeta $x_0$=45 cm, zamenivši vrednosti u jednačinu (8) dobija se:

$$0{,}5 = 5(\frac{5}{45+5}) \cdot \tag{9}$$

Da bi se dobijena zavisnost primenila pri testiranju vida potrebno je da bude poznato fizičko rastojanje od oka posmatrača do uređaja za testiranje u trenutku testiranja. Stoga korisnik treba da izmeri to rastojanje i zada programu pred testiranje. Uređaj za testiranje može biti računar, pri čemu se rastojanje meri od ekrana do tačke neposredno između očiju kad je ispitanik u nepomičnom položaju. Ukoliko je uređaj za testiranje mobilni telefon najpreciznija vrednost rastojanja se dobija ukoliko je korisnik izmeri alatom za merenje (slika 4 i 5) od tačke neposredno između očiju i ispružene ruke pri čemu je duž rastojanja normala očiju na alat i zada programu opcijom koja je omogućena.




Slika 4. Način korišćenja programa u vidu aplikacije za mobilne uređaje.



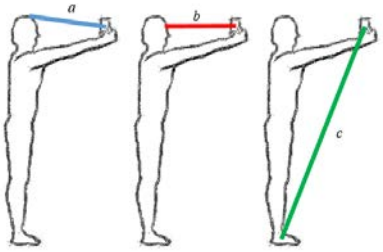
Slika 5. Merenje rastojanja korišćenjem alata.

Razmatran je i slučaj da korisnik zada samo dužinu ruke koju je nešto lakše izmeriti i računato je nad 4 ispitanika (visok muškarac, visoka žena, nizak muškarac i niska žena) da li je moguće zanemariti rastojanje obeleženo na slici 4 sa *b* ili da li se može smatrati proporcijom dužine ruke *c*. Traženo rastojanje *a* se može izračunati Kosinusnom teoremom:

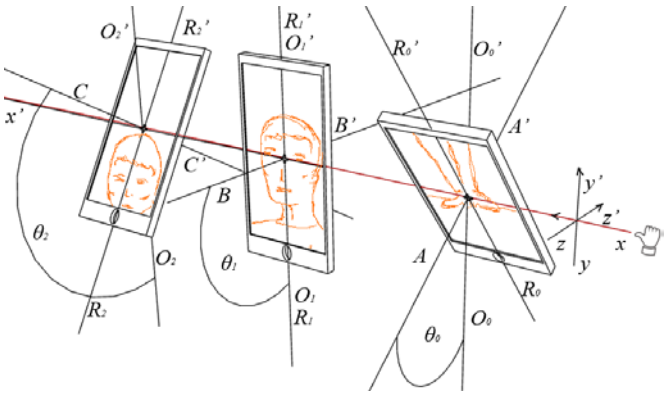
$$a = \sqrt{b^2 + c^2 - 2bc\cos\omega}\,. \qquad (10)$$

Ugao $\omega$ obeležen na slici 4 se može smatrati da je isti za sve ispitanike budući da odstupa od srednje vrednosti za $3.2^0$, ali u svakom slučaju dužina ruke ispitanika nije bila dovoljna informacija za tačno računanje rastojanja *a* stoga je ovakvo razmatranje odbačeno.

Razvijen je način merenja rastojanja uređaja za testiranje od oka posmatrača uz pomoć žiroskopskog senzora i prednje kamere kod uređaja koji ih poseduju. Budući da većina ljudi zna koliko su visoki, ispitanik zadavanjem svoje visine *l* i korišćenjem programa na telefonu može bez fizičkog alata za merenje da izmeri odnosno zada potrebno rastojanje. Korisnik drži telefon kao na slici 4, odnosno ekran u ravni sa očima pri čemu mu je ruka ispružena. Na ekranu je prikaz slike sa prednje kamere a na sredini ekrana je obeležen pokazivač u vidu ukrštenih linija (+). Korisniku je objašnjeno da treba da namesti telefon pod takvim uglovima i da obavesti kada pokazivač pokazuje na vrh korisnikove glave, zatim na tačku između očiju i na tačku na tlu na kojoj čovek stoji (slika 6a i 6b).

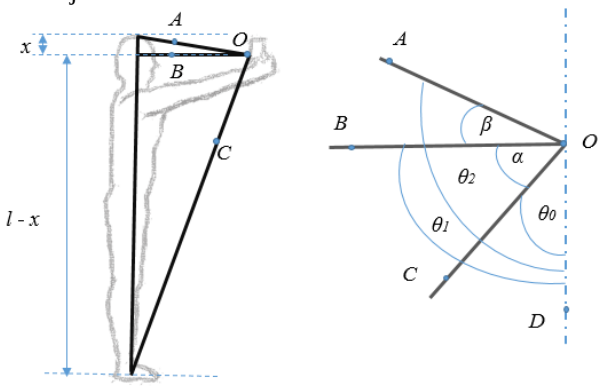



Slika 6a. Merenje rastojanja pomoću kamere i žiroskopa – karakteristični položaji rotacije telefona.



Slika 6b –karakteristični položaji rotacije telefona – uzimani su suplementni uglovi u odnosu na uglove koje izveštava žiroskop radi lakšeg uočavanja.

Žiroskop senzor pokazuje pod kojim uglom se telefon nalazi u odnosu na vertikalni odnosno horizontali položaj, pa su poznati uglovi obeleženi na slici 7 i 8 i izraženi jednačinama (12) i (13). Dovoljno je izmeriti uglove $\theta_0$ i $\theta_2$ za određivanje rastojanja *R*, a ugao $\theta_1$ se meri da bi se ukoliko nije prav sugerisalo korisniku da povisi ili spusti ruku kojom drži telefon.



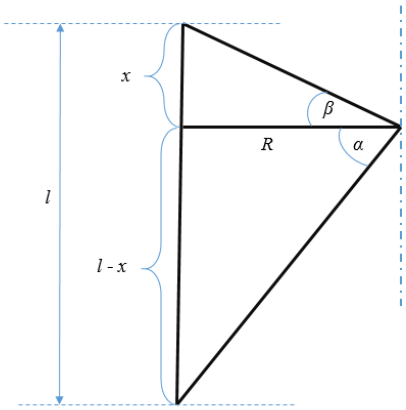
Slike 7. Karakteristične dužine i uglovi prilikom merenja.

$$\alpha \equiv \angle(BOC); \beta = \angle(AOB);$$
$$\theta_2 = \angle(AOD); \theta_0 = \angle(COD); \qquad (11)$$
$$\theta_1 = \angle(BOD) = 90°;$$

$$\alpha = \theta_1 - \theta_0 = 90° - \theta_0 \qquad (12)$$

$$\beta = \theta_2 - \theta_1 = \theta_2 - 90° \qquad (13)$$

Budući da su nam poznati sada uglovi $\alpha$ i $\beta$, kao i zadana visina *l* (slika 8) na sledeći način se može izračunati nepoznato rastojanje *x* a zatim i traženo rastojanje *R*:




Slika 8. Prilog uz jednačine od (14) do (20).

$$\tan\beta = \frac{x}{R} \tag{14}$$

$$\tan\alpha = \frac{l-x}{R} \tag{15}$$

$$R = \frac{x}{\tan\beta} = \frac{l-x}{\tan\alpha} \tag{16}$$

$$\frac{x(\tan\alpha+\tan\beta)}{\tan\alpha\tan\beta} = \frac{l}{\tan\alpha} \tag{17}$$

$$x = \frac{l\tan\beta}{\tan\alpha+\tan\beta} \tag{18}$$

$$x = \frac{l}{\frac{\tan\alpha}{\tan\beta}+1}\,. \tag{19}$$

Sada kada je poznato i rastojanje *x* iz jednačine (16) možemo proračunati traženo R:

$$R = \frac{l}{\tan\alpha+\tan\beta}\,. \tag{20}$$

Tangens ugla možemo izračunati algoritmom Maklorenovog razvoja ukoliko nemamo funkciju za njegovo računanje:

$$f(x) = \tan x = \sum_{k=0}^{\infty}\frac{f^{(n)}(0)}{k!}x^k \bmod 2 = x + \frac{x^3}{3} + \ldots \tag{21}$$

Razvijan je i način merenja traženog rastojanja uz pomoć obezbeđenog dodatka i kamere uređaja. Ukoliko bi kornisnik imao jednobojan kvadratić papira jarke boje koja se ne nalazi na licu i ukoliko bi se taj kvadratić nalazio neposredno uz oči, što se može postići korišćenjem npr. obezbeđenog okvira za naočare bez stakala koji na spojnom delu preko nosa ima pričvršćen pomenut kvadratić (slika 9), korišćenjem kamere koja snima tu konstrukciju (prednja kamera mobilnih uređaja) može se proračunati razmera stvarnih dimenzija kvadratića i njegovog lika na rastojanju R.



Slika 9. Model, realizacija i primer upotrebe konstrukcije.

Program za sliku sa kamere može korišćenjem dve petlje proći kroz sve piksele te slike i tražiti piksele koji su boje koja je rezervisana za kvadratić konstrukcije uzevši u obzir i tolereanciju odstupanja usled lošeg osvetljenja i sl. Kada program proračuna kolike su dimenzije kvadratića na slici u pikselima, budući da zna njegove stvarne fizičke dimenzije (kvadratić je dimenzija 2cm x 2cm) može izračunati razliku dimenzija na udaljenosti ali ipak ne može znati koliko je kamera umanjila lik načinom na koji beleži sliku stoga se ovakav sistem ne može uspešno upotrebiti na ovakav način za traženu primenu.

Budući da su sada poznati postupci prikupljana potrebnih informacija kreiran je aplikativni program.

*A. Opis funkcionisanja i korišćenja aplikacije*

Korisnik aplikacije može da vrši testiranje svog vida više puta tokom dana, poželjno je na svakih sat vremena, a aplikacija će rezultate obezbediti njegovom izabranom medicinskom stručnjaku koji je zadužen za njihovo praćenje. Korisnik svaki put kada odabere opciju „Novo merenje“ pristupa stranici za merenje gde potom treba da odabere da li trenutno testira levo oko, desno oko ili oba oka. Ispod tih izbornih dugmića se nalazi prikaz optotipa koji su u ovoj aplikaciji tekst odnosno smislene reči. Nasumično se pokažu tri reči koje korisnik treba da pročita. Potrebno je da zatim uvećava, ukoliko testira dalekovidost odnosno umanjuje, ukoliko testira kratkovidost, izborom odgovarajućih dugmića i da na onoj granici kada više ne može da raspozna šta piše, da znak pritiskom na završno dugme. Da bi se sprečilo da korisnik upamti reči pa da mu se čini da još uvek vidi jasno reči jer zna koje su, svaki put kada se promeni veličina teksta promene se i reči, odnosno budu prikazane nove tri nasumične reči. Izbor reči se vrši iz baze podataka koju korisnik može i da izmeni stoga se može sprovesti i usputno učenje, kao sekundarna primena aplikacije, tako što će se recimo prikazivati parovi reči na engleskom ili nemačkom i njihov prevod na srpski. Tako se može kombinovati i učenje stranog jezika uz testiranje vida. Varijante su brojne, korisnik može uneti reči koje želi da mu se prikazuju pa tako učiti recimo fizičke formule, hemijska jedinjenja, glavne gradove država, brojeve telefona najbližih i slično. Ukoliko su reči na nekom od podržanih jezika i ukoliko uređaj podržava tehnologiju HTML5SpeechSynthesis korisnik takođe može pritiskom na reč da čuje tu reč izgovorenu i tako proveri da li je uspešno pročitao. Kada korisnik više ne može da raspozna reči, jer su previše krupne ili sitne (preblizu su ili predaleko), on bira završno dugme a program računa njegovu oštrinu vida i potrebnu dioptriju i to beleži u bazu.

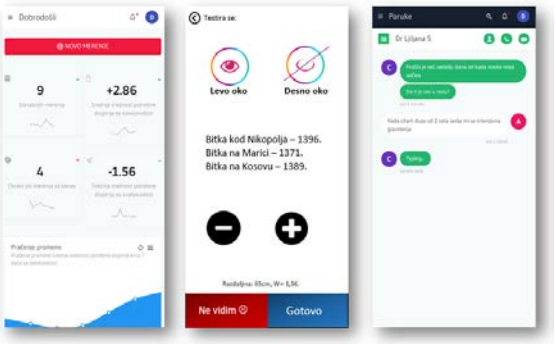
Zna se da će se vrednost menjati tokom dana a takođe postoji i polje u koje korisnik može upisati komentar relevantan za to konkretno merenje. U komentaru korisnik može upisati da je iscrpljen u tom momentu, ili da naznači da je malo spavao prethodne noći, da je u prostoriji sa slabim ili izrazito jakim svetlom, da je u pokretu i kojom brzinom ukoliko je u kolima i nije vozač i slično. Svakako će biti zabeleženo i vreme u koje je testiranje izvršeno. Na osnovu srednjih vrednosti merenja i komentara medicinski stručnjak može da isprati promenu korisnikove oštrine vida i dioptrije u periodu jutra, večeri, noći, u periodu opuštenosti i umora itd. i da revizira zaključak koji mu predlaže program. Program računa oštrinu vida i potrebnu dioptriju po sledećim jednačinama:

$$V \equiv \frac{6}{d} = \frac{6}{x_0 + x} \tag{22}$$

$$\omega = \frac{1}{x_0} - \frac{1}{d} = \frac{1}{x_0} - \frac{1}{x_0 + x} \quad . \tag{23}$$

Sa V je obeležena vrednost oštrine oka pozivajući se na proporciju optotipa da se pod zdravim smatra vid slova visine 8.75 mm na rastojanju[1] od 6 m. Pri pokretanju programa obezbeđeno je da se prikažu slova fizičke visine od 8.75mm i to se smatra početnom vrednošću. V dobijamo proporcijom referentne vrednosti rastojanja od optotipa za zdrav vid (6m) i onog na kome vidi ispitanik (*d*) [2]. Optička moć koje korektivno sočivo naočara treba da poseduje, mereno u dioptrijama se meri kao razlika recipročnih vrednosti rastojanja sa kojeg ispitanik čita i rastojanja na kome ispitanik vidi tekst[2].

Budući da su podaci poverljivi i da bi mogao isti korisnik da pristupa sa različitih uređaja, kao i da bi više korisnika moglo da koristi program na istom uređaju razvijen je sistem za kreiranje naloga odnosno registrovanje putem interneta. Takođe, osim korisnika koji testiraju svoj vid, nalog prave i medicinski stručnjaci kojima je povereno praćenje podataka, odnosno oni koji i upućuju pacijente na korišćenje aplikacije na određeno vreme. Medicinksi stručnjak može pratiti statistiku svih svojih pacijenata putem platforme ili može izabrati da mu stižu izveštaji na mejl. Korisnik-pacijent i korisnik-medicinsko osoblje mogu poslati međusobno zahtev za praćenje, pretražiti ostale javne korisnike i razmenjivati poruke. Moguće je i zakazivanje sastanaka putem platforme. Puna funkcionalnost platforme zahteva povezanost na internet ali moguće je i bez konekcije izvršiti merenja pa ih dodati prvim povezivanjem u bazu ili je prikazati u vidu dokumenta. Jezici kojima je programirana platforma su PHP, JavaScript, Java i ostalo, uz korišćenje paketa Cordova. Na slici 10 prikazan je izgled nekih strana aplikacije.




Slika 10. Izled nekih strana aplikacije.

## VI. ZAKLJUČAK

Razviće ovakve platforme, dostupne prvenstveno današnjoj omladini, pruža veliku podršku u sprečavanju loše propisanih korekcionih sredstava za vid pogotovo mladima do 21. godine do kada se oko intenzivno menja i kada mu treba prilagoditi korekciono sredstvo, a ne dozvoliti da se njegovo stanje trajno pogorša lošim izborom korekcionog sredstva. Razvijena platforma je nazvana OPTIS, od engleskih reči „Optical Power Tracking Information System“ i otvorena je za dalja unapređivanja i proširenja sa ciljem da omogući korisniku, koji ima ili sumnja da ima probleme sa vidom, adekvatnu podršku kroz umreženost sa medicinskim osobljem i programima koje će biti moguće koristiti samostalno.

## ZAHVALNICA

Zahvaljujem se profesorki fizike Emiliji Živanović na pruženoj nezamenljivoj podršci i podsticaju i profesorki fizike Snežani Zdravković na podlozi u znanju i prenesenoj motivaciji, kao i profesorki fizike Snežani Golubović.

## LITERATURA

[1] Nikola Smiljanić, Ispitivanje vidnih funkcija, Zavod za udžbenike i nastavna sredstva, Beograd,2001.
[2] K. Čupak, Oftalmologija, Jugoslovenska medicinska naklada
[3] Prim. dr Ljiljana Kovač, Institut za očne bolesti, Kabinet za kontaktna sočiva i refrakciju oka, Klinički centar Srbije, Beograd, članak na sajtu Stetoskop otvoren dana 1.8.2017. ( http://www.stetoskop.info/Sta-je-kratkovidost-4688-c92-content.htm )
[4] Wikipedia, Slobodna internet enciklopedija, https://www.wikipedia.org/
[5] Jelena Čolić, „Određivanje oštrine vida pomoću različitih vrsta optotipa“ – diplomski rad, Univerzitet u Novom Sadu, Prirodno-matematički fakultet, Departman za fiziku
[6] S. Gallup, Journal of Behavioral Optometry (5(5):115-120), 1994.

[1] Vrednost visine jednog slova u SAD-u gde je referentno udaljenje 20 stopa iznosi 8.86 mm, a u Evropi gde je ref. udaljenje 6m, iznosi 8.75mm, jer je 1m ≈ 3.28084 feet, tj. 20 stopa ≈ 6.096, gde je izvršeno aproksimiranje na 6m.
[2] Uzima se prethodna vrednost od *d*, budući da na rastojanju *d* ispitanik već ne može da raspozna tekst.

# Realizacija jednog ASK/PSK modulatora

Ana Veljanovska, Nikola Ilić, Lazar Tešić

*Sadržaj* – Postoji više načina na koji se realizuju modulatori u telekomunikacijama. Neki od njih su suštinski analogni, a digitalna modulacija se može postići dovođenjem odgovarajućih analognih oblika digitalnih signala na njihov ulaz. Postoje i modulatori koji se ne mogu koristiti kao analogni modulatori, već se koriste isključivo za digitalne signale. Prilikom realizacije ovog projekta korišćena je upravo jedna takva metoda. Modulator je realizovan kroz multiplekser na čiji ulaz se dovodi analogni signal oblika digitalizovanog pravougaonog signala. Filtarske sekcije realizovane su u RLC tehnici, dok se fazni modulator realizuje kao i amplitudski sa prethodnim obrtom faze, koji se postiže upotrebom klasičnog kola invertora.

## I. UVOD

Digitalna modulacija predstavlja nelinearnu operaciju pri kojoj se analogni signal nosioca moduliše diskretnim signalom. Procesom modulacije se digitalni ili analogni signal u opsegu niskih frekvencija konvertuje u uskopojasni analogni signal na višim frekvencijama. Originalni signal se obično naziva signalom u osnovnom opsegu, dok se modulisani signal naziva signalom u transponovanom opsegu. Postoji nekoliko prednosti modulisanog signala u pogledu prenosa: najpre, za bežični prenos visokih frekvencija su potrebne antene manjih dimenzija, u skladu sa manjom talasnom dužinom. Sa druge strane, frekvencijski opseg na visokim frekvencijama omogućava frekvencijsko multipleksiranje, odnosno simultani prenos više signala.

Takođe, digitalna modulacija ima višestruke prednosti nad analognom. U ove prednosti spadaju: znatno bolja robusnost i imunost na smetnje u kanalu, jednostavnije multipleksiranje, bolju mogućnost zaštite i kriptovanja, kao i bolju fleksibilnost.

Postoji više različitih mogućnosti za digitalnu modulaciju nosioca, a najčešće se za binarne signale koriste digitalna amplitudska modulacija (ASK – *Amplitude Shift Keying*), digitalna frekvencijska modulacija (FSK – *Frequency Shift Keying*) i digitalna fazna modulacija (PSK – *Phase Shift Keying*). Ove osnovne modulacije menjaju amplitudu, frekvenciju ili fazu u skladu sa digitalnim modulišućim signalom. Naravno postoje i višenivojske modulacije, ili kombinacije pomenutih modulacija, ali one spadaju u naprednije modulacione šeme koje imaju za cilj povećanje informacionog kapaciteta telekomunikacionog prenosa.

Ana Veljanovska, Nikola Ilić i Lazar Tešić su student master akademskih studija na Elektronskom fakultetu Univerziteta u Nišu, A. Medvedeva 14, 18000 Niš, Srbija. E-mailovi: ana.d.veljanovska@gmail.com, ilic6nikola3@gmail.com, lazartesic@gmail.com

Digitalni modulator se može, na primer, posmatrati kao 2x1 (analogni, a u mnogim slučajevima i digitalni) multiplekser koji na svojim ulazima ima dva različita kontinualna oblika analognih signala koji predstavljaju digitalne binarne signale „0“ i „1“, dok kontrolni ulaz multipleksera kontroliše ulazni digitalni signal.

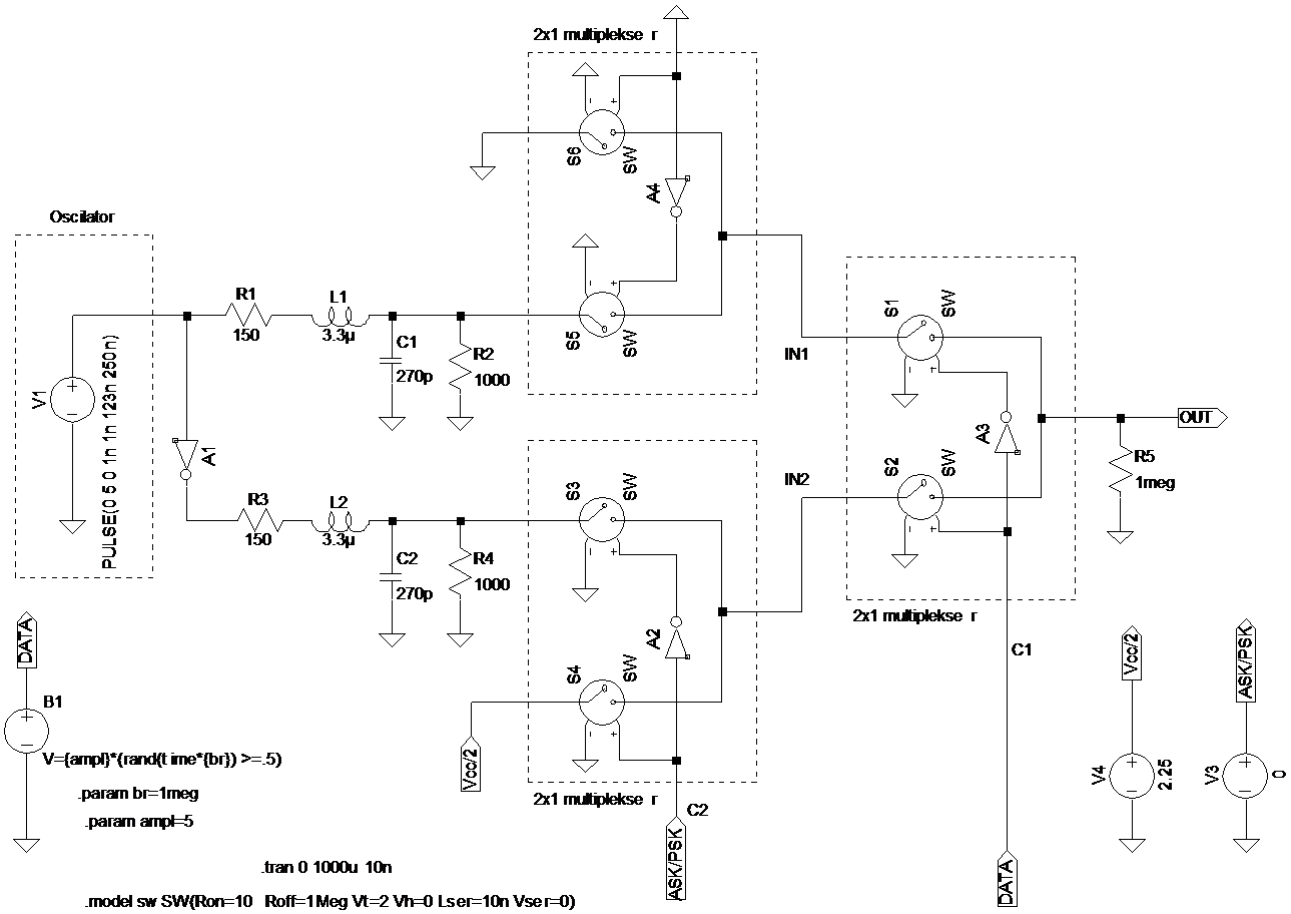
Polazne pretpostavke u realizaciji konkretnog projekta ASK/PSK modulatora su sledeće:

- Preporuka je da koncept modulatora bude jednostavan kako ne bi zahtevao previše komponenti i trošio više energije nego što je to neophodno.
- Potrebno je da nosilac bude frekvencije 4MHz, a bitske brzine koje se očekuju su ispod 1Mb/s, u skladu sa konceptom bežičnih senzorskih mreža male snage i brzine.
- Takođe je potrebno da modulator može da se koristi i kao ASK i kao PSK modulator, ali nije nužno da generiše oba modulisana signala u isto vreme.
- Realizovano kolo ima ugrađeni odgovarajući stabilizator napona kako bi moglo da se napaja preko standardnog ispravljača koji daje napon 12V.

## II. SIMULACIJA KONCEPTA MODULATORA

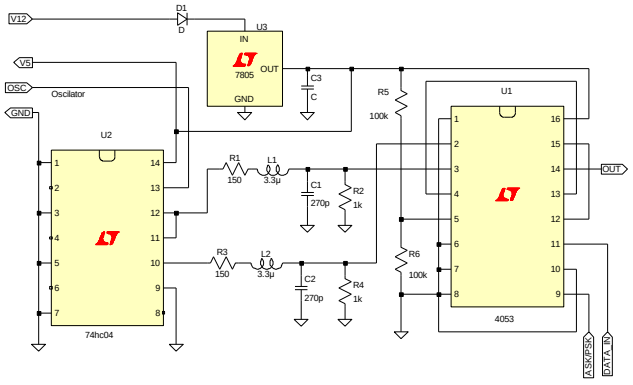
Kolo modulatora realizovano je uz pomoć dva multipleksera, 2 filtra i par invertora, sa stabilizatorom napona (Slika 1). Za simulaciju je korišćen simulator LT-Spice koji je besplatan za korišćenje, ali ima dosta mogućnosti. Komponenta koju nismo mogli da pronađemo u bibliotekama modula jeste analogni multiplekser, tako da je on zamenjen jednostavnim naponom kontrolisanim prekidačima. U kasnijoj praktičnoj relizaciji je korišćen integrisani multiplekser.

U gornjoj grani, direktno sa takt-oscilatora, signal se vodi na niskopropusni filtar granične frekvencije 4MHz. Filtar je realizovan u RLC tehnologiji, gde su parametri pojedinačnih komponenti utvrđeni računanjem kompleksne impedanse RLC kola i nalaženjem granične frekvencije. Zatim su parametri podešavani postepeno kako bi se postiglo odsecanje najbliže 4MHz. Filtriran signal se vodi na ulaz multipleksera. Multiplekser je realizovan kao dvostruko prekidačko kolo između kojih se nalazi jedan invertor. U gornjoj grani je multiplekser postavljen jedino iz razloga simetrije, kako bi kašnjenje u granama bilo jednako.




Slika 1. Izgled principske šeme modulatora za simulaciju.

Radi lakše realizacije date šeme u praksi, potrebno je prilagoditi istu karakteristikama komponenti kojih ima u slobodnoj prodaji. Kako se multiplekseri i invertori, kao i oscilatori kupuju kao gotove komponente koje sadrže više pojedinačnih elemenata, radi lakšeg povezivanja na štampanoj pločici kompletnu šemu potrebno je realizovati nešto drugačije (Slika 2).




Slika 2. Detaljnija šema povezivanja kola.

Na datoj novoj šemi vidimo da je potrebno upotrebiti jedan stabilizator napona, jedno invert kolo sa 14 pinova, kao i samo jedan multiplekser sa 16 ulaza. Uzimajući u obzir karakteristike komponenti, to je sasvim dovoljno. Filtarske sekcije se realizuju na isti način kako je i planirano u početku.

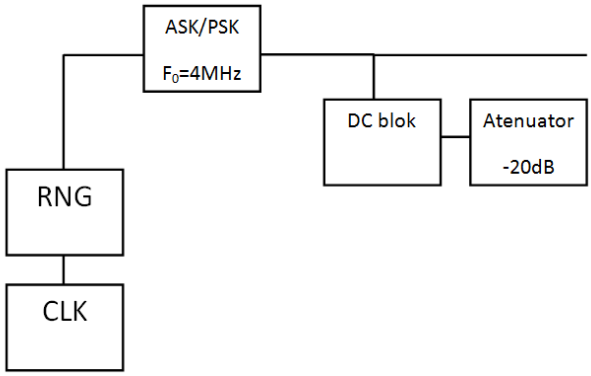
## III. REALIZACIJA REALNOG MODULATORA

Realni modulator treba u najvećoj mogućoj meri da da rezultate približno iste onima koji su dobijeni putem računarske simulacije. Radi lakšeg praćenja rada modulatora, na narednoj slici izložena je uprošćena principska šema kompletnog kola. Blok označen kao RNG predstavlja generator pseudoslučajnog niza bitova, koji simulira digitalne podatke koje dovodimo na ulaz modulatora. Oscilator koji generiše signal nosioca je ugrađen u kolo modulatora, što nije prikazano na slici 2. U svakom slučaju, za ovu svrhu je korišćen integrisani kvarcni oscilator koji ima 3 funkcionalna pina: napajanje, masu i izlaz, tako da je njegova primena krajnje jednostavna. Za takt oscilator smo koristili generator funkcija koji je dostupan u laboratoriji, i koji može da obezbedi jednostavnu promenu frekvencije, odnosno bitske brzine modulišućeg signala.

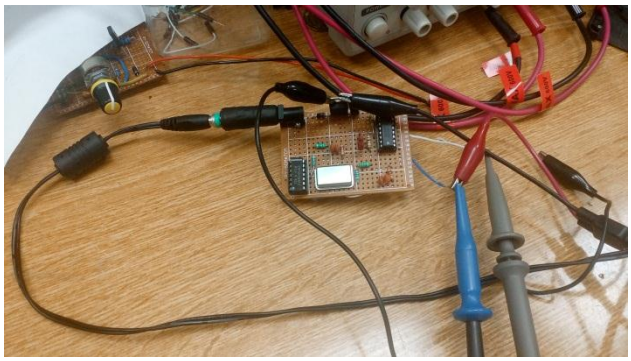
Fotografija povezivanja realizovanog kola u laboratorijskim uslovima prikazana je na slici 4.

Nakon dovođenja napajanja u kolo modulatora, na stabilizator napona paralelno je povezan unimer radi praćenja amplitude napona na izlazu istog. Ovo je prvi korak u postupku testiranja, jer ukoliko stabilizator ne daje zahtevani izlaz, može u najgorem slučaju doći do oštećenje komponenti. Sa druge strane, dijagnostika ostalih potencijalnih problema u kolu nije moguća ako ne postoji potrebno napajanje. U konkretnom slučaju nismo imali značajnijih problema u funkcionisanju. Jedini

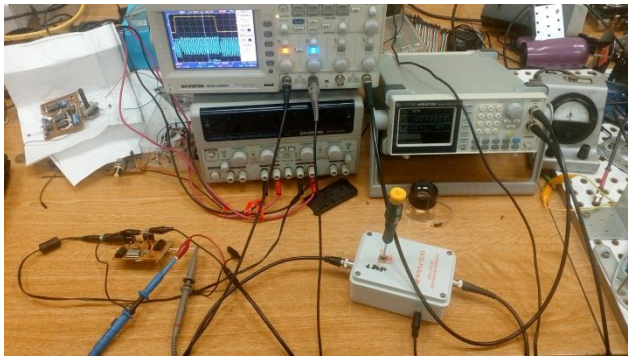
problem koji smo uočili jeste neadekvatni propusni opseg filtera. Naime, prilikom nabavke komponenti, dobijeni su kalemovi koji imaju 10 puta manju induktivnost od potrebne. Nakon identifikovanja ovog problema i zamene induktivnosti odgovarajućim vrednostima koje su bile predviđene, talasni oblici u kolu su bili potpuno u skladu sa simulacionim rezultatima.




Slika 3. Šema povezivanja modulatora prilikom testiranja.

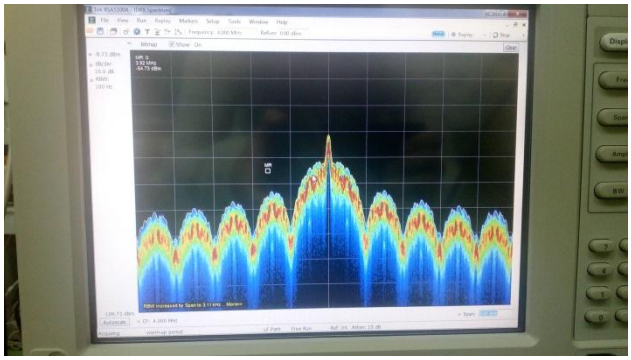


Slika 4. Realno realizovano kolo u laboratoriji.



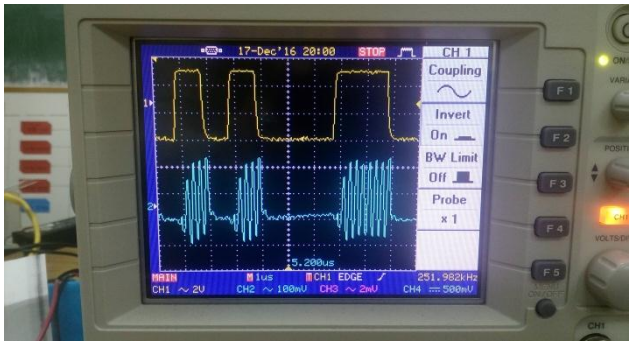
Slika 5. Izlaz modulatora povezan na laboratorijsku opremu.

Izlaz kola ASK/PSK modulatora vezuje se na analizator spektra odnosno osciloskop, radi sticanja uvida u oblik i spektar signala koji je na njegovom izlazu. Analizatori spektra su veoma osetljivi uređaji koji funkcionišu sa veoma niskim nivoima ulaznih signala. Da ne bi došlo do eventualnog oštećenja analizatora spektra signalom iz modulatora, modulisani signal je doveden na ulaz analizatora preko kola za blokiranje DC signala, kao i atenuatora koji slabi signal za 20dB. Spektar signala u transponovanom opsegu jeste osnovno merilo ispravnog rada modulatora (Slika 5). Analiza rada kola poprilično je jednostavna. Ono što treba uočiti su određene promene u spektru signala i promene u ponašanju izlaznog signala u vremenskom domenu. Počevši od ASK signala, spektar je prikazan na slici 6.



Slika 6. Spektar ASK signala sa izlaza kola.

Amplitudski spektar izlaznog ASK signala, čitan sa analizatora spektra prikazuje sve osobine jednog klasičnog amplitudski modulisanog signala. Centriran je oko frekvencije od 4MHz što odgovara uslovu zadatka. Nosilac se jasno vidi, nalazi se tačno na 4MHz u spektru i ima najviši nivo. Funkcija amplitudskog spektra odgovara matematičkoj definiciji abs(sin $x$/$x$) što je i trebalo očekivati. Ponašanje izlaza kola u vremenskom domenu prikazano je na slici 7.

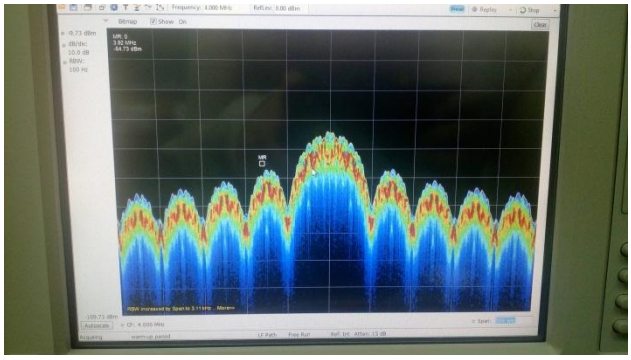



Slika 7. Izgled ulaznog digitalnog signala i izlaznog ASK signala kola na osciloskopu.

Prikaz na oscilokopu pokazuje da ulaz apsolutno odgovara izlazu kola u skladu sa uslovom zadatka. Gornji signal je ulazni i on predstavlja analogni pravougaoni signal koji oponaša digitalnu povorku impulsa. Amplitudska digitalna modulacija sva stanja „1“ prikazuje delom signala maksimalne amplitude, u vremenu, dok stanja „0“ prikazuje odsustvom amplitude signala u vremenu (amplituda je minimalna). U tom smislu izlaz, odnosno modulisani signal zaista sadrži i „verno“ prenosi takva stanja sa ulaza.

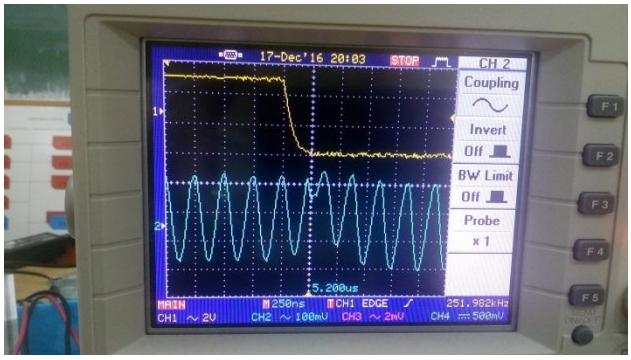
ASK signal ima pogodnu osobinu da se može jednostavno demodulisati nekoherentnom demodulacijom uz pomoć detektora anvelope. Sa druge strane, postojanje diskretnog nosioca u spektru modulisanog signala znači da energetska efikasnost modulacije nije optimalna, jer sam nosilac ne nosi informaciju o podacima koji se prenose, već se informacioni sadržaj nalazi u bočnim opsezima. Bolje performanse, po ceni komplikovanije demodulacije nudi digitalna fazna modulacija.

U nastavku biće prikazan rad „fazne grane“ kola modulatora (Slika 8).



Slika 8. Amplitudski spektar fazno modulisanog signala sa izlaza kola.

Spektar fazno modulisanog signala je centriran oko 4MHz, nema signala nosioca, shodno tipu modulacije. Poznato je da je u spektru PSK signala nosilac potisnut, što se dobro može uočiti sa slike. Što se spektra izlaza tiče i „fazni“ deo ASK/PSK modulatora funkcioniše bez problema. Kratka vremenska analiza prikazana je na slici 9, gde se vidi promena faze za 180° prilikom promene ulaznog bita.

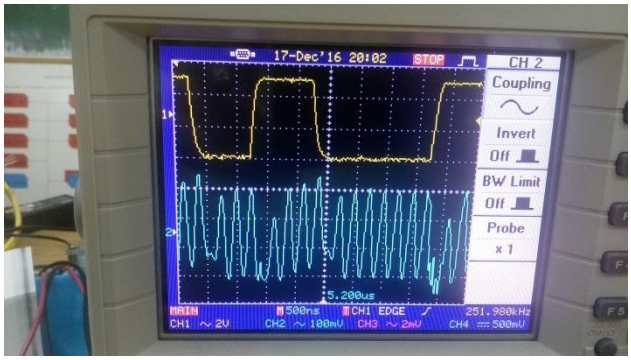



Slika 9. Fazno modulisani signal sa izlaza modulatora prikazan u vremenskom domenu.

Fazna modulacija sve promene stanja digitalnog signala koduje (prikazuje) promenom faznog stava modulisanog signala u vremenskom domenu. Prostije rečeno, menja se tačka odakle signal „počinje“. Sa slike 9 uočavamo da promenu stanja ulaznog signala (nisponska ivica takta) prati i promena faze izlaza. Na mestu gde amplituda informacionog signala pada na minimum, modulisani signal menja fazu. To je jasan znak da fazni deo modulatora radi. Slika 10 prikazuje isti slučaj na primeru prikaza dužeg dela informacione sekvence.

## IV. ZAKLJUČAK

Kompletan rad motivisan je željom da se uz minimalna ulaganja samostalno konstruiše digitalni modulator kao jedno od najtraženijih kola na tržištu telekomunikacione opreme danas. Polazeći od takve jedne ideje, računarska simulacija je morala biti jedan logičan početak ovakvog mini projekta. Poslužila je da se u softverskom okruženju ideja sprovede u delo i isproba pre samog ozbiljnijeg rada na realnim komponentama. Uz minimalan utrošak vremena softverski je ispitan budući modulator i proračunate su mu sve komponente. Ovo je jako bitno jer, danas u industriji opreme koja, shodno svrsi može biti manje ili više bitna, smanjujemo marginu greške pre nego što do bilo kakvih grešaka dođe. Shodno nameni opreme, takve greške posle finalizacije proizvoda, mogu biti preskupe, u kranjoj liniji i kobne (ponovo, zavisno od namene opreme). Takođe, šema i simulacija poslužile su kao svojevrstan vodič za realnu konstrukciju modulatora i kao „reper“ za ispitivanje rada istog. Analizom rada modulatora u vremenskom i frekvencijskom domenu, dolazimo do zaključka da kolo funkcioniše bez većih problema. Danas, u svetu integrisanih, programabilnih kola i softverski definisanih telekomunikacionih uređaja, ovakvo diskretno kolo bilo bi glomazno i verovatno komercijalno neisplativo. Međutim, ovaj rad mogao bi poslužiti kao kratko uputstvo za samostalan rad studenata, i mladih stručnjaka, na realizaciji jednog od osnovih kola u digitalnoj elektronici i telekomunikacionoj tehnici.




Slika 10. Fazno modulisan signal sa izlaza kola u vremenu.

## ZAHVALNICA

Ovaj studentski projekat je izveden pod mentorstvom prof Dejana Milića, uz pomoć katedre za telekomunikacije i laboratorije Elektronskog fakulteta u Nišu. Autori se ovim putem zahvaljuju profesoru Dejanu Miliću i katedri na ukazanoj pomoći.

## LITERATURA

[1] Garvey, B. (2013). Link Budget Analysis: Digital Modulation, Part 2. Atlanta RF Services, Software & Designs

[2] Vasiljević, V. Pavić, B. Kostić, I. (2012). Digitalne Telekomunikacije, priručnik za laboratorijske vežbe. Visoka škola elektrotehnike i računarstva strukovnih studija, Beograd.

[3] Mohammed, E. Abu Foul, K. (n.d.). ASK Modulation. Islamic Univerzity of Gaza, Faculty of Engineering, Electrical Department

[4] Digital Communication Training System ST 2137, Learning Material version 1.1

[5] Tokle, T. Serbay, M. Rosenkranz, W. Jeppsen, P. (2006). 1 Gb/s Inverse RZ - ASK - DQPSK Modulation with Low Implementation Penalty. IEEE proceedings (pg. 490-491).

# Uređaj za inteligentno upravljanje radom električne ograde

Luka Matović

*Sadržaj* - Cilj projekta je inteligentno upravljanje radom električne ograde. Praktično iskustvo ovih sistema u poljoprivredi pokazuju da postoji potreba za različitim režimima rada električnih ograda neposredno nakon instaliranja i u kasnijim vremenskim periodima eksploatacijee. U odnosu na komercijalne, često analogne, realizacije električnih ograda koje rade u ustaljenom režimu rada predstavljen način upravljanja električne ograde omogućava veću efikasnost primene i smanjuje pražnjenje akumulatora. Projekat je namenjen za upotrebu u poljoprivredi kako bi se zaštitila poljoprivredna dobra.

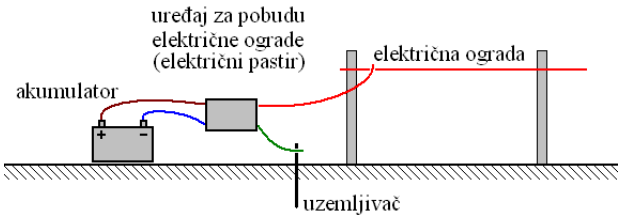
## I. Uvod

Električne ograde (električni pastiri) [1] koriste se u poljoprivredi za sprečavanje nastanka štete na poljoprivrednim dobrima od domaćih i divljih životinja. Strujnim udarima ograničenog intenziteta i vremena trajanja životinje se odvraćaju od namere da pređu prepreku. Praktična upotreba ovih sistema u poljoprivredi pokazala je da je neposredno nakon instaliranja električne ograde potrebno češće dovoditi kratkotrajne impulse napona na ogradu kako bi životinje primetile prisustvo električne ograde u svom okruženju. U kasnijim vremenskim periodima upotrebe ne postoji potreba za učestalim generisanjem impulsa čime se smanjilo pražnjenje akumulatora. Komercijalne električne ograde nemaju mogućnost samostalnog menjanja režima rada, tj. promene ritma slanja impulsa na ogradu. Cilj projekta je inteligentno upravljanje radom električne ograde za primenu u poljoprivredi kako bi se smanjila angažovanost čoveka i smanjila šteta na poljoprivrednim dobrima.

## II. Alati i metodika rada

### *A. Komponente sistema*

Komercijalne realizacije električnih ograda uglavnom koriste transformatore koji nisu zaliveni, hermetički izolovani, pa su izloženi uticaju vlage. Radni uslovi na mestima gde se u poljoprivredi koristi električna ograda karakteristični su zbog uticaja vlage (rosa, magla, kiša). Zbog toga je ova vrsta transformatora sklona proboju izolacije, odnosno otkazu rada celog uređaja. U projektu je za dobijanje visokog napona korišćena bobina [3] iz putničkog automobila. Bobina je transformator koji ne samo da je hermetički izolovan već je otporan na vibracije, hemijske i druge uticaje kakve postoje ispod haube putničkog automobila. Napajanje uređaja ostvareno je akumulatorom napona 12V. Ceo uređaj može se smestiti u PVC hermetičku kutiju iz koje priključni kablovi prolaze kroz hermetičke uvodnike čime je u potpunosti sprečen prodor vlage u kutiju, odnosno sprečen uticaj vremenskih uslova na rad uređaja. Uređaj poseduje priključne kablove za akumulator, provodnik koji se uzemljuje u odnosu na koga uređaj generiše visok napon i provodnik visokog napona koji se vezuje za električnu ogradu (Slika 1). Frekvenciju generisanja naponskih impulsa zadaje mikrokontroler ATtiny13 [2] na osnovu unetog programa i trenutnih podešavanja.




Slika 1. Povezivanje uređaja sa akumulatorom i električnom ogradom.

### *B. Upravljanje*

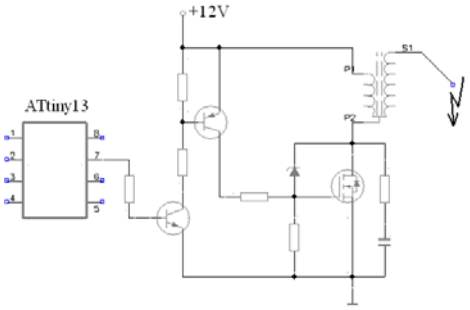
Mikrokontroler Attiny13 vrši pobudu bobine kroz kratkotrajne vremenske impulse trajanja τ. Mikrokontroler setuje pin čime uključuje MOS tranzistor čime dovodi napon na bobinu. Struja primara bobine kreće da raste a u bobini se akumulira elektromagnetna energija. Nakon proteklog vremena τ mikrokontroler resetuje stanje na svom pinu čime gasi MOS tranzistor. Energija akumulirana u magnetnom polju bobine predaje se električnoj ogradi. Promenom vremena τ može se softverskim putem menjati intenzitet električne energije predate električnoj ogradi. Nakon generisanja visokonaponskog impulsa mikrokontroler odbrojava vreme do sledeće pobude bobine definisano. Programom se definiše način rada uređaja i intenzitet električne energije poslate električnoj ogradi.

## III. Rezultati

Uređaj se napaja sa 12V. Bobina se pobuđuje preko pobudnog kola sa MOS tranzistorom kao što je to predstavljno na Slici 2. Radi zaštite MOS tranzistora zbog visokih napona u kolu je dodato snubber kolo [4] i zener

Luka Matović,obrazovni profil elektrotehničar računara, Tehnička škola Loznica, Trg Jovana Cvijića 3, 15300 Loznica, Srbija, E-mail: lukamatovic45@gmail.com

dioda postavljena između drejna i gejta MOS tranzistora. Uloga zener diode je da otvori MOS tranzistor kada napon na drejnu postane visok. Napon proboja zener diode definiše amplitudu izlaznog napona na sekundaru bobine. S obzirom na princip rada bobine, za razliku od pobuda drugih potrošača induktivnog karaktera, u datom kolu nije moguće koristiti brze zaštitne diode.




Slika 2. Upravljanje bobinom pomoću mikrokontrolera ATtiny13.

U toku testiranja uređaja snimani su naponi na drejnu MOS tranzistora kako bi se proverila bezbednost rada MOS tranzistora zbog prisustva visokih napona na bobini. Amplituda izlaznog napona na sekundaru bobine podešava se hardverski, promenom probojnog napona zener diode. Na Slici 3. prikazan je oblik napona na drejnu MOS tranzistora. Dužina autonomnog rada uređaja, do ispražnjenog stanja akumulatora zavisi od režima rada, odnosno unetog programa kao i od kapaciteta akumulatora. Hermetičko kućište uređaja obezbeđuje rad uređaja koji je nezavisno od vremenskih prilika.




Slika3. Vremenski oblik napona na drejnu MOS tranzistora (5V/div, oslabljivačka sonda 1/10, 1ms/div).

## IV. Zaključak

Upotreba mikrokontrolera doprinosi boljem iskorišćenju energije akumulatora, manjoj angažovanosti čoveka oko korišćenja električne ograde, dovijanju snalažljivosti životinja koje su sposobne da uče i smanjenju štete koje životinje mogu da načine na poljoprivrednim usevima. Svrha uređaja je primena električnih ograda isključivo u poljoprivredi kako bi se smanjile štete koje nastaju od slobodnog kretanja divljih i domaćih životinja.

## Zahvalnica

Za pomoć pri izradi ovog projekta, podršku i stručnu pomoć, zahvalio bih se profesoru Tomislavu Mijiću.

## Literatura

[1] https://en.wikipedia.org/wiki/Electric_fence
[2] http://www.atmel.com/images/doc2535.pdf
[3] https://en.wikipedia.org/wiki/Ignition_coil
[4] https://en.wikipedia.org/wiki/Snubber

# Brojač realizovan korišćenjem senzora prolaza

Dejan Anđelković

*Sadržaj* – U ovom radu prikazana je realizacija brojača realizovanog korišćenjem senzora prolaza. Fotodioda je korišćena za formiranje snopa IR svetlosti dok je fototranzistor imao ulogu detekcije prekida istog. Logika rada je realizovana korišćenjem mikrokontrolera PIC16f870 dok je diplej formiran korišćenjem dva sedmosegmentna displeja.

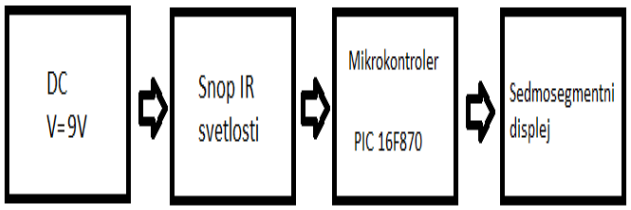
## I. UVOD

Moderno doba stvara sve veću potrebu za praćenjem tačnog broja posetilaca organizovanih dešavanja (sajma, festivala, i sl.). Tačnost takve informacije omogućava prijatnije skupove, smanjene gužve, a i statističke podatke za buduće skupove. Da bi se povećala tačnost tih informacija, realizovan je veliki broj brojača (senzora) prolaza baziranih na različitim principima rada. Cilj ovog rada jeste da se realizuje brojač prolaza baziran na principu korišćenja fotodiode i fototranzistora kao samog senzora.

Realizacijom laboratorijskih vežbi studenti se osposobljavaju da samostalno realizuju složenije električne šeme, a kasnije i da sami naprave neki elektronski uređaj.

## II. TEORIJSKI PREGLED

Na slici 1 možemo videti da se brojač prolaza sastoji od četiri osnovna bloka: blok napajanja od 9V, blok prekida IR (infra-crvenog) snopa fotodiode [1] i fototranzistora [2], blok mikrokontrolera Pic 16F870 [3] koji će obraditi ulazne signale, i od bloka dva sedmosegmentna displeja.



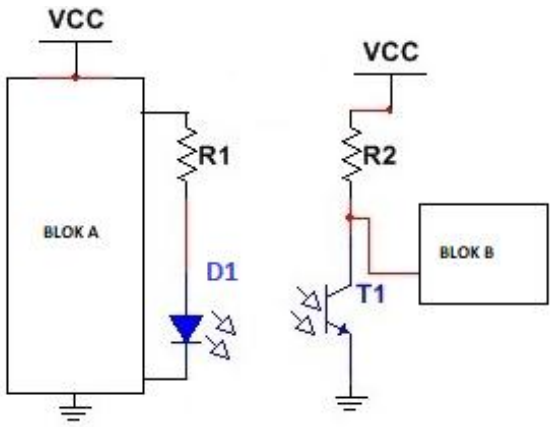

Slika 1. Blok dijagram brojača prolaza.

Detaljnije objašnjenje principa rada uređaja kao i detaljniji prikaz mikrokontrolera i električne šeme sledi u nastavku rada.

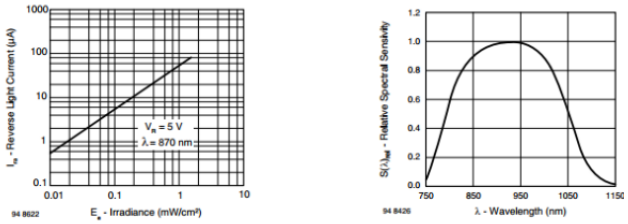
### *A. Princip rada*

Dejan Anđelković, student Elektronskog fakulteta u Nišu, modul Elektronske komponente i mikrosistemi, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: theyan.andjelkovic@gmail.com

Kako bi brojač prolaza funkcionisao, neophodan je senzor baziran na snopu IR svetlosti. Ovaj IR snop formiramo korišćenjem IR diode, a fototranzistor koristimo kao spregu. Na slici 2 je fototranzistor označen sa T1 dok je dioda označena sa D1. Princip rada je jednostavan. Dioda D1 šalje IR zrake bez prestanka. Fototranzistor T1 je konfigurisan tako da pri IR osvetljenju sa diode (zrak nije prekinut), za naše kolo daje logičku nulu. Kada se snop prekine (neki entitet ga prekine svojim prolazom), fototranzistor tada za naše kolo daje logičku jedinicu.



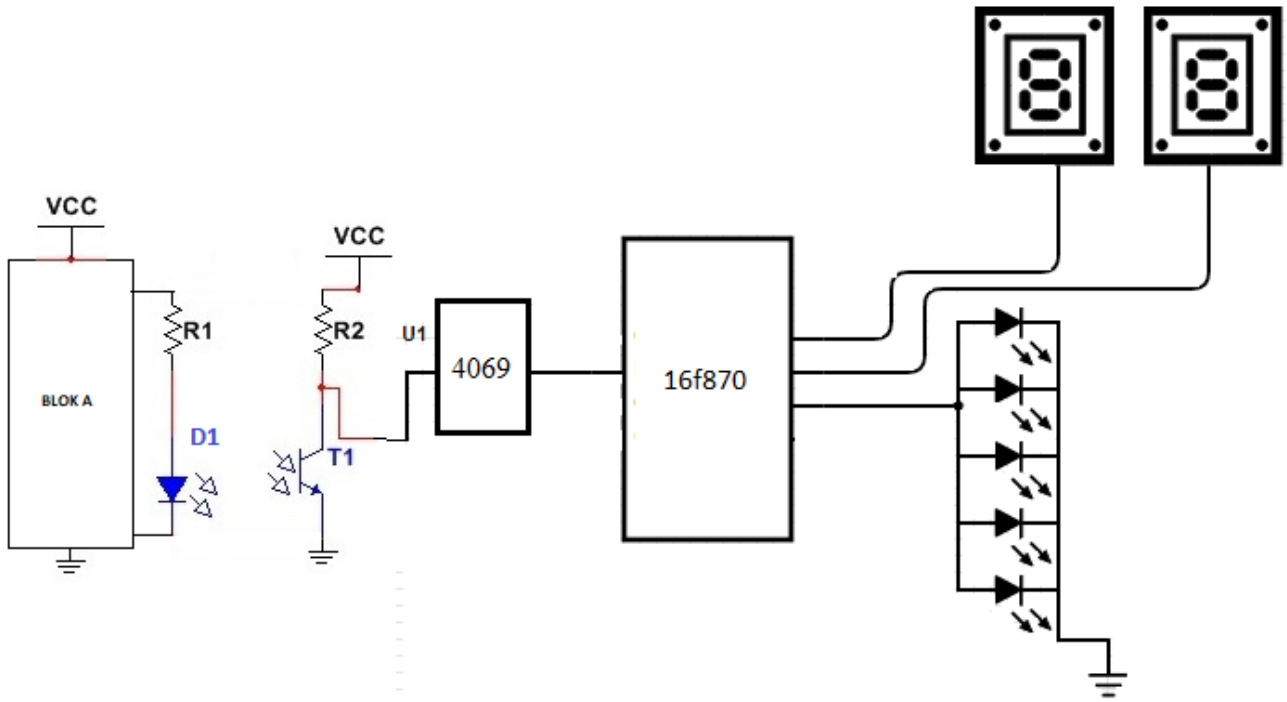

Slika 2. Šema formiranog IR snopa.



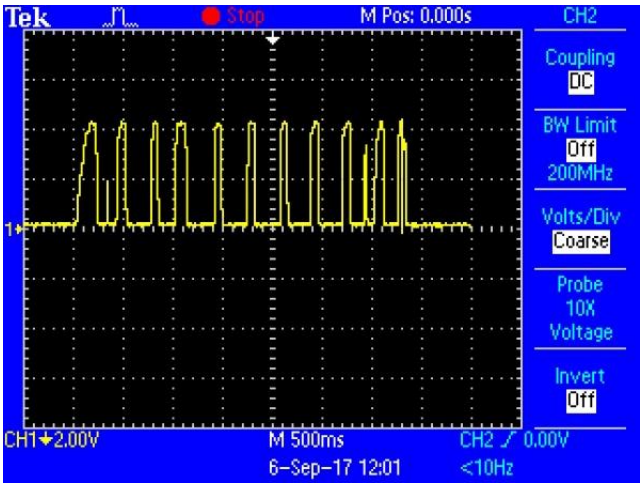

Slika 3. Odnos spektralne osetljivosti i talasne dužine IR snopa fotodiode BPV10NF.

Sa logičke strane gledano, kada nema fizičkih prekida IR snopa između fotodiode i fototranzistora, izlazni napon pojačavača je 0V i vodi se do mikrokontrolera. Mikrokontroler ne obavlja interrupt funkciju i time se brojčano stanje ne menja sa početnog (koje je podešeno da bude 00 na paru displeja). Kada dođe do prekida, sa fototranzistora se vodi 5V, što izaziva interrupt i stanje na paru displeja se menja u 01, prva dioda od 5 u nizu se isključuje i druga se uključuje, pa se oglašava i buzzer. Svakim drugim prekidom se cifra na paru povećava za 1, dioda u nizu se isključuje a naredna počinje da svetli, i isto

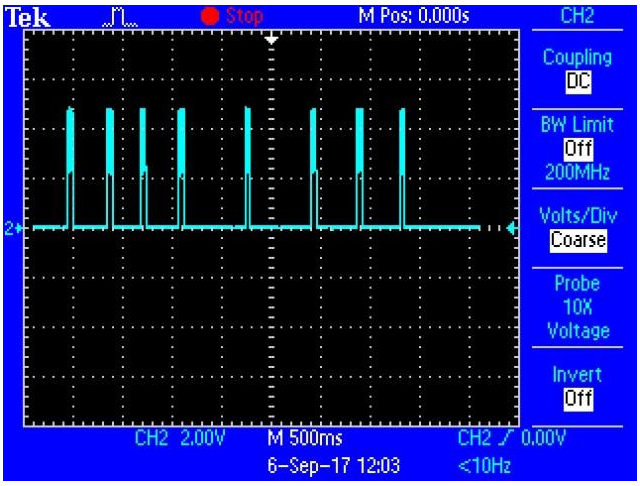
tako se svaki put buzzer oglašava. Kada brojčano stanje pređe broj 99, uređaj se resetuje i počinje ponovno od 00.




Slika 4. Električna šema brojača prolaza razvijena za bolju preglednost u toku rada.




Slika 5. Signal fototranzistora pri prekidu snopa.




Slika 6. Signal doveden buzzer-u nakon prekida snopa.

Sklop je sastavljen na protoploči i priključen na 7V, pri čemu je sonda osciloskopa TDS2024B priključena za kolektor fototranzistora. Zatim je mehanički prekidan snop IR svetlosti sa diode. Slika 5 prikazuje rezultat prekida IR svetlosti u obliku analognog signala koji se generiše na kolektoru fototranzistora.

Na slici 6 vidimo signal doveden buzzer-u, usnimljen osciloskopom. Uređaj je sniman pri naponu od 7V a sonda osciloskopa je priključena na ulaznom pinu buzzer-a. Svaki od skokova na slici se može posmatrati kao logička jedinica, a buzzer proizvodi kratak i bučan zvuk za svaku od logičkih jedinica.

## III. PROGRAMIRANJE I PODEŠAVANJE MIKROKONTROLERA

Mikrokontroler korišćen za realizaciju ovog projekta jeste firme PIC, serije 16F870. Ova komponenta predstavlja mikroprocesor srednje klase. Memorija programa je reči veličine 1Kb što predstavlja 1024 instrukcija, kako je svaka 14 bitna memorijska reč iste dužine kao jedna instrukcija. Memorija podataka (RAM) je veličine 64 bajta. EEPROM podataka je takođe 64 bajta. Mikrokontroler ima 13 I/O pina koji se moraju manualno

podesiti programskim kodom. Neki od njih su predviđeni za više funkcija, ali mogu obavljati samo jednu od tih u datom trenutku. Programskim kodom se podešava koju od tih funkcija mikrokontroler treba na njima obavljati. Neki od ovih funkcija su: eksterni interrupt, interrupt izazvan promenom na portu B, input za timer0.

Memorija mikrokontrolera je organizovana u dva bloka. Jedan blok predstavlja programska memorija a drugi blok predstavlja memorija podataka. Svaki blok ima svoj bus, tako da se oba bloka mogu pristupiti u istom ciklusu oscilatora.

Memorija podataka se dalje može podeliti u memoriju generalne primene kao i u memoriju registara specijalne funkcije (Special Funcion Registers, SFRs). Blok memorije podataka takođe sadrži i EEPROM memoriju podataka. Ova memorija je indirektno mapirana u samoj komponenti, pa indirektni određivač adrese podešava adresu EEPROM-a za čitanje i/ili pisanje. EEPROM memorija podataka je 64 bajta i prima veličine od 0h do 3fh.

```
void interrupt intcode() {
  if (INTCONbits.T0IF == 1) {
    if (index== 1) {
      DISPLAY = get_code(D1);
      PORTAbits.RA0 = 1;
      PORTAbits.RA1 = 0;
      index=0;
    } else {
      DISPLAY = get_code(D2);
      PORTAbits.RA1 = 1;
      PORTAbits.RA0 = 0;
      index=1;
    }
    INTCONbits.T0IF = 0;
```

Slika 7. Interrupt blok programskog koda.

Kod se sastoji od više blokova. Prvi blok čine linije konfiguracije, koje se jasno prepoznaju komandama config, include i define. Tu se određuje rad multipleksiranih pinova, kao i unutrašnjih funkcija. Definisanje porta za displej i za button (što je u stvari prekid IR snopa) takođe pripada ovom bloku.

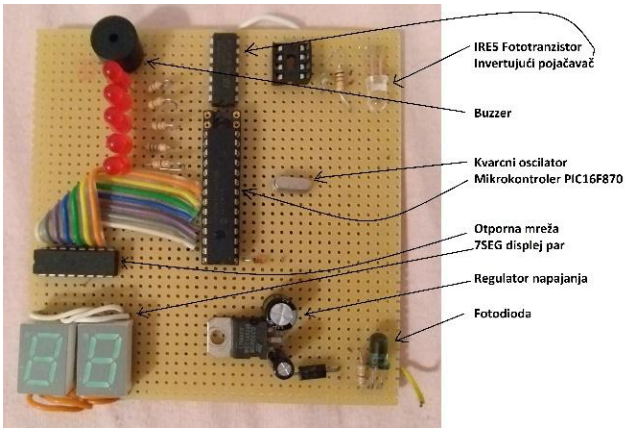
Sledi blok početnih instrukcija. Korišćenjem promenljive brojač kontroliše se izlazna vrednost na portu C i time se niz od 5 LED dioda uključuje redom. Pri svakoj promeni promenljive brojač, program briše sve vrednosti na portu C (i time isključuje poslednju uključenu diodu) i šalje napon narednoj diodi. Nakon toga, stoji niz instrukcija za uključenje buzzera pri svakoj promeni. Ovaj deo koda je prilično jednostavan, najosnovniji oblik instrukcija assemblera.

Glavni deo programa je naredni blok koda sa kombinacijom interrupt linije. Prvo se mikrokontroleru dozvoljava hardverski interrupt i definiše se reakcija na isti. Zatim se podešava naizmenični rad dva sedmosegmentna displeja. Niz „if (++D2 == 10)“ proverava da li je vrednost prvog „displeja“ prevazišao devet, i ako jeste, program ga u narednim nizovima vraća na nulu a povećava D1 za 1. Onda dolazi deo definisanja interrupta, što predstavlja svaki signal logičke jedinice doveden na port A, pin2, kao interrupt. Kombinacija ove dve instrukcije predstavlja naizmenično proveravanje promenljivih D1 i D2, i ispisivanje istih na odgovarajući displej (D2 su jedinice, a D1 desetice).

## IV. PRAKTIČNA REALIZACIJA BROJAČA PROLAZA

Kako je sam uređaj eksperimentalno realizovan, najefikasnije je bilo koristiti raster ploču. Raster je štampana ploča bez lead-ova ili bilo kakvog rutiranja, već je puna samo rupica standardnih veličina za through-hole komponente sa bakarnim slojem (kako bi se mogle komponente lemiti). Glavna mana raster ploča je to što se svaka konekcija mora ostvariti višestrukim lemovima kao i korišćenjem tankih bakarnih provodnika što stvara dodatnu „gužvu“.

Kako je programski kod uspešno testiran, neophodno je isti i isprogramirati u mikrokontroler. Za to je korišćen programator firme *Microchip,* korišćenjem računarskog programa PicKitV2.



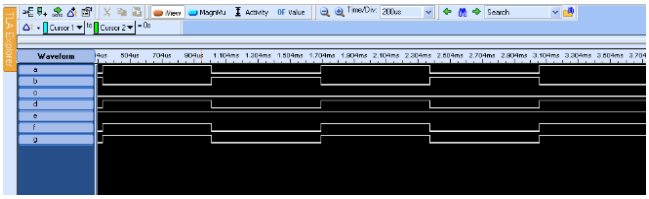

Slika 8. Prikaz realizovanog brojača prolaza.

## V. ANALIZA UREĐAJA LOGIČKIM ANALIZATOROM TLA5202B

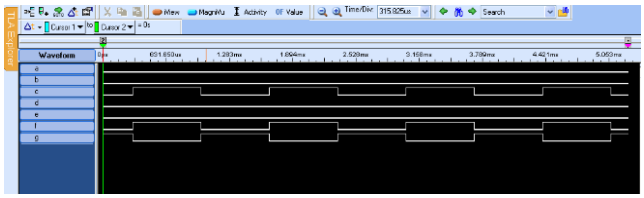
Nakon fizičke realizacije brojača pristupljeno je testiranju uređaja. Uređaj je priključen na napajanje od 7V u kućnim uslovima i uočeno je normalno funkcionisanje. Testiran je do zadnjeg broja 99 (zbog samo dva primenjena sedmosegmentna displeja). Takođe, rad samog uređaja i testiran je i logičkim analizatorom TLA5202B koji proizvodi firma Tektronix.

Za snimanje je bilo bitno odabrati određene brojeve koji će jasno prikazati multipleksiranje na

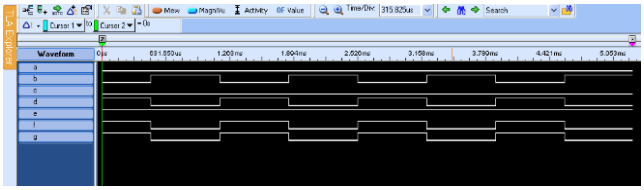
sedmosegmentnim displejima koje konstanto obavlja mikrokontroler. Odabrani su brojevi 15, 20 i 57.



Slika 9. Multipleksiranje broja 15.



Slika 10. Multipleksiranje broja 20.



Slika 11. Multipleksiranje broja 57.

Uređaj je kroz analizu pokazao logiku multipleksiranja dva broja mikrokontrolerom u realnom vremenu.

## VI. ZAKLJUČAK

U ovom radu prikazana je simulacija rada elektronskog brojača realizovanog korišćenjem senzora prolaza. Ovo je jednostavan primer realizacije tehnike modernog doba i predstavlja dobru osnovu za dalja istraživanja.

## ZAHVALNICA

Autor se zahvaljuje doc.dr Danijelu Dankoviću na izdvojenom vremenu, sugestijama i pomoći pri realizaciji projekta i pisanju rada.

## LITERATURA

[1] „BPV10NF datasheet“, dostupno na: http://www.vishay.com/docs/81503/bpv10nf.pdf
[2] „IRE5 datasheet“, dostupno na: http://www.bucek.name/pdf/ire5.pdf
[3] „PIC 16F870 datasheet“, dostupno na: http://ww1.microchip.com/downloads/en/DeviceDoc/30569C.pdf

# Efekat primene visokog napona na asimetrični kondenzator

Božidar Nikolić, Petar Mirković, Ilija Petrović, Violeta Dimić

*Sadržaj* - Ovaj rad se bavi efektom fenomena mehaničke sile generisane visokim naponom asimetričnog kondenzatora. Takozvani Biefeld-Brown efekat je fascinantan efekat na osnovu koga levitacija može da se postigne bez pokretnih ili rotirajućih elemenata. Visoki (DC) napon se primenjuje između asimetričnih elektroda pri čemu se generiše sila ka elektrodi manjih dimenzija. Koristeći ovaj efekat, urađen je eksperiment kako bi se prikazao odnos između geometrije modela i indukovane sile. Rezultati pokazuju jasan odnos proizvedene sile na strukturu modela i dimenzije. Ukoliko je asimetrija jača i sila je jača. Cilj ovog rada je da prikaže primenu ovog efekta sa osnovnim modelom asimetričnog kondenzatora kao i sa složenijim modelom kako bi se pojačao efekat.

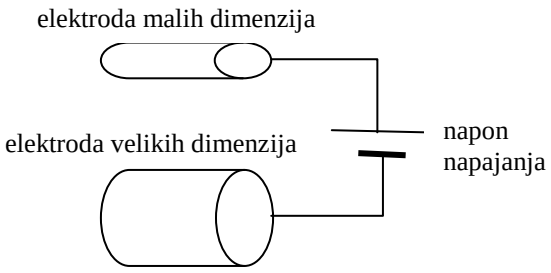
## I. UVOD

Visokonaponski podizač (u daljem tekstu koristićemo opšteprihvaćeni pojam "lifter") predstavlja asimetrični kondenzator koji koristi visoki napon (veći od 10 kV) da bi proizveo silu podizanja. Sastoji se od same konstrukcije asimetričnog kondenzatora i visokonaponskog napajanja. Ovu pojavu otkrio je Thomas Townsend Brown 1928. godine kada je napravio prvu vrstu ovakvog liftera i patentirao ga 1960. godine. Ovaj projekat je gotovo 50 godina bio zaboravljen, kada je 2000. i 2001 godine postao predmet interesovanja NASE u svetlu istraživanja novih vrsta pogona. NASA je izjavila da nije postigla očekivane rezultate, pa zato oni nikada nisu objavljeni. Naravno ovo je zainteresovalo javnost koja je počela više da se bavi istraživanjem ovog fenomena. Godinu dana nakon toga, projekat je proučio Jean-Louis Naudin te napravio replikaciju koja je uspešno poletela. Rezultate svog istraživanja objavio je na web stranici i time podstako oko 350 ljudi širom sveta koji su uspešno napravili replike i različite konstrukcije asimetričnog kondenzatora radi postizanja što boljih rezultata [1] .

## II. PRINCIP RADA

Sam princip rada do danas nije dovoljno objašnjen, mada je najpribližnije objašnjenje teorija o jonskom vetru. Postoji i mišljenje, da je za potisak odgovorno strujanje dielektrika koje je uzrokovano jakim električnim poljem. Eksperimenti sprovedeni u vakuumu idu u prilog teoriji jonskog vetra, mada se javljaju anomalije kod uspostavljanja i prekidanja napajanja. Ovo vodi ka zaključku da je potrebna neka dielektrična sredina za rad liftera. Lifter nije ništa drugo nego kondenzator koji se sastoji od 2 elektrode različitih dimenzija i nekog dielektrika. Na početku istraživanja primećeno je da se sila guranja javlja kod kondenzatora sa simetričnim elektrodama, a kasnije da je efekat izraženiji kod elektroda sa većom asimetrijom. Uočeno je i da je efekat izraženiji, ako je elektroda većih dimezija uzemljena.

Eksperimentisanjem se došlo do zaključka da se najbolji rezultati dobijaju (pri istim uslovima) ako je elektroda manjih dimenzija pozitivno, a elektroda većih dimenzija negativno naelektrisana i uzemljena (slika 1) [2] .




Slika 1. Asimetrični kondenzator.

Nadalje, ustanovljeno je da je efekat izraženiji ako je površina negativne elektrode veća, a takođe i napon . Zbog probojnog napona važan je i oblik electrode većih dimenzija čiji krajevi treba da budu zaobljeni zbog efekta korone. Ovim se može postići manje rastojanje između elektroda, pa samim tim primeniti i veći napon. Zbog Biefield-Brown efekta smatralo se da je na neki način pronađena veza između elektriciteta magnetizma i gravitacije [3].

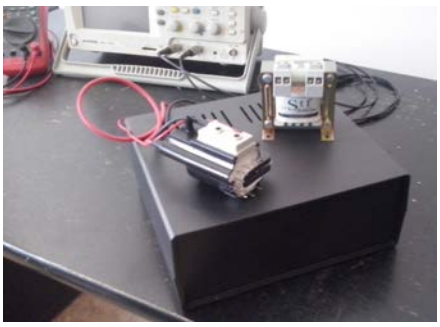
## III. POTREBNA OPREMA ZA DEMONSTRACIJU EFEKTA

### *A. Visokonaponski generator*

Visokonaponsko napajanje sastoji se od nekoliko osnovnih delova i to: dvostrani ispravljač, regulator napona, oscilator za dobijanje pravougaonog napona, kao i visokonaponski transformator (koristi se u TV tehnici).

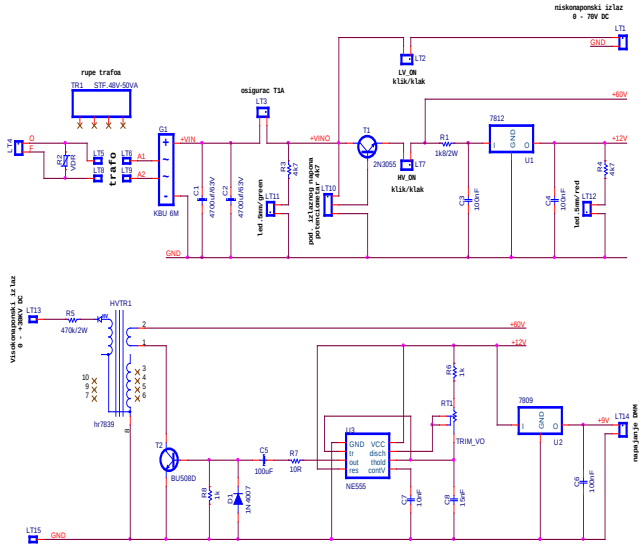
Razliku između mrežnog i visokonaponskog transformatora vidimo na slici 2.

Božidar Nikolić i Petar Mirković su učenici IV razreda obrazovnog profila Elektrotehničar informacionih tehnologija, Ilija Petrović je učenik IV razreda obrazovnog profila Elektrotehničar računara i Violeta Dimić je mentor i profesor stručnih predmeta u školi ETŠ "Nikola Tesla", Aleksandra Medvedeva 18, 18000 Niš, Srbija, E-mail: bozidar.nikolic1999@gmail.com



Slika 2. Visokonaponski (levo) i mrežni (desno) transformator.

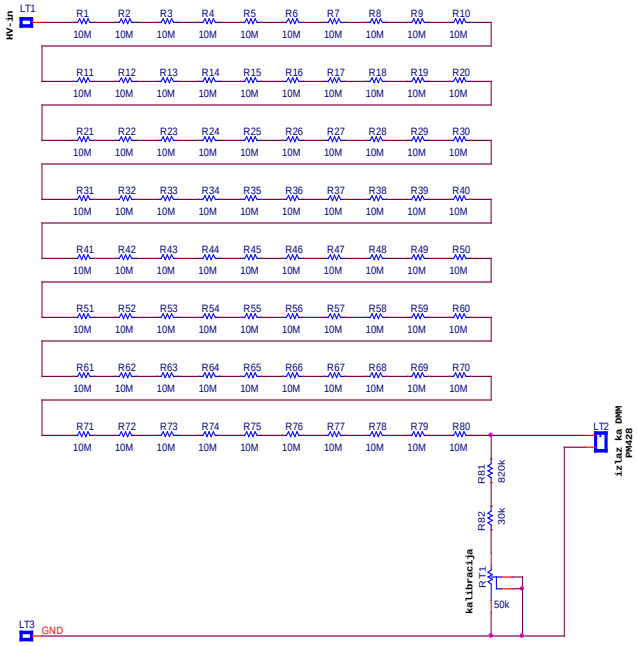
Dvostrani ispravljač sastoji se od mrežnog transformatora 48V/50VA i Grecovog spoja 200V/5A. Napon se reguliše preko tranzistora 2N3055 u granicama 0÷60V, a to je pogonski napon za visokonaponski transformator. Visokonaponski transformator HR7839 koristi se za povećanje napona na izlazu do ~30 kV i struje reda veličine 1mA (slika 3). Oscilator se sastoji od NE555 i potenciometra kojim regulišemo frekvenciju u opsegu 10 ÷ 20kHz. Kao pogonski tranzistor za HR7839 koristi se tranzistor BU508.




Slika 3. Električna šema HV napajanja.

Generator ima ugrađenu visokonaponsku sondu sa displejem za prikazivanje veličine primenjenog napona (slika 4). Napon na izlazu generatora može se menjati u opsegu 0÷30kV.

Kod primene generatora na asimetrični kondenzator izlaz (+) se dovodi na tanku bakarnu žicu koja predstavlja anodu dok se drugi kraj izlaza (-) dovodi na Al foliju odnosno katodu [4] .



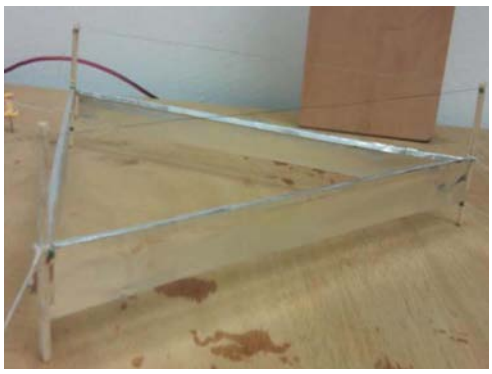

Slika 4. Električna šema za prikaz HV napajanja.

Pri uključenju, potenciometrom za regulaciju napona, menjamo napon na izlazu primaru visokonaponskog transformatora 0 ÷ 60V, a time automatski regulišemo i napon na izlazu što u asimetričnom kondenzatoru izaziva jako električno polje usled kojeg nastaje pojava polarizacije dielektrika (vazduha). Budući da elektrode nisu simetričnih dimenzija javlja se jača privlačna sila prema negativnoj eletrodi tj. katodi (aluminijumska folija), pa se molekuli u vazduhu kreću prema njoj i javlja se strujanje vazduha što daje silu guranja. Da bi se postigla optimalna sila guranja, napon u prostoru između anode i katode kondenzatora trebalo bi da bude na granici probojne čvrstoće vazduha (u našem slučaju uz korišćeni napon sekundara od približno 30kV, razmak je približno 3cm). Veličina probojnog napona zavisiće od vlažnosti vazduha i oblika katode, što se kompenzuje promenom izlaznog napona.

*B. Asimetrični kondenzator*

Prilikom konstrukcije asimetričnog kondenzatora koristili smo: tanku bakarnu žicu prečnika 0.1mm (pozitivna elektroda), tanku aluminijumsku foliju debljine 0.03mm i štapiće balse debljine 2mm

Osnovna konstrukcija je oblika trougla sa stranom od 200 x 40mm, visine 75mm (slika 5). Razmak između folije i tanke Cu žice je 3cm. Težina konstrukcije liftera je ~2g.



Slika 5. Osnovni model liftera.

Kada se visokonaponski pozitivan izlaz priključi na tanku Cu žicu, a negativan na Al foliju, lifter se podiže vertikalno naviše [5]. Na slici 6, prikazan je napravljen složeniji model kako bi se došlo do boljih performansi i pojačao efekat.



Slika 6. Složeniji model liftera.

Ovaj uređaj upotrebljava visoki napon koji je prisutan na kondenzatoru tako da pri isključenju uređaja, izlaz bi trebalo spojiti na masu preko otpornika 10k/2W, pre dodirivanja modela radi zaštite od elektrošoka.

## III. ZAKLJUČAK

Cilj ovog rada je da se prikaže primena Biefeld-Brown-og efekta u eksperimentu sa osnovnim asimetričnim kondenzatorom tzv. lifterom, a zatim i sa složenijim konstrucijama kako bi se došlo do boljih performansi istog (slika 6). Tokom eksperimenata primećeno je da vlažnost vazduha prilično utiče na pojavu električnih lukova tj. proboja koji se javljaju u obliku iskre.

## ZAHVALNICA

Autori se posebno zahvaljuju na stručnoj pomoći mentoru Violeti Dimić kao i stručno-finansijskoj podršci firmi “INA Design&Engineering” iz Novog Pazara.

## LITERATURA

[1] http://jnaudin.free.fr/lifters/hvps.htm
[2] Bahder, TB, "Force on an Asymmetric Capacitor", Retrieved 9 October 2011.
[3] Mallove, Eugene. *"The "Lifter" Phenomenon"* September–October 2002.
[4] *Borg, Xavier (Aug 3, 2004). "What is an EHD thruster?".* http://blazelabs.com/l-intro.asp
[5] Moshe Einat and Roy Kalderon, “High efficiency Lifter based on the Biefeld-Brown effect”, AIP Advances 4, 2014

# Solarna meteorološka stanica

Nikola Minić, Aleksa Mitrović, Jovan Jovanović, Marko Petrović, Radmila Mitić

*Sadržaj* - Solarna meteorološka stanica je sklop električnih elemenata koji je projektovan i konstruisan sa ciljem očitavanja i slanja vremenskih parametara. Upotreba ovakve meteo stanice može da ima široki spektar primene, jer nadogradnjom dodatnih senzora njen rad može da se unapredi. U ovom radu je prikazan jedan od načina realizacuje meteorološke stanice koja je predviđena da se koristi u vinogradarstvu. Rad je realizovan u laboratorijskim uslovima korišćenjem *Arduino Uno* razvojnog okruženja. U radu je predstavljen postupak realizacije Solarne meteorološke stanice.

## I. REZIME

Vreme je trenutno stanje atmosfere na određenom prostoru, što znači da se neprekidno mora pratiti stanje najvažnijih meteoroloških pojava i elemenata (parametara). Da bismo što preciznije prikazali neke meteorološke vrednosti (temperaturu vazduha, vlažnost vazduha, brzinu i pravac vetra) koristimo merne instrumente.

Povećana potreba za očuvanjem poljoprivredne kulture, koja je posledica nagle promene klime, izazvala je potrebu za novim sistemima zaštite. Da bi se ona adekvatno sprovela potrebno je prikupiti podatke o klimatskim karakteristikama datog područja. Na osnovu naših istraživanja u prikupljanju podataka o meteorološkim parametrima značajno mesto dobijaju bežične meteorološke stanice, koje se najčešće koriste u poljoprivredi. Njihova uloga je da se na osnovu prikupljenih podataka sprovede adekvatna mera zaštite i nege biljaka. Takođe, treba naglasiti da je velika prednost ove meteorološke stanice u samostalnom izvoru napajanja, koji je realizovan u vidu solarnog panela. Prednost je i u tome što korisnik ima stalni bežični pristup podacima merenja preko Web stranice. Ali, kao što je poznato, svaka mašina ima neku manu. Kod solarne meteorološke stanice jedan od problema je da dolazi do problema sa WiFi modulom, u slučaju jakog nevremena jer su moguci prekidi u slanja podataka.

## II. UVOD

Meteorologija je nauka koja se bavi proučavanjem Zemljine atmosfere, odnosno proučava promene vremenskih uslova oko nas [1]. Sam početak meteorologije se bazirao na posmatranju sve do XVII veka kada su pronađeni prvi instrumenti za merenje tih pojava (barometar, hidrometar i termometar). Uključivanjem i nauke posebno Njutnovih zakona, Paskalovih zakona i drugih naučnika ova nauka je dobila na značaju i napretku.

Počeci moderne meteorologije su vezani za 1948-u godinu kada je Jule Charney primenio pojednostavljene jednačine dinamičke ravnoteže. Korišćenje računara u meteorologiji doneo je znatan napredak jer je omogućio da je na osnovu merenih podataka moguće prognozirati vreme rešavajući sisteme jednačina. Meteorološki podaci su dobili veleiki značaj u svakodnevnom životu čoveka kao i u brojnim industrijjskim granama.. Da bi se obezbedio dovoljan broj podataka postavljene su brojne meteorološke stanice koje su međusobno povezane na globalnom nivou u Svetsku meteorološku organizaciju (World Meteorological Organization (WMO)).

## III. PRINCIP RADA I PRAKTIČNA REALIZACIJA

Kod nas u Srbiji do meteoroloških podataka je moguće doći preko meteoroloških stanica koje je postavio meteorološki zavod Srbije. To predstavlja problem jer one ne mogu, odnosno ne pokrivaju celu teritoriju Srbije. Da bi bilo moguće u svakom trenutku dobiti informaciju o meteorološkim informacija sa bilo kog mesta potrebno je imati na raspolaganju pokretnu meteorološku stanicu sa sopstvenim napajanjem. Jedno od takvih rešenja je solarna meteo stanica čiji način izrade smo prikazali u ovom radu.

Solarna meteo stanica radi na principu iskorišćenja solarne energije za pokretanje meteoroloških senzora i bežično šalje podatake merenja korisniku. Uz pomoć meteorološke stanice postavljene npr: u voćnjaku pratimo većinu za nas važnih meteoroloških parametara, kao što su: brzina i pravac vetra, temperaturu i vlažnost vazduha. Stanica se napaja preko "Solarnog pratilaca" , a preko Web stranice, koja je uparena sa WiFi modulom ESP12, je povezana sa potrošačem. Rad je realizovan u laboratorijskim uslovima.

Za realizovanje ovog projekta korišćeni su sledeći delovi:

1. "Solarni pratilac" i baterija (Slika 1.)
2. Senzor brzine i pravca vetra
3. Senzor temperature i vlažnosti vazduha
4. WiFi modul ESP12

Nikola Minić, Aleksa Mitrović, Jovan Jovanović, Marko Petrović su učenici IV razreda smera Tehničar mehatronike, Elektrotehnička škola "Nikola Tesla", Aleksandra Medvedeva 18, 18000 Niš, Serbia, E-mail: etstesla@etstesla.ni.ac.rs

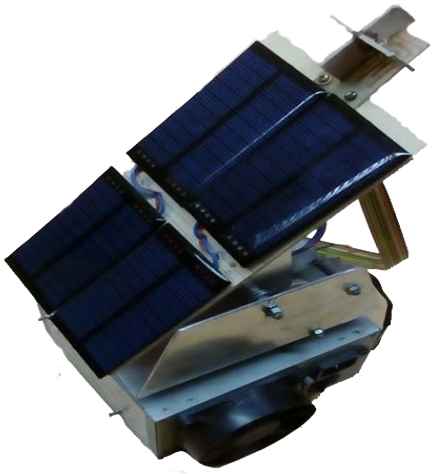
Radmila Mitić, profesor stručnih predmeta, Elektrotehnička škola "Nikola Tesla", Aleksandra Medvedeva 18, 18000 Niš, Serbia, E-mail: racamitic@gmail.com

## IV. SOLARNI PRATILAC

Solarni pratilac [2] je uređaj konstruisan tako da automatski zauzima položaj, u odnosu na sunčevu svetlost pod uglom od 90 stepeni, tokom celog dana.

Solarni pratilac radi na pricipu maksimalnog iskorišćenja sunčeve svetlosti koja se postiže zahvaljujući fotootpornicima koji detektuju maksimalno zračenje. Ta informacija smanjenja otpornosti odnosno povećavanja struje se prenosi do arduina koji pokreće motore panela i postavlja ih u položaj maksimalnog osvetljenja. Na taj način je ostvarena maksimalna efikasnost rada solarnih panela. Fotootpornici imaju i ulogu automatskog prekidača. Kada je svetlost jako slabog intenziteta koja prouzrokuje otpornost otpornika oko 250kΩ, da se ne bi trošila energija za pokretanje celog sistema, fotootpornik svojom otpornošću daje signal upravljačkoj komponenti (motorima) da ostanu u stanju mirovanja sve do pojave jačeg inteziteta svetlosti, odnosno do postizanja otpornosti od 40kΩ. Za to vreme ostale komponente se napajaju iz baterije koju je fotonaponski izvor punio tokom dana svojim proizvedenim viškom energije.

Opremanjem solarnog pratioca senzorima koji služe za merenje meteoroloških parametara ovakav uređaj moguće je koristiti kao samostalnu pokretnu meteo stanicu.



Slika 1. “Solarni pratilac”.

## V. SENZOR BRZINE I DUVANJA VETRA

Za merenje jačine vetra moguće je koristiti anemometer [3] sa lopaticama (Slika 2.), koji je i najčešće u upotrebi. Njihova cena na tržištu je niža u odnosu na druge vrste anemometara, što potvrdjuje njihov izbor od ponuđenih senzora na tržištu. Ovi senzori se sastoje od 3 ili 4 lopatice koje su smeštene vertikalno na horizontalne nosače. Horizontalni nosači smešteni su na zajedničku osovinu koja se okreće zajedno sa lopaticama. Osovina je spojena na električni pretvarač, u našem slučaju na enkoder, koji proizvodi naizmenicni električni izlazni signal čiji napon i/ili frekvencija je proporcionalna brzini merenog vetra. Srednja brzina vetra izračunava se na osnovu poznavanja broja impulsa u određenom vremenskom intervalu i ta informacija u digitalnom obliku se šalje do arduina a zatim i do korisnika. Neke od radnih karakteristika vezane za ovaj tip anemometra su: ± 4%, rezolucija $0{,}1\ m/s$ i merni opseg od$1m/s$ do $75m/s$

Kao senzor za odredjivanje pravca vetra može da se koristi i Arduino kompas. Senzor podržava procedure automatske kalibracije. Radni napon ovog senzora je 3.3V-5V.

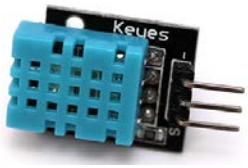
Ovaj senzor prikazuje ugao od 0 ° do 360 ° sa tačnošću od ± 5 ° i rezolucijom od 1 °.



Slika 2. Anemometar.

## VI. SENZOR TEMPERATURE I VLAŽNOSTI

Za merenje temperature i vlažnosti vazduha (Slika 3.) može da se koristi i senzor DHT-11 [4]. Ovo je senzor sa jednožičnim digitalnim interfejsom. Senzor je fabrički kalibrisan i ne zahteva dodatne komponente tako da se može odmah upotrebiti za merenja. Sastoji se od kapacitivnog senzora vlažnosti vazduha, termistora za merenje temperature i elektronike za komunikaciju sa mikrokontrolerom.



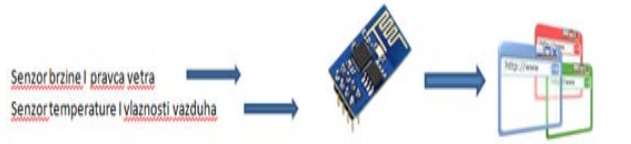

Slika 3 Senzor temperature i vlažnosti vazduha.

Tehničke karakteristike:

- napajanje: 3.3 ~ 6.0V
- potrošnja struje za vreme meranja: 1 ~ 1.5mA
- standby potrošnja struje: 40 ~ 50 uA
- opseg meranja vlažnosti: 0 - 100% RH
- opseg merenja temperature: -40 - +80 °C
- tačnost merenja: ±2% RH i ±0.5°C

## VII. WIFI MODUL ESP8266 (ESP-12)

Za potrebe praćenja i kontrole parametara od strane korisnika razvijena je Web stranica, pa se sistem moze predstaviti blok šemom na Slici 4.




Slika 4. Blok šema realizovanog sistema.

WiFi modul ESP8266 [5] (Slika 5.) je nezavistan sistem na čipu sa integrisanim TPC/IP protokolom koji omogućava pristup WiFi mreži. U samom procesu proizvodnje izvršeno je preprogramiranje svakog modula sa AT skupom firmware komandi što omogućava povezivanje serijskom vezom sa bilo kojim mikrokontrolerom. Ovaj modul ima odlične mogućnosti obrade i sklapanja podataka koje mu dozvoljavaju da se integriše sa senzorima i drugim uredjajima.

Visok stepen integracije na čipu sa minimalnim dodatnim kolima omogućava da modul zauzme mali prostor.

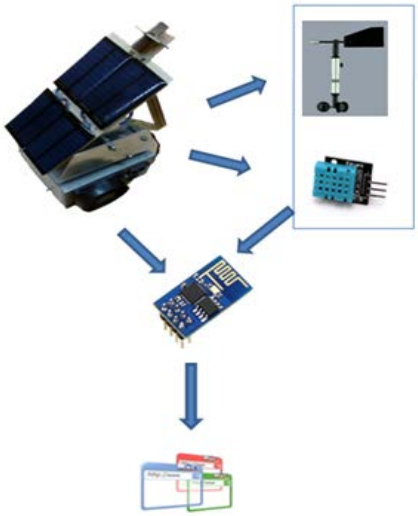
Značajne karakteristike modula ESP8266 su:

- $802{,}11b/g/n$ protokol,
- WiFi direktni (P2P), soft AP,
- Integrisani TCP/IP protokol,
- $+19{,}5\ dbm$ izlazna snaga u $802{,}11b/g/n$ modu,
- Integrisani temperaturni senzor,
- Integrisani $32-$bitni procesor male snage koji može biti korišćen kao procesor aplikacija ( radna frekvenca $52\ MHZ$ )
- SDIO 1.1/2.0 SPI, UART, ADC, PWM,
- STBC, $1 \times 1$ MIMO, $2 \times 1$ MIMO,
- Prenos paketa $< 2\ ms$,
- Pasivna potrošnja $< 1.0\ mW$,
- Potrošnja u predaji $215\ mA$,
- Potrošnja u prijemu $62\ mA$

Povezanost predhodno navedenih uređaja i tok signala prikazan je na slici 6.



Slika 5. Fizički izgled WiFi modula ESP8266.



Slika 6. Šema povezivanja komponenti i prikaz toka signala.

## IX. ZAKLJUČAK

Zadatak projekta je bio raelizacija meteorološke stanice koja ima veliki učinak u zaštiti biljaka. Analizom zadatka se došlo do rešenja o realizaciji solarne meteorološke stanice čija je efikasnost zadovoljavajuća. Uspešno smo realizovali solarnu meteorološku stanicu koja je prikazana na slici 7. Smatramo da je rad na projektu bio uspešan i da će budućom nadogradnjom služiti povrtarima i vinogradarima za bolji prinos u poslu, kao i drugim zainteresovanim korisnicima.



Slika 7. Solarna meteorološka stanica.

## LITERATURA

[1] Viktor Mandić, Diplomski rad
[2] www.solarnipaneli.org
[3] sr.wikepedia.org
[4] www.saperel.com
[5] J. Krstic, WiFi web multisenzorski sistem u energetski efikasnim objektima

# Multifunkcionalni časovnik

Uroš Stojković, Marko Nikolić, Jovan Cekić, Boris Brzački, Marko Marjanović, Radmila Mitić, Dejan Stamenković

*Sadržaj* - Multifunkcionalni časovnik je sklop električnih elemenata koji je projektovan i konstruisan sa ciljem merenja, prikazivanja vremena, snimanja i reprodukcije zvučnih poruka. Rad je realizovan u laboratoratorijskim uslovima korišćenjem *Arduino Uno* i drugih mikrokintrolera. U radu je predstavljen jedan od načina realizacije Multifunkcionalnog časovnika.

## I. UVOD

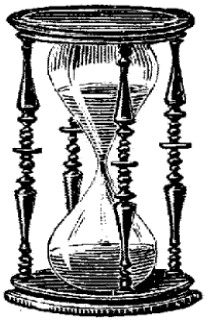
Uz sve brži napredak tehnologije ljudi imaju sve više obaveza i sve manje vremena za njihovu realizaciju. Zbog toga je važno da u svakom trenutku znaju koliko im je vremena ostalo za ostvarenje svojih obaveza. Radi lakšeg i bržeg funkcionisanja u planiranju svog vremena potrebno je da u svakom momentu znamo još koliko vremena imamo na raspolaganju, odnosno koliko je sati.

Od nastanka prvih civilizacija ljudi su imali potrebu za računanjem vremena. Prvo računanje vremena vršeno je uz pomoć ***sunčanog sata*** još u starom Egiptu. Sunčani sat (slika 1) je bacao senku uz pomoć fiksiranog objekta na ravnu površinu na kojoj su bili označeni sati. U daljem periodu korišćen je ***peščani sat*** (slika 2). Merenje vremena je bilo postignuto prolaskom sitnih zrna peska kroz uzani otvor u staklenoj posudi. Stari Egipćani su koristili ***klepsidu*** za merenje vremena. Klepsida (slika 3) je merila vreme uz pomoć posude i vode koja je kroz uzani otvor tekla. Pretpostavlja se da je ovo bio jedan od najtačnijih instrumenata za merenje vremena. Smatra se da prvi sat sa mehanizmom datira još od 87. god p.n.e. Prvi prilično tačni časovnici, satovi na tornjevima iz 13. veka, pravljeni su za italijanske kaludjere. Najraniji stoni satovi koji su preživeli do dan danas datiraju još iz 16. veka. ***Satovi sa klatnom*** napravljeni su još 1657. godine. Dodatno su dorađeni 1670. godine kada su mu dodate kazaljke za minute i sekunde [1]. Razviće elektronike 20. veka donosi satove bez ikakvog satnog mehanizma koji su se napajali pomoću raznih izvora energije bez potrebe za navijanje sata.

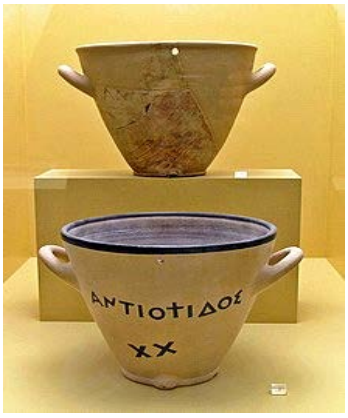
Danas u 21. veku postoji široki spektar dostupnih uređaja za merenje vremena. Međutim, naša istraživanja su pokazala, da na trenutno dostupnom tržištu postoji malo ili ne postoje uređaji koje bi mogli koristiti osobe sa hendikepom. Zato smo realizovali Multifunkcionalni časovnik koji mogu da koriste i slepe osobe. Način njegove realizacije prikazacemo u ovom radu.



Slika 1. Sunčani časovnik.



Slika 2. Peščani časovnik.




Slika 3. Klepsida.

## II. ARHITEKTURA SISTEMA

Za realizaciju Multifunkcionalnog časovnika potreban je softverski i hardverski deo.

Hardversku osnovu sistema čine:

Uroš Stojković, Marko Nikolić, Jovan Cekić, Boris Brzački, Marko Marjanović su učenici III razreda smera Tehničar mehatronike, Elektrotehnička škola "Nikola Tesla", Aleksandra Medvedeva 18, 18000 Niš, Serbia, E-mail: etstesla@etstesla.ni.ac.rs

Radmila Mitić, Dejan Stamenković, Profesori stručnih predmeta, Elektrotehnička škola "Nikola Tesla", Aleksandra Medvedeva 18, 18000 Niš, Serbia, E-mail: racamitic@gmail.com

- 6-cifarski LED displej
- Konektori
- Potenciometar
- Kondenzatorski mikrofon
- Zvučnici
- Tasteri
- Adapter
- Mikrokontroler **AT89S8252**
- Čip **Winbond ISD4004-12M**
- Čip **RTC-72423A**
- Pojačavač **TDA7052**
- Integrisani stabilizatori **LM317, 7809, 7805**
- Senzor udaljenosti **HC-SR04**

Za realizaciju programa je korišeno Arduino softversko okruženje.

Ovaj rad se sastoji iz nekoliko glavnih celina koje funkcionišu kao sklop, a takođe mogu se sagledati i koristiti kao samostalno funkcionalni delovi. Opis funkcionisanja ovih delova kao samostalne celine navodi se u nastavku.

## III. NAPAJANJE

Napajanje je moguće realizovati uz pomoć akumulatora i kontrolera napajanja (punjenja). Sklop koji je zadužen za kontrolu punjenja ima zadatak da u zavisnosti od tipa baterije ne dozvoli prepunjavanje baterije. Neke od njegovih karakteristika su:

-uključuje punjenje ako je nizak napon baterije i ako je ulazni napon koji dolazi od solarne ćelije veći od 12V

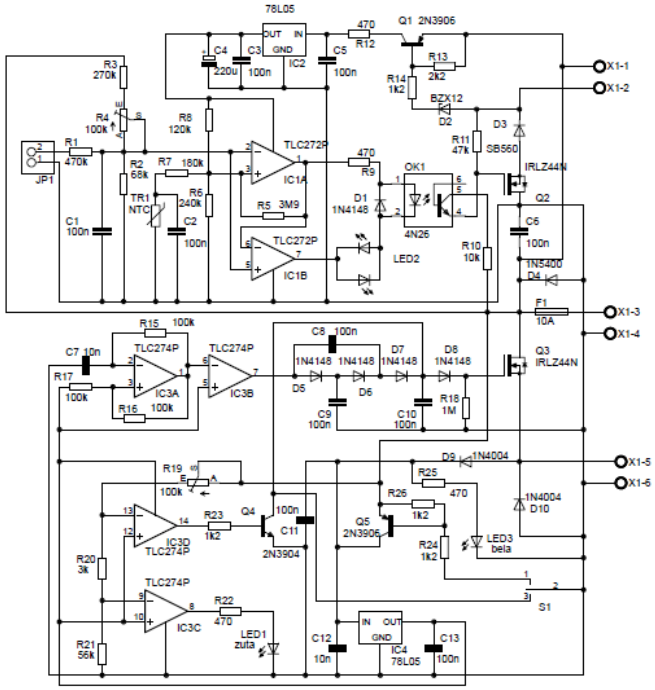
-kontroliše struju punjenja da ne bi došlo do „over current“ prepunjavanja i oštećenja akumulatora

-punjenje se isključuje ukoliko se baterija pregrejala ili ako je dostigla željeni napon (merenje temperature baterije se vrši preko NTC otpornika postavljenog uz samu bateriju, a napon koji isključuje punjač obično je od 13,8V do 14,2V) [2].

Priključenjem solarnog panela za vreme dok ima svetlosti na diodi D2 će se pojaviti napon veći od 12V, koji otvara tranzistor Q1 i dovodi na stabilizator IC2 potreban napon za pokretanje upravljačke elektronike.

Dovođenjem napona na IC1, a da je pri tome zadovoljen uslov da je napon u tački R4-R2 niži od zadate vrednosti ispražnjenosti baterije počinje ciklus punjenja, izlaz na nožici 1 (IC1) prelazi na visoki logički nivo i preko optokaplera (OK1) otvara tranzistor Q2 koji pušta struju iz solarne ćelije u bateriju. Dioda D3 služi kao zaštita od inverznog napona na solarnoj ćeliji ili generatora. Potenciometrom R4 se određuje gornja granica punjenja baterije. Punjenje i prekid punjenja baterije se vrši automatski.

Uređaj poseduje i sklop koji kontroliše pražnjenje baterije i on se startuje preko prekidača S1 i aktivira tranzistor Q5 koji dovodi napon na stabilizator IC4. Stabilizator IC4 služi za napajanje kola IC3 koje je jednim delom oscilator, a drugim komparator napona. Oscilatorski deo IC3 preko utrostručivača napona otvara tranzistor Q3 koji prosleđuje napon iz baterije prema potrošaču.




Slika 4. Šema kontrolora napajanja.

Uslov za ovu funkciju je da je napon na bateriji iznad već utvrđenog minimalnog napona koji se podešava preko potenciometra R19 i koji se preko svetleće diode LED1 prikazuje. Prilikom pada napona baterije tokom korišćenja, nizak napon će se prvo prikazati preko LED1, a zatim posle izvesnog vremena tranzistor Q3 će prestati da provodi prema potrošaču. Prekidač S1 ima i mogućnost potpunog isključenja potrošača, ako nema potreba za stalnim napajanjem istog. Dioda LED3 označava da je uređaj u funkciji. Dioda D4 i osigurač F1 štite uređaj od pogrešnog priključenja baterije tako da u slučaju zamene polova na bateriji, dioda D4 postaje potpuno provodna i dolazi do pregorevanja osigurača F1. Na slici 4 prikazana je šema koja prikazuje način povezivanja datih komponenata porebnih za realizaciju ovog projekta.

## IV. ULTRAZVUČNI SENZOR UDALJENOST

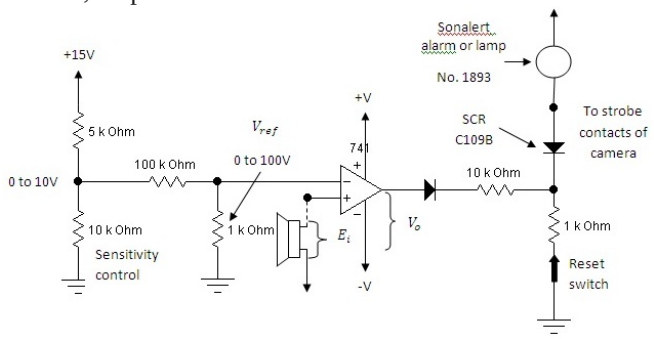
Ultrazvučni senzor udaljenosti se koristi za detekciju prisustva objekta na određenom rastojanju (od 0,02m do 5m). Rad ovog senzora se prati uz pomoć dve LED diode (npr. crvena i zelena). Kada senzor detektuje objekat daje se električni signal i uključuje se zelena LED dioda, dok je crvena ugašena. Drugo moguće stanje sistema je kad ispred senzora nemamo objekat tada se električnim signalom uključuje crvena LED dioda dok je zelena ugašena. Crvena LED dioda ostaje uključena sve dok ne dodje do promene i

detekcije objekta. U ovom radu je korišćen senzor **HC-SR04** kao prekidač. Premda, kada senzor detektuje objekat, u našem slučaju čoveka, senzor šalje električni signal koji se dalje koristi za aktiviranje uređaja i služi za reprodukciju glasovne poruke, odnosno glasovnog obaveštenja satnice ili uključivanja displeja. Ovakva primena je iskorišćena radi uštede vremena ili ekonomičnije potrošnje električne energije [3].

## V. PREKIDAČ AKTIVIRAN ZVUKOM

Prekidač aktiviran zvukom, može upravljati različitim uređajima. Na primer, uređaj se može podesiti tako da reaguje na zvuk kucanja o vrata ili pljeskanja rukama i da na taj način uključuje zadati uređaj, u našem slučaju časovnik[4].

Princip rada ovog uređaja bazira se na činjenici da bilo kakav šum proizvodi naizmenični napon na mikrofon. Prva pozitivna oscilacija sa Ei na operacionom pojačavaču (Slika 4.), veća od Vref dovodi Vo na +Vsat. Dioda sada provodi strujni impuls od 1mA na gejt SCR-a (Silicon-Controled Rectifier). U početnim uslovima, anoda i katoda SCR-a se ponašaju kao otvoreni prekidač. Strujni impuls čini SCR otvorenim. Audio ili vizuelni alarm su sada aktivirani. Alarm će biti uključen, jer kada se SCR jednom aktivira, on provodi dok se anoda i katoda ne otvore.




Slika 5. Šema prekidača aktiviranog zvukom.

## VI. HARDVERSKA REALIZACIJA GLASOVNOG ČASOVNIKA

Multifunkcionalni časovnik prikazuje vreme i/ili datum na 6-cifarskom LED displeju. Zeleno svetli displej u normalnom režimu rada, a crveno kada je na uređaj priključen modul za programiranje. Na bočnom delu časovnika nalazi se konektor za priključivanje modula kojim se on programira, kao i ručica potenciometra kojom se podešava željena jačina zvuka. Na gornjoj strani nalazi se kondenzatorski mikrofon koji služi za snimanje audio poruka,koje mogu da se reprodukuju pomoću zvučnika. Interfejs uređaja može se podeliti na dva glavna dela[5]:

1) Interfejs pri normalnom radu:

Ovaj deo čine tri tastera koja se nalaze sa gornje strane uređaja.

Taster1 – Ovaj taster se koristi za ulazak u režim nameštanja vremena i datuma.

Taster2 – Ovim tasterom se forsira izmena prikaza vremena i datuma.

Taster 3 – Aktivira govor vremena ili datuma u zavisnosti od toga šta je trenutno prikazano na displeju.

Snimanje poruke započinje se pritiskom na taster 1 i odmah potom na taster 2. Snimanje poruke traje dok se ne pritisne ili otpusti neki od tastera ili dok ne prođu dva minuta. Reprodukcija poruke aktivira se pritiskom na taster 1 pa zatim na taster 3. Reprodukcija traje dok se ne pritisne ili otpusti neki taster ili do završetka poruke.

2) Intefejs pri priključenom modulu

Programiranje uređaja aktivira se priključivanjem modula za programiranje. Ovaj mod omogućava sledeće:

1. Podešavanje trajanja prikaza vremena na displeju
2. Podešavanje vremena prikaza datuma na displeju
3. Snimanje i reprodukcija glasova neophodnih za govor datuma i vremena
4. Podešavanje intervala automatskog aktiviranja datuma i vremena.
5. Podešavanje dnevnog perioda u kome će se automatski izgovarati datum i vreme.
6. Podešavanje parametara vezanih za alarm

Da bi uređaj pravilno radio neophodno je snimiti 104 reči na srpskom jeziku. Primeri reči su jedan, prve, januar, itd. Uređaj koristi veoma mali broj tastera da bi olakšao rad korisnicima sa oslabljenim vidom. To je razlog za korišćenje raznih kombinacija tastera i dodatnog modula umesto dodavanja novih. Bitno je da pri uobičajenom radu broj tastera bude što manji. Tasteri na kutiji su razmešteni tako da ih desnoruke osobe intuitivno nalaze.

Pri realizaciji multifunkcionalnog časovnika uz hardversku realizaciju glasovnog časovnika dodate su i druge komponenete. Uz pomoć tih komponenti (ultrazvučni senzor udaljenosti, prekidač aktiviran zvukom i napajanje) dobili smo dodatne funkcije koje olakšavaju upotrebu i štede električnu energiju.

## VII. SOFTVERSKA REALIZACIJA MULTIFUNKCIONALNOG ČASOVNIKA

Za softversku realizaciju moguće je koristiti razlicite programske jezike. Za naš rad izabrali smo program Arduino UNO jer smo ovaj program imali na raspolaganju u školi.

Fizičke karakteristike Arduina su:

- 14 Digitalnih inputa (od kojih 6 mogu da se koriste kao PMW output)
- 6 Analognih inputa
- Kvarcni oscilator od 16MHz
- USB konekciju
- Dugme za resetovanje

Pozitivne karakteristike Arduna UNO su:

- Relativno niska cena na tržištu
- Potrebe za manjim napajanjem (u odnosu na Rpi)

- Lako se počinje rad sa aktivnom onlajn podrškom dok kod Rpi treba imati dosta znanja o Lynux sistemima da bi se pokrenuo rad.
- Za programiranje se moze koristiti C, Ruby, Python i IDE. Potpuno je open-source.

Negativne karakteristike:

- Ima malo mesta za skladišćenje podataka u odnosu na Rpi
- Ne moze da podrži teške algoritme ili interfejs kao ekran na dodir bez dodatnih shieldova

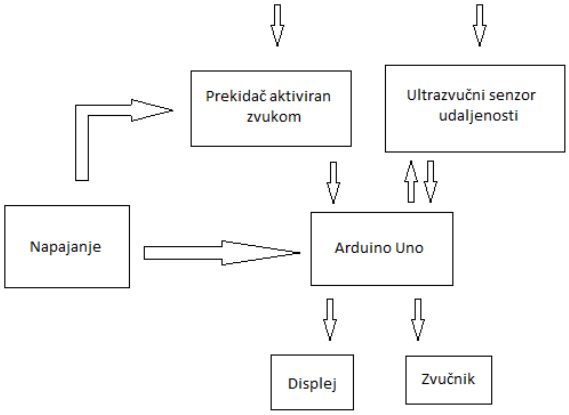


Slika 6. Arduino UNO.

Arduino predstavlja razvojni sistem otvorenog koda zasnovan na 8-bitnom Atmel AVR ili 32-bitnom AtmelAMR mikroprocesoru. Ploča Arduino razvojnog okruženja (slika 2.) u osnovi sadrži: oscilator odnosno kristal koji reguliše vremenske uslove procesora (clock), naponske regulatore vrednosti 5V i 3.3V i USB-to-serial konvertor. Za ovaj projekat je korišćena platforma Arduino UNO. Arduino Uno je mikrokontrolerska ploča sa ATmega 328P. Ona sadrži 16 digitalnih ulazno/izlaznih pinova (od kojih 6 mogu da se koriste kao PWM izlazi a 6 kao analogni ulazi), kristalni oscilator od 16MHz, mikro USB konektor, priključak za napajanje, ICSP konektor i reset taster. Kontroler je isprogramiran u Arduino IDE razvojnom okruženju korišćenjem C/C++ programskog jezika.

## VII. ZAKLJUČAK

Analizom rada predhodno pomenutih komponenti može se zaključiti da njihovim povezivanjem je moguće realizovati multifunkcionalni časovnik, kao što je rikazano na slici 7. Centralni deo datog sistema pedstavlja Arduino koji prima sigale iz prekidača aktiviranog zvukom i ultrazvučnog senzora, a sam šalje signale ultrazvučnom senzoru, displeju i zvučniku. Izvor energije za rad Arduina i prekidača aktiviranim zvukom je napajanje (baterija od 12 V). Nakon intenzivog istraživanja, projetovanja i testiranja uspešno smo realizovali u laboratorijskim uslovima ovaj uređaj čime smo ispnili sebi postavljen zadatak. Jedan od mogućih budućih zadtaka je usavršavanje ovog uređaja što je moguće postići dodavanjem senzora koji bi proširili njegovu funkcionalnost, kao što su sensor temperature, sensor pritiska…Nadamo se da će ova ideja biti prepoznata od strane industrijskog menađmenta i da zaživi u praksi.




Slika 7. Blok šema multifunkcionalnog časovnika.

## ZAHVALNICA

Rad je realizovan u laboratorijama elektrotehničke škole „Nikola Tesla“. Autori se zahvaljuju profesorima škole na pruženoj pomoći i podršci u radu, posebno profesorima Radmili Mitić, Dejanu Stamenkoviću ,kao i profesorima sa katedre za mikroelektroniku Elektronskog fakulteta u Nišu.

## LITERATURA

[1] Istorijat https://sr.wikipedia.org/sr-el/%D0%A7%D0%B0%D1%81%D0%BE%D0%B2%D0%BD%D0%B8%D0%BA
[2] Kontrolor napajanja-https://www.automatika.rs/projekti/svi-projekti/kontroler-punjenja-akumulatora-solarnim-celijama.html
[3] Ultrazučni senzor udaljenosti [ARDUINO]-https://www.automatika.rs/projekti/upotreba-ultrazvucnog-senzora-daljine-arduino.html
[4] Prekidač aktiviran zvukom - https://www.automatika.rs/projekti/svi-projekti/prekidac-aktiviran-zvukom.html
[5] Hardverska realizacija govornog časovnika - https://www.automatika.rs/projekti/svi-projekti/hardverska-realizacija-govornog-casovnika.html
[6] Arduino UNO - https://www.quora.com/What-are-the-pros-and-cons-of-Arduino-and-Raspberry-Pi-for-IoT-hardware-solutions

# Daljinska kontrola i nadgledanje “pametne” kuće

Nikola Stanković, Bogdan Ranđelović, Radmila Mitić, Violeta Dimić, Dejan Stamenković

*Sadržaj* – „Pametna” kuća je sklop električnih elemenata koji je projektovan i konstruisan sa ciljem kontrole i nadzora nad svim električnim potrošačima u kući (rasveta, bela tehnika, grejanje, senzore) Rad je realizovan u laboratorijskim uslovima korišćenjem *Arduino Mega* razvojnog okruženja. U radu je predstavljen postupak realizacije „pametne” kuće. Kontrola različitih vrsta kućnih uređaja se realizuje preko Android aplikacije, dok se komunikacija sa tim uređajima odvija preko Interneta.

## I. UVOD

Za kontrolu kućnih uređaja potrebna je aplikacija čija bezbednost treba da je na visokom nivou, kako ne bi došlo do zloupotrebe i neovlašćenog pristupa kući.

Centralni deo projekta „Pametna” kuća je Arduino [1]. Na njemu postoji veliki broj povezanih komponenti, koje treba da odgovore zantevima korisnika (uključivanje/ isključivanje svetla, regulacija grejanja, otvaranje/ zatvaranje vrata garaže, podizanje/spuštanje roletni itd.). Bezbednost ovakve kuće se sagledava sa više aspekata, npr. mod „Izlazim iz kuće” koji se aktivira putem tastera u blizini ulaznih vrata ili jednim klikom na pametnom telefonu. Mod „Izlazim iz kuće” isključuje šporet, utičnicu za peglu, sva svetla, spušta roletne, što uključuje i automatsku aktivaciju alarma. Time se postiže veliki nivo bezbednosti koji pored uštede električne energije i obezbeđenja kuće od provalnika, eliminiše ljudsku grešku zaboravnosti, npr. ako nije isključena ringla, pegla za veš/kosu, grejno telo itd. Ovim bezbednim modom se sprečava nastanak velikih gubitaka, kako materijalnih tako i ljudskih.

Ugrađeni alarmni sistem, prilikom detektovanja bilo kakvih nepravilnosti šalje SMS poruku vlasniku i aktivira sirenu za uzbunu. Sistem bezbednosti obuhvata sigurnosni sistem kao i sistem za detekciju dima (protiv požarni sistem).

Sistem „pametne” kuće ima ugrađen detektor kiše i senzor brzine vetra, što ga stavlja u red sistema sa većim nivoom bezbednosti i inteligencije. Ako korisnik izađe iz kuće i zaboravi prozor da zatvori ili ga je ostavio zbog provetravanja, u slučaju pogoršanja vremena, sam sistem automatski vrši obezbeđivanje kuće, odnosno zatvaranje prozora.

Pored bezbednosti projekat „Pametna” kuća definiše praktične funkcije koje omogućavaju da sa bilo kog mesta upravlja uređajima koji se nalaze u kući. Jedna od njih je i daljinsko uključivanje grejanja radi postizanja određene temperature u prostorijama kuće.

Jedan od zanimljivih sistema je i sistem za zalivanje travnjaka i cveća, kao i sistem za automatsku regulaciju svetlosti u prostorijama kuće. Na osnovu merenog intenziteta svetlosti i unapred postavljenih parametara željenog intenziteta svetlosti u prostoriji, sistem sam održava taj intenzitet regulacijom položaja roletni. Sistem automatski podiže roletne, odnosno spušta u zavisnosti od kretanja sunca u toku dana. U večernjim časovima, kada dolazi do znatnog smanjenja intenziteta svetlosti, sistem automatski uključuje unutrašnju rasvetu u željenom delu kuće. Takođe je moguće i kontrolisanje uključivanja spoljašnje rasvete, solarnih panela, bojlera, regulacije grejanja bazena i sauna.

Mod „EKO” je mod u kome korisnik dobije obaveštenje da je niža tarifa struje pri čemu on treba da odluči da li će neki od velikih potrošača (bojler, kotao, peći) iti uključen ili će u nekom drugom terminu koristiti sopstveni izvor električne energije (alternativni izvor energije).

Mod „Vraćam se kući” uključuje sve uređaje koji se koriste na osnovu zahteva korisnika i prilagođava okruženje njegovom prisustvu (slika 1) [2].



Slika 1. Sistem povezanih uređaja u kući.

## II. ARHITEKTURA SISTEMA

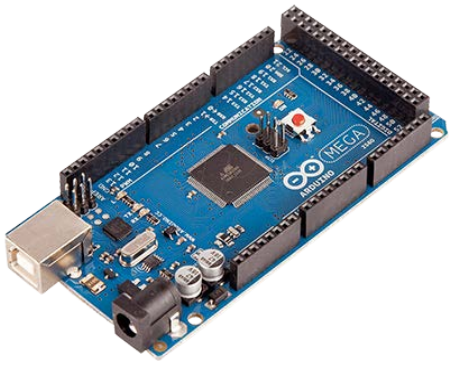
Realizacija cele ideje „pametne“ kuće je osmišljena kroz softverski i hardverski deo. Hardversku osnovu sistema čini Arduino Mega (slika 2.) sa komponentama:

- Senzor pokreta
- Senzor dima
- Detektor kiše
- Senzor vetra

Nikola Stanković je učenik IV razreda smera Tehničar mehatronike, Elektrotehnička škola “Nikola Tesla”, Bogdan Ranđelović je učenik IV razreda smera Tehničar informacionih tehnologija, Elektrotehnička škola “Nikola Tesla”, Aleksandra Medvedeva 18, 18000 Niš, Srbija, E-mail: etstesla@etstesla.ni.ac.rs

Radmila Mitić, Violeta Dimić, Dejan Stamenković, profesori stručnih predmeta, Elektrotehnička škola “Nikola Tesla”, Aleksandra Medvedeva 18, 18000 Niš, Srbija, kontakt e-mail: racamitic@gmail.com

- Senzor vlažnosti
- Senzor količine svetlosti
- Senzor temperature
- Sirena
- Magnetni prekidači
- Motori
- Tiristori
- Trijak
- WiFi mikroprocesor ESP8266
- DC napajanje
- Kontaktori



Slika 2. Arduino Mega.

*A. Arduino Mega*

Arduino predstavlja razvojni sistem otvorenog kôda zasnovan na 8-bitnom Atmel AVR ili 32-bitnom Atmel AMR mikroprocesoru. Ploča Arduino razvojnog okruženja (slika 2.) u osnovi sadrži: oscillator, odnosno kristal koji reguliše radni takst (taktni signal) procesora, naponske regulatore vrednosti 5V i 3.3V i *USB-to-serial* konvertor. Za ovaj projekat je korišćena platforma Arduino Mega. Arduino Mega je mikrokontrolerska ploča koja se bazira ATmega328P integrisanom kolu. Ona sadrži 54 digitalnih ulazno/izlaznih pinova (od kojih 12 mogu da se koriste kao PWM izlazi a 16 kao analogni ulazi), kristalni oscilator od 16MHz, mikro USB konektor, priključak za napajanje, ICSP konektor i reset taster. Kontroler je isprogramiran u Arduino IDE razvojnom okruženju korišćenjem C/C++ programskog jezika. Tehničke specifikacije razvojnog okruženja su date u tabeli 1 [3].

TABELA 1
KARAKTERISTIKE MIKROKONTROLERA

| **Mikrokontroler** | **ATmega328P** |
|---|---|
| Radni napon | 5V |
| Napon napajanja (preporučeno) | 7-12V |
| Maksimalni napon napajanja (ne preporučuje se) | 20V |
| Digitalni I/O pinovi | 54 (od kojih 12 obezbeđuje PWM izlaz) |
| Analogni ulazni pinovi | 16 |
| DC (jednosmerna) struja po I/O pinu | Max 500mA |
| DC (jednosmerna) struja za pin na 3.3V | 50mA |
| Flash memorija | 32KB ATmega328P) od kojih je 0.5KB iskorišćeno za bootloader |
| SRAM | 2KB (ATmega328P) |
| EEPROM | 1KB (ATmega328P) |
| Takt procesora | 16MHz |

*B. Wi-Fi mikroprocesor ESP8266*

Wi-Fi modul – ESP8266 je samostalan SOC (sistem na čipu) sa integrisanim TCP/IP protokolom koji omogućava pristup WiFi mreži [4]. Svaki modul je fabrički programiran AT komandama i može se serijskom vezom povezati sa bilo koji mikrokontrolerom. Ovaj modul ima mogućnosti obrade i skladištenja podataka koje mu dozvoljavaju da se integriše sa senzorima i drugim uređajima preko njegovih GPIO pinova. Visoki stepen integracije na čipu sa minimalnim dodatnim kolima omogućava da modul zauzme minimalni prostor na štampanoj ploči.




Slika 3. Aplikacija za kontrolu i nadzor kuće [5].

Bitne karakteristike modula su:
-802.11b/g/n protokol
-integrisani TCP/IP protokol
-+19.5 dbm izlazne snage u 802.11b/g/n modu
-integrisani temperaturni senzor
-integrisani 32-bitni procesor (klok 52MHZ)
-spio 1.1/2.0 spi, UART, ADC, PWM
-STBC, 1x1MIMO, 2X1 MIMO
-prenos paketa < 2 ms
-pasivna potrošnja < 1.0mW
-potrošnja u predaji 215mA
-potrošnja u prijemu 62mA

Mikrokontroler očitava podatke sa senzora i preko Wi-Fi modula (slika 4) šalje signal do krajnjeg korisnika..



Slika 4. WiFi mikroprocesor ESP8266.

## III. POBOLJŠANJA POSTIGNUTA PROJEKTOM „PAMETNA“ KUĆA

Izrada projekta „Pametna“ kuća, pruža mogućnost daljinske kontrole, kao i proveru stanja ispravnosti povezanih sistema u kući u bilo kom trenutku, čime se bezbednost podiže na visok nivo (slika 5). Osim toga, štedi se vreme i energija za izvršenje zadataka kao što je polivanje cveća, otvaranje i zatvaranje garažnih vrata, roletni, uključivanje i isključivanje svetla. Primenom ovog projekta može se uprvaljati celim sistemom „jednim klikom“, tako da korisnik ima više vremena za posao, porodicu i slobodne aktivnosti.

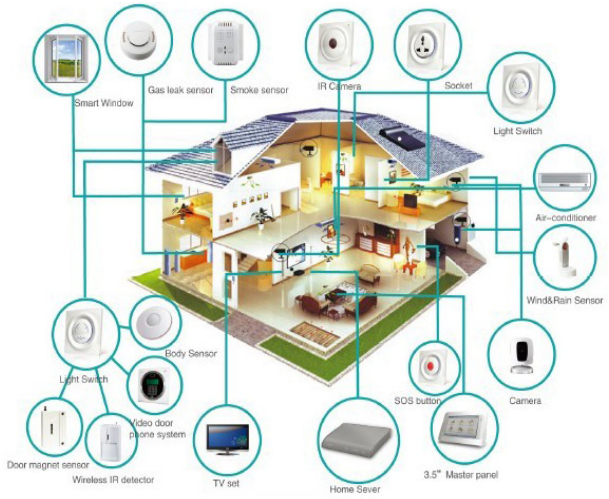
Nove tehnologije su dovele od ideje za ušavršavanje doma do projekta „Pametna“ kuća.

Karakteristike postojećih dosadašnjih kuća su:

1. nepraktičnost
2. nizak nivo bezbednosti
3. nemogućnost daljinske kontrole, odnosno provere trenutnog stanja ispravnosti u kući
4. nedostatak vremena za druge aktivnosti
5. neracionalna potrošnja električne energije

Karakteristike „pametne“ kuće:

1. daljinska kontrola i nadgledanja rada uređaja u kući,
2. precizno i lako podešavanje uređaja.,
3. pouzdanost i bezbednost objekta na visokom nivou,
4. mogućnost korišćenja objekta u dva rezima rada, sa i bez sistema „pametne“ kuće,
5. ušteda električne energije,
6. smanjenje vremena za regulisanje rada uređaja,
7. smanjenje obaveza korisnika,
8. izvršenje ekonomičnih stvari (regulacija i automatsko paljenje svetla),
9. dodatna zaštita od provale modom „Na odmoru sam“ koji simulira prisustvo korisnika (povremeno gasi i pali svetla pomera roletne).




Slika 5. Povezani sistemi – projekat “Pametna” kuća

## IV. ZAKLJUČAK

„Pametna“ kuća je uređaj koji se izdvaja svojim dobrim karakteristikama u odnosu na do sada korišćene standardne kuće. Realizacijom ovog složenog uređaja, postignut je napredak sinergijom informacionih tehnologija i mehatronike. Sve ovo nam pruža mogućnost daljinske kontrole i praćenja rada sa bilo kog mesta. Primena ovog sistema bi se trebalo proširiti na bolnice, kiničke centre i ostale ustanove gde je potrebma zaštita korisnika i uređaja datih objekata kao i njihovo ekonomično korišćenje.

## ZAHVALNICA

Rad je realizovan u laboratorijama elektrotehničke škole „Nikola Tesla“. Autori se zahvaljuju profesorima škole na pruženoj pomoći i podršci u radu, posebno profesorima Radmili Mitić, Dejanu Stamenkoviću kao i profesorima sa katedre za mikroelektroniku Elektronskog fakulteta u Nišu.

## LITERATURA

[1] Arduino Uno http://arduino.cc/en/main/software
[2] J.Krstić, D.Stajić: “Web Home Controller”, 2nd Olimpics in technical creativity of young people in Southestern Europe , Zrenjanin, 2014.
[3] Arduino IDE http://arduino.cc/en/main/software
[4] https://en.wikipedia.org/wiki/ESP8266
[5] https://www.pinterest.com/pin/201325045826484896/

# Ispitivanje uticaja debljine tunel oksida kod Flash EEPROM ćelije pomoću TCAD alata

Stefan Ilić, Miloš Marjanović

*Sadržaj* - Struktura Flash EEPROM ćelije sadrži plivajući gejt koji ima ulogu skladištenja naelektrisanja. Dovedeno naelektrisanje povećava napon praga i pomera prenosnu karakteristiku što omogućava razlikovanje logičke "1" i logičke "0". Pokazano je da razlika napona praga pri "1" i "0" direktno zavisi od količine naelektrisanja dovedenog na plivajući gejt. Izračunata je kapacitivnost između gejtova za datu strukturu. Pokazano je da debljina od 12 nm ima najbolje osobine prilikom upisa i brisanja podataka, manje debljine oksida imaju problem zadržavanja naelektrisanja, a veće debljine problem brisanja.

## I. UVOD

MEMORIJE koje sadržaj pamte i nakon nestanka napajanja se nazivaju permanentne memorije (eng. Non-Volatile Memories). Prvo je nastala Mask ROM (Read Only Memory) memorija, tj. maskom programiran ROM. Na čipu se formira ceo niz memorijskih ćelija u vidu matrice, zatim se prema zahtevu korisnika, formira maska za fotolitografski postupak koji definiše otvore za nanošenje linija veza između ćelija i napajanja. Ćelije koje treba da sadrže logičku "1" vezuju se za napajanje (nanosi se, preko otvora na masci, jako dopiran polisilicijum koji stvara kontakt), dok ćelije koje treba da sadrže logičku "0" ostaju nevezane za napajanje. U Mask ROM-u se podaci trajno smeštaju, proces je veoma skup i svaka izmena zahteva promenu maske za fotolitografiju.

Da bi se rešio problem skupe proizvodnje, nastaje PROM (Programmalbe ROM) memorija, ovde podatke smešta sam korisnik specijalizovanim električnim uređajima. Naime, PROM se proizvodi tako da sve ćelije imaju logičku "1", dakle sve su vezane za napajanje, prekidanje linija veza između pojedinih ćelija vrši se propuštanjem struje velike gustine, koja istopi datu liniju. Jednom programiran PROM nije moguće reprogramirati.

Sledeća memorija je EPROM (Erasable & Programmable ROM), kod nje je prvi put moguće brisanje podataka, koje se vrši izlaganjem memorije UV zracima. EPROM je prva memorija koja u strukturi ima plivajući gejt (eng. floating gate). Prilikom pisanja i čitanja memorija je bajt-adresibilna, dok proces brisanja obuhvata celu memoriju. Prilikom brisanja memorijski čip je neophodno izvaditi iz matične ploče i izložiti UV zračenju u trajanju od 20 minuta [1].

Stefan Ilić, student treće godine na modulu Elektronske komponente i mikrosistemi, Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: stefan.ilic@yahoo.com

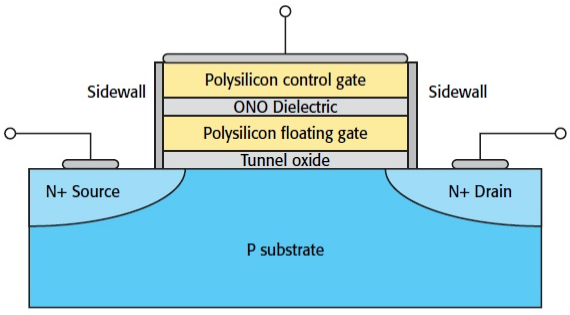
Miloš Marjanović, student doktorskih studija na modulu Elektronske komponente i mikrosistemi, Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija

Ovaj proces potpunog brisanja je neophodno uraditi iako je potrebno promeniti samo jedan bajt. Problem je prevaziđen razvojem sledeće vrste memorija tzv. EEPROM (Electrically Erasable PROM) memorija koje imaju mogućnost brisanja podataka električnim signalom. Sve operacije (upis, čitanje i brisanje) se odvijaju putem električkih signala, pri čemu je memorija bajt-adresibilna. Zbog veće funkcionalnosti, memorija ima veću dimenziju memorijske ćelije koju čine memorijski tranzistor (u kome se smešta informacija) i selektni trazistor (koji omogućuje da sve operacije budu bajt-adresibilne). EEPROM memorija ima tzv. dvo-tranzistorsku memorijsku ćeliju, koja je veća od memorijske ćelije EPROM-a (ima samo memorijski tranzistor), pa samim tim EEPROM ima manju gustinu pakovanja od EPROM-a.

Potom nastaju Flash EEPROM memorije kao kombinacija EPROM-a koja ima mogućnost brzog upisa i veliku gustinu pakovanja i EEPROM-a koja ima mogućnost brisanja podataka električnim signalom. Ove memorije su prvobitno nastale za potrebe vojske, gde je bilo potrebno napraviti memoriju koja će velikom brzinom (eng. flash, pa otuda i naziv) obrisati sve podatke kako ne bi podaci pali u ruke neprijatelja [2].

## II. PRINCIP RADA FLASH EEPROM ĆELIJE

Struktura Flash EEPROM ćelije je bazirana na posebnoj strukturi n-kanalnog MOS tranzistora, gde između gejta koji se ovde naziva kontrolni gejt (eng. control gate) i kanala postoji još jedan gejt koji je izolovan sa svih strana tzv. plivajući gejt, kao što je prikazano na slici 1.

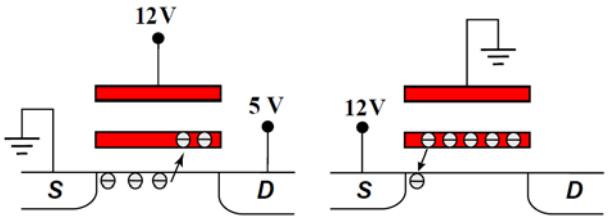



Slika 1. Struktura Flash EEPROM ćelije u 2D.

Između plivajućeg gejta i kanala se nalazi tanak oksid (eng. Tunnel oxide) kroz koji protiču elektroni prilikom upisa i brisanja informacija. Debljina ovog oksida se kreće

ispod 20 nm [1]. Između kontrolnog gejta i plivajućeg gejta se nalazi izolator, obično je to strukutura oksid/nitrid/oksid ($SiO_2/Si_3N_4/SiO_2$, skraćeno ONO). Da bi plivajući gejt bio potpuno izolovan sa strane se vrši depozicija oksida (eng. Sidewall). Kontrolni i plivajući gejt su obično napravljeni od polisilicijuma koji je jako dopiran (n-tip).

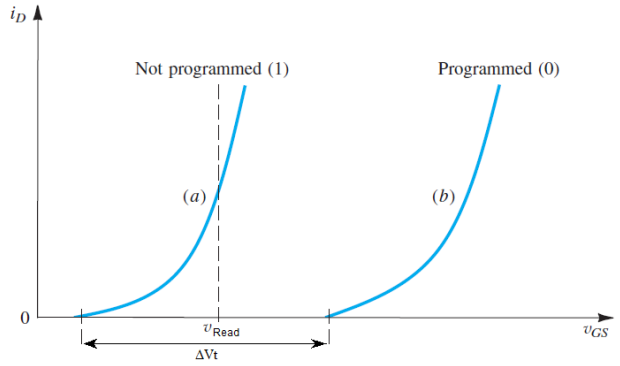
Prilikom upisa na kontrolni gejt se dovodi napon upisa, obično 12 V. Na drejn je doveden napon od 5 V, a sors je uzemljen, tako da struja protiče kroz tranzistor, elektroni koji se nađu u kanalu pod dejstvom električnog polja kontrolnog gejta koje je jače od električnog polja drejna bivaju privučeni, zbog tankog oksida i jakog električnog polja određeni broj elektrona dolazi do plivajućeg gejta, gde ostaju zarobljeni (slika 2 levo).




Slika 2. Proces upisa (levo) i brisanja podataka (desno).

Prilikom brisanja podataka kontrolni gejt je uzemljen. Na sors je doveden napon od 12V, a drejn je otkačen (slika 2 desno). Elektroni pod dejstvom električnog polja sorsa bivaju odvučeni sa plivajućeg gejta, na taj način je memorijska ćelija ispražnjena.

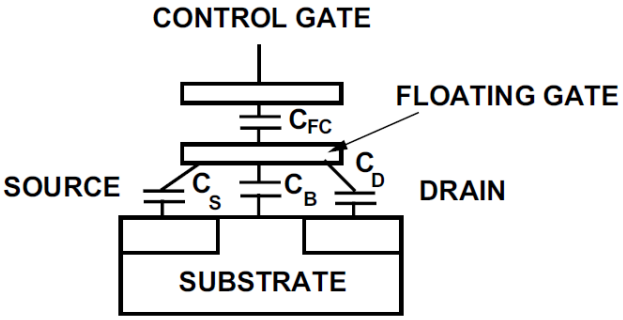
Na slici 3 su prikazane prenosne karakteristike Flash EEPROM ćelije pre i posle dovođenja naelektrisanja na plivajući gejt tj. pre i posle upisa (programiranja).




Slika 3. Prenosna karakteristika pre (a) i posle upisa podataka (b).

Kada na plivajućem gejtu nema naelektrisanja, tada će pri dovođenju napona čitanja ($V_{read}$) tranzistor provesti i detektovaće se struja na izlazu, što se naziva logičkom "1". Nakon dovođenja naelektrisanja na plivajući gejt (upis podataka), prenosna karakteristika se pomera u desno za $\Delta V_t$, tako da pri dovođenju napona čitanja tranzistor ne može da provede, što predstavlja logičku "0". Napon čitanja ima vrednost između napona praga pri "1" i napona praga pri "0" [3]. Razlog pomerene prenosne karakteristike nakon upisa je zato što elektroni koji se nalaze na plivajućem gejtu odbijaju elektrone koji treba da oforme kanal, samim tim je potrebno dovesti veći napon na kontrolni gejt kako bi se nadjačalo električno polje koje potiče od elektrona na plivajućem gejtu i tako došlo do formiranja kanala.

Pri analizi Flash EEPROM ćelije korišćen je jednodimenzionalni model plivajućeg gejta (slika 4) [2].




Slika 4. Jednodimenzionalni model plivajućeg gejta.

Iz ovog modela ekvivalentna kapacitivnost plivajućeg gejta ($C_T$):

$$C_T = C_{FG} + C_D + C_S + C_B \quad (1)$$

gde je:
$C_{FG}$ - kapacitivnost između kontrolnog i plivajućeg gejta.
$C_D$ - kapacitivnost između drejna i plivajućeg gejta.
$C_S$ - kapacitivnost između sorsa i plivajućeg gejta.
$C_B$ - kapacitivnost između supstrata i plivajućeg gejta.

Sve kapacitivnosti su vezane paralelno sa plivajućim gejtom, zbog toga je ekvivalentna kapacitivnost jednaka zbiru svih kapacitivnosti.

Na osnovu datog modela dolazimo do sledeće jednačine, postupak izvođenja ovde nećemo obrađivati, koja povezuje količinu naelektrisanja na plivajućem gejtu ($Q$) i razliku između napona praga pre i posle upisa [2]:

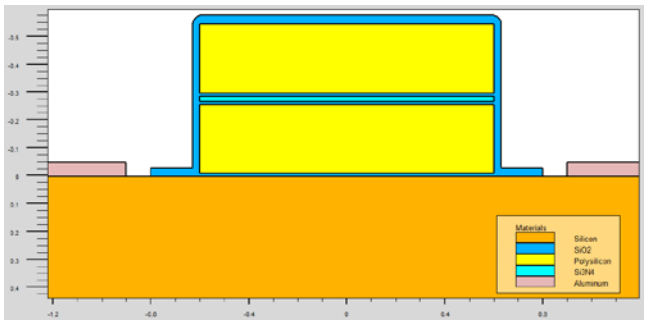
$$\Delta V_t = -\frac{Q}{C_{FG}} \quad (2)$$

Pošto je kapacitivnost konstantna veličina za datu strukturu memorijske ćelije, ova jednačina predstavlja linearnu zavisnost razlike napona praga pre i posle upisa od dovedenog naelektrisanja na plivajući gejt. Kako napon praga pre upisa ima konstantnu vrednost pri fiksnom naponu na drejnu [4], to znači da se sa povećanjem negativnog naelektrisanja na plivajućem gejtu nakon upisa prenosna karakteristika pomera u desno, tj. napon praga se povećava. Ova jednačina ima važnu ulogu u objašnjavanju načina rada Flash EEPROM memorijske ćelije.

## III. Model i detalji proračuna

Svi rezultati su dobijeni korišćenjem sledećih simulatora u Silvaco TCAD alatu: Athena, DevEdit i Atlas

[5]. Tehnološki parametri za dobijanje strukture su preuzeti iz primera koji se nalazi u Silvacu [6], a struktura je prikazana na slici 5. Korišćena je velika gustina mreže kako bi se dobili što precizniji rezultati, rastojanje između linija mreže u kritičnim oblastima je bila 5 nm, dok je u tunel oksidu bilo 0.1 nm.

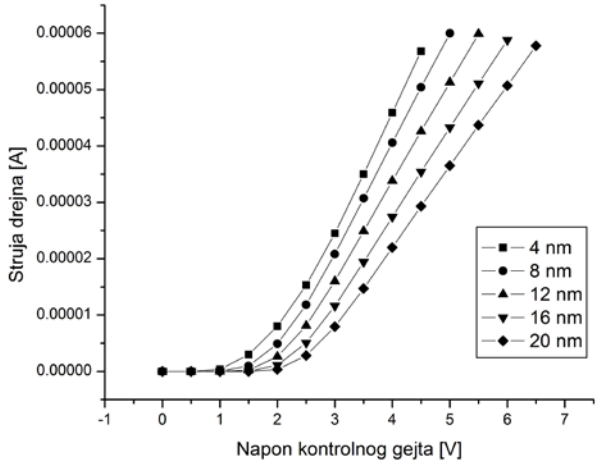



Slika 5. 2D struktura Flash EEPROM memorijske ćelije.

Prilikom simuliranja date strukture vršena je varijacija debljine tunel oksida od 4 nm do 20 nm. Debljine ispod 4 nm nisu razmatrane jer tada nije dominantno Fowler-Nordheim tunelovanje [7], čiji je model korišćen pri simulaciji.

## III. REZULTATI I DISKUSIJA

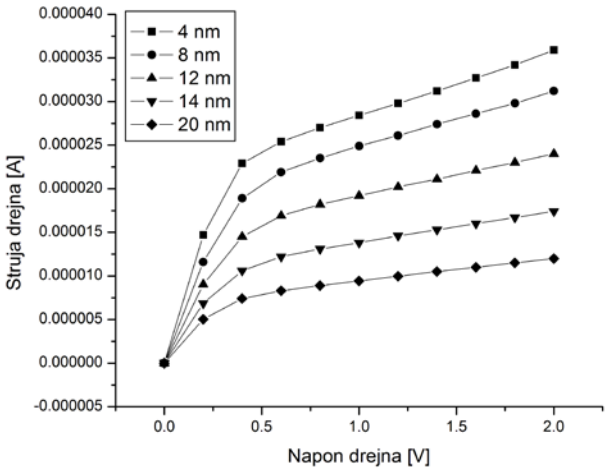
Na slici 6 predstavljene su prenosne karakteristike Flash EEPROM ćelija pre dovođenja naelektrisanja na plivajući gejt ("1") za različite debljine oksida, pri $V_{DS} = 0.5V$. Primećujemo da sa povećanjem debljine oksida raste napon praga, to je posledica slabijeg delovanja električnog polja kontrolnog gejta na kanal zbog veće udaljenosti od kanala.




Slika 6. Prenosne karakteristike Flash EEPROM ćelija pri "1" sa različitim debljinama tunel oksida.

Iz tog razloga je, sa povećanjem debljine tunel oksida, potrebno dovesti sve veći napon na kontrolni gejt kako bi se oformio kanal i tranzistor proveo struju.

Takođe, primećujemo da se prenosne karakteristike sve više krive sa povećanjem debljine oksida. To je posledica sve veće otpornosti u kanalu. Ako posmatramo izlazne karakteristike Flash EEPROM ćelije za različite debljine tunel oksida pri stalnom naponu na kontrolonom gejtu od $V_{cg} = 3V$ (slika 7), možemo uočiti da otpornost kanala raste sa povećanjem debljine tunel oksida.

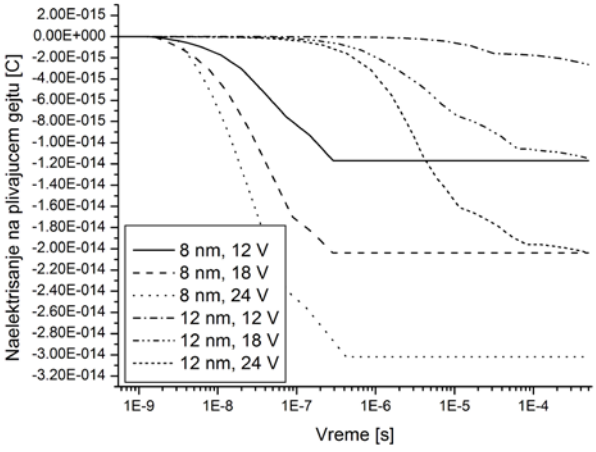



Slika 7. Izlazne karakteristike Flash EEPROM ćelija pri Vcg=3V sa različitim debljinama tunel oksida.

Pri malom naponu na drejnu kanal se može posmatrati kao otpornik, čija je otpornost određena dimenzijama i koncentracijom elektrona u kanalu [4]. Razlog sve veće otpornosti, sa povećanjem debljine oksida, leži u sve slabijem transverzalnom električnom polju kontrolnog gejta koje deluje na elektrone u kanalu.

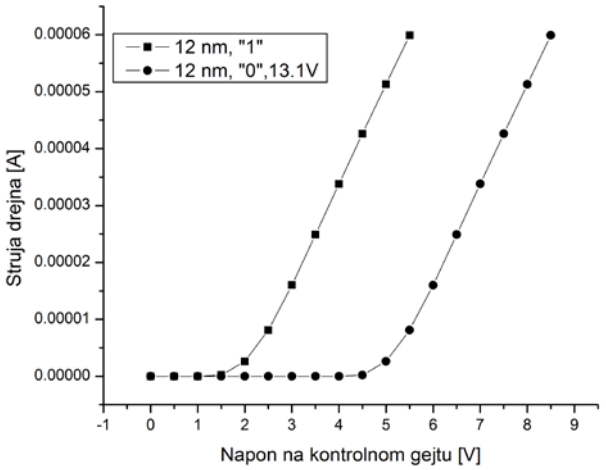
### *A. Upis podataka*

Prilikom upisa podataka dovodi se naelektrisanje na plivajući gejt pod dejstvom jakog električkom polja kontrolonog gejta, pri $V_{DS} = 5V$. Na slici 8 su prikazane zavisnosti količine naelektrisanja koje su dovedene na plivajući gejt u toku vremena, za dve debljine oksida, pri različitim naponima na kontrolnom gejtu.




Slika 8. Zavisnost količine naelektrisanja na plivajućem gejtu u od vremenu, za različite debljine tunel oksida i napona na kontrolnom gejtu.

Primećujemo da sa povećanjem napona na kontrolnom gejtu raste količina naelektrisanja koja je dovedena na plivajući gejt. Takođe, uočavamo da sa povećanjem debljine tunel oksida, za isti napon na kontrolom gejtu, dolazi manja količina naelektrisanja na plivajući gejt. Na osnovu jednačine (2) izračunata je kapacitivnost između kontrolnog i plivajućeg gejta na osnovu prethodnih rezultata: $C_{FG} = (1.458 \pm 0.002) fF$. Ova vrednost je potvrđena određivanjem ekivalentne kapacitivnosti tri serijski vezana kondenzatora tj. dielektrika koji se nalaze između gejtova (ONO), na osnovu poznatih debljina slojeva, dielektričnih konstanti ($\varepsilon_{SiO_2}$= 3.9; $\varepsilon_{Si_3N_4}$= 7.5) i površine gejtova (ploče kondenzatora).

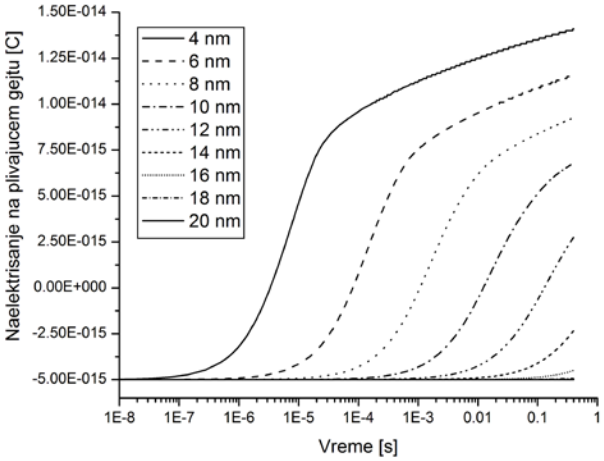



Slika 9. Prenosne karakteristike Flash EEPROM ćelije pri "1" i pri "0" za debljinu tunel oksida od 12 nm i 13.1V napona na kontrolnom gejtu prilikom upisa.

Pri proizvodnji memorijskih ćelija uzima se da je $\Delta V_t = 3V$ [2]. Na osnovu prethodno izračunate vrednosti kapacitivosti između gejtova, možemo na osnovu jednačine (2) dobiti koliku količinu naelektrisanja je potrebno dovesti na plivajući gejt $Q = 4.374 \times 10^{-15}\ C$. Empirijskim putem je dobijeno da sa vrednošću napona $V_{cg} = 13.1\,V$ na kontrolnom gejtu, pri debljini oksida od 12 nm, dolazi tačno izračunato naelektrisanje na plivajući gejt. Na slici 9 je to i prikazano, za ovakve karakteristike napon čitanja bi iznosio oko $V_{read} = 3.5\,V$. Za korišćenje tanjih tunel oksida za dobijanje datog naelektrisanja bilo bi potrebno dovesti manji napon, ali nastaje problem zadržavanja naelektrisanja na plivajućem gejtu.

*B. Brisanje podataka*

Smanjenje količine naelektrisanja na plivajućem gejtu se konstantno odvija, ali je ona zanemarljiva za deblje okside. Donja granica je oko 7-8 nm [7]. Prilikom brisanja podataka kontrolni gejt je uzemljen, drejn je otkačen (u simulaciji se za drejn vezuje otpornik od $R = 10^{20}\ \Omega$), dok se na sors dovodi pozitivan napon od 12 V. Ispitivanje brzine brisanja podataka za različite debljine tunel oksida je prikazano na slici 10. Uočavamo da se sa povećanjem debljine tunel oksida plivajući gejt sporije prazni.




Slika 10. Zavisnost količine naelektrisanja na plivajućem gejtu od vremena, prilikom brisanja, za različite debljine tunel oksida.

Primećujemo da debljine tunel oksida veće od 12 nm ne mogu da se isprazne za dati napon na sorsu i predviđeno vreme.

## IV. Zaključak

Ispitivanjem strukture Flash EEPROM ćelije sa različitim debljinama oksida došli smo do zaključka da debljina tunel oksida od 12 nm predstavlja najbolji kompromis između energije sa kojom je potrebno dovesti željeno naelektrisanje na plivajući gejt kao i vreme potrebno za zadržavanje naelektrisanja kako bi memorijska ćelija ispravno funkcionisala tokom dugog perioda.

## Zahvalnica

Zahvaljujemo se Katedri za mikroelektroniku na Elektronskom fakultetu, Univerziteta u Nišu na pruženoj velikoj pomoći i saradnji.

## Literatura

[1] William Brown, Joe Brewer, *Nonvolatile Semiconductor Memory Technology*, New York, Institute of Electrical and Electronics Engineers: IEEE Press, 1998.
[2] G. Campargo, R. Micheloni, D. Novosel, *VLSI - Design of Non-Volatile Memories*, Springer-Verlag Berlin Heidelberg: Springer, 2005.
[3] A. Sedra, K. Smith, *Microelectronics Circuits*, 6th edition: Oxford University Press, 2010.
[4] Zoran Prijić, Aneta Prijić, *Uvod u poluprovodničke komponente i njhovu primenu*, Niš, Elektronski fakultet, Univerzitet u Nišu: Edicija Osnovni udžbenici, 2014.
[5] Silvaco, T. C. A. D. "ATLAS Device Simulator."
[6] Silvaco TCAD Examples: Flash EEPROM Programming and Erasing. Dostupno na: https://www.silvaco.com/examples/tcad/section17/example1/index.html
[7] Dragica Vasileska, *Modeling of Non-Volatile Memories With Silvaco*, Arizona State University. Dostupno na: https://nanohub.org/resources/11612/download/memories_lecture_notes.pdf.

# DTW algoritam u formi glasovnog prepoznavanja

Risto Keković

*Sadržaj* – U ovom radu je predstavljen i objašnjen DTW algoritam i način na koji se on primenjuje u tehnikama za prepoznavanje govora. U praktičnom delu biće objašnjena implementacija ovog algoritma kao i rezultati koji se dobijaju.

## I. UVOD

Govor je najprirodniji način komunikacije kod ljudi. Iako je ova tvrdnja istinita još od nastanka same civilizacije, pronalazak i proširena upotreba telefona, audio-foničnih medija za smeštanje podataka, radija i televizora su dali dodatnu važnost glasovne komunikacije i glasovnog procesiranja. Napredak u tehnologiji obrade digitalnog signala doprineo je korišćenju obrade govora za razne potrebe kao što su kompresija govora, poboljšanje, sinteza i prepoznavanje.

U ovom radu je implementiran DTW (Dynamic Time Warping) algoritam koji koristi MFCC (Mel Frequency Cepstral Coefficients) za izdvajanje karakteristika iz govora. Prikazani su i rezultati eksperimenata dobijenih primenom ovog algoritma.

## II. IZDVAJANJE KARAKTERISTIKA

Srž rešenja leži u tome da se zvuci koje proizvodi čovek filtriraju na osnovu oblika vokalnog trakta, što uključuje jezik, zube, itd. Ovaj oblik određuje kakav će zvuk nastati. Oblik vokalnog trakta se može predstaviti spektrom, a zadatak MFCC-a jeste da ga prikaže na pogodan način. U nastavku je opisano dobijanje **Melovih frekventnih "kepstralnih" koeficijenata** (*Mel Frequency Cepstral Coefficients*, MFCC).

Reč "kepstar" je igra reči i dolazi od okretanja redosleda prva četiri slova reči "spektar", jer predstavlja inverzne Furijeove transformacije logaritma spektra signala. Operacije nad kepstrom se slično često nazivaju analiza kvefrence (od "frekvence") i lifterovanje (od "filterovanje").

MFCC je u širokoj upotrebi za automatsko prepoznavanje govora, jer daje reprezentaciju izgovorenih fonema koje ne zavise od boje glasa i visine tona. Uspeh MFCC-a je i u mogućnosti kompaktnog zapisa amplitudskog spektra govora. Svaki korak u izdvajanju MFC koeficijanata motivisan je perceptualnim doživljajem ljudskog govora od strane čoveka.

Risto Keković je student na Elektronskom fakultetu, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija. E-mail: risto.kekovic@gmail.com

Računanje MFCC-a se može opisati kroz sledeći niz koraka. Polazi se od talasnog oblika.

- Deljenje govornog signala na okvire (frejmove), obično primenom prozora nad fiksniranim intervalima. Cilj je modelovanje malih (obično oko 20ms dugih) odeljaka signala za koje se statistički može reći da su stacionarni. Naravno, zvučni signal se konstantno menja, tako da nikad nije zapravo stacionaran. Zato se i uzimaju ovoliki intervali. Eksperimentalno je dokazano da su dovoljno dugi da sadrže dovoljno uzoraka za procenu spektra, a dovoljno kratki da se ne menjaju previše tokom vremena. Odmerci se provlače kroz funkciju prozora, uglavnom Hamingovu. Na ovaj način se uklanjaju ivični efekti. Rezultat ove faze je kepstralni faktor za svaki okvir.
- Sledeći korak je računanje diskretne Furijeove transformacije (DTF) za svaki okvir. Ovde je od interesa samo logaritam spektra amplitude. Faza se odbacuje jer su perceptualna istraživanja pokazala da amplituda igra mnogo značajniju ulogu, tj. da nosi mnogo više informacija o govoru nego faza. Uzima se logaritam zbog načina na koji ljudsko uho doživljava jačinu zvuka.
- Posle ovoga se "pegla" spektar i naglašavaju se samo frekvence koje su u perceptualnom smislu od značaja. Ovo se postiže skupljanjem (na primer) 256 spektralnih komponenti u (na primer) 40 frekventnih banaka. Ove banke nisu na jednakim rastojanjima jedne od druge, jer čovekov slušni aparat mnogo bolje razlikuje promenu u višim nego u nižim frekvencijama. Zato su banke mnogo tesnije poređane u oblastima nižih frekvencija. Skala u koju se prevodi originalni spektar se naziva Melova skala. Skoro je linearna izpod jednog kiloherca, a posle toga postaje jasno logaritamska. Prevođenje frekvence u Melovu skalu obavlja se formulom $M(f) = 1125 * \ln(1 + f/700)$. Naravno, moguće je i inverzno prevođenje.
- Među komponentama koje se dobijaju javlja se velika korelacija. Utvrđeno je da je dovoljna samo polovina kepstralnih karakteristika. Ovime se smanjuje broj parametara sistema što ubrzava izračunavanje i smanjuje neophodni memorijski prostor.
- Dobijene karateristike se transformišu. Mada je moguće primeniti različite transformacije, za

prepoznavanje govora se najbolje pokazala diskretna kosinusna transformacija (DCT).

## III. DTW ALGORITAM

*Dynamic time warping* (dinamičko savijanje vremena, DTW) je tehnika za nalaženje optimalnog načina na koji se dve vremenski zavisne sekvence mogu poravnati, tako da budu što sličnije. Treba da otkloni vremensku zavinost, odnosno da "savije" vreme u smislu da proizvoljne sekvence produži, odnosno skrati, kako bi se postiglo bolje poklapanje. [1] Ovo se uklapa u potrebe poklapanja uzoraka ljudskog glasa, jer se na taj način zanemaruje trajanje izgovorene reči, kao i akcenat. Ilustracije radi, izgovor reči "elfaak" i "eelfak" bi imao 100% poklapanje (odnosno imao bi cenu jednaku nuli) jer se produžavanjem slova *a*, i skraćenjem slova *e* iz prve reči dolazi u drugu.

Formalno rečeno, cilj DTW-a je da uporedi dve vremenski zavisne sekvence X = (x1, x2, ... xN) dužine N i Y = (y1, y2, ..., yM), dužine M. U opštem slučaju, ovi nizovi mogu biti karakteristike odmerene nad tačkama u vremenu na jednakoj razdaljini. U praksi, ovo znači da se radi o diskretnim signalima koji su opisani nekim vektorom. Ako se prostor karakteristike obeleži sa F, može se reći reći da DTW algoritam treba da kao rezultat da cenu c, koja predstavlja preslikavanje Dekartovog proizvoda dva ovakva skupa (F × F) u skup ne-negativnih realnih brojeva. Kaže se da između dva podskupa ovih prostora (npr. dva različita snimka izgovorene reči) postoji dobro preklapanje ako je cena mala, i loše preklapanje ako je cena velika.

Niz uređenih parova (x, y), pri čemu su x i y elementi vektora X i Y iz prostora F čije se upoređivanje vrši, naziva se **savijenom putanjom**. Ovaj lanac može se predstaviti matricom. Na primer, za pomenute reči "eelfak" i "elfaak", putanja bi se savijala na sledeći način.

```
  E E L F A K
E x x . . . .
L . . x . . .
F . . . x . .
A . . . . x .
A . . . . x .
K . . . . . x
```

Horizonalna linija predstavlja deo gde je "vreme stalo" za sekvencu *EE* (ili, posmatrano iz drugog ugla, "produžilo" se za *E*), kako bi se dobilo dobro preklapanje za sekvencu *LF*. Slično se zatim "vreme produžava" za *A*.

Mada su intuitivna, u nastavku su navedena ograničenja koja mora da zadovolji savijena putanja kako bi bila od koristi za prepoznavanje govora.

- Granični uslov: prvi uređeni par je (1, 1) a poslednji uređeni par je (N, M), gde su N i M, dužine vektora između kojih se vrši preklapanje.
- Uslov monotonosti: niz sačinjen od prvih, odnosno drugih, elementa uređenih parova savijene putanje mora biti neopadajući.

Prvi uslov se može ilustrovati poređenjem reči *elfak* i *elfakultet*. Jasno je da putanja glatko "klizi" niz sekvencu slova *elfak*, ali ne može da se završi tu: mora se izvršiti "savijanje" u smislu da se sekvenca *ultet* "skrati" i "umetne" zajedno sa *k*.

```
  E L F A K U L T E T
E x . . . . . . . . .
L . x . . . . . . . .
F . . x . . . . . . .
A . . . x . . . . . .
K . . . . x x x x x x
```

Drugi uslov je takođe očigledan: nema vraćanja unazad. Na primer, u rečima "ana" i "banana", put ne može izgledati kao što je prikazano s leve strane na sledećoj slici, već kao što je prikazano zdesna.

```
  B A N A N A        B A N A N A
A x x . . . .      A x x . . . .
N . . x . x .      N . . x . . .
A . . . x . x      A . . . x x x
```

Ovi uslovi su potrebni i dovoljni u opštem slučaju, ali se obično nameće još jedan uslov, a to je uslov veličine koraka. Odnosno, koliko se daleko u matrici može "skočiti". Ovo zavisi od konkretne potrebe u zavinosti od toga kolika je preciznost i brzina potrebna. Standardno se uzimaju koraci (1, 0), (0, 1) i (1, 1) (desno, dole, desno-dole dijagonalno).

Nije uvek moguće naći savršeno poklapanje (sa cenom jednakom nuli), kao što je primer u rečima *eelfak* i *elfaak*. Na primer, između reči *truba* i *gruba* nemoguće je preklopiti prva slova, ali je putanja i dalje očigledno strogo dijagonalna.

```
  T R U B A
G x . . . .
R . x . . .
U . . x . .
B . . . x .
A . . . . x
```

Zato se, osim putanje, govori i o njenoj ceni. Jasno je da je neophodno definisati metriku na osnovu koje će se porediti dva uzorka koja se trenutno ispituju. Na primer, za rastojanje između dva vektora se može uzeti Euklidova distanca između njih.

Cena puta se definiše kao suma svih pojedinačnih cena između odgovarajućih vektora na putanji. Zadatak DTW algoritma je da pronađe optimalnu cenu i odgovarajuću optimalnu putanju. U opštem slučaju optimalna savijena putanja ne mora da bude jedinstvena, ali se ovakve situacije retko sreću u praksi.

## III. IMPLEMENTACIJA

Za praktičnu implementaciju DTW algoritma je iskorišćen programski jezik *Python*. Razlog izbora ovog programskog jezika je njegova portabilnost, odnosno mogućnost izvršavanja na različitim platformama (dobar primer je *Raspberry Pi*), kao i sve veća popularnost u poslednjih nekoliko godina. Pored toga, *Python* poseduje veliki broj modula i biblioteka koji se mogu iskoristiti za proširivanje funkcionalnosti jezika i daje dodatne mogućnosti za kompleksna matematička izračunavanja, kao i prikaz dijagrama i drugih vizuelnih reprezentacija. Ovo je, između ostalog, jedan od razloga zbog kojih je *Python* izuzetno popularan u oblasti *Data Science*-a, obzirom da može da procesira velike količine podataka u relativno kratkom vremenskom periodu.

Pored samog jezika, u primerima koji su navedeni korišćeno je i interaktivno okruženje *IPython* koje predstavlja nadskup samog programskog jezika *Python*, obzirom da dodaje još neke funkcionalnosti, a takođe omogućava i CLI (eng. *Command Line Interface*), odnosno interaktivno unošenje komandi i štampanje rezultata i međukoraka. Ono što on nudi pored standardnih prikaza rezultata je i interaktivni prikaz dijagrama i slika u okviru samog *shell*-a (takozvani *inline* pristup).

Sistem za prepoznavanje govora može se implementirati korišćenjem tehnika MFCC-a i DTW-a.

Sam eksperiment se izvršava na sledeći način: pre svega, za testiranje prepoznavanja govora će biti potrebna baza podataka sa nekoliko različitih govornika i nekoliko različitih reči reči, i to će biti baza znanja koju algoritam koristi u fazi testiranja. Za potrebe eksperimenta je korišćena baza podataka od 12 Francuskih reči izgovorenih između 20 do 25 puta (broj uzoraka varira za svaku reč) od strane različitih govornika u različitim uslovima. Pored toga se učitava i lista labela za sve izgovorene reči. Te labele se koriste kasnije da bi se proverilo da li je došlo do pogotka, odnosno da li je reč koju je vratio algoritam ista kao i izgovorena reč.

U suštini, ovaj eksperiment simulira testiranje nove izgovorene reči i poređenje tog izgovora sa izgovorima koji se nalaze u bazi podataka. Nasumično se za svaku od 12 reči bira po jedan izgovor i zatim se traži najbolje poklapanje u bazi, odnosno najniža cena puta. Nakon što se ona pronađe, proveravaju se labele izgovorene reči i reči za koju je dobijena najmanja cena. Ukoliko su one iste, dobija se pogodak. U suprotnom, došlo je do promašaja. Na kraju se gleda odnos pogodaka i ukupnog broja uzoraka u eksperimentu, i na osnovu toga se određuje preciznost algoritma.

Izabrana baza podataka daje dovoljnu raznovrsnost na osnovu kojih se mogu sprovesti realni testovi.

Korišćenjem ovih funkcija može se istestirati algoritam na većem broju uzoraka, pri čemu se dobijaju sledeći rezultati:

```
Number of samples: 12
Number of recognized words: 10
Number of misses: 2
Recognition rate 83.3333333333%
```

Kao što se može videti, od 12 testiranih uzoraka, 10 je uspešno prepoznato uz 2 promašaja. Dobijeni procenat prepoznavanja varira u određenom opsegu. Ovde će biti prezentovani rezultati nekoliko uzastopnih iteracija prepoznavanja korišćenjem DTW algoritma sa ovom bazom podataka.

Ukoliko se udvostruči broj uzoraka (odnosno uzme se po 2 uzorka za svaku reč), doći će do smanjenja procenta prepoznavanja, ali je i dalje pretežno veći broj pogodaka u odnosu na promašaje.

```
Number of samples: 24
Number of recognized words: 18
Number of misses: 6
Recognition rate 75.0%
```

Sa po 3 uzorka za svaku reč dobija se 36 ulaznih uzoraka. Kod ovog eksperimenta dolazi do jednog zanimljivog zapažanja. Naime, i pored povećanja broja uzoraka, broj promašaja je u odnosu na prethodni eksperiment sa manjim brojem uzoraka ostao isti. To dovodi do većeg procenta uspešnosti za ovaj eksperiment. Dakle, od 36 uzoraka dobija se 30 pogodataka, odnosno procenat prepoznavanja oko 83%

```
Number of samples: 36
Number of recognized words: 30
Number of misses: 6
Recognition rate 83.33333333334%
```

Najzad, sa 48 test uzoraka (odnosno po 4 uzorka za svaku reč), procenat je i dalje iznad 80%, što pokazuje da uspešnost algoritma ne opada sa povećanjem broja uzoraka koji se testiraju.

```
Number of samples: 48
Number of recognized words: 39
Number of misses: 9
Recognition rate 81.25%
```

U proseku ovo daje približno oko 80.73% preciznosti algoritma. U ostalim iteracijama koje su testirane dobijaju se neznatno male promene u procentu uspešnosti, ali u proseku je to i dalje negde oko 80%.

Razlog za ovakve varijacije u različitim iteracijama pre svega leži u sličnosti određenih izgovora nekih reči. To je između ostalog i razlog zbog čega je izabran Francuski jezik sa kompleksnim izgovorom reči što dovodi do toga da neke reči zvuče veoma slično. Samim tim se testira i efikasnost algoritma. Ukoliko se promeni baza podataka na neki drugi jezik ili na neki drugi fond reči, to može da dovede do boljih ili lošijih performansi. Iz ovoga sledi da

performanse sistema za prepoznavanje ne zavise od samog algoritma i njegove implementacije već i od trening podataka nad kojima se on izvršava.

Dobar sistem mora da vodi računa o tome da ključne reči koje on poznaje i može da prepozna ne budu previše slične da ne bi došlo do rizika da se jedna glasovna komanda ne prepozna kao neka druga. Ovo očigledno može da predstavlja problem u bazama sa velikim brojem reči gde je nekada nemoguće rešiti ovaj problem. Međutim, nije uvek potrebno da postoji stoprocentno prepoznavanje. U nekim situacijama procenat prepoznavanja preko 80% može da bude sasvim zadovoljavajući. Naravno, ako bi se ovakav sistem koristio za biometrijsku identifikaciju na bankomatima ili nekim drugim sistemima koji zahtevaju izuzetno veliku preciznost prepoznavanja, tada bi bilo neophodno da procenat prepoznavanja bude veći.

Drugi razlozi za ovakav procenat uspešnosti leže i u spoljnim efektima, odnosno u šumovima i buci koja se javlja u nekim od uzoraka, kao i različite brzine i dužine izgovaranja samih reči. To je nešto na šta često ne može da se utiče, obzirom da je retko neko u prilici da snimi glasovnu komandu bez bilo kakve interferencije spolje sredine (primer za to je bankomat koji se nalazi na ulici gde stalno prolaze automobili i mogu se čuti glasovi drugih ljudi koji su u prolazu).

## IV. ZAKLJUČAK

U ovom radu testiran je pristup korišćenjem MFCC-a i DTW algoritma za prepoznavanje. Na osnovu dobijenih rezultata, može se zaključiti da on daje dobre rezultate prepoznavanja. U sistemima u kojima je od ključne važnosti da postoji izuzetno visok procenat uspešnosti prepoznavanja (preko 95%), ovo se ipak ne bi moglo koristiti. Ovakvi sistemi zahtevaju mnogo veću preciznost, tako da ono što definitivno daje bolje rezultate su sistemi koji implementiraju neku formu mašinskog učenja u njihovim algoritmima. Ovo im obezbeđuje da se „naviknu“ na određenu osobu i način na koji ona priča i da prepoznaju karakteristike samog glasa i govora, što daje mnogo bolje rezultate u procesu testiranja odnosno prepoznavanja. Međutim, implementacija jednog ovakvog sistema zahteva mnogo složeniji pristup i samim tim predstavlja temu za sebe.

## LITERATURA


[1] Lazar Ljubenović “Prepoznavanje glasa korišćenjem DTW i MFCC-a“, Maj 2017.
URL: https://github.com/lazarljubenovic/pi-proj (Preuzeto: 15.09.2017.)

[2] Aleksandar Ilijev / mentor prof.dr.Radomir Stanković “Primena korelacije za proveru autentičnosti korisnika na osnovu glasa“, diplomski rad na katedri za računarstvo Elektronskog fakuleta Univerziteta u Nišu, Oktobar 2016.

[3] Shivanker Dev Dhingra, Geeta Nijhawan, Poonam Pandit “Isolated Speech Recogniton Using MFCC and DTW”, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 2, Issue 8, Avgust 2013.

[4] Jan Cernocký, Valentina Hubeika “Speech Recognition: Intro and DTW”, lecture on Faculty of Information Technology, *Brno* University of Technology, Februar 2013.

[5] Darshan Mandala, Pravin Gareta “Speaker Recognition Using MFCC and Vector Quantization Model”, Department of Electrical Engineering Electronics & Communication Engineering Program Institute of Technology, Nirma University, Maj 2011.

[6] Pierre Rouanet “DTW (Dynamic Time Warping)”, Jul 2014.
URL: https://github.com/pierre-rouanet/dtw (Preuzeto: 22.09.2017.)

[7] Dr. James Glass, Prof. Victor Zue “Dynamic Time Warping & Search”, lecture on course Automatic Speech Recognition, Massachusetts Institute of Technology (MIT), Mart 2003.

# Razvoj Pygame aplikacije na Raspberry Pi-u

Marko Stanković

*Sadržaj* - Raspberry Pi je embeded računar koji sadrži sistem na čipu sa ARM procesorom. U ovom radu su ispitane mogućnosti Raspberry Pi platforme za razvoj igara korišćenjem programskog jezika Pyton. Kreirana je Pong igra pomoću Pygame modula. Prikazana je realizacija aplikacija na Raspberry Pi platformi.

## I. UVOD

Razvoj sistema na čipu (System on Chip - SoC) sa niskom cenom omogućio je sve veću upotrebu male Raspberry Pi platforme. Korišćenje Linux-a kao operativnog sistema čini Raspberry Pi platformu pristupačnom, prilagodljivom i moćnom za veliki broj aplikacija. Embeded sistemi zajedno sa Linux-om omogućavaju lak razvoj uređaja koji mogu odgovoriti budućim izazovima u pametnim zgradama, pametnim gradovima, Internet of Things, robotici, interakciji između ljudi i računara, naprednim sistemima vozila i u mnogim drugim aplikacijama. Svaki embeded sistem se sastoji iz namenski projektovanog hardvera i pratećeg softvera koji se na njemu izvršava. Da bi se mogao projektovati embeded sistem koji rešava konkretan inženjerski problem, neophodno je poznavanje složenosti problema i raspoloživih algoritama za njegovo rešavanje.

U ovom radu je kreirana igra nazvana Pong korišćenjem programskog jezika Python i modula Pygame. Na praktičnom primeru je pokazan razvoj jednostavne multimedijalne aplikacije kao i programiranje Raspberry Pi-a. Realizacija ovog zadatka obuhvata sledeće faze: kreiranje praznog ekrana, crtanje objekta na ekranu, kretanje objekata, definisanje interakcije između objekata i testiranje krajnje verzije kôda.

Struktura ovog rada organizovana je u pet poglavlja. Nakon uvodnog dela, u drugom poglavlju rada predstavljeni su karakteristike Raspberry Pi platforme. Treće poglavlje je posvećeno programskom jeziku Python i Pygame modulu. Četvrto poglavlje prikazuje realizaciju Pong igre pomoću modula Pygame, pri čemu je kao programski jezik korišćen Python. Na kraju je dat kratak zaključak ovog rada.

## II. RASPBERRY PI

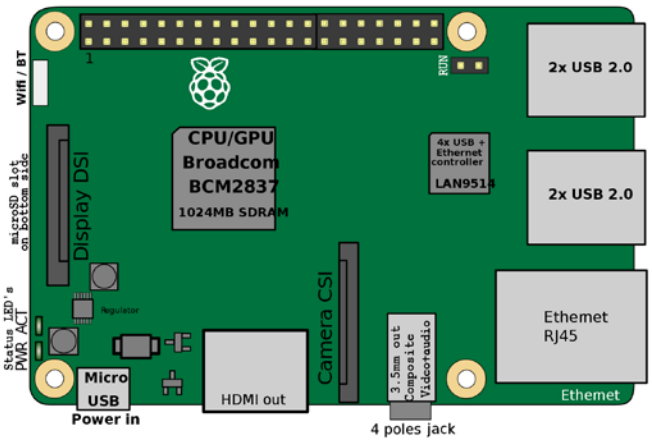
Raspberry Pi je moćan računar veoma malih dimenzija razvijen u Engleskoj. On predstavlja potpuno opremljen računar sa kojim se može uraditi većina stvari kao na personalnom računaru [1].

Marko Stanković je student Elektronskog fakulteta, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: marko031988@gmail.com

Kada su se pojavili procesori koji su se koristili za mobilne uređaje, koji su bili pristupačni po ceni i dovoljno snažni da obezbede dobru podršku grafičkom okruženju, omogućili su da se ubrzo nakon toga na tržištu pojavi prvi primerak računara Raspberry Pi, koji je svojim izgledom i mogućnostima privukao veliku pažnju kod programera početnika.

Njegova mala cena, dobre karakteristike i veoma velika dostupnost informacija pružaju mogućnost za realizaciju mnogih ideja. Razvijen je operativni sistem Raspbian koji je jednostavan za pokretanje, ima dobro grafičko okruženje i veliki broj programa namenjenih za obuku mladih naučnika. Okruženja koja su već dostupna u okviru operativnog sistema Raspbian namenjena su osobama bez iskustva u programiranju.

Na slici 1 prikazana je opšta blok šema Raspberry Pi platforme. Svi modeli sadrže Broadcom sistem na čipu koji podrazumeva ARM (Advanced RISC Machine) kompatibilni CPU (Central Processing Unit) i ugrađeni grafički čip GPU (Graphics Processing Unit). CPU dostiže brzinu do 1.2 GHz. Veličina RAM (Random Access Memory) memorije iznosi do 1 GB. SD (Secure Digital) kartice se koriste za čuvanje operativnog sistema i programa. Većina ploča ima između jednog i četiri USB priključka, HDMI (High-Definition Multimedia Interface) video izlaz, 3.5 mm audio izlaz, GPIO (General-Purpose Input/Output) pinove, Ethernet port, a neke WiFi i Bluetooth priključak [2].




Slika 1. Blok šema Raspberry Pi platforme.

## III. PYTHON

Python je programski jezik koji je razvio Guido van Rossum 90-tih godina prošlog veka u Holandiji na Nacionalnom razvojnom institutu za matematiku i

računarske tehnike. Nastao je kao programerski jezik za učenje programiranja.

Python je slobodan softver otvorenog kôda čiji je razvoj veoma intenzivan i može se besplatno preuzeti sa Interneta. Napravljen je tako da može raditi na više platformi kao što su na primer Windows, Unix, Linux, Mac OS itd. Python se koristi u mnogim sferama primene računara, od zamene za kalkulator pa do kompleksnih korisničkih programa, stvaranja veb-prezentacija itd. Postoji ogromna biblioteka dodataka za Python [3].

Python je dobar izbor za početak učenja programiranja, jer obezbeđuje bogat set alata za kodiranje, omogućavajući da jednostavni programi budu napisani bez ikakve zabune. To omugućava početnicima da se upoznaju sa konceptima i metodama na kojima su zasnovani moderni programski jezici, bez potrebe da poznaju sve jezike na samom početku učenja. Python je veoma modularan, sa puno dodatnih biblioteka, koje mogu biti importovane za brzo proširivanje funkcionalnosti.

Python je interpreterski jezik, što znači da je moguće napisati program i izvršiti ga direktno umesto kompajliranja u mašinski kôd. Ovo čini Python lakšim za programiranje jer ne moramo da brinemo da li su odgovarajuće biblioteke povezane i učitane. Ovo je dobro i zbog toga što u Python-u ne morate eksplicitno naglasiti da li je promenljiva broj, lista ili niz karaktera, interpreter to zaključuje sam prilikom izvršavanja programa [1].

Python podržava objektno orijentisane tehnike programiranja. To znači da je naš program izgrađen od objekata koji kombinuju podatke i funkcionalnost.

Postoje mnogi razlozi zašto odabrati Python kao inženjerski alat, a neki od njih su sledeći:

- Odličan je za početnike, a istovremeno moćan za stručnjake
- Pogodan za velike projekte, kao i za one male
- Omogućuje brz razvoj
- Omogućuje razvoj i izvršavanje kôda na gotovo svim platformama i arhitekturama
- Ugradiv je u kodove pisane u drugim programskim jezicima
- Proširiv sa drugim programskim jezicima
- Objektno orjentisan
- Uredna i elegantna sintaksa
- Stabilan
- Bogatstvo dostupnih paketa i biblioteka
- Otvoreni besplatni kôd koji održava velika zajednica programera i naučnika

Python se koristi za razvoj veb-aplikacija, pristup bazama podataka, izradu desktop aplikacija sa grafičkim korisničkim interfejsom, mrežno i sistemsko programiranje, integraciju, razvoj softvera i kompjuterskih igara, numeričke proračune i za razvoj GIS (Geographic Information Systems) aplikacija.

Nova nauka zahteva i nove programske alate i metode i tu je zapaženo mesto našao i programski jezik Python. Najnovije verzije Python-a imaju moćnu podršku za asinhroni rad, omogućavajući sajtovima da sa pravim bibliotekama rešavaju na desetine hiljada zahteva u sekundi.

Python se koristi za metaprogramiranje. U jeziku Python sve je objekat, uključujući i Python module i same biblioteke. Ovo omogućava Python-u da radi kao visoko efikasan generator kôda, omogućavajući pisanje aplikacija koje manipulišu sopstvenim funkcijama i imaju vrstu proširljivosti koja bi bila teško izvodljiva u drugim jezicima.

### *A. Pygame*

Pygame je skup Python modula koji su napravljeni za programiranje igrica. Pygame proširuje funkcionalnosti SDL (Simple DirectMedia Layer) biblioteke i na taj način omogućuje izradu igrica i multimedijskih aplikacija u programskom jeziku Python. Pygame, kao i Python, je kompatibilan sa skoro svim platformama i operativnim sistemima. Pygame je besplatan i izdat je pod GPL (General Public License) licencom. Stoga sa Pygame-om možete izraditi i besplatne i komercijalne igrice, igrice otvorenog kôda itd.

Pygame je, kao i Python, jednostavan i lak za korišćenje. Koristi optimiziran C kôd i asemblerski kôd za ključne funkcije što čini naš program mnogo bržim.

Dobra osobina Pygame modula je to što ovaj modul ne sadrži hiljade linija kôda koje većina korisnika nikad neće koristiti. Sadrži samo ključne stvari a dodatne stvari se razvijaju izvan Pygame modula.

Pygame sadrži naredbe pomoću kojih možemo:

- Crtati grafičke oblike na ekranu
- Prikazivati slike i bitmape
- Praviti animacije
- Upravljati događajima pomoću tastature, miša ili joysticka
- Reprodukovati zvuk
- Odrediti kada se desila kolizija objekta

Pygame modul pozivamo naredbom *import*, tj. samo na početku programskog kôda napišemo *import pygame* i sve funkcije i objekti iz tog modula biće dostupne za korišćenje.

## IV. PONG IGRA

Kako bi se objasnilo na praktičnom primeru programiranje Raspberry Pi-a na programskom jeziku Python, pomoću modula Pygame, kreirana je igra koju nazivamo Pong. Pong je igra koju je još davne 1971. godine razvila kompanija Atari. Ova igra je bila mnogo jednostavnija za igru od prethodnih igara koje su se pojavile i zbog toga je postala veoma popularna. Smatra se da je ova igra pokrenula novo poglavlje u računarskoj industriji.

Pri programiranju veoma je bitna analiza zadatka. Radi lakšeg programiranja podelićemo naš zadatak u nekoliko koraka, s tim da u svakom koraku zadržavamo prethodno programiran kôd i dodajemo neki novi deo kôda. Koraci našeg projektnog zadatka su [6]:

- Kreiranje praznog ekrana. Tu se koriste naredbe za pravljenje i inicijalizovanje ekrana na kome projektujemo našu igru.
- Crtanje objekta na ekranu. Ovde se treba pozabaviti pravljenjem klase objekta i naredbama za crtanje.
- Kretanje objekata. U ovom delu definišemo kretanje objekata i naredbe sa tastature.
- Definisanje interakcije između objekata. Ovde kreiramo logiku kretanja nakon interakcije između dva objekta, u našem slučaju reketa i lopte.
- Krajnja verzija kôda. Tu se ubacuju neki dodaci kao što je brojanje i prikaz rezultata na ekranu. I tu je prikazana krajnja verzija kôda koja je prethodno testirana.

*A. Kreiranje praznog ekrana*

Prvi korak programiranja aplikacije u Pygame-u je kreiranje praznog ekrana, na čijoj površini crtamo objekte i dalje razvijamo aplikaciju. U ovom delu pored toga što su postavljene početne vrednosti ekrana kao što su dimenzije, naziv, boja i ažuriranje ekrana, definiše se i uslov za izlazak iz našeg programa, kao i vreme osvežavanja ekrana. Kao primer korišćene naredbe navešćemo *pygame.display.set_mode(size)* koju koristimo da bi kreirali ekran i podešavali njegovu veličinu. Naredbom *pygame.display.set_caption("ime_ekrana")* definišemo ime prozora [5]. Naredbu *pygame.time.Clock()* koristimo da bismo mogli u igri da pratimo i kontrolišemo broj frejmova u sekundi. Naredbom *screen.fill(color)* se boji pozadina ekrana. Pomoću *pygame.display.update()* obnavlja se slika ekrana u slučaju da su se neki parametri promenili. Naredbom *clock.tick(frames_per_second)* podešava se broj frejmova u sekundi.

*B. Crtanje objekta na ekranu*

U ovom delu korišćene su naredbe za crtanje, kao i pravljenje klasa i funkcija. U svakoj klasi su po dve funkcije, jedna za inicijalizaciju i druga za crtanje objekta. Ova igra ima tri objekta, loptu i dva reketa, i svaki od objekata smešten je u klasu radi lakšeg programiranja. Prvi objekat koji predstavlja jedan od reketa i nalazi se na levoj strani ekrana nazvan je *Igrac_levo*, drugi objekat odgovara reketu na desnoj strani ekrana i nazvan je *Igrac_desno* a treći objekat koji predstavlja loptu smešten je u klasu *Lopta*. Kao primer korišćene naredbe izdvojićemo *pygame.draw.rect(surface, color, rect)* kojom crtamo naše objekte koji su pravougaonog oblika [5]. *surface* je naziv ekrana na kojem crtamo objekte, *color* je boja našeg objekta, i *rect* predstavlja koordinate i veličinu objekta. Ovde imamo nekoliko novih promenljivih koje definišu veličinu objekata, to su promenljive *reket_sirina*, *reket_visina* i *velicina_lopte*. Njihove parametre možemo menjati kako bi se igra učinila težom ili lakšom. Kako bi se crtali objekti pri svakoj promeni njihovih parametara u dodate su naredbe *ime_objekta.draw()* da bi se pozvale funkcije *draw* iz klasa.

*C. Kretanje objekta*

Sledeći korak u kreiranju ove igre je programiranje funkcija za kretanje objekata. Imamo tri objekta koja se kreću Igrac_levo, Igrac_desno i Lopta.

Objekti Igrac_levo i Igrac_desno u suštini predstavljaju rekete kojima upravljaju igrači sa leve i desne strane. Reketi se kontrolišu komandama sa tastature i logika je da se pritiskom određenih tastera reket pomeri gore ili dole za određeni broj piksela. To možemo postići naredbom *pygame.key.get_pressed()* kojom smeštamo indeks dugmeta sa tastature koje smo pritisnuli u promenljivu *keys* [5], a zatim u *if* petlji ispitujemo da li se taj indeks poklapa sa indeksom znaka kojim smo definisali kretanje reketa. Ako se indeksi poklapaju onda pomeramo reket za onoliko piksela koliko smo definisali promenljivu *reket_brzina*.

U ovom delu projekta programirana je funkcija za kretanje lopte. U ovoj funkciji je definisana brzina i način kretanja nakon udara lopte o ivicu ekrana. Za sada je kretanje lopte definisano bez interakcije sa reketom. Promenljive *self.brzina_x* i *self.brzina_y* određuju smer i brzinu kretanja loptice. Kretanje lopte definišemo tako što pomeramo loptu za određeni broj piksela po x i y osi istovremeno. Tu smo postavili granice da kada lopta udari u neku od ivica ekrana lopta promeni smer i nastavi dalje sa kretanjem. Za sada smo samo projektovali kretanje lopte bez interakcije sa reketom.

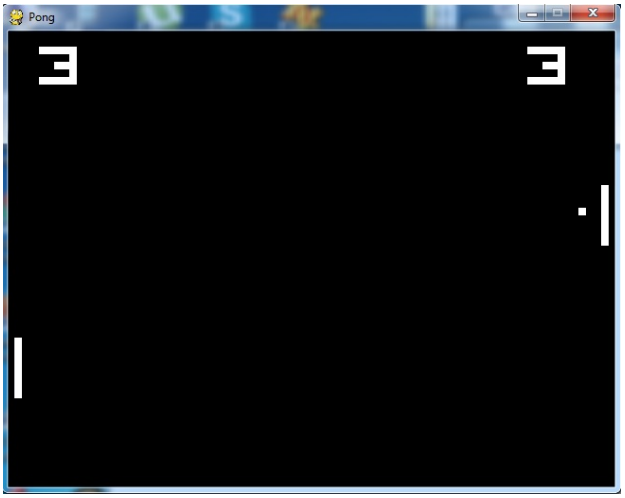
*D. Definisanje interakcije između objekata*

U ovom koraku funkcija za kretanja lopte je proširena nekim novim uslovima koji se odnose na interakciju između reketa i lopte, kao i uslovima za izlazak lopte sa ekrana. Uslove treba pažljivo podesiti kako ne bi došlo do greške u koliziji reketa i lopte.

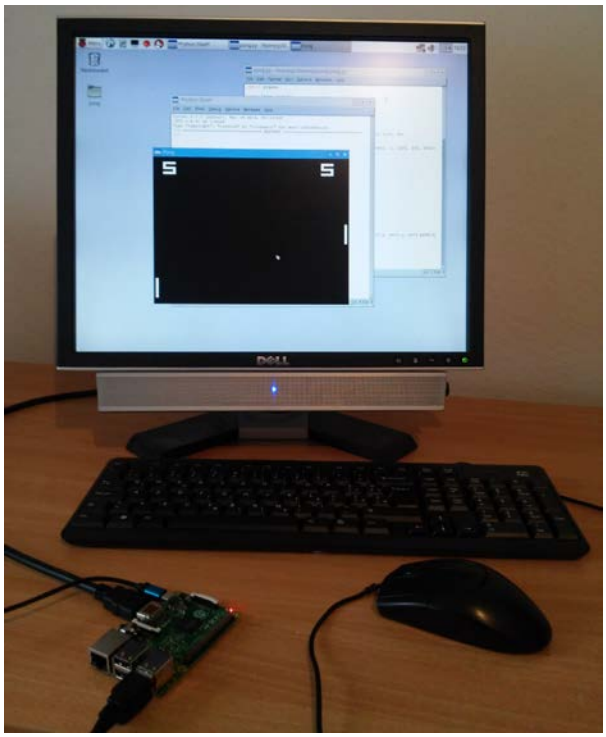
Dodali smo u deo koda za kretanja lopte neke nove uslove koji se odnose na interakciju između reketa i lopte. Dodali smo takođe po jednu *for* petlju za levu i desnu stranu koje će ispitivati da li je lopta udarila u reket ili ne. To možemo postići tako što petlja prolazi celom dužinom reketa, i ako se ta oblast poklapa sa pozicijom lopte, lopta će promeniti smer. Takođe smo dodali uslove ako lopta ne udari u reket i dođe do leve ili desne ivice ekrana da se sve vrednosti vraćaju na početne (__init__).

*E. Krajnja verzija kôda*

U zadnjem koraku projekta, dodaje se funkcija za brojanje i prikaz rezultata, i vrši se testiranje krajnjeg verzije kôda aplikacije. Treba naglasiti da se testiranje vrši nakon svakog koraka u projektovanju, s tim da se na kraju vrši detaljna provera rada aplikacije pri specijalnim slučajevima.




Slika 2. Krajnja verzija Pong igre.



Slika 3. Pong igra na Raspberry Pi platformi.

## V. ZAKLJUČAK

U ovom radu je prikazana realizacija jednostavne multimedijalne aplikacije na Raspberry Pi platformi. Kreirana je igra nazvana Pong korišćenjem programskog jezika Python i modula Pygame. Ovaj zadatak je realizovan kroz sledeće faze: kreiranje praznog ekrana, crtanje objekta na ekranu, kretanje objekata, definisanje interakcije između objekata i testiranje krajnje verzije kôda.

Na osnovu svega prikazanog u radu može se zaključiti da je Raspberry Pi, zbog svoje niske cene i veoma dobrih performansi, odličan uređaj za sve početnike u oblasti računarske nauke. Raspberry Pi ima besplatan softver koji poseduje sve potrebne alate za projektovanje i razvoj aplikacija. Standardni operativni sistem je Raspbian koji promoviše Python kao osnovni program.

Za početak učenja programiranja nije toliko bitno učenje pravila nekog programskog jezika, već je bitnije usvajanje načina razmišljanja pri programiranju. Zbog toga je programski jezik Python, između ostalog, idealan za programere početnike jer je veoma lak za učenje i savladavanje. Zbog toga se više pažnje može posvetiti projektovanju postavljenog zadatka.

## ZAHVALNICA

Autor želi iskreno da se zahvali prof. dr Tatjani Nikolić sa katedre za elektroniku, Elektronski fakultet, Univerzitet u Nišu, za neizmernu podršku i pomoć tokom izrade ovog rada.

## LITERATURA

[1] Dogan Ibrahim, *Raspberry Pi: Hardware Projects*, Elektor, 2014.
[2] Bert van Dam, *Raspberry Pi: Explore the RPi in 45 Electronics Projects*, Elektor, 2014.
[3] Tim Cox, *Raspberry Pi Cookbook for Python Programmers*, Packt Publishing,2014.
[4] Matt Richardson and Shawn Wallace, *Getting Started with Raspberry Pi*, Maker Media, 2013.
[5] Gabriel Alejandro Zorrilla, *Pygame 1.5.5 Reference Manual*
[6] http://trevorappleton.blogspot.rs/2014/04/writing-pong-using-python-and-pygame.html

# Sistem za daljinsko upravljanje

Marko Milanović

*Sadržaj-* U ovom radu prikazana je realizacija sistema za daljinsko upravljanje. Sistem za daljinsko upravljanje sastoji se iz dva dela, odnosno od predajnika i prijemnika. Pritiskom na taster koji se nalazi u kolu predajnika, uređaj koji je priključen na sistem za daljinsko upravljanje se uključuje. Električna šema je projektovana u programu LTSpiceIV [1], dok je štampana ploča projektovana u programu *Altium Designer* [2].

## I. UVOD

U godinama koje su iza nas, tehnologija je toliko napredovala da se život bez elektronskih uređaja gotovo i ne može zamisliti. Na svakom koraku se možemo susresti sa tehnološkim inovacijama, koje znatno unapređuju i olakšavaju rad i život čoveka.

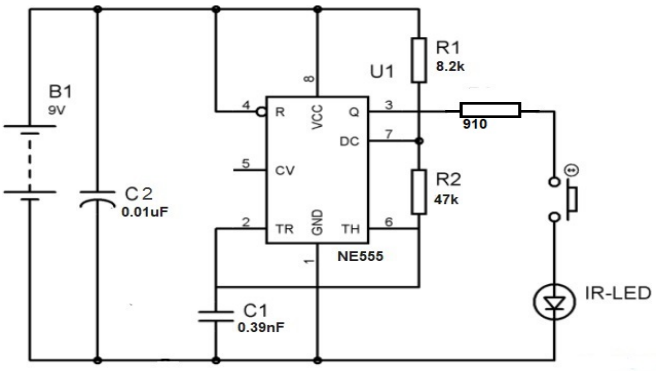
U ovom radu biće opisan uređaj koji predstavlja sistem za daljinsko upravljanje. Uređaj se zapravo sastoji iz dva odvojena dela, od kojih jedan predstavlja predajnik (transmitter), dok drugi deo predstavlja prijemnik (receiver). Predajnik predstavlja neki vid daljinskog upravljača, gde se pritiskom na taster šalje signal prijemniku. Prijemnik predstavlja deo uređaja koji detektuje signal koji stiže od predajnika, i po prijemu signala uključuje odgovarajući uređaj koji je priključen (sijalica, ventilator, zvučnik, itd.).

## II. TEORIJSKI PREGLED

Sistem za daljinsko upravljanje sastoji se iz dva dela, tj. od predajnika (daljinskog upravljača), i prijemnika koji detektuje signal, a zatim uključuje odgovarajući uređaj. U ovom delu biće pojedinačno objašnjen svaki od delova sistema, sa električnim šemama, i ulogom pojedinih komponenata u kolu. Na slici 1 prikazana je električna šema predajnika.

Najvažniju komponentu u kolu predajnika (slika 1.) čini integrisano kolo NE555 [3]. Da bi se razumela uloga NE555 integrisanog kola, dat je kratak teorijski pregled o njemu. NE555 integrisano kolo, ili kako se još naziva NE555 tajmer, služi za kreiranje preciznih vremenskih kašnjenja, ili za generisanje signala određene frekvencije. U zavisnosti od toga da li je potrebno određeno vremensko kašnjenje signala, ili signal određene frekvencije, NE555 integrisano kolo može da radi u dva režima rada, a to su: monostabilni režim rada i astabilni režim rada.

Marko Milanović je student na smeru Elektronske komponente i mikrosistemi, Elektronski fakultet, Univezitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: markoscholes@gmail.com

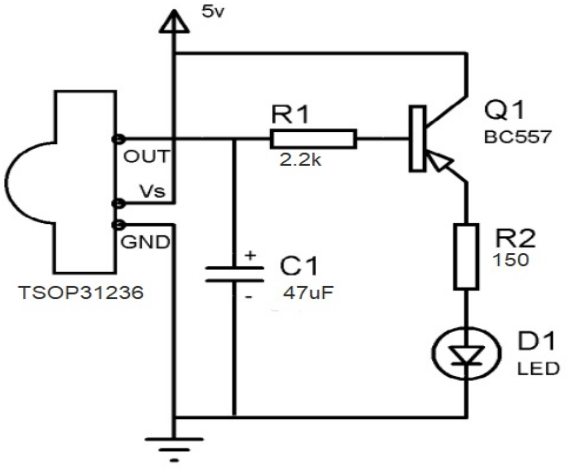



Slika 1. Električna šema predajnika sistema za daljinsko upravljanje.

Nakon kratkog teorijskog pregleda integrisanog kola NE555, znatno je lakše objasniti njegovu ulogu u kolu predajnika sa slike 1. Sa električne šeme predajnika (slika 1.) može se videti da je NE555 povezan tako da radi u astabilnom režimu rada, tj. u režimu rada koji na njegovom izlazu daje signal određene frekvencije, u ovom slučaju to je 36kHz. Signal ove frekvencije je potreban jer se u kolu prijemnika nalazi TSOP31236, koji zapravo predstavlja najvažniju komponentu u kolu prijemnika.

Pored NE555 u kolu predajnika nalaze se još i otpornici *R1* i *R2*, kao i kondenzator *C1*, koji služe za podešavanje frekvencije izlaznog signala iz NE555, zatim nalazi se jedan taster, jedna infracrvena dioda (IR dioda). Pritiskom na taster dioda emituje infracrvenu svetlost.

Drugi deo sistema za daljinsko upravljanje predstavlja prijemnik, čija je električna šema prikazana na slici 2.




Slika 2. Električna šema prijemnika sistema za daljinsko upravljanje.

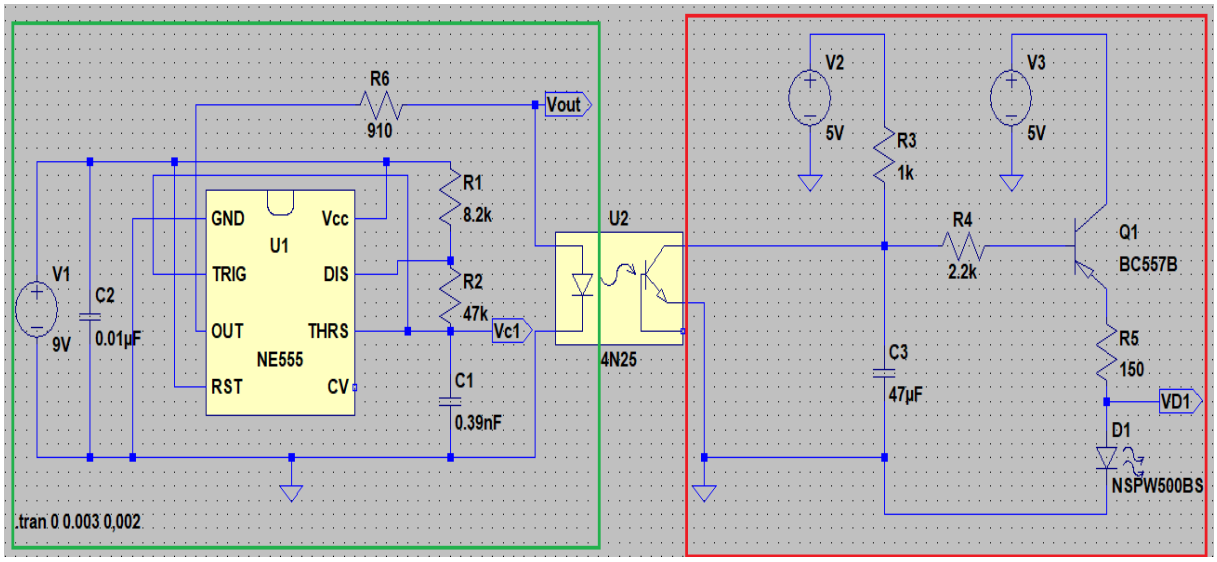
Najvažniju komponentu u kolu prijemnika sistema za daljinsko upravljanje (slika 2.) predstavlja TSOP31236 [4]. TSOP31236 zapravo predstavlja infracrveni senzor koji pripada familiji TSOP312xx infracrvenih senzora, čija se radna frekvencija kreće u opsegu od 30kHz do 56kHz. Senzor TSOP31236 ima radnu frekvenciju od 36kHz.

Uloga infracrvenog senzora TSOP31236 u kolu prijemnika jeste da detektuje infracrvenu svetlost koju emituje infracrvena (IR dioda) u kolu predajnika. Pošto ovaj infracrveni senzor ima radnu frekvenciju od 36kHz, bilo je neophodno da se infracrvena dioda u kolu predajnika pobuđuje signalom baš te frekvencije.

Pored infracrvenog senzora TSOP31236, za realizaciju kola prijemnika potreban je još i jedan PNP bipolarni tranzitor (BC557). Uloga ovog PNP tranzistora je da pojača signal koji dolazi na LED. U ovom slučaju LED služi kao indikator umesto nekog konkretnog uređaja kao što je ventilator, lampa, zujalica, itd. Jako bitnu ulogu u kolu prijemnika ima kondenzator *C1*, kako na potrošač ne bi dolazio impulsni signal.

## III. Simulacija rada u programu LTSpiceIV

Na slici 3 prikazana je električna šema sistema za daljinsko upravljanje u programu LTSpiceIV. Pošto nekih komponenata nije bilo u već postojećoj biblioteci koju poseduje program LTSpiceIV, električna šema je malo modifikovana, ali princip na kome radi sistem za daljinsko upravljanje je verno prikazan.




Slika 3. Električna šema sistema za daljinsko upravljanje u programu LTSpiceIV.

Sa električne šeme (slika 3.) može se videti da je glavna modifikacija u odnosu na originalnu šemu sistema za daljinsko upravljanje ta što se sistem za daljinsko upravljanje sastoji iz dva dela (kolo predajnika i kolo prijemnika), dok su u ovom slučaju kola predajnika i prijemnika spojena u jednu celinu. U okviru zelene boje nalazi se kolo predajnika sa svim komponentama (NE555, otpornici *R1* i *R2*, kao i konednzatori *C1* i *C2*, itd. ), dok se u okviru crvene boje nalazi kolo prijemnika. Glavna modifikacija u odnosnu na originalnu šemu sistema za daljinsko upravljanje je sledeća: Infracrvena dioda iz kola predajnika (CQY99), kao i infracrveni senzor (TSOP31236) iz kola prijemnika zamenjeni su jednim optokaplerom (u ovom slučaju 4N25), pomoću koga je ostvarena optička sprega koju je trebalo da formiraju infracrvena dioda i senzor TSOP31236.

LED dioda u kolu prikazanom na slici služi kao indikator. Merenjem napona na diodi i struje koja protiče kroz diodu (oko 11.8mA) dolazi se do zaključka da indikatorska dioda svetli, odnosno da kolo funkcioniše.

## IV. Praktična realizacija

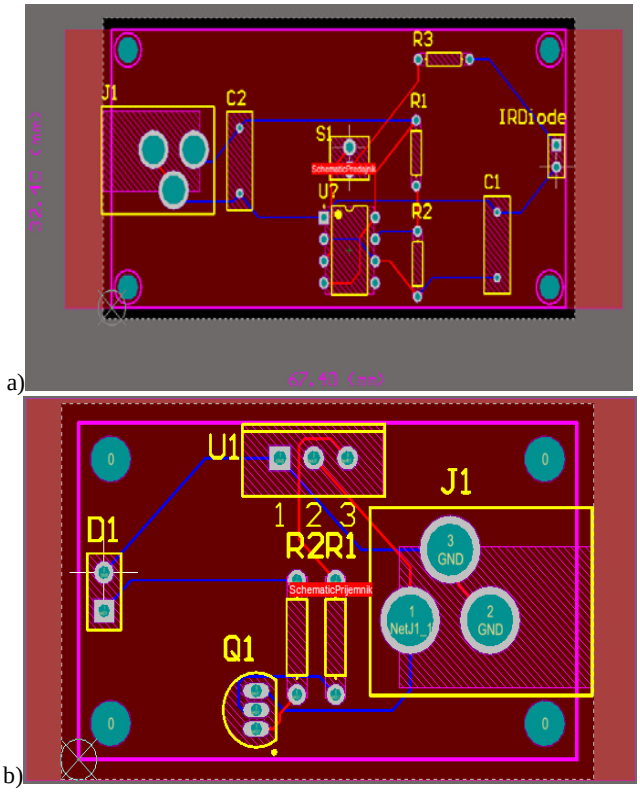
Nakon teorijske analize električnog kola, simulacije i analize strujnih i naponskih signala u programu LTSpiceIV, sistem za daljinsko upravljanje realizovan je i praktično na štampanoj ploči.

Spisak svih komponenata potrebnih za praktičnu realizaciju sistema za daljinsko upravljanje dat je u sledećoj tabeli.

TABELA 1.
SPISAK KOMPONENATA POTREBNIH ZA PRAKTIČNU REALIZACIJU SISTEMA ZA DALJINSKO UPRAVLJANJE

| Naziv komponente | Oznaka | Količina |
|---|---|---|
| Integrisano kolo | NE555 | 1 |
| Kondenzator | 390pF, 100nF | 2 |
| Otpornik | 47kΩ, 8.2kΩ, 910Ω, 220Ω | 5 |
| Infracrveni senzor | TSOP31236 | 1 |
| Infracrvena dioda | CQY99 | 1 |
| LED | / | 1 |
| Bipolarni PNP tranzistor | BC557 | 1 |

Za projektovanje štampanih ploča (potrebne dve, jer se sistem sastoji iz dva dela) ovog sistema za daljinsko upravljanje, korišćen je programski paket Altium Designer [5]. Kada se radi projektovanje nekog sistema u Altium-u, potrebno je da komponente koje se koriste za izradu datog sistema imaju svoj simbol, otisak (Footprint) i 3D model. Kada je završeno crtanje električne šeme u programu Altium Designer, potrebno je ažurirati šematik na pločicu, što je prikazano na slici 4.

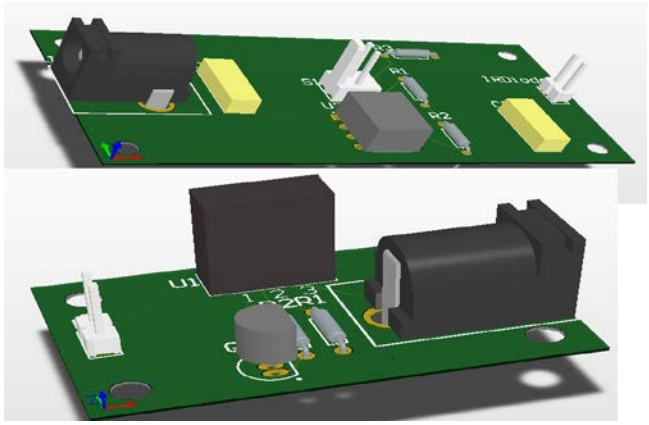



Slika 4. Prikaz otisaka komponenata a) predajnika i b) prijemnika.

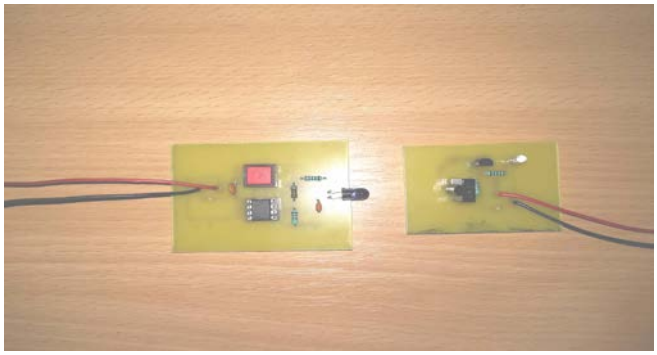
Kada je šematik prenet na pločicu, jasno se mogu videti otisci svih komponenata. Pored toga, potrebno je komponente što bolje rasporediti kako bi njihovo povezivanje bilo što jednostavnije. Kada su prethodno navedeni koraci obavljeni, potrebno je svakoj komponenti dodeliti i 3D model. Nakon što su 3D modeli dodeljeni moguće je videti 3D model celokupne pločice.

Nakon što je projektovanje štampanih ploča završeno, neophodno je generisanje tzv. Gerber fajlova na osnovu kojih je moguće izvršiti praktičnu realizaciju pločica. Praktično realizovane štampane pločice predajnika i prijemnika sistema za daljinsko upravljanje prikazane su na slici 6.

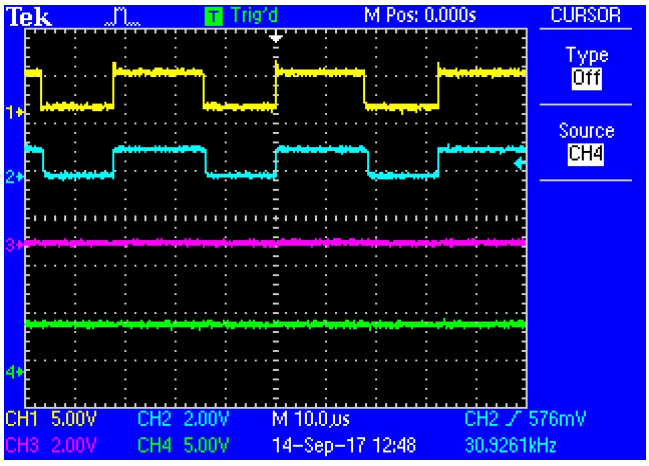
Nakon što je praktična realizacija sistema za daljinsko upravljanje završena, neophodno je izvršiti proveru ispravnosti rada ovog sistema. Najbolji način da se to proveri jeste korišćenjem osciloskopa. Snimanje signala osciloskopom vršeno je u dva navrata. Prvi put kada je senzor TSOP31236 neosvetljen, a drugi put kada je osvetljen. Signali usnimljeni osciloskopom, mogu se videti na sledećim slikama.



Slika 5. 3D model pločice predajnika (gore) i prijemnika (dole).



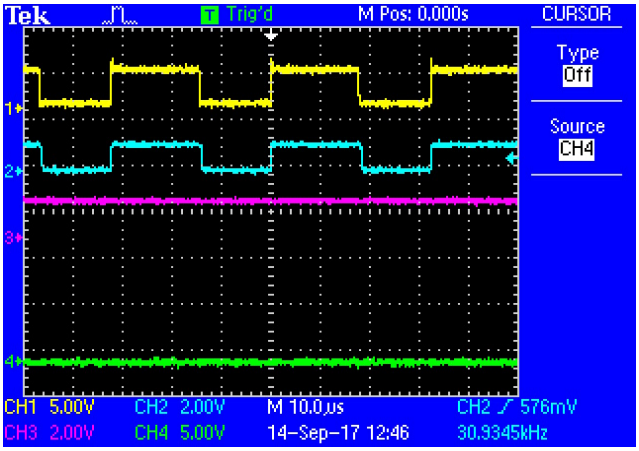
Sika 6. Praktično realizovana štampana ploča predajnika (levo) i prijemnika (desno).




Slika 7. Signali snimljeni osciloskopom kada je senzor TSOP31236 neosvetljen.

Prvi signal predstavlja signal na izlazu iz NE555 integrisanog kola, i sa slike 7 se jasno vidi da se na izlazu NE555 javlja impulsni signal odgovarajuće frekvencije i amplitude. Signal na infracrvenoj diodi predstavljen je

drugim signalom, i kao što se može videti na infracrvenu diodu stiže impulsni signal frekvencije iste kao i signal na izlazu NE555. Poslednji signal predstavlja signal na izlazu senzora TSOP31236, može se videti da se na izlazu senzora javlja kontinualni signal od oko približno 5V, na osnovu toga dolazi se do zaključka da senzor TSOP31236 radi u obrnutoj logici, tj. da kada senzor nije osvetljen na izlazu senzora nalazi se visok naponski nivo, a dok kada je osvetljen na izlazu se nalazi nizak naponski nivo. I na kraju treći signal predstavlja signal na LED-u. Napon na LED-u je 0V, što znači da senzor TSOP31236 nije osvetljen (taster nije pritisnut), i LED neće emitovati svetlost.




Slika 8. Signali snimljeni osciloskopom kada je senzor TSOP31236 osvetljen.

Kao i u prethodnom slučaju, prvi signal predstavlja signal na izlazu NE555, a drugi signal predstavlja signal na infracrvenoj diodi. I u ovom slučaju poslednji signal predstavlja signal na izlazu iz TSOP31236 senzora, a pošto senzor radi u obrnutoj logici, sa slike se može videti da se na izlazu senzora nalazi nula. Napon na LED-u predstavljen je trećim signalom i kao što se može videti na LED stiže kontinualni signal od oko 2V što je dovoljno da LED provede i na taj način emituje svetlost. Na ovaj način proverena je ispravnost ovog sistema.

## V. ZAKLJUČAK

U ovom radu, kao što se iz prethodno navedenog može zaključiti, akcenat je na principu rada sistema za daljinsko upravljanje, kao i na rad pojedinih komponenata u kolima predajnika i prijemnika. Pored toga, veoma važno je bilo predstaviti način na koji se optička sprega između dve jedinice sistema formira. Simulacijom u programu kao što je LTSpiceIV pokriveni su teorijski principi rada ovog sistema. U programu Altium Designer je projektovana štampana ploča ovog sistema, a zatim po realizaciji štampane ploče usnimljeni su signali osciloskopom i na taj način sistem je testiran.

## ZAHVALNICA

Autor se zahvaljuje doc. dr Danijelu Dankoviću na izdvojenom vremenu i sugestijama pri izradi ovog rada.

## LITERATURA

[1] LTSpiceIV, dostupno na: http://www.linear.com
[2] Altium Designer, dostupno na: http://www.altium.com/
[3] NE555, dostupno na: http://www.ti.com/ne555.pdf
[4] TSOP31236,dostupno na: https://www.vishay.com/tsop312
[5] Analogna mikroelektronika, Zoran Prijić
[6] Komponente za telekomunikacije praktikum, Danijel Danković

# Život i studije u Lisabonu

Marko Nikolić

*Sadržaj* – U ovom radu opisan je način studiranja u Portugalu, na NOVA (Universidade Nova de Lisboa) univerzitetu u Lisabonu. Iskustvo je stečeno na osnovu šestomesečnog boravka u Lisabonu pod pokroviteljstvom Erasmus stipendije. Opisan je studentski život i aktivnosti na ovom univerzitetu, kao i predispitne i ispitne obaveze.

## I. Uvod

Nova Univerzitet u Lisabonu [1] (Universidade Nova de Lisboa) ili NOVA je portugalski univerzitet čiji je rektorat smešten u oblasti Campolide u Lisabonu. Osnovan je 11. avgusta 1973. godine i najmlađi je od tri državna univerziteta u Lisabonu. Osnivanje NOVA je bio odgovor na sve veću potražnju za visokim obrazovanjem u Portugalu uopšte i posebno u gradu Lisabonu. Podeljena je na devet akademskih jedinica koje su raspoređene u Lisabonu, Oeirašu i Almadi:

- FCT NOVA - Naučno-tehnološki fakultet
- FCSH - Fakultet društvenih i humanističkih nauka
- NOVA SBE – Škola za biznis i ekonomiju
- NMS/FCM - Medicinska škola/Medicinski fakultet
- FD - Pravni fakultet
- NOVA IMS - Škola za upravljanje informacijama
- ITKB - Antonio Ksavier Institut za hemijsku i biološku tehnologiju
- IHMT - Institut za tropsku higijenu i medicinu
- ENSP - Nacionalna škola za javno zdravlje



Slika 1. Zgrada univerziteta.

Ova institucija ima više od 19000 studenata, 1491 profesora i 804 članova osoblja raspoređenih na pet fakulteta, tri instituta i jednu školu, pružajući niz kurseva iz nekoliko oblasti znanja.




Slika 2. Zvanični grb univerziteta.

## II. Vodič kroz Lisabon

Smešten na samom zapadu Iberijskog poluostrva, kao najzapadnija prestonica evropskog kontinenta, brdoviti Lisabon svojim neodoljivim prizorima privlači nove putnike, avanturiste i sve one koji su u potrazi za lepim, šarenolikim i drugačijim od svega viđenog. Sa jedne strane stari deo grada, romantičan, šarmantan, zbijene zgradice kao sardine, fenjeri, sa druge strane neboderi, veliki tržni centri, futurističke zgrade. Lisabon je topao, srdačan grad, raskošan u svojim kulturno istorijskim znamenitostima. To je najveći grad u Portugalu, prestonica i velika luka na obalama Atlantskog okeana. Smešten je na ušću reke Težo u okean. Pored toga, on je izuzetno značajan univerzitetski grad, kao i sedište vlade i celokupne uprave Portugalije. Najstariji univerzitet je osnovan krajem XIII veka.



Slika 3. Brdoviti Lisabon i tramvaj kao obeležje grada.

Grad obiluje lepim građevinama svih stilova, uglavnom nastalih nakon zemljotresa iz 1755. godine. Staro jezgro grada je pod zaštitom UNESKA kao svetska baština. Lisabonska katedrala potiče iz 1147. godine. Izdvaja se i kula Belem, koju je u čast prekomorskih otkrića podigao između 1515. i 1521. godine kralj Manuel I Portugalski, kako bi branila gradsku luku. Brojni su i muzeji, od kojih se ističu oni posvećeni morima i okeanima. Grad ima i poznatu operu.

Marko Nikolić je student na smeru Računarstvo i informatika, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: nikolicmarko93@gmail.com



Slika 4. Kula Belem.

Osim što obiluje znamenitostima koje svedoče o raskošnoj i bogatoj prošlosti, pre svega Portugalije kao kolonijalne velesile, Lisabon je raj savremene umetnosti i dizajna.



Slika 5. Glavni trg.

Poznati su po muzičkom žanru „Fado“ koji se prati od 1820-ih u Portugalu. Karakterišu ga tužne melodije i tekstovi, često o moru i životu siromašnih. Pored ribljih specijaliteta, dobrog vina, Lisabon je poznat i po dezertu pod nazivom „Pastéis de nata“.



Slika 6. Most „25 de Abril“ preko reke Težo.




Slika 7. Mural posvećen Fado muzici.

## III. STUDENTSKI ŽIVOT

S obzirom da su fakulteti raspoređeni na različitim mestima u Lisabonu i okolini, smeštaj se traži na osnovu toga gde je najbliže fakultetu. Studenti mogu izabrati život u domu ili naći smeštaj u nekom od stanova. Što se tiče studentskih domova, postoje tri koji su raspoređeni na različitim mestima. Prioritet se daje portugalskim stipendistima i stranim studentima na programima razmene. Fakultet na kom sam ja studirao je FCT [2] (Fakultet za nauku i tehnologiju) i u neposrednoj blizini kampusa fakulteta nalazi se studentski dom „Fraústo da Silva“ [3]. U domu pored Erasmus studenata, većinu čine domaći studenti koji dolaze iz raznih krajeva Portugala. Život u studentskom domu je najbolji način za upoznavanje i druženje sa ljudima iz celog sveta. Pored toga, na fakultetu postoje organizacije koje često organizuju roštiljanje u kampusu fakulteta, kao i tematske žurke. Naravno, tu su i studentske organizacije vezane za sve studente u Lisabonu kao što su ESN Lisboa i Erasmus Life Lisboa koje svakog dana organizuju žurke i druženja na raznim lokacijama, kao i ekskurzije kroz ceo Portugal. S obzirom na sve ove organizacije, svaki dan je zagarantovano ispunjen druženjem.




Slika 8. Žurka na brodu.

Još jedna od prednosti je ta što je Lisabon blizu okeana pa je druženje često bilo i na dugačkim i prelepim plažama.



Slika 9. Druženje na plaži.



Slika 10. Večera sa internacionalnim prijateljima.

## IV. AKADEMSKE AKTIVNOSTI

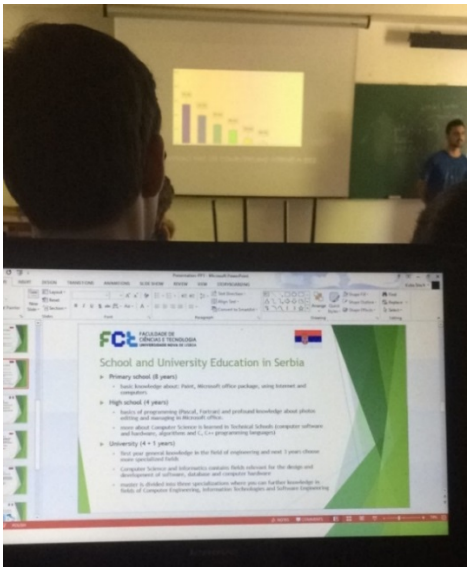
Kao što sam i ranije naglasio fakultet na kojem sa studirao je FCT koji je vezan za nauku i tehnologiju. Fakultet je podeljen u nekoliko katedra. Tu su katedre za životnu sredinu i inženjerstvo, za materijale, mašinsko i industrijsko inženjerstvo, za fiziku, matematiku, hemiju, elektrotehniku, građevinu, biologiju, kao i računarstvo na kojem sam ja bio. Pre odlaska na razmenu bilo je potrebno odabrati kurseve koje ću slušati tokom razmene. Na sajtu fakulteta bio je dostupan detaljan spisak kurseva sa opisima i periodima u kojima su dostupni. S obzirom da može doći do toga da se neki od kurseva održavaju u istom terminu, moguća je i promena ugovora o učenju u toku razmene. Sistem učenja na ovoj katedri se nije mnogo razlikovao u odnosu na naš sistem. Ocene su za razliku od naših u rasponu od 0 do 20, gde je potrebno najmanje 10 da bi se ispit položio.

### *A. Predavanja i vežbe*

Sva predavanja se održavaju usmeno, uz pomoć projektora. Za svaki kurs materijal postoji u elektronskom obliku i bio je dostupan nakon predavanja. Predavanja iz nekih predmeta su bila obavezna s obzirom da se određen broj poena dobijao prisustvom. Što se tiče vežbi bile su organizovane na različite načine u zavisnosti od kursa. Iz nekih predmeta su se održavale računske vežbe koje su bile slične predavanjima iz tog predmeta s tim što su bile vezane isključivo za zadatke. Iz nekih to se svodilo na rad u laboratorijama koje su opremljene velikim brojem računara.

### *B. Predispitne obaveze*

Predispitne obaveze su iz nekih predmeta jedini način za dobijanje ocena, dok kod ostalih utiču na konačnu ocenu studenta. Kontinuirani rad je bio obavezan jer je skoro svake nedelje bio rok za predaju nekog dela projekta ili domaćeg zadatka. Na svim predmetima su predispitne obaveze bile u vidu individualnih i timskih zadataka. Za svaki projekat je bilo potrebno pisanje izveštaja, a ponekad i više njih raspoređenih tokom semestra. Ti izveštaji služe da se predstavi napredak u radu na projektu. Prednost ovih parcijalnih izveštaja je što studenti mogu da čuju mišljenje profesora i na vreme unaprede svoj rad. Pored slanja zadataka profesoru, studenti su prezentovali svoj rad. Iz nekih predmeta je bila obavezna prezentacija na projektoru od petnaestak minuta pred celom grupom, dok je iz nekih bilo prezentovanje samo profesoru u učionici.




Slika 11. Prezentacija radova.

Iako ima dosta predispitnih obaveza, svi rokovi za predaju naznačeni su na početku, i ravnomerno su raspoređeni tokom semestra. Odstupanja od ovih rokova uglavnom nije bilo, osim u situaciji kada je cela grupa tražila odlaganje zbog velikog broja obaveza.

*C. Ispiti*

U toku semestra održavaju se iz skoro svakog kursa po dva testa. Ukoliko student položi oba testa, sa položenim predispitnim obavezama, student je položio ispit. Taj sistem podseća u mnogome na naš. Polaganje ispita iz jednog predmeta je bilo moguće i samo preko domaćih zadataka i projekta. Princip polaganja pismenih ispita jako je sličan našem fakultetu – sastoji se od pitanja koja zahtevaju teorijsko znanje, ali i primenu tog znanja na jednostavnijim praktičnim primerima. Za pripremu ispita dovoljno je koristiti materijal koji je profesor delio studentima. Primeri ispita iz ranijih godina bili su dostupni radi bolje pripreme, često uz priložena rešenja zadataka. Ispiti se polažu na kraju semestra u unapred navedenim terminima.

## V. ZAKLJUČAK

Boravak na inostranom univerzitetu omogućava upoznavanje sa različitim metodama rada i učenja. Pored toga, život i rad u internacionalnoj sredini omogućava vam da proširite vidike, upoznate različite kulture i načine razmišljanja. Razmena je jedno neprocenjivo iskustvo koje vam omogućava da sazrite na ličnom i profesionalnom planu. Ako želite drugačiji, ispunjeniji, srećniji život, morate da izađete iz svoje zone komfora, a to je ono što između ostalog razmena studenata nudi.



Slika 12. Ulaz u kampus fakulteta.

## LITERATURA

[1] http://www.unl.pt/en, Web sajt NOVA univerziteta
[2] https://www.fct.unl.pt/en, Web sajt FCT fakulteta
[3] http://sas.unl.pt/alojamento/rfs/frausto-da-silva-halls-of-residence, Web sajt studentskog doma

# Automatizirani sustav navodnjavanja upravljan mikrokontrolerom

Filip Vlajsović

*Sadržaj* – Rad opisuje sustav navodnjavanja upravljan Arduino mikroupravljačem. Opisuje način rada komponenti te njihovu ulogu unutar sustava. Sve komponente rade na niskom naponu 5V-12V, ali princip rada je isti za veće sustave na višim naponskim razinama. Izrada rada je provedena u dvije faze.

Prva faza bila je fizička izrada kućišta i postolja sa svim potrebnim komponentama. Za izradu kućišta i postolja korišteno je drvo, a za obradu su se koristili ručni električni strojevi. Po izradi kućišta i postolja, postavljali su se električki moduli i komponente.

Druga faza izrade bila je ožičenje postavljenih modula i komponenti. Nakon toga je uslijedilo programiranje sustava te otklanjanje grešaka i podešavanje samog sustava. U članku je opisana druga faza izrade projekta.

Ideja izrade automatiziranog sustava navodnjavanja upravljanog Arduinom nastala je iz potrebe unapređenja i pojednostavljenja navodnjavanja okućnica i vrtova koja se obavljala ručno. Ovakav sustav uvelike smanjuje posao ljudima s velikim okućnicama ili vrtovima te donosi velike uštede na temelju praćenja stanja vlage u zemlji.

## I. UVOD

Sustavi navodnjavanja sve prisutniji su u makropoljoprivredi, ali i u mikropoljoprivredi. Oni služe za dovod vode u sušnim razdobljima te se na taj način sprječava da nasad ostane bez vode. Također se sve više koriste automatizirani sustavi navodnjavanja. Oni olakšavaju i smanjuju posao poljoprivrednicima i uzgajivačima raznih kultura jer često ti sustavi automatski prate stanje vlage u zemlji te uspoređuju s potrebnom vlagom i po potrebi puštaju vodu u nasade.

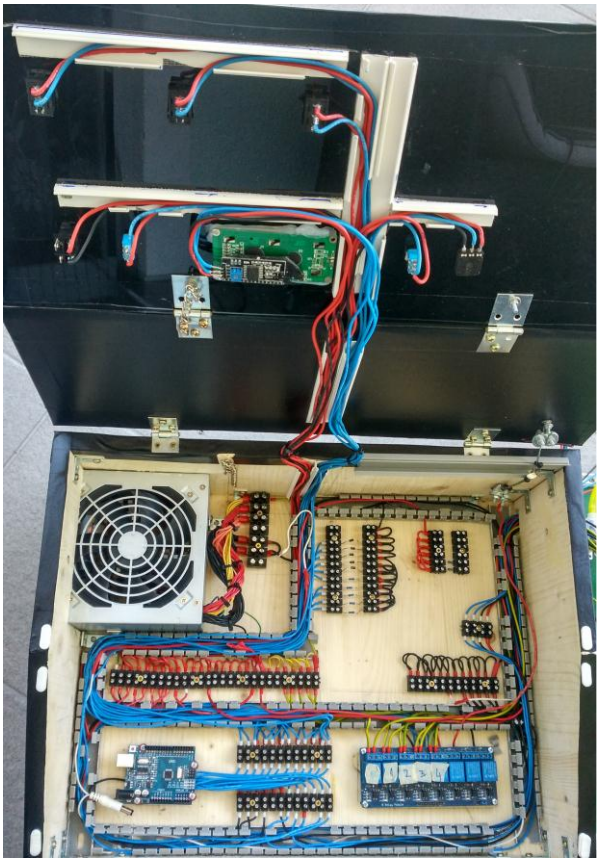
Razgovarajući s manjim proizvođačima, nastao je zaključak da se zalijevanje kultura u sušnim i vrućim razdobljima treba vršiti ili u ranojutarnjim satima prije izlaska sunca, što je preporučljivo, ili u kasnovečernjim satima kada je sunce već na zalasku. Također zalijevanje se vrši prema procjeni, što znači da to može biti premalo, što nije dobro za određene kulture jer smanjuje prinos. Često puta zalijeva se s prevelikom količinom vode, što opet nije dobro jer se na taj način troškovi vode povećavaju i mogućnosti pojave bolesti je već. Rješavanje problema ranojutarnjeg ili kasnovečernjeg "zalijevanja", kao i odgovarajuće količine vode je automatizirano navodnjavanje s praćenjem vlage u polju (nasadu). Istraživajući po raznim internetskim stranicama stvorio se dojam da su svi ti sustavi poprilično skupi za manje uzgajivače koji sade nasade (vrtove) radi svojih potreba. S obzirom da velika populacija starijih ljudi koji sade te nasade nije informatički pismena, potrebno je stvoriti funkcionalan i jednostavan sustav prikladan široj populaciji te za manje nasade s više kultura. Sumirajući sve te činjenice proizašla je ideja o stvaranju vlastitog sustava navodnjavanja koji će otkloniti sve navedene probleme te smanji posao i troškove uzgajivačima.

Filip Vlajsović je student na Elektrotehničkom odjelu Tehničkog veleučilišta u Zagrebu, Konavoska 2, 10000 Zagreb, Hrvatska, E-mail: fvlajsovi@tvz.hr

## II. OPIS ELEKTRONIČKOG MODULA AUTOMATIZIRANOG SUSTAVA NAVODNJAVANJA UPRAVLJANOG MIKROKONTROLEROM

Elektronički modul se sastoji od nekoliko ključnih dijelova u koje spada:

- napajanje
- upravljački dio
- izvršni dio
- mjerni dio
- dodatne komponente



Slika 1. Unutrašnjost kućišta upravljačke jedinice.

Svaka el. komponenta unutar sustava zahtjeva određenu naponsku razinu. Ovaj sustav zahtjeva dvije istosmjerne naponske razine; 12V i 5V. Zbog ograničenja prostora unutar kućišta, tražilo se rješenje kojim bi se zadovoljile obje naponske razine u jednog sklopa. Kao dovoljno dobro rješenje pokazalo se računalno napajanje. Ono se sastoji od 3 naponske razine: 3.3V, 5V, 12V; od kojih je korišteno samo 5V i 12V. 5V naponska razina većinom je korištena za upravljačke i mjerne komponente te za uklapanje releja dok je 12V korišteno za izvršne elemente (pumpa i ventili).

Upravljački dio se svodi na mikrokontroler koji radi na sličan način kao i Arduino mikrokontroler. Za ovaj projekt korištena je replika Arduino Uno pod nazivom UNO R3 s 8 bitnim čipom ATmega328P. Pločica se napaja s konstantnim izvorom napajanja od 5V.

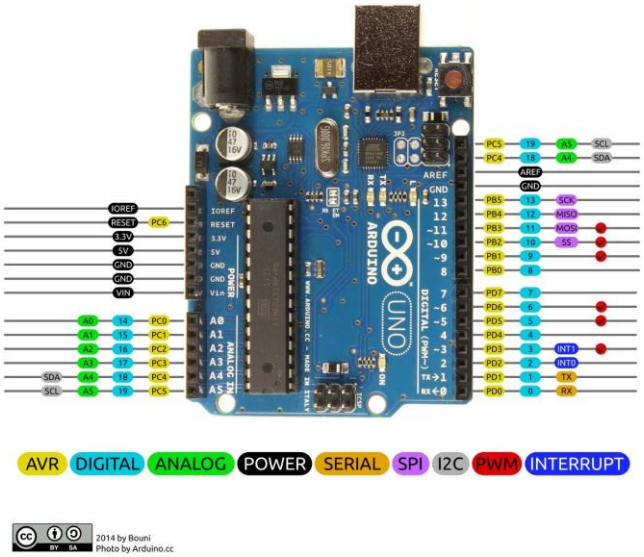
Kao i svaki mikrokontroler, ima ulazne i izlazne pinove. Model koji je korišten ima 14 digitalnih ulaza/izlaza (svaki digitalni pin može biti ili ulazni ili izlazni pin, ovisno o tome kako je definiran u samom programu). Također, ima i 6 analognih ulaznih pinova. Na samoj pločici postoji izvod za 5V, 3.3V napajanje.

Na mikrokontroleru iskorišteni su svi digitalni pinovi. Tako su za digitalne ulaze korišteni pinovi 7-12, dok su za digitalne izlaze korišteni pinovi od 2-6 i 13. pin. Na digitalne ulaze spajani su prekidači s upravljačke ploče koja se nalazi na kućištu upravljačke jedinice sustava navodnjavanja. Prekidači se koriste za upravljanje sustavom na način određivanja režima rada, upravljanje tokovima vode, itd. Također na digitalni ulaz spojena je i ECHO nožica ultrazvučnog senzora udaljenosti pomoću kojeg se mjeri visina vode u akumulacijskom spremniku (ako se koristi). Na digitalne izlaze spojeni su releji kojima se uklapaju i isklapaju ventili i pumpa. TRIG pin ultrazvučnog senzora također je spojen na digitalni izlaz mikrokontrolera.

Kao što je spomenuto na raspolaganju nam je 6 analognih ulaza, no korišteno je njih 5. Tri potenciometra spojena su na tri analogna ulaza. Dva potenciometra korištena su za simulaciju vlage sa dva različita mjesta, te jedan potenciometar za zadavanje referentne vlage. Dva analogna ulaza korištena su za $I^2C$ komunikaciju kojom je ostvaren ispis važnih podataka na LCD ekran.

Po završetku fizičkog spajanja pristupilo se programiranju sustava. Da bi se mikrokontroler mogao programirati prije svega ga je potrebno spojiti i konfigurirati s PC-jem. Komunikacija između PC-ja i mikrokontrolera obavlja se preko USB kabela.

Pri programiranju pazilo se na što manji broj linija koda kako bi program što stabilnije i što brže radio. Sustav je zamišljen i sastoji se od dva načina rada: automatski te ručni način rada.




Slika 2. Prikaz pinova Arduina.




Slika 3. Upravljačka ploča.

## III. NAČIN RADA

### *A. Automode*

Funkcionira tako da se preko potenciometra s upravljačke ploče postavi željena vlaga (na LCD ekranu "ref") u postocima. Sustav sam uspoređuje vrijednost trenutne vlage u polju i prema potrebi pušta i zatvara ventil prema polju1 ili polju2. Polje1 i polje2 mogu biti dva različita nasada.

Na LCD zaslonu prikazuju se podaci o:

- načinu rada ("AM" za automode)
- stanju spremnika "v" u postotcima
- zadanoj vlazi "ref" u postocima
- stanje vlage u polju1 "p1" i u polju2 "p2"

*B. Manualmode*

Razlikuje se od automode-a po tome što postoji mogućnost manipulacije vlage u svakom polju posebno. Manualmode ima još mogućnost izbora između upravljanja VLAGOM ili VODOM.

Ukoliko se izabere upravljanje VLAGOM, tada se može regulirati vlaga u svakom nasadu (polju) posebno. To se izvodi na način da se preko potenciometra zada željena vlaga za npr. polje1 te se omogući istjecanje vode u polje1 postavljanjem prekidača POLJE1 na komandnoj ploči u poziciju ON. Nakon što se postigne željena vlaga u polju1 pumpa i ventil se automatski gasi, ali je potrebno prekidač vratiti na stanje OFF kako sve daljnje promjene referentne vlage ne bi utjecale na vlagu u polju1. Isti postupak vrijedi i za postavljanje vlage u polju2.

Na LCD ekranu prikazuju se podaci kao u automatskom načinu rada osim što će umjesto "AM" pisati "MM" što označava način rada u kojem se sustav nalazi.

Upravljanje VODOM sastoji se od dva procesa:

a) miješanja
b) istakanja

Proces miješanja vode radi na principu cirkulacije vode kroz spremnik pomoću pumpe. Miješanje se aktivira postavljanjem prekidača MIJEŠANJE na ON, a deaktivira postavljanjem prekidača na OFF.

Proces istakanja interesantan je u slučaju prihrane nasada. Dakle, ukoliko treba istočiti točno određenu količinu vode tada se potenciometrom zadaje željena količina vode. Ukoliko je željena količina veća od količine vode u spremniku, tada će istakanje biti onemogućeno i na LCD ekranu će se ispisati "MM NEDOVOLJNO VODE". Ako postoji dovoljna količina vode u spremniku tada se aktiviraju ventili gdje treba istočiti vodu. Sustav po istakanju željene količine vode gasi pumpu i ventile. Ventili se u bilokojem trenutku mogu deaktivirati.

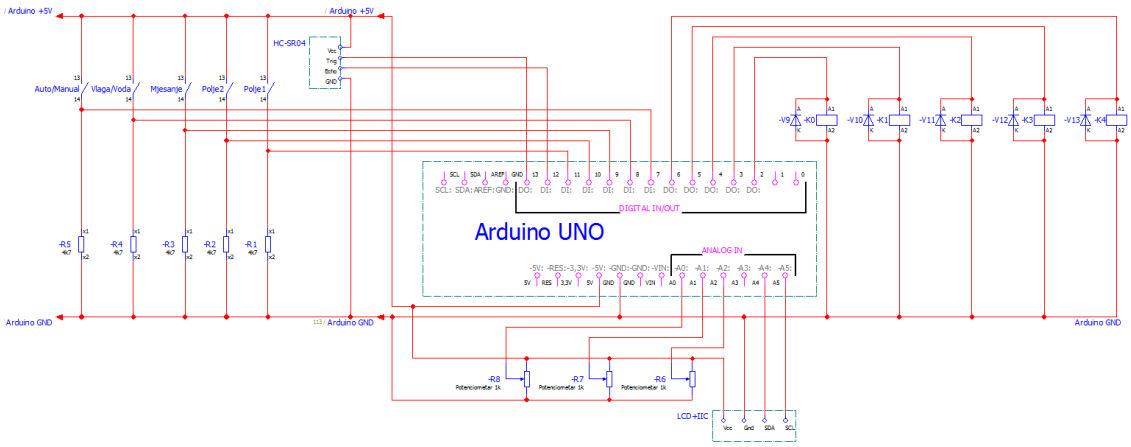
Na LCD ekranu u tom procesu prikazuju se podaci o:

- načinu rada ("MM" za manualmode)
- stanje spremnika "v" u postotcima
- zadanoj količini vode "ref" u litrama
- stanje spremnika "ras" u litrama
- preostalo od istakanja

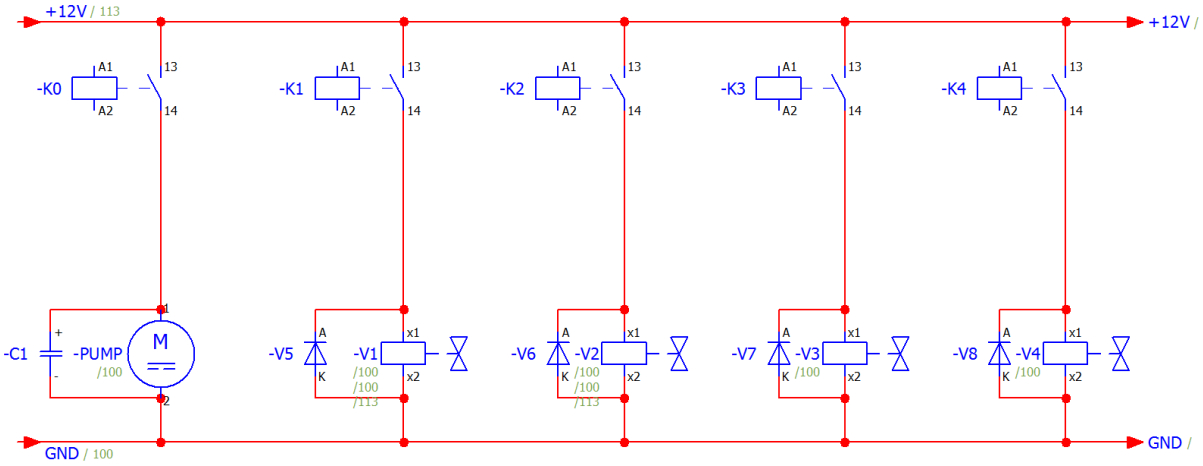
## IV. ZAKLJUČAK

Odabir Arduino mikrokontrolera pokazao se kao odličan odabir kod projektiranja manjeg sustava kao što je odabran u radu. Prednost takvog mikrokontrolera je sama cijena te jednostavnost upravljanja i programiranja. Nedostatak mu je malo opterećenje koje može podnijeti. Stoga je potrebno maksimalno rasteretiti samu pločicu tako da se koliko je to moguće, što više potrošača prebaci na neki drugi izvor napajanja kao što je to napravljeno u ovom radu.

Po završetku programiranja i pri puštanju u pogon sustava bez opterećenja odnosno u situaciji kada releji nisu bili opterećeni elektroventilima i pumpom, sustav je radio kako je i zamišljeno, no nakon spajanja opterećenja uglavnom pri isklopu pumpe ili elektroventila mikrokontroler se počeo ponašati nepredviđeno. Digitalni izlazi su se počeli nekontrolirano paliti, LCD ekran je počeo ispisivati ono što mu nije zadano, vanjske naredbe nisu radile te se nakon nekoliko sekundi mikrokontroler resetirao i ponovno startao. Zaključak je bio da se na releju pri isklapanju ventila, koji u sebi ima zavojnicu, pojavi napon uzrokovan pražnjenjem zavojnice što stvara udar na mikrokontroler. Taj udar je iznad naponske razine koju mikrokontroler u normalnim uvjetima može izdržati te je njegov "obrambeni" sistem reset svih funkcija. Problem je bio riješen postavljanjem povratnih dioda između digitalnih izlaza mikrokontrolera i nule te između nožica elektroventila, no nakon duljeg testiranja počele su se javljati iste smetnje samo smanjenog intenziteta. Smetnje nisu bile prisutne kada pumpa nije bila priključena na relej. Analizom pumpe ustanovilo se da pumpa ima kolektorski motor koji iskrenjem između četkica stvara smetnje. Kako bi otklonili te smetnje postavljen je filter, tj. kondenzator između pozitivne i negativne nožice pumpe. Na koncu je sve radilo kao što je i zamišljeno.




Slika 3. Shema spajanja digitalnih ulaza i izlaza na mikrokontroler.




Slika 4. Shema spajanja elektroventila i pumpe na DC izvor preko releja.

Svaki sustav ima mane, pa tako i ovaj. Trenutno je sustav ograničen s brojem ventila odnosno releja kojima se može upravljati zato što nema mjesta na izlazima iz mikrokontrolera za dodatne elektroventile. Također postoji ograničenje vezano za mjerenje vlage. Prema sadašnjoj situaciji mjeri se vlaga samo s dva senzora (simulira se, ali princip je isti), no moguće je dodati još jedan senzor vlage.

Mane ovakvog sustava, kakav je trenutno, mogu se riješiti uz dodatna ulaganja. Dobavljanjem većeg Arduino mikrokontrolera te malom prilagodbom poštujućeg rješenja.

S obzirom na namjeru korištenja ovog sustava nastojat će se vršiti konstantna poboljšanja sve dok se ne dobiju željene karakteristike i mogućnosti sustava. Neka od poboljšanja koja bi se trebala napraviti prije realizacije i postavljanja u stvarne uvijete rada su svakako postavljanje mjerača protoka, postavljanja senzora vlage te sklopa za regulaciju brzine vrtnje pumpe. Naravno, postavljanje većeg LCD ekrana kako bi se više informacija moglo prikazati istovremeno. Također, onaj krajnji cilj koji se želi postići je spajanje kompletnog sustava na server te upravljanje i praćenje navodnjavanja iz bilokojeg dijela svijeta te mogućnost upravljanja preko mobilne aplikacije.

## LITERATURA

[1] http://www.obz.hr/hr/pdf/poljoprivredni_info_pult/2010/Navodnjavanje.pdf
[2] https://e-radionica.com/hr/blog/2015/10/08/sto-je-arduino-i-croduino/
[3] https://www.scribd.com/doc/26600926/Releji-maturski-rad
[4] http://howtomechatronics.com/tutorials/arduino/control-high-voltage-devices-arduino-relay-tutorial/
[5] https://e-radionica.com/hr/ultrazvucni-modul-hc-sr04.html
[6] https://e-radionica.com/hr/lcd-ekran-16x2-hd44780.html

# Linearni modulator laserske diode

Nikola Raičević, Žarko Jovanović

*Sadržaj* – U radu je prikazan pokušaj da se realizuje jedan linearni drajver za lasersku diodu. S obzirom na linearnu karakteristiku, drajver može da služi i kao modulator koji direktno menja injekcionu struju lasera kada se na njegovom ulazu menja dovedeni napon.

## I. UVOD

Linearni drajver laserske diode je u idealnom smislu izvor konstantne struje, koji je linearan, bešuman i precizan, tako da daje tačno definisanu struju laseru, potrebnu za njegovu odgovarajuću primenu. Najveći broj uređaja koji koriste drajvere lasera se napajaju iz električne mreže, ali postoje i drajveri koji koriste baterije kao izvor struje.

U najvećem broju slučajeva, drajver treba da obezbedi napajanje konstantnom strujom, čime se postiže kontinualno zračenje lasera, sa približno konstantnom optičkom izlaznom snagom. Usled znatno nelinearne strujno-naponske karakteristike laserskih dioda, sa malom diferencijalnom impedansom (odnosno, visokim odnosom dI/dV), koja značajno zavisi od temperature spoja, obično nije dovoljno da se laseru dovede konstantna struja. Umesto toga, struja mora da se stabilizuje putem automatske promene napona. Ovaj mod konstantne struje jeste način funkcionisanja strujnog izvora. Kada se povećava temperatura potreban je manji napon za istu struju. Ako se umesto toga napon drži konstantnim, struja se može znatno povećati, što sa druge strane doprinosi povećanju temperature. Ova situacija se može pretvoriti i u otkaz, pri čemu laserska dioda biva uništena ako struja nije ograničena.

Treba primetiti da struja, a ne napon, određuje brzinu kojom se nosioci ubacuju u spoj laserske diode. Prema tome, optička izlazna snaga je jako korelisana sa strujom drajvera, a znatno manje sa naponom.

Da bi se predupredilo oštećenje laserske diode, važno je izbeći bilo kakvo premašenje struje drajvera. Čak i veoma kratki strujni impulsi mogu uništiti lasersku diodu, na primer u vidu katastrofalnog optičkog oštećenja usled preteranog optičkog intenziteta na izlaznim stranicama rezonatora. Posebno ako drajver laserske diode nije projektovan za specifične diode sa zadatom maksimalnom strujom, potrebno je da on ima mogućnost posebne kontrole ove maksimalne struje koja se može ručno podesiti. Ovaj podešeni limit mora biti ispoštovan od strane svih ostalih kontrola koje drajver može da ima. Ukoliko se dostigne strujni limit, struja se može ograničiti tako da dostigne zasićenje, ili se može potpuno isključiti sve dok korisnik ne aktivira drajver ponovo, pritiskajući taster, na primer. Na taj način se može značajno smanjiti rizik da korisnik slučajno uništi diodu tako što će okrenuti dugme koje kontroliše intenzitet previše. Posebna pažnja je potrebna jedino u početku kada se podešava ovaj strujni limit.

U nekim slučajevima, laserske diode se mogu namerno preopteretiti, odnosno opteretiti strujom koja je preko preporučenog maksimuma. Na ovaj način se može postići veća izlazna saga, ali po cenu skraćenog životnog veka lasera, i sa rizikom trenutnog nastajanja oštećenja.

Očigledno, drajver laserske diode treba da bude relativno imun na spoljašnje uticaje kao što su naponski impusli u električnoj mreži ili prelazni režimi struje usled loših električnih kontakata.

Laserske diode u opštem slučaju nisu pogodne za direktno konektovanje i diskonektovanje, odnosno nisu "hot-plug" komponente. Laser treba priključivati i isključivati jedino kada je drajver isključen i kada su preduzete potrebne mere predostrožnosti kako bi se izbeglo oštećenje usled elektrostatičkog pražnjenja. Na primer, pinovi lasera se mogu kratkospojiti pre nego što se laser fizički odvoji od drajvera.

Pre korišćenja drajvera unutar laboratorije ili nekog sistema, potrebno je proveriti da se ne pojavljuju problemi usled neodgovarajućeg uzemljenja. Neki drajveri imaju jedan pin koji je spojen sa kućištem i efektivno uzemljen, a tada se može desiti da postoji interferencija sa dodatnim uzemljenjima ostalih uređaja. Čak i kada je isti pin spojen sa uzemljenjem na različitim lokacijama, mogu se pojaviti problemi usled "petlje uzemljenja" u kojoj magnetna polja mogu indukovati značajne strujne smetnje.

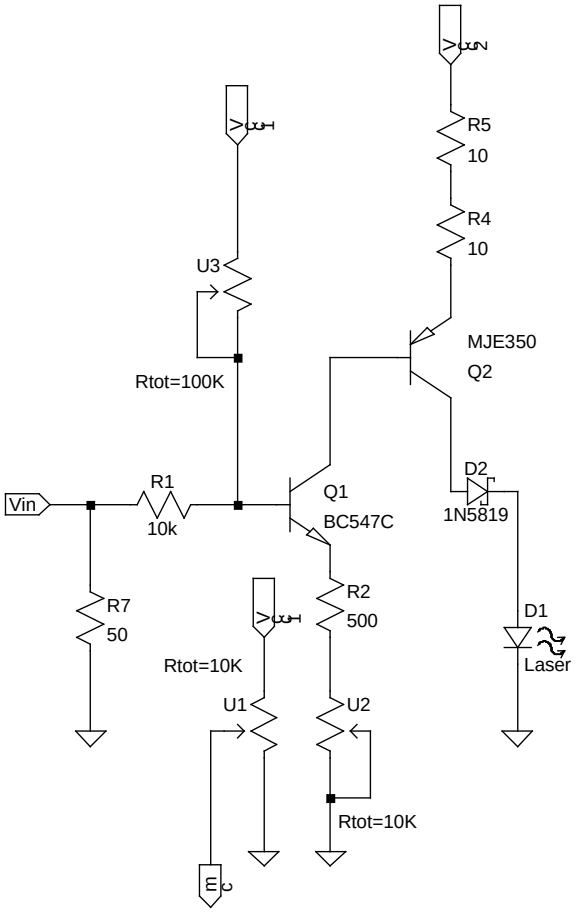
U ovom radu smo pokušali da projektujemo i realizujemo jedan jednostavan drajver za lasersku diodu, koji će biti u stanju da ulazni napon pretvori u proporcionalnu izlaznu struju lasera. Na ovaj način je moguće precizno kontrolisati struju i postići modulaciju optičke snage lasera prostim menjanjem ulaznog napona. Početna ideja je da drajver bude što jednostavniji, ali i da ima mogućnost modulacije koja ima zadovoljavajući odziv u frekvencijskom opsegu do nekoliko MHz.

## II. IDEJNO REŠENJE DRAJVERA

S obzirom da smo odlučili da pokušamo realizaciju koja će omogućiti relativno veliki propusni opseg modulacije, ograničili smo se na tranzistore kao aktivne

Nikola Raičević i Žarko Jovanović su studenti na modulu Telekomunikacije, Elektronski fakultet Univerziteta u Nišu, A. Medvedeva 14, 18000 Niš, Srbija. E-mail: nikolaraicevic@gmail.com, zaree9994@gmail.com

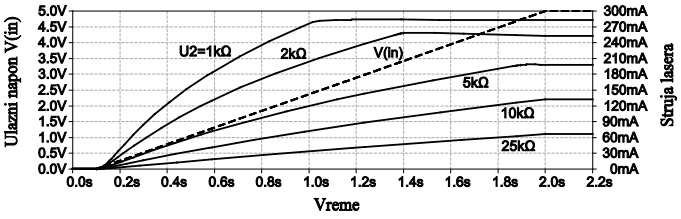
komponente i izbegli operacione pojačavače koji se obično koriste u kolima analognih drajvera. Takođe smo zbog jednostavnosti izbegli povratnu spregu za kontrolu struje, ali se ova funkcionalnost može dodati i kasnije. Laser koji smo imali na raspolaganju u laboratoriji ima struju koja je ograničena na 330 mA, tako da izlazni stepen mora imati odgovarajući tranzistor koji omogućava ovoliku struju. Koncept drajvera koji koristimo jeste dvostepeni pojačavač sa bipolarnim tranzistorima, koji je prikazan na slici 1.




Slika 1. Kompletna simulaciona šema drajvera.

Drajver koji je prikazan predstavlja dvostepeni pojačavač u kome prvi stepen obezbeđuje prilagođenje naponskih nivoa i impedanse, dok drugi stepen omogućava potrebnu izlaznu snagu za pobudu lasera. Ulazna impedansa kola je 50 Ω, da bi bila kompatibilna sa generatorom funkcija koji ima ovoliku izlaznu impedansu. Kolo ima potenciometar U2 koji služi za podešavanje osetljivosti modulatora, odnosno koliko će miliampera varirati izlazna struja sa promenom ulaznog napona za 1 V. Predviđeno je da ova vrednost može da se prikaže na npr. sedmosegmentnom displeju, pa je ovaj potenciometar zamišljen kao stereo potenciometar, kod koga će drugi deo (U1) služiti za indikaciju. U kolu postoji i trimer-potenciometar U3 koji služi za podešavanje jednosmerne radne tačke drajvera.

Rezultati simulacije u LT-Spice simulatoru su prikazani na slici 2. Pretpostavljeno je da se ulazni napon menja linearno od 0 do 5 V u vremenu od 0.1-2 s. Izlazna struja je prikazana u zavisnosti od otpornosti podešene na U2. Vidi se da pri malim otpornostima struja ima veći nagib i brže se povećava sa naponom, ali zato i brže dostigne maksimalnu vrednost. Kada su vrednosti otpora veće, maksimalna struja je ograničena na manju vrednost i nagib je sve manji. Na ovaj način smo pokazali da se nagib struje može jednostavno kontrolisati. Takođe se može primetiti da je struja znatno linearnija za veće vrednosti ovog otpora.

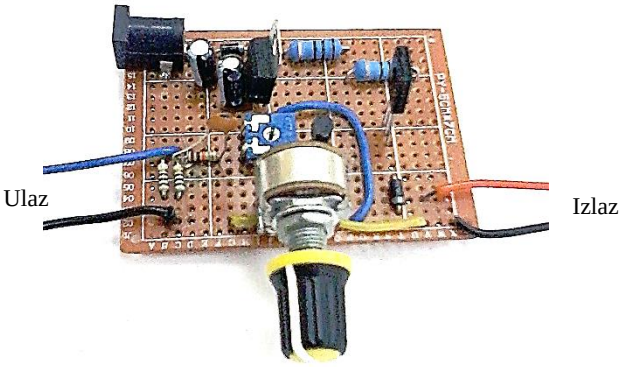



Slika 2. Simulacijaizlazne struje, u zavisnosti od ulaznog napona, za različite vrednosti položaja potenciometra U2.

Maksimalna struja u kolu lasera je ograničena uz pomoć rednog otpora u kolu koji iznosi 20 Ω. Ako pretpostavimo da je napon napajanja u kolu lasera 9 V, a pad napona na laserskoj diodi oko 2 V, to nam preliminarno kazuje da je maksimalna struja 7 V/20 Ω = 350 mA. Naravno, ovde nismo uračunali pad napona na izlaznom tranzistoru, kao ni pad napona na Šotkijevoj diodi, tako da će struja biti nešto manja od toga. Na slici 2. vidimo da maksimalna struja iznosi oko 280 mA, što je u skladu sa ovim pojednostavljanim razmatranjem.

## III. PRAKTIČNA REALIZACIJA I TESTIRANJE PROTOTIPA

Za realizaciju našeg projekta smo iskoristili raster-ploču veličine 4×5cm, slika 3. Pored postojećih rupica sa rasterom 2.54 mm, bilo je potrebno i dodatno proširiti neke od njih jer su pinovi dostupnog DC konektora bili nešto veći po dimenzijama. Prototip sadrži stabilizator napona na 5 V koji služi za napajanje prvog stepena drajvera. Drugi deo se napaja direktno od jednosmernog napona iz ispravljača ispred stabilizatora. Funkcionisanje prototipa je provereno jednostavnim vezivanjem LED umesto lasera. Tom prilikom je potenciometar okrenut maksimalno udesno, odnosno prema većim vrednostima otpora, kako bi struja bila ograničena na manje vredsnoti. Dovodjenjem jednosmernog napona na ulaz kola, LED se pali i počinje da svetli progresivno intenzivnije kako se povećava napon na ulazu.

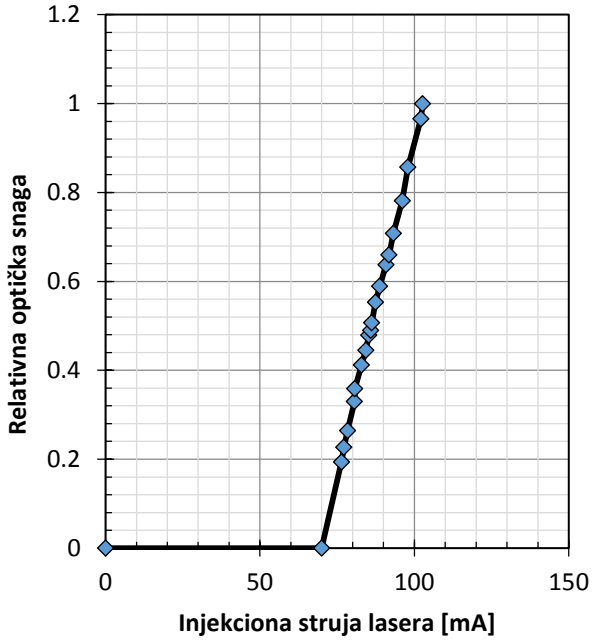



Slika 3. Izgled realizovanog prototipa na raster ploči.

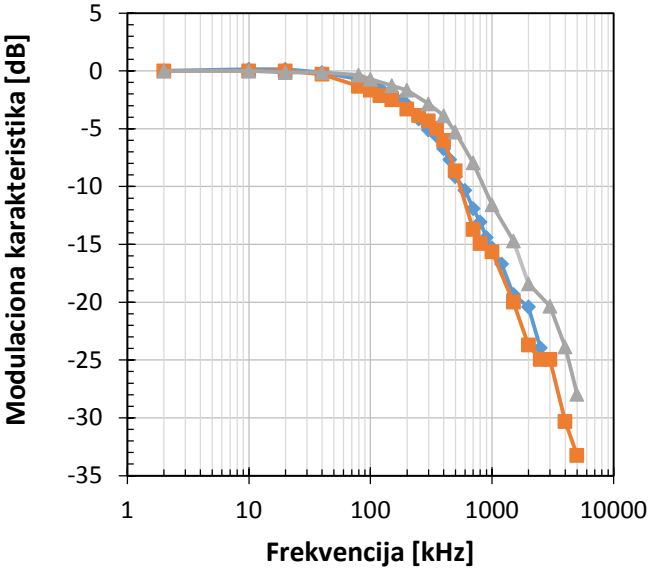
Nakon što smo se uverili da kolo funkcioniše, pokušali smo da umesto LED pobudimo lasersku diodu. U laboratoriji je bila dostupna jedna laserska dioda koja emituje svetlost zelene boje, talasne dužine oko 532 nm, i sa maksimalnom preporučenom strujom od 330 mA. Međutim, nismo uspeli da pobudimo ovu diodu da svetli. Prag struje pri kome laser počinje da svetli iznosi oko 190 mA, tako da naše kolo drajvera nije bilo u stanju da dostigne ovu vrednost izlazne struje. To nam govori da simulacija i pored preciznih modela tranzistra nije u potpunosti u mogućnosti da opiše sve efekte koji se mogu pojaviti u praktičnom kolu.

Povezivanjem ampermetra u kolu umesto lasera smo ustanovili da realizovani prototip može da da struju od 160 mA. Uzrok ovome može da bude pad napona na samom ispravljaču, jer je ova struja relativno bliska nazivnoj struji ispravljača. Umesto toga, odlučili smo da isprobamo drugi laser koji ima manju struju praga i koji bi mogao da fujnkcioniše sa kolom u već realizovanom obliku. Ovaj drugi laser predstavlja lasersku diodu iz starog DVD uređaja i ima talasnu dužinu zračenja od oko 650 nm, što odgovara crvenoj svetlosti. Kada su u pitanju laseri, veoma je pogodno koristiti lasere koji emituju vidljivu svetlost, jer postoji vidljiva indikacija izlazne snage. S obzirom da infracrveni laseri mogu imati znatnu snagu, pri čemu su nevidljivi za ljudsko oko, oni mogu biti izuzetno opasni po vid. Da bismo se uverili da ova crvena laserska dioda može da funkcioniše ispravno sa našim drajverom, najpre smo izmerili njegovu karakteristiku izlazne optičke snage u zavisnosti od injekcione struje.

Karakteristika lasera je prikazana na slici 4. Ova karakteristika je dobijena tako što je snop laserskog zračenja uz pomoć sočiva kolimiran i usmeren na fotodiodu uređaja koji meri fotostruju. Fotostruja je direktno srazmerna optičkoj snazi koja je incidentna na fotodiodu, tako da očitavanje fotostruje služi kao indikacija snage lasera. Konstantu srazmernosti smo zanemarili tako da smo na slici prikazali relativnu optičku snagu normalizovanu maksimalnom vrednošću u toku testiranja. Očigledno je da ovaj laser ima prag laserskog efekta koji počinje da se pojavljuje iznad injekcione struje 70 mA.




Slika 4. Izgled realizovanog prototipa na raster ploči.




Slika 5. Injekciona struja lasera struja u zavisnosti od frekvencije.

Nakon što smo se uverili da ovaj laser može da se pobudi realizovanim drajverom, nastavili smo sa testiranjem tako što smo pokušali da proverimo modulacionu karakteristiku. Podesili smo jednosmernu vrednost ulaznog napona na vrednost koja odgovara izlaznoj struji od 85 mA, a zatim smo na ovu vrednost superponirali naizmenični sinusni signal iz generatora funkcija. Amplituda naizmenjičnog signala je bila relativno mala, odnosno oko 100 mV, kako se ne bi pojavljivali nelinearni efekti u izlaznom signalu. Amplitudu optičke snage smo merili tako što smo svetlost lasera uperili u inverzno polarizovanu fotodiodu sa malom impedansom

opterećenja kako bi propusni opseg ovakvog prijemnika bio dovoljno veliki. Dobijena modulaciona karakteristika je prikazana na slici 5.

Sa slike se može uočiti da optička snaga prati ulazni napon do frekvencija od oko 350 kHz. U slučaju detektovanja optičke snage, fotostruja je direktno proporcionalna optičkoj snazi tako da je optički propusni opseg definisan pragom od -6 dB. Ova vrednost optičkog propusnog opsega je relativno mala i ne odgovara pretpostavkama koje smo učinili na početku. S obzirom da je kolo drajvera dosta jednostavno, potrebno je da se detaljnije prouči njegovo ponašanje u frekvencijskom domenu i da se precizno identifikuje uzrok pojavljivanja polova u prenosnoj funkciji, kako bi ovaj uzrok ograničenog propusnog opsega mogao da se kompenzuje. Jedna mogućnost za povećanje propusnog opsega je dodavanje kondenzatora paralelno R1, tako da se rezultujuća impedansa smanjuje sa povećanjem frekvencije. Kada smo paraleleno sa R1 stavili kondenzator kapacitivnosti 100 nF, dobili smo prenosnu karakteristiku sa nešto većim optičkim propusnim opsegom od 550 kHz, što se može videti na slici 5. Pretpostavka je da bi se propusni opseg mogao značajnije povećati sličnom operacijom u kolu emitora, ali je za svaku vrednost otpora potenciometra U2 potrebno odrediti odgovarajuću vrednost paralelnog kondenzatora, tako da se gubi mogućnost jednostavnog podešavanja nagiba modulacione karakteristike.

## IV. Zaključak

U ovom radu smo realizovali jedan jednostavni linearni modulator laserske diode, čime smo potvrdili pretpostavljeni princip funkcionisanja drajvera. Drajver je u mogućnosti da pobudi laserske diode do maksimalne struje od 160 mA, ali se može modifikovati tako da radi i sa većim strujama. Propusni opseg drajvera u režimu linearnog modulatora nije veliki, ali se takođe može poboljšati preciznijim razmatranjem i odgovarajućim modifikacijama.

## Zahvalnica

Ovaj rad napisan je kao deo studentskog projekta na ispitu Optičke komunikacije, na smeru Telekomunikacije, Elektronskog fakulteta Univerziteta u Nišu. Ispit je u 7. semestru osnovnih akademskih studija. Hardverska realizacija projekta izvedena je pri laboratoriji za telekomunikacione sisteme, pod mentorstvom prof. dr Dejana Milića.

## Literatura

[1] Dejana Milić „Optičke telekomunikacije“, Niš 2016
[2] Aleksandar Marinčić „Osnovi optoelektronskih komunikacija“, Gradjevinska knjiga, Beograd, 1986.
[3] Rüdiger Paschotta, „Laser Diode Drivers“, RP Photonics Encyclopedia, RP Photonics Consulting GmbH, dostupno putem interneta na: https://www.rp-photonics.com/laser_diode_drivers.html
[4] Bruce Soileau, „Linear Laser Diode Driver Design“, Analog Modules Inc, dostupno putem interneta na: http://analogmodules.com/linear-laser-diode-driver-design/
[5] ON Semiconductors, "BC546 / BC547 / BC548 / BC549 / BC550, NPN Epitaxial Silicon Transistor", Fairchild Semiconductor Corporation, 2002, kataloški podaci, dostupno putem interneta na: http://www.onsemi.com/pub/Collateral/BC550-D.pdf
[6] ON Semiconductors, "MJE350G, Plastic Medium-Power PNP Silicon Transistor", 2013, Semiconductor Components Industries, LLC, 2013, kataloški podaci, dostupno putem interneta: http://www.onsemi.com/pub/Collateral/MJE350-D.PDF

# Duplo klatno

Jovan Dimitrijević

*Sadržaj-* U ovom radu prikazano je opisivanje kretanja duplog klatna matematički i način merenja uglova koje zahvataju pojedina klatna. Prikazana su dva teorijska pristupa: analitički i numerički. Za merenje uglova korišćena su dva para IR diode i prijemnika iz optičko-mehaničkog miša.

## I. UVOD

Kretanje duplog klatna se može predvideti na više načina. Za male uglove (kada kretanje nije haotično) je moguće naći analitičko rešenje za zavisnost uglova $\theta_1$ i $\theta_2$ od vremena. Takođe je moguće koristiti i numeričku simulaciju. Oba slučaja zanemaruju uticaj trenja, tako da su to samo aproksimacije realnog sistema.

## II. ANALITIČKO REŠENJE ZA MALE UGLOVE

Koordinate centra masa prvog i drugog klatna su:

$$x_1 = l_{cm1}\sin(\theta_1),$$
$$y_1 = -l_{cm1}\cos(\theta_1),$$
$$x_2 = l_1\sin\theta_1 + l_{cm2}\sin(\theta_2),$$
$$y_2 = -l_1\cos\theta_1 - l_{cm2}\cos(\theta_2),$$

gde su $l_{cm1}$ i $l_{cm2}$ rastojanja od početka klatna do centra mase.

Prema tome su odgovarajuće brzine po x i y osi:

$$\dot{x}_1 = l_{cm1}\cos(\theta_1)\dot{\theta}_1,$$
$$\dot{y}_1 = l_{cm1}\sin(\theta_1)\dot{\theta}_1,$$
$$\dot{x}_2 = l_1\cos(\theta_1)\dot{\theta}_1 + l_{cm2}\cos\theta_2\dot{\theta}_2,$$
$$\dot{y}_2 = l_1\sin(\theta_1)\dot{\theta}_1 + l_{cm2}\sin\theta_2\dot{\theta}_2.$$

Langražijan je dat kao razlika kinetičke i potenijalne energije sistema:

$$L = \left(\frac{m_1 l_{cm1}^2}{2} + \frac{m_2 l_1^2}{2}\right)\dot{\theta}_1^2 + \frac{m_2}{2} l_{cm2}^2\dot{\theta}_2^2 +$$
$$+ m_2 l_1 l_{cm2}\dot{\theta}_1\dot{\theta}_2\cos(\theta_1 - \theta_2) +$$
$$+ \frac{1}{2}\left(I_1\dot{\theta}_1^2 + I_2\dot{\theta}_2^2\right) + g\cos(\theta_1)(m_1 l_{cm1} + m_2 l_1) +$$
$$+ m_2 g l_{cm2}\cos(\theta_2),$$

Jovan Dimitrijević je učenik na smeru Elektrotehničar računara, Tehnička škola „Nikola Tesla", Tome Ivanovića 24, 17530 Surdulica, Srbija, E-mail: jovandimitrijevic50@gmail.com

$I_1$ i $I_2$ su momenti inercija prvog i drugog klatna.

Da bi se moglo naći analitičko rešenje potrebno je koristiti aproksimaciju malih uglova. Razvijanjem u Tejlorov red funkcije cos dobija se:

$$\cos(\theta_1) \approx 1 - \frac{\theta_1^2}{2},$$
$$\cos(\theta_2) \approx 1 - \frac{\theta_2^2}{2},$$
$$\cos(\theta_1 - \theta_2) \approx 1 - \frac{(\theta_1 - \theta_2)^2}{2}.$$

Nakon zamene ovih izraza dobija se langražijan za male uglove. Sada se može napisati sistem diferencijalnih jednačina:

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{\theta}_1} - \frac{\partial L}{\partial \theta_1} = 0,$$
$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{\theta}_2} - \frac{\partial L}{\partial \theta_2} = 0.$$

Kada se nađu izvodi ovaj sistem se može zapisati u matričnom obliku.

$$\boldsymbol{M\ddot{\theta} + K\theta = 0}$$
$$\boldsymbol{M} = \begin{bmatrix} m_1 l_{cm1}^2 + m_2 l_1^2 + I_1 & m_2 l_1 l_{cm2} \\ m_2 l_1 l_{cm2} & m_2 l_{cm2}^2 + I_2 \end{bmatrix}$$
$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$$
$$\boldsymbol{K} = \begin{bmatrix} (m_1 l_{cm1} + m_2 l_1)g & 0 \\ 0 & m_2 g l_{cm2} \end{bmatrix}$$
$$\boldsymbol{0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Sopstvene frekvencije se mogu naći iz jednačine:

$$\det(K - M\omega^2) = 0.$$

$$\omega_1 = (((m_2 l_{cm2}(m_1 l_{cm1}(l_{cm1}+l_{cm2})+l_1 m_2(l_{cm2}+l_1)+I_1)+I_2(m_1 l_{cm1}+l_1 m_2))^{2}-4m_2 l_{cm2}(m_1 l_{cm1} + l_1 m_2)(m_2(l_{cm2}^2(m_1 l_{cm1}^2+I_1)+I_2 l_1^2)+I_2(m_1 l_{cm1}^2+I_1)))^{0,5}+m_2 l_{cm2}(m_1 l_{cm1}(l_{cm1} + l_{cm2})+l_1 m_2(l_{cm2}+l_1)+I_1)+I_2(m_1 l_{cm1} + l_1 m_2))^{0,5}\sqrt{\frac{g}{2m_2(l_{cm2}^2(m_1 l_{cm1}^2+I_1)+I_2 l_1^2)+2I_2(m_1 l_{cm1}^2+I_1)}}$$

$$\omega_2 = (-((m_2 l_{cm2}(m_1 l_{cm1}(l_{cm1}+l_{cm2})+l_1 m_2(l_{cm2}+l_1)+I_1)+I_2(m_1 l_{cm1} + l_1 m_2))^{2}-4m_2 l_{cm2}(m_1 l_{cm1}+l_1 m_2)(m_2(l_{cm2}^2(m_1 l_{cm1}^2+I_1)+I_2 l_1^2)+I_2(m_1 l_{cm1}^2 + I_1)))^{0,5}+m_2 l_{cm2}(m_1 l_{cm1}(l_{cm1}+l_{cm2})+l_1 m_2(l_{cm2}+l_1)+I_1)+I_2(m_1 l_{cm1} + l_1 m_2))^{0,5}\sqrt{\frac{g}{2m_2(l_{cm2}^2(m_1 l_{cm1}^2+I_1)+I_2 l_1^2)+2I_2(m_1 l_{cm1}^2+I_1)}}$$

Za potpuno opisivanje zavisnosti ugla od vremena potrebno je naći i sopstvene vektore iz jednačine:
$(\boldsymbol{K}-\boldsymbol{\omega}^2\boldsymbol{M})\boldsymbol{H}=\boldsymbol{0}$.

$$H_{21} = (-((m_2 l_{cm2}(m_1 l_{cm1}(l_{cm1}+l_{cm2})+l_1 m_2(l_{cm2}+l_1)+I_1)+I_2(m_1 l_{cm1} + l_1 m_2))^{2}-4m_2 l_{cm2}(m_1 l_{cm1}+l_1 m_2)(m_2(l_{cm2}^2(m_1 l_{cm1}^2+I_1)+I_2 l_1^2)+I_2(m_1 l_{cm1}^2 + I_1)))^{0,5}-m_2 l_{cm2}(m_1 l_{cm1}(l_{cm1}-l_{cm2})+l_1 m_2(l_1-l_{cm2})+I_1)+I_2(m_1 l_{cm1} + l_1 m_2))\frac{1}{2l_1 m_2^2 l_{cm2}^2}$$

$$H_{22} = (((m_2 l_{cm2}(m_1 l_{cm1}(l_{cm1}+l_{cm2})+l_1 m_2(l_{cm2}+l_1)+I_1)+I_2(m_1 l_{cm1} + l_1 m_2))^{2}-4m_2 l_{cm2}(m_1 l_{cm1}+l_1 m_2)(m_2(l_{cm2}^2(m_1 l_{cm1}^2+I_1)+I_2 l_1^2)+I_2(m_1 l_{cm1}^2 + I_1)))^{0,5}-m_2 l_{cm2}(m_1 l_{cm1}(l_{cm1}-l_{cm2})+l_1 m_2(l_1-l_{cm2})+I_1)+I_2(m_1 l_{cm1} + l_1 m_2))\frac{1}{2l_1 m_2^2 l_{cm2}^2}$$

Zavisnost ugla od vremena je data sledećom formulom:
$$\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix} = C_1\begin{bmatrix}1\\ H_{21}\end{bmatrix}\cos(\omega_1 t+\varphi_1) + C_2\begin{bmatrix}1\\ H_{22}\end{bmatrix}\cos(\omega_2 t+\varphi_2).$$
Konstante $C_1$, $C_2$, $\varphi_1$ i $\varphi_2$ se mogu odrediti iz početnih uslova [1].

## III. NUMERIČKA SIMULACIJA

Zavisnost ugla od vremena se može odrediti numerički Runge-Kuta metodom, rešavanjem sistema od 4 jednačine dobijene od parcijalnih izvoda hamiltonijana [2].

$$\dot{\theta}_1 = \frac{\partial H}{\partial p_1}$$
$$\dot{\theta}_2 = \frac{\partial H}{\partial p_2}$$
$$\dot{p}_1 = -\frac{\partial H}{\partial \theta_1}$$
$$\dot{p}_2 = -\frac{\partial H}{\partial \theta_2}$$

Kompletan kod iz programskog paketa Mathematica je prilicno obima, pa iz tih razloga nije dostupan u samom radu.

## IV. NAČIN MERENJA

Uglovi $\theta_1$ i $\theta_2$ se mogu meriti rotacionim senzorima, koji se mogu naći u mehaničko-optičkom mišu. Oni se sastoje od infracrvenih dioda i fotodetektora. Prilikom rotacije disk na osovini (Slika 1) stvara pulsirajući signal koji se obrađuje i šalje mikrokontroleru. Mikrokontroler Arduino Uno i mehanički miš komuniciraju pomoću biblioteke PS2Mouse. Drugi sensor meri relativan ugao između dva klatna pa je za računanje otklona drugog klatna potrebno oduzeti otklon prvog klatna od relativnog. Prikazan je kod za merenje uglova:

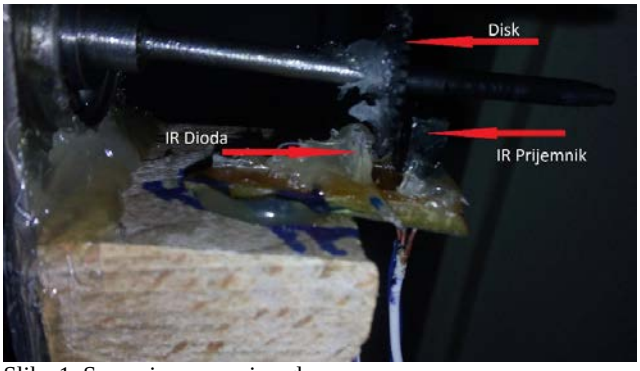
```
#include <PS2Mouse.h>
#define MOUSE_DATA 5
#define MOUSE_CLOCK 6

PS2Mouse mouse(MOUSE_CLOCK, MOUSE_DATA, REMOTE);

int x=0;
int y=0;
int r=0;

void setup()
{
 Serial.begin(38400);
 mouse.initialize();
 mouse.set_resolution(3);
 mouse.set_sample_rate(200);
}

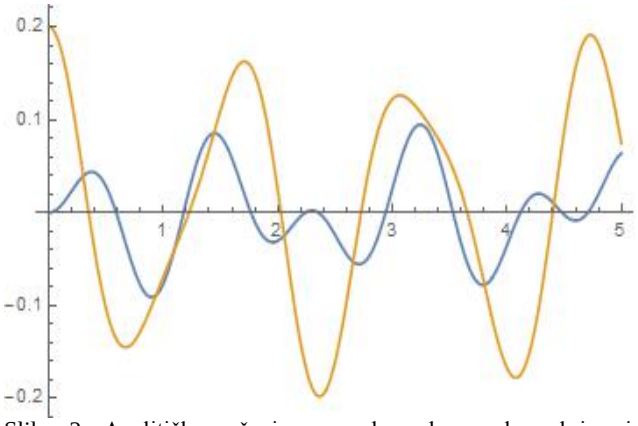
void loop()
{
 int data[2];
 mouse.report(data);
 x+=data[1];
 y+=data[2];
 r=y-x;
 Serial.print(x);
 Serial.print(",");
 Serial.print(y);
 Serial.print(",");
 Serial.print(r);
 Serial.println();
}
```



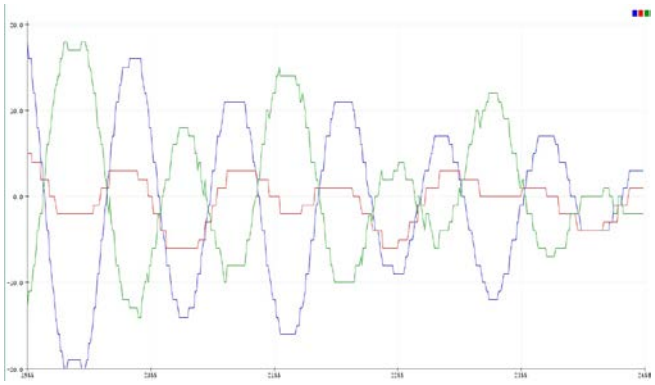

Slika 1. Senzori za merenje uglova.

## V. REZULTATI I DISKUSIJA

Kod analitičkog rešenja postoji greška jer se trigonometrijska funkcija $\cos$ predstavlja samo kvadratnim i slobodnim članom. Numerička simulacija nije ograničena malim uglovima, ali takođe ima grešku koja zavisi od metode i koraka vremena. Slika 2 predstavlja rezultate analitičkog rešenja za male uglove. Početni uslovi su: $\theta_1$ = 0 rad; $\theta_2$ = 0,2 rad; $m_1$ = 1 kg; $m_2$ = 0,5 kg; $l_1$ = 0,3 m; $l_{cm1}$ = 0,15 m; $l_{cm2}$ = 0,2 m; $I_1$ = 0.025 $\mathrm{kgm^2}$; $I_2$ = 0.034 $\mathrm{kgm^2}$ i klatno kreće da osciluje iz mirovanja. Numerička simulacija daje slične rezultate kada se je broj koraka dovoljno veliki (npr. 10000). Slika 3 prikazuje rezultate merenja. Greška pri merenju veoma zavisi od toga koliko su precizno postavljeni senzori.




Slika 2. Analitičko rešenje za male uglove, plava kriva je zavisnost ugla prvog, a žuta kriva drugog klatna.



Slika 3. Rezultati merenja, crvenom je označen ugao prvog klatna, plavom relativan ugao između prvog i drugo klatna i zelenom apsolutni ugao drugog klatna.

## VI. ZAKLJUČAK

Merenja nisu precizna zato što senzori nisu idealno postavljeni, prisutan je i drift, ali se može prepoznati oblik kriva koje opisuje kretanje klatna. Za analitičko rešenje ovaj rad je proširio model koji je opisivao duplo klatno sastavljeno od niti za koje su zakačene kuglice na model koji uračunava i pozicije centra mase i moment inercije. Dalji razvitak bi mogao predstavljati upoređivanje teorijskih modela sa realnim merenjima.

## ZAHVALNICA

Veliku zahvalnost na pruženoj pomoći dugujem Veselinović Filipu i profesoru Bratislavu Radovanoviću.

## LITERATURA

[1] Double Pendulum, https://www.math24.net/double-pendulum/

[2] Double Pendulum, http://scienceworld.wolfram.com/physics/DoublePendulum.html

# Sistem pametne kuće

Nikola Miljković

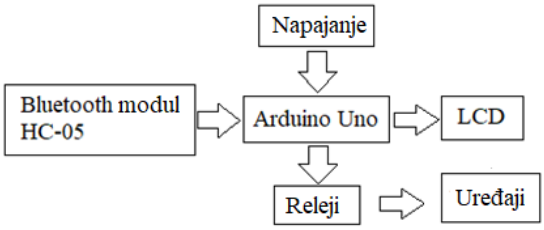
*Sadržaj* - Pametne kuće su moderno i inventivno rešenje za sve koji žele da uživaju u potpunom komforu i kontroli doma. Automatizacija i daljinska kontrola uređaja u domu je revolucionaran koncept koji olakšava pristup i kretanje u domu. Cilj rada bila je realizacija pametne kuće uz pomoć Arduino razvojnog okruženja, mobilnog telefona sa Android operativnim sistemom i Bluetooth-a. Najpre je kreirana Android aplikacija uz pomoć koje bi se davale komande sistemu, a za to je iskorišćena aplikacija App Inventor. Nakon kreiranja aplikacije napisan je kod u Arduino IDE softverskom paketu. Električna šema je odrađena u programu Fritzing.

## I. Uvod

Sve ono što se nekada činilo nemogućim, upravljanje kućnim aparatima daljinskim upravljačem, glasom, senzorima, postala je naša realnost i izvesna budućnost. Pametna kuća (Smart home) je spoj tehnologije i stila življenja. Sistemi pametne kuće integrišu sistem grejanja, hlađenja, protivprovalni i protivpožarni sistem, kućnu rasvetu i audio/video sistem u jedinstvenu celinu koja omogućava optimalno korišćenje energije i komforno upravljanje kućom. Takav sistem povećava vrednost i sigurnost doma, čini ga lepšim i komfornijim za život, a ekonomično trošenje energije smanjiće troškove života.

## II. Princip rada

Šema uređaja za kontrolu je prikazana na slici 1 i sastoji se od šest blokova: napajanje, Arduino Uno [1], bluetooth modul HC-05 [2], LCD [3], releji [4] i na kraju uređaji koje mi kontrolišemo. Arduino Uno je mozak sistema i zato se nalazi u središtu. Preko njega se napajaju ostale komponente sistema, vrši prijem podataka sa bluetooth modula i u zavisnosti od primljenog podatka uključuje određeni uređaj preko releja. LCD ima ulogu da nam pokaže trenutno stanje, odnosno da se na njemu ispiše koji je uređaj u tom trenutku uključen/isključen.




Slika 1. Blok šema uređaja.

Nikola Miljković je student master studija na modulu Elektonika i mikrosistemi, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: nikola3991@live.com

### *A. Kreiranje aplikacije*

App Inventor za Android je otvorena izvorna web aplikacija koju je originalno obezbedio Google, a sada je održava Massachusetts Institute of Technology (MIT). Omogućava korisnicima (programerima) da kreiraju softverske aplikacije za Android operativni sistem.

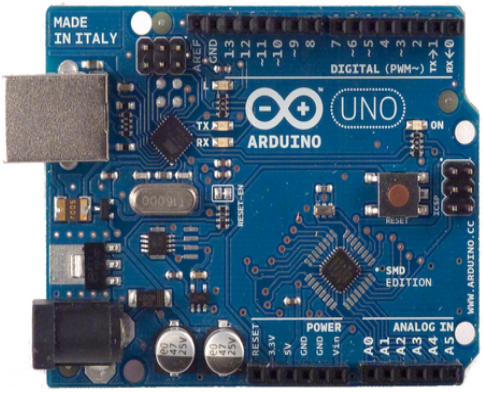


Slika 2. App Invertor.

### *B. Komponente potrebne za realizaciju uređaja*

1. Arduino UNO

Arduino Uno je ploča mikrokontrolera zasnovana na ATmega328 (prikazana na slici 3). Ima 14 digitalnih ulaza/izlaza (od kojih se 6 mogu koristiti kao PWM izlazi), 6 analognih ulaza, kristalni oscilator od 16 MHz, USB vezu, priključak za napajanje, ICSP zaglavlje i dugme za resetovanje.

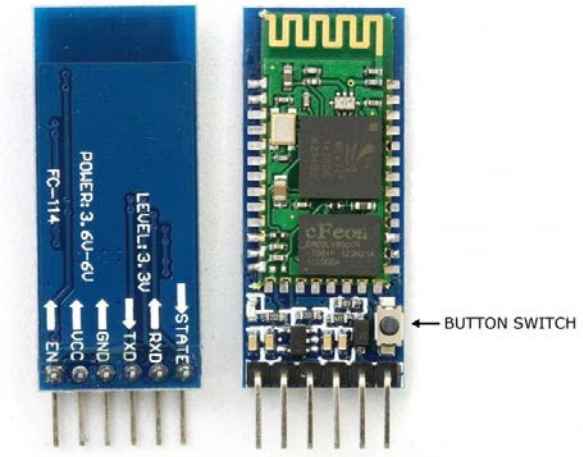



Slika 3. Arduiino UNO.

2. Bluetooth modul HC-05

Modul HC-05 je jednostavan za korišćenje bluetooth SPP (Serial Port Protocol), dizajniran za transparentnu serijsku vezu bežičnog povezivanja. Bluetooth modul HC-

05 se može koristiti u konfiguraciji master ili slave, što ga čini odličnim rešenjem za bežičnu komunikaciju. Ovaj modul serijskog porta je u potpunosti kvalifikovan bluetooth V2.0 + EDR (Enhanced Data Rate) 3Mb/s modulacija sa kompletnim 2.4GHz radio prijemnikom i baseband-om. Koristi CSR Bluecore 04- eksterni bluetooth sistem čip sa CMOS tehnologijom i sa AFH (Adaptive Frequency Hopping).




Slika 4. Bluetooth modul HC-05.

3. LCD modul sa I2C interfejsom

LCD ekran (engl. Liquid Crystal Display) je elektronski modul displeja i pronalazi široku primenu u aplikacijama [4]. 16x2 LCD ekran je veoma jednostavan modul i vrlo često se koristi u različitim uređajima i kolima. Ovi moduli se češće koriste od sedmosegmentnih i drugih višesegmentnih LED displeja. Razlozi:

- LCD je ekonomičan;
- lako se programira;
- nema ograničenja u prikazu posebnih ili čak prilagođenih znakova (za razliku od sedmosegmentnog displeja).




Slika 5. LCD modul sa I2C interfejsom.

4. Releji

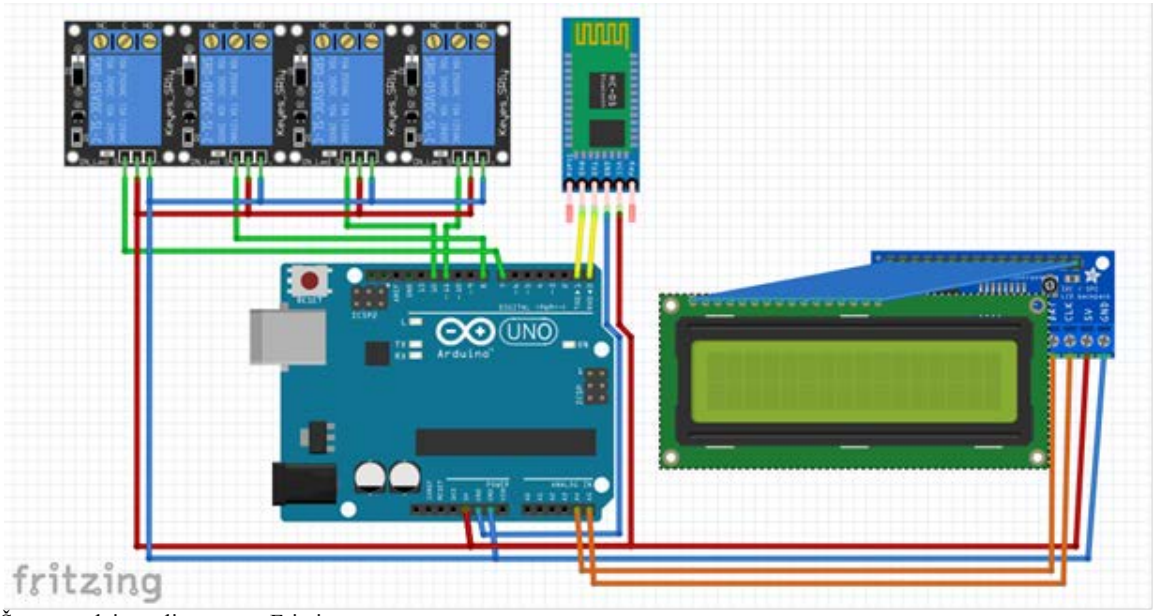
Relej je naprava koja se koristi za prekidanje ili uspostavljanje strujnog kola putem elektromagneta koji otvara i zatvara strujne kontakte.




Slika 6. Relejni modul SRD-05VDC-SL-C.

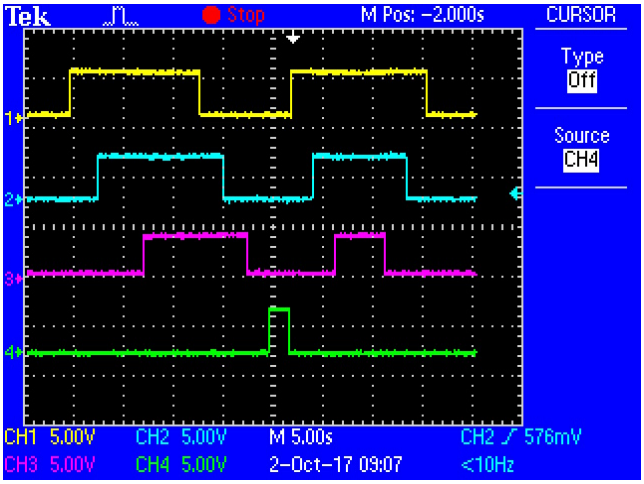
## III. ELEKTRIČNA ŠEMA SISTEMA

Električna šema sistema izrađena je u programu Fritzing.




Slika 7. Šema uređaja realizovana u Fritzing-u.

Praktično realizovan uređaj testiran je u laboratoriji. Signali koji se dovode do releja usnimljeni su korišćenjem osciloskopa (slika 8). Žutom bojom predstavljen je izlaz za svetlo, plavom je predstavljen izlaz za pozadinsko osvetljenje, ljubičastom bojom predstavljen je izlaz za ventilator i zelenom bojom je predstavljen kompjuter.




Slika 8. Rezultat simulacije snimljen na osciloskopu.

Uređaji su nasumično uključivani i isključivani. Kod zelenog signala se vidi kako simulira rad tastera za uključivanje kompjutera, nakon pritiska izlaz je u stanju logičke jedinice neko vreme i zatim se vraća u stanje logičke nule.

## IV. ZAKLJUČAK

U ovom radu prikazan je jedan od osnovnih uređaja za kontrolu kućnih aparata, princip rada i njegova primena. Data je električna šema, kompletna lista materijala potrebnih za realizaciju, a projekat je takođe praktično realizovan. Razlog korišćenja ovakvog sistema je višestruk, pre svega povećanje vrednosti doma, čini ga lepšim i komfornijim za život.

## ZAHVALNICA

Zahvaljujem se profesoru Danijelu Dankoviću na celokupnom angažovanju prilikom realizacije projekta.

## LITERATURA

[1] https://www.arduino.cc/en/Guide/ArduinoUno
[2] https://wiki.eprolabs.com/index.php?title=Bluetooth_Module_HC-05
[3] https://www.engineersgarage.com/electronic-components/16x2-lcd-module-datasheet
[4] https://arduino-info.wikispaces.com/file/view/SRD-05VDC-SL-C-Datasheet.pdf

# Operacioni pojačavač u konfiguraciji diferencijatora i integratora

Sandra Veljković, Danilo Đorđević, Marija Živković

*Sadržaj* – U ovom radu vršena je analiza funkcionisanja diferencijatora i integratora. Pre svega, razmtrana je primena i zastupljenost ovih malih ali značajnih kola u mnogim uređajima, a fokus je stavljen na sama kola, način na koji ona rade i matematićke proračune koji pojašnjavaju njihov rad. S obzirom da između matematićkih proračuna, simulacija i praktičnih realizacija vrlo često postoji odstupanje, ovde je prikazano upoređenje rezultata koji su dobijeni korišćenjem i jedne i druge metode.

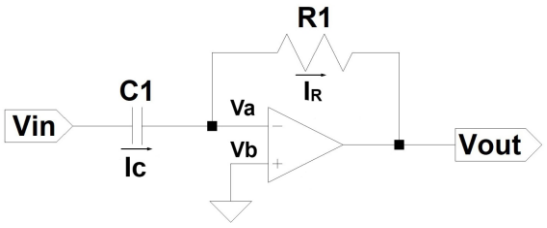
## I. UVOD

Diferencijator i integrator su kola koja imaju jako jednostavnu arhitekturu [1], ali zbog mogućnosti manipulacije različitim oblicima signala imaju široku primenu u velikom broju uređaja [2]. Na primer, diferencijator može da se koristi u analognognim kolima kako bi obezbedio manipulaciju diferencijacijom na ulaznom analognom naponu [3]. Još jedna interesantna primena bi bila u sistemima za monitoring. U ovom slučaju može da se koristi za praćenje promene vrednosti merenih tačaka i ako izmerena vrednost rezultira izlaznim naponom većim od standardne vrednosti, to se detektuje komparatorom. Zato može da se koristi kod projektovanja alarma i sistema upozorenja [3]. Takođe, i integrator može da se koristi u analognim kolima. Može da bude sastavni deo analogno-digitalnih konvertora, generatora testerastih signala, i aplikacijama koje se koriste za oblikovanje signala. Pored toga, još jedna interesantna primena bi bila u okviru kola koje se koristi za merenje protoka vode [4].

## II. DIFERENCIJATOR

Pri realizaciji diferencijatora koristi se operacioni pojačavač, u našem slučaju kolo LM2902N, kondenzator koji je postavljen između ulaza kola i invertujućeg ulaza operacionog pojačavača i otpornik koji predstavlja povratnu spregu. Diferencijator (slika 1) predstavlja pojačavačko kolo koje vrši matematičku operaciju diferenciranja, i proizvodi izlazni napon koji je direktno proporcionalan brzini promene ulaznog napona u vremenu [4].

Drugim rečima, ako je veća ili brža promena ulaznog signala, u tom slučaju je i amplituda izlaznog signala veća, što će biti pokazano matematičkim postupkom. Kada se na ulaz dovede određeni signal, kondenzator blokira bilo koju DC komponentu signala, dozvoljavajući samo promene AC tipa signala, tako da je izlazni napon proporcionalan diferencijalu ulaznog [5].




Slika 1. Osnovna konfiguracija diferencijatora.

Kako bi se matematički odredio izlazni napon, primenjuje se sledeći matematički postupak (relacije 1-5).

Korišćenjem uslova 1 i 2

$$i_{C1} = C_1 \frac{d(V_{in} - V_a)}{dt}, \quad i_{R_1} = \frac{V_a - V_{out}}{R_1} \tag{1}$$

$$V_a = V_b = 0 \quad \text{i} \quad i_{C1} = i_{R1} \tag{2}$$

dobija se:

$$C_1 \frac{d(V_{in} - V_a)}{dt} = \frac{V_a - V_{out}}{R_1}, \quad C_1 = \frac{dV_{in}}{dt} = -\frac{V_{out}}{R_1} \tag{3}$$

$$V_{out} = -C_1 R_1 \frac{dV_{in}}{dt}, \quad V_{in} = A\sin\omega t = A\sin 2\pi f t \tag{4}$$

$$V_{out} = -C_1 R_1 A\omega \cos\omega t = -C_1 R_1 A 2\pi f \cos\omega t \,. \tag{5}$$

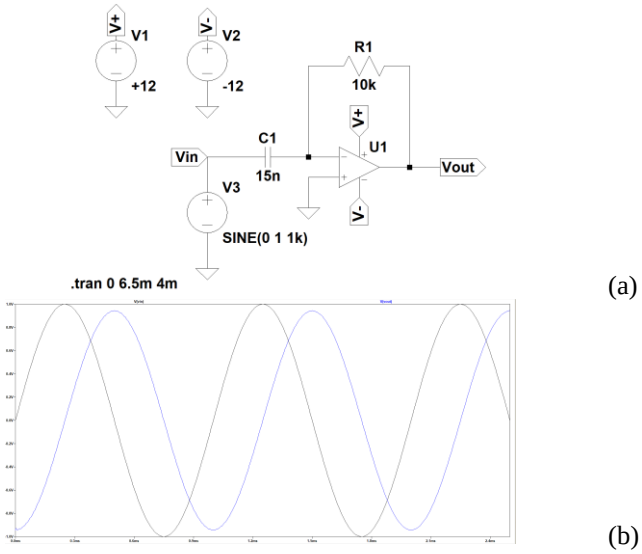
Za simulaciju ovog kola koristi se program LT Spice. Vrednost koja se koristi za kondenzator je $C_1 = 15$ nF a za otpornik $R_1 = 10$ kΩ. Ulazni signal se dovodi na jedan kraj kondenzatora, čija je amplituda 1 V i frekvencija 1 kHz. Rezultat simulacije je prikazan na slici 2.

Izlazni signal nije potpuno stabilan jer se javljaju visokofrekventni šumovi. Nakon simulacije kola, rađena je praktična realizacija kola. Izlaz kola je snimljen osciloskopom, kao što je prikazano na slici 3.

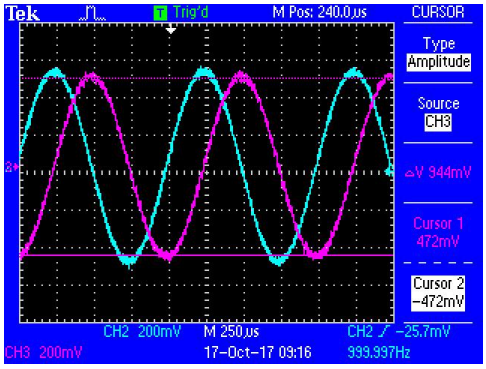
Teorijski, odnosno korišćenjem proračuna, predviđena vrednost izlaznog napona je 942 mV. Upoređivanjem rezultata dobijenih i simulacijom i praktičnom realizacijom dolazi se do zaključka da su odstupanja jako mala, do nekoliko mV, jer praktično dobijena vrednost izlaznog napona iznosi 944 mV. S obzirom da je na ulaz diferencijatora doveden signal sinusnog oblika, a da kolo vrši operaciju diferenciranja, na izlazu se javlja kosinusni signal koji počinje negativnom poluperiodom jer se

Sandra Veljković, Danilo Đorđević i Marija Živković su studenti na modulu Elektronske komponente i mikrosistemi na Katerdri za mikroelektroniku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: sandra.veljkovic@elfak.rs, danilo.djordjevic@elfak.rs, marija.zivkovic@elfak.rs.

prilikom proračuna javlja negativni predznak koji figuriše do kraja.




Slika 2. Standardna konfiguracija diferencijatora (a), ulazni i izlazni signal (b).

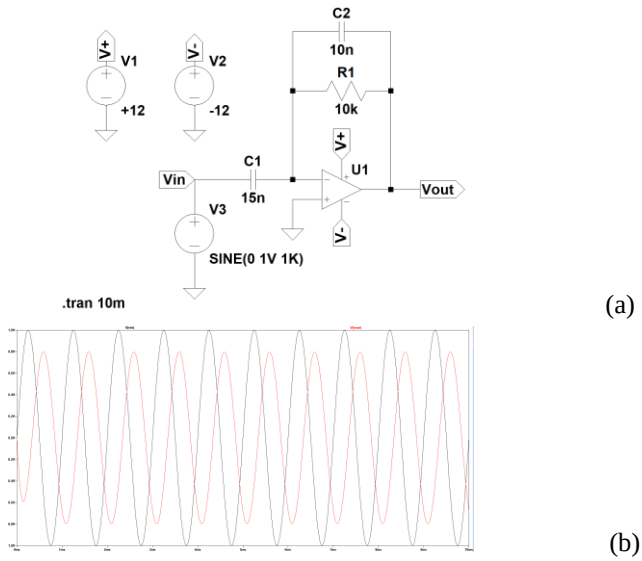



Slika 3. Ulazni i izlazni signali standardne konfiguracije diferencijatora snimljni osciloskopom.

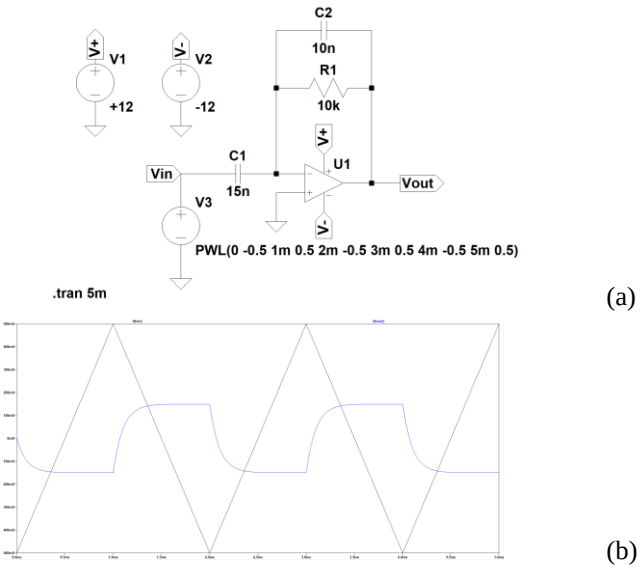
Važno je naglasiti da postoji poboljšana verzija diferencijatora, pri čemu je jedina razlika u odnosu na početnu šemu u kondenzatoru koji se postavlja iznad otornika. Taj kondenzator se dodaje kako bi se potisnule visoke frekvencije u povratnoj sprezi. U simulaciji korišćen je kondenzator vrednosti $C_2 = 10$ nF, i ta vrednost kondenzatora je tipična vrednost koja se u praksi koristi za odklanjanje visokih frekvencija. Simulacija poboljšane verzije diferencijatora je prikazana na slici 4.

Kao što može da se primeti, kod izlaznog signala poboljšanog kola diferencijatora nema smetnji i šumova, kao kod početnog kola.

Pored sinusnog signala, na ulaz kola može da se dovede i signal nekog drugog oblika, pravougaonog ili testerastog. Kada se na ulaz kola dovede pravougaoni signal, na izlazu se dobija signal testerastog oblika, a kada je na ulazu signal testerastog oblika, na izlazu je signal pravougaonog oblika. Na slici 5 je prikazana simulacija diferencijatora kada se na ulaz dovodi signal testerastog obika.

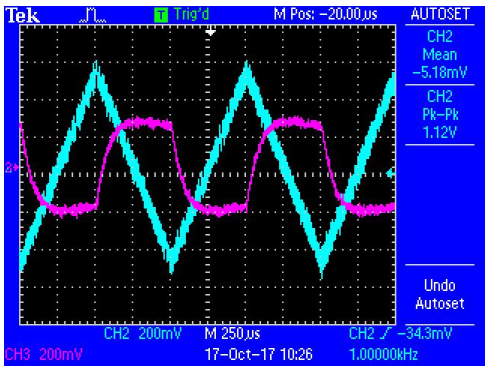



Slika 4. Poboljšana konfiguracija diferencijatora (a), ulazni i izlazni signal (b).




Slika 5. Poboljšana konfiguracija diferencijatora (a), ulazni i izlazni signal (b).

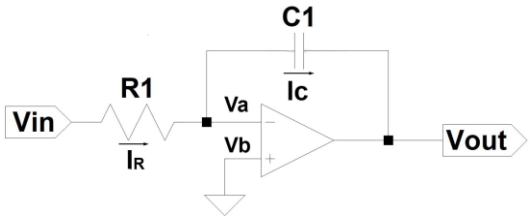
Međutim, iako teorijski gledano na izlazu treba da se javi signal pravougaonog oblika, ni u simulaciji ni u praktičnoj realizaciji to nije slučaj a razlog tome je kondenzator. Vrednost kondenzatora ne sme da bude ni previse mala, jer će u tom slučaju brzo da se napuni i isprazni, a ni previse velika, jer tada neće stići da se napuni dovoljno a već će početi da se prazni. Zbog toga je jako teško biti i blizu idealnog teorijskog slučaja. Slika 6 prikazuje rezultat praktične realizacije ovog primera.




Slika 6. Rezultat praktične realizacije poboljšane konfiguracije diferencijatora sa testerastim ulaznim signalom.

## III. INTEGRATOR

Kolo koje je po svom izgledu jako slično diferencijatoru je integrator (slika 7). Fizički gledano, razlika je u pozicijama na kojima se kondenzator i otpornik nalaze. Takođe, i u ovoj realizaciji koristi se operacioni pojačavač LM2902N. Integrator predstavlja kolo koje vrši matematičku operaciju integracije, i proizvodi izlazni napon koji je direktno proporcionalan brzini promene ulaznog napona u vremenu. Kada se na ulazu kola pojavi signal, kondenzator je prazan, u tom slučaju ima malu otpornost, i ponaša se gotovo kao kratak spoj koji omogućava maksimalnu struju kroz ulazni otpornik [6]. Vremenom kondenzator počinje da se puni, a samim tim i njegova impedansa počinje polako da raste. Impedansa kondenzatora, $X_C$, je u tom trenutku veoma mala, tako da je pojačanje $X_C/R_1$ takođe veoma malo, što dovodi do malog pojačanja izlaznog signala, ($A$<1). Koliko će kondenzator da se napuni zavisi od RC konstante.




Slika 7. Osnovna konfiguracija integratora.

Kako bi matematički bio određen izlazni napon, primenjuje se matematički postupak (relacije 6-10).

Korišćenjem uslova 6 i 7

$$i_{in} = \frac{V_{in} - V_a}{R_1}, \quad i_{C1} = C_1 \frac{d(V_a - V_{out})}{dt} \tag{6}$$

$$V_a = V_b = 0 \quad \text{i} \quad i_{in} = i_{C1} \tag{7}$$

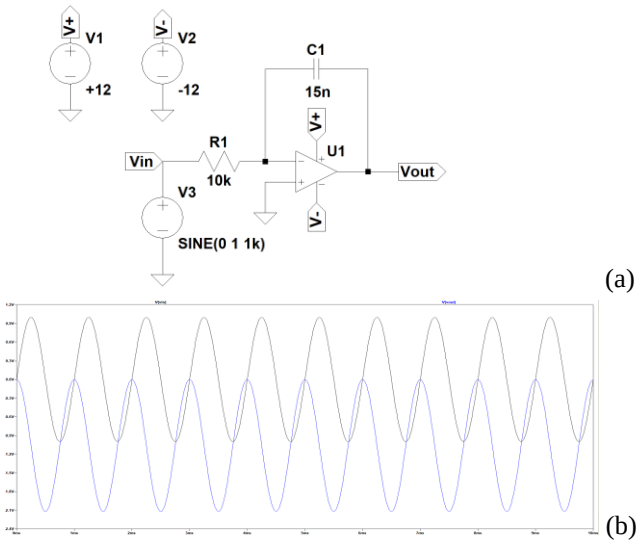
dobija se:

$$-C_1 \frac{dV_{out}}{dt} = \frac{V_{in}}{R_1}, \quad \int \frac{dV_{out}}{dt} dt = -\frac{1}{R_1 C_1} \int V_{in} dt \tag{8}$$

$$V_{in} = A \sin \omega t = A \sin 2\pi f t, \; V_{out} = -\frac{1}{R_1 C_1} \int_0^t \sin \omega t \, dt \tag{9}$$

$$V_{out} = V_{C_1} - \frac{A}{R_1 C_1 2\pi f} + \frac{A}{R_1 C_1 2\pi f} \cos \omega t \cdot \tag{10}$$

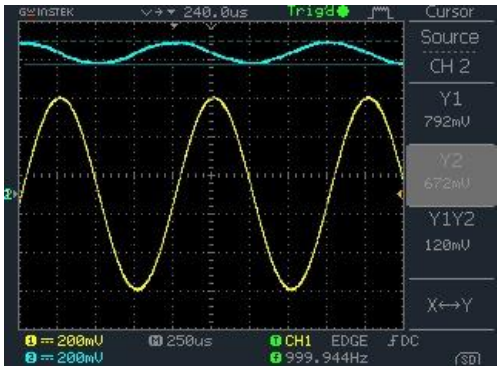
I prilikom simulacije ovog kola (integratora) korišćen je progam LT Spice, a od komponenata korišćen je otpornik $R_1 = 10\ \text{k}\Omega$ i kondenzator $C_1 = 15\ \text{nF}$. Ulazni signal, koji je sinusnog obilika i ima amplitudu 1 V i frekvenciju od 1 kHz, je doveden na jedan kraj otpornika. Rezultat simulacije je prikazan na slici 8.




Slika 8. Standardna konfiguracija integratora (a), ulazni i izlazni signal (b).

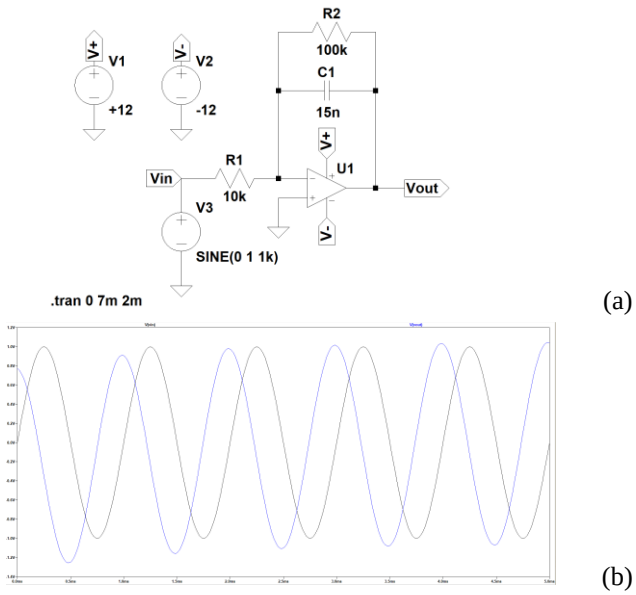
S obzirom da se na ulaz dovodi signal sinusnog oblika, a kolo vrši operaciju integraljenja, na izlazu kola javlja se kosinusni signal, koji počinje negativnom poluperiodom, što potiče od minusa koji se dobija pri integraljenju. Primećuje se offset koji potiče od punjenja kondenzatora u povratnoj sprezi, što utiče na virtuelnu masu.

Nakon simulacije kola, rađena je praktična realizacija kola. Izlaz kola je snimljen osciloskopom (slika 9).



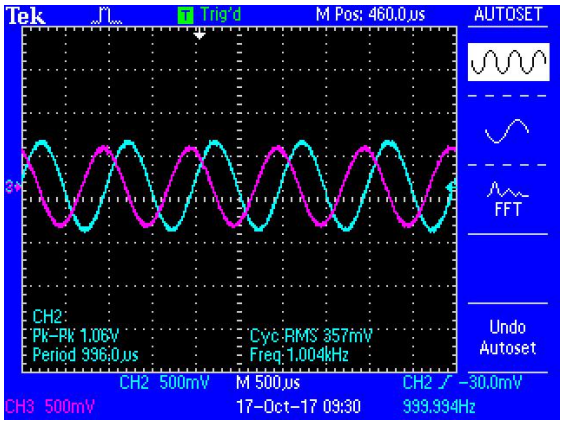

Slika 9. Ulazni i izlazni signali standardne konfiguracije integratora snimljni osciloskopom.

Međutim, s obzirom da je ovde izlazni signal pomeren, može da se izvrši korekcija tako što se postavi još jedan otpornik iznad kondenzatora. Otpornik služi kako bi se kondenzator praznio, što smanjuje offset izlaznog signala. Vrednost novog otpornika treba da bude veća 10 puta od vrednosti otpornika $R_1$. U ovom slučaju korišćen je otpornik $R_2$ = 100 kΩ. Simulacija poboljšane konfiguracije integratora je prikazana na slici 10.




Slika 10. Poboljšana konfiguracija integratora (a), ulazni i izlazni signal (b).

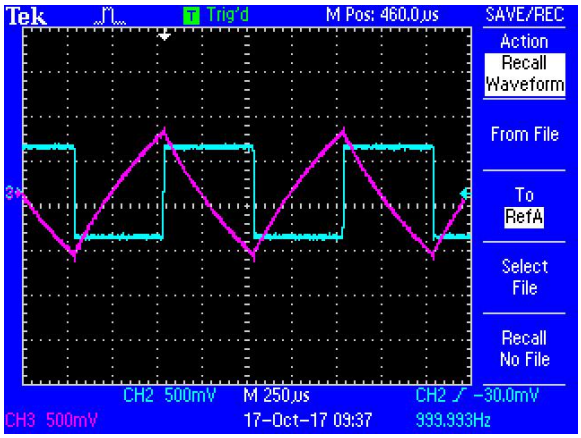
Sa slike može da se primeti da je samo u početnom trenutku signal pomeren jer je kondenzator počeo da se puni, a zatim i da se prazni preko otpornika $R_2$ što traje sve dok kolo ne postane stabilno. Nakon kompjuterske simulacije, izvršena je praktična realizacija (slika 11).




Slika 11. Ulazni i izlazni signali poboljšane konfiguracije integratora snimljni osciloskopom.

Pored sinusnog signala, na ulaz kola može da se dovede i signal nekog drugog oblika, pravougaonog ili testerastog. Kada se na ulaz kola dovede pravougaoni signal, na izlazu kola javlja se signal testerastog oblika, kao što je prikazano na slici 12. Naime, integraljenjem pravougaonog signala praktičcno se vrši operacija integraljenja:

$$\int a\,dx = ax + c\,. \qquad (11)$$




Slika 12. Rezultat praktične realizacije poboljšane konfiguracije integratora sa pravougaonim ulaznim signalom.

## III. ZAKLJUČAK

I ako danas živimo u digitalnoj eri, vremenu kada je prioritet razvijanje digitalnih sistema i maksimalno ulaganje u njih, ne treba zaboraviti jednostavna analogna kola poput diferencijatora i integratora. Naizgled ne tako bitni, ova mala kola mogu u mnogome da pojednostave rad velikog broja uređaja, smanje cenu izrade i poboljšaju efikasnost. Iz tog razloga se treba upoznati sa njihovim funkcionisanjem i primenom, kako bi se njihove prednosti maksimalno iskoristile.

## ZAHVALNICA

Autori rada se zahvaljuju docentu dr Danijelu Dankoviću pre svega na ideji, a zatim i na izdvojenom vremenu, sugestijama i značajnoj pomoći pri praktičnoj realizaciji i pisanju rada.

## LITERATURA

[1] Z. Prijić, "Analogna mikroelektronika", predavanja, 2016, http://mikroelektronika.elfak.ni.ac.rs/analogna/index_sr.php
[2] V. B. Litovski, "Osnovi elektronike, Teorija, rešeni zadaci i ispitna pitanja" Beograd, 2006.
[3] http://www.radio-electronics.com/info/circuits/opamp-circuits/operational-amplifier-differentiator.php
[4] http://www.electronicshub.org/operational-amplifier-as-integrator/
[5] http://www.electronics-tutorials.ws/opamp/opamp_7.html
[6] http://www.electronics-tutorials.ws/opamp/opamp_6.html

# Vektorsko upravljanje mašinom sa stalnim magnetima

Bratislav Trojić

*Sadržaj* – Zadatak rada je da se oformi matematički model model sinhrone mašine sa permanentnim magnetima (SMPM), pogodan za modelovanje u paketu Matlab/Simulink. Potom izvršiti proračun parametara regulatora struje i brzine i rezultate proveriti simulacijama. Sličan postupak sprovesti i eksperimentalno u laboratoriji.

## I. UVOD

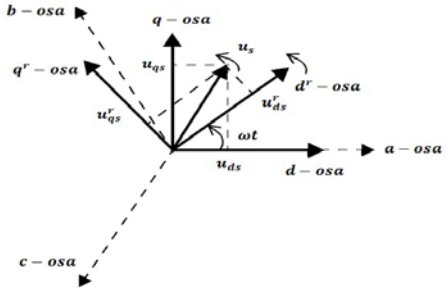
Razvojem digitalnih procesora postalo je moguće primeniti složene upravljačke algoritme i na motore naizmenične struje. Tako su se stvorile olakšane mogućnosti upravljanja pogona sa naizmeničnim motorima tj. karakteristike upravljanja su postale približne karakteristikama pogona sa jednosmernim motorima.

U poređenju sa drugim motorima, sinhrona mašina sa permanentnim magnetima (SMPM) ima bolje dinamičke performanse, manji je po veličini i ima veću efikasnost. SMPM je višenamenski, nelinearan i viskokospregnut sistem. Izlazni moment i struja statora su u složenoj funkcionalnoj relaciji. Magnetno polje se može razdvojiti da bi se dobile dobre, upravljive karakteristike.

## II. MATEMATIČKO MODELOVANJE SMPM-A

### *A. Klarkina i Parkova transformacija*

U industriji se više koristi trofazni SMPM, međutim upravljanje motorom preko trofaznih naizmeničnih veličina je dosta složen proces, pa se zbog toga pribegava transformacijama trofaznog sistema u dvofazni rotacioni sisitem. Međutim da bi se uspelo u tome, mora da se prođe kroz dve transformacije referentnog sistema. Prvo prebacivanje trofaznog stacionarnog sistema u dvofazni stacionarni, a potom prebacivanje dvofaznog stacionarnog referentnog sistema u dvofazni rotacioni, čime se postiže da se veličine posmatraju kao jednosmerne.




Slika 1. Klarkina i Parkova transformacija.

Bratislav Trojić je student master studija na Elektronskom fakultetu, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: bratislav.trojic@gmail.com

Na slici 1. vidimo vektorski prikaz transformacije veličina iz jednog referentnog sistema u drugi, pa potom u treći. Promenljive $u_{ds}{}^{r}$ i $u_{qs}{}^{r}$ su naponi u *dq* rotacionom referentnom sistemu, potom $u_{ds}$ i $u_{qs}$ u *dq* stacionarnom referentnom sistemu, a $u_{as}$, $u_{bs}$ i $u_{cs}$ naponi u *abc* stacionarnom referentnom sistemu. Vektor napona $u_s$ ima određeni izraz u jednom referentnom sistemu, a drugačiji u drugom, pa se izjednačavanjem dobija relacija (1) Klarkine transformacije kojom prebacujemo trofazni stacionarni sistem u dvofazni stacionarni:

$$\begin{bmatrix} u_{ds} \\ u_{qs} \end{bmatrix} = \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}\begin{bmatrix} u_{as} \\ u_{bs} \\ u_{cs} \end{bmatrix} \quad (1)$$

Sada želimo da prebacio dvofazni stacionarni referentni sistem u dvofazni rotacioni. To postižemo primenom Parkove transformacije, čija je relacija izvedena i data u jednačini (2):

$$\begin{bmatrix} u_{ds}^{r} \\ u_{qs}^{r} \end{bmatrix} = \begin{bmatrix} \cos\theta_r & \sin\theta_r \\ -\sin\theta_r & \cos\theta_r \end{bmatrix}\begin{bmatrix} u_{ds} \\ u_{qs} \end{bmatrix} \quad (2)$$

gde je $\theta_r$ ugao za koji je pomeren novi referentni sistem u odnosu na stari.

### *B. Matematički model*

Električni matemtaički model SMPM-a je jednostavno modelovati. Njegov stator po konstrukciji je isti kao i stator asinhronog motora ili svakog drugog sinhronog motora. Relacaije napona i fluksa trofaznog SMPM-a date su pod (3) i (4):

$$\begin{bmatrix} u_{as} \\ u_{bs} \\ u_{cs} \end{bmatrix} = R_s\begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix} + s\begin{bmatrix} \lambda_{as} \\ \lambda_{bs} \\ \lambda_{cs} \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} \lambda_{as} \\ \lambda_{bs} \\ \lambda_{cs} \end{bmatrix} = \begin{bmatrix} L_{aa} & L_{ab} & L_{ac} \\ L_{ba} & L_{bb} & L_{bc} \\ L_{ca} & L_{cb} & L_{cc} \end{bmatrix}\begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix} + \lambda_{af}\begin{bmatrix} \cos\theta_r \\ \cos\left(\theta_r - \frac{2\pi}{3}\right) \\ \cos\left(\theta_r + \frac{2\pi}{3}\right) \end{bmatrix} \quad (4)$$

gde je sa *L* označena induktivnost, a sa $\lambda_{af}$ fluks permanentnog magneta na rotoru, $i_{as}$, $i_{bs}$ i $i_{cs}$ su vrednosti struja u trofaznom stacionarnom sistemu, dok su $i_{qs}{}^{r}$ i $i_{ds}{}^{r}$ struje u dvofaznom rotacionom. Primenom transformacija iz prethodnog poglavlja, dobijamo relacije pogodne za

modelovanje motora u dvofaznom (*dq*) rotacionom referentnom sistemu, one su date u relacijama (5) i (6).

$$\begin{bmatrix} u_{ds}^r \\ u_{qs}^r \end{bmatrix} = \begin{bmatrix} R_s + sL_d & -\omega_r L_q \\ \omega_r L_d & R_s + sL_q \end{bmatrix} \begin{bmatrix} i_{ds}^r \\ i_{qs}^r \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_r \lambda_{af} \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} \lambda_{ds}^r \\ \lambda_{qs}^r \end{bmatrix} = \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \begin{bmatrix} i_{ds}^r \\ i_{qs}^r \end{bmatrix} + \begin{bmatrix} \lambda_{af} \\ 0 \end{bmatrix} \quad (6)$$

Mehanički deo mašine se modeluje preko Njutnove jednačine, koja je data pod (7):

$$J \frac{d\omega_r}{dt} = T_e - T_l \quad (7)$$

gde je $J$ moment inercije, $T_l$ moment opterećenja, a $T_e$ elektromagnetni moment mašine, koji se računa po relaciji (8).

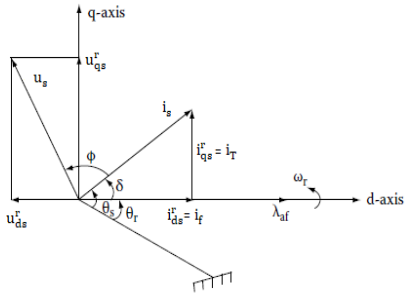
$$T_e = \frac{3}{2}\frac{P}{2}\left[\lambda_{af} \sin\delta + \left(L_q - L_d\right) i_{ds}^r \sin 2\delta\right] i_{qs}^r \quad (8)$$

gde je $P$ broj polova. Iz ove jednačine vidimo zavisnost momenta od struje statora.

## III. VEKTORSKO UPRAVLJANJE

Vektorsko upravljanje je drugačije poznato i kao upravljanje po polju pobude (engl. *Field oriented control*). Vektorsko upravljanje pretvara trofaznu struju statora u dvofaznu rotacionu struju po *dq* osama, jedna proizvodi fluks, dok druga moment. Ovo omogućava direktno upravljanje fluksa i momenta. Ovom vrstom upravljanje se postiže da SMPM bude praktično ekvivalentan mašini jednosmerne struje sa nezavisnom pobudom po karakteristikama. Model SMPM-a je nelinearan, ali se vektorskim upravljanjem on praktično linearizuje.

### *A. Princip upravljanja*




Slika 2. Fazorski dijagram SMPM-a.

Transformacijama se sistem dovede do toga da se barata sa jednosmernim veličinama, pa se tako sprovede da napon i fluks ne zavise od položaja rotora. Vektor fluksa permanentnog magneta je postavljen po *d* osi, pa se na taj način, fluksom može manipulisati promenom struje po *d* osi. Kvadraturna komponenta je pomerena za 90 stepeni i ona direktno utiče na elektromagnetni moment mašine.

Prebacivanjem struja Klarkinom i Parkovom transformacijom, dobija se relacija (9).

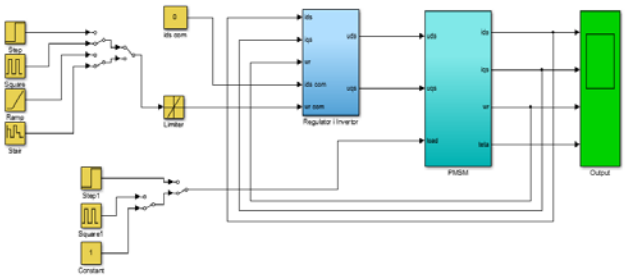
$$\begin{bmatrix} i_{ds}^r \\ i_{qs}^r \end{bmatrix} = i_s \begin{bmatrix} \cos\delta \\ \sin\delta \end{bmatrix} \quad (9)$$

gde je $\delta$ tzv. ugao momenta i pokazuje položaj struje statora (Slika 2.), gde je on konstantan za dato opterećenje, pa se tako dobije vrednost elektromagnetnog momenta:

$$T_e = \frac{3}{2}\frac{P}{2}\lambda_{af} i_{qs}^r \quad (10)$$

Tako smo sproveli koncept upravljanja mašinom jednosmerne struje sa nezavisnom pobudom na upravljanje SMPM-om, jer se vrši nezavisno upravljanje fluksom i momentom.

Dakle, model SMPM vektorskog upravljanja je postao širokorasprostranjen interes. Analizom matematičkog modela SMPM-a, uz pomoć jakih simulacija u *Matlab/Simulink*-u, upravljački sistem SMPM-a će biti podeljen na nekoliko nezavisnih funkcionalnih modula, kao što su: modul samog SMPM-a, modul regulatora (struje i brzine), modul invertora, modul transformacije koordinata itd. Kombinovanjem ovih modula, može se izgraditi model simulacije vektorskog upravljanje SMPM-a. U mom modelu upravljanja izostavljeni su blokovi transformacija iz razloga što je sam model SMPM-a pravljen direktno u *dq* rotacionom referentnom sistemu.



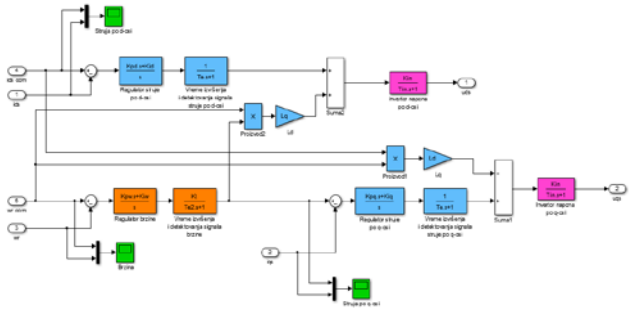
Slika 3. Model upravljanja u paketu *Matlab/Simulink*.

### *B. Projektovanje regulatora i invertora*

Zarad finijeg upravljanja, koristimo PI regulatore u kaskadnoj regulaciji brzine i struje. U našem slučaju unutrašnja petlja je na najnižem nivou i sa negativnom povratnom spregom po struji statora $i_q$. Regulatorom struje deluje se na invertor i motor, uzimajući za komandni ulaz izlaz iz regulatora brzine. Regulaciona petlja po brzini je na višem nivou čiji izlaz predstavlja komandni ulaz unutrašnje petlje.

Projektovanje, puštanje u rad i podešavanje parametara regulacionog sistema može da se vrši postupno, krećući se od najnižeg ka višim nivoima, uz uprošćavanje modela prethodnog nivoa. Konkretno u mom slučaju, projektovanje regulatora struje je vršeno metodom tehničkog optimuma preko propusnog opsega, dok su parametri pojačanja regulatora brzine izračunati metodom simetričnog optimuma. Vrednosti parametara nisu prikazane u radu, već samo krajnji produkt tj. simulacije, ali u literaturi pod [6], se može videti način na koji se došlo do njih.

Invertor je u ovom slučaju prikazan jednim blokom sa vremenskim članom.



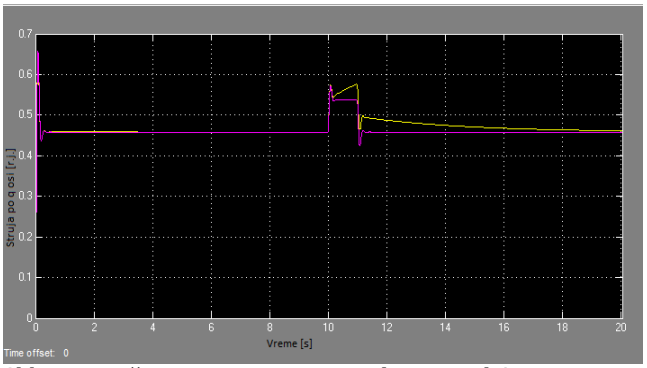
Slika 4. Kaskadna regulacija brzine i struje sa invertorom.

Za projektovanje cele šeme upravljanja korišćeni su kataloški podaci motora (SMPM) iz Laboratorije za elektromotorne pogone i vuču, Katedre za elektroenergetiku. To je tip servo motora proizvodnje Schneider Electric pod kataloškim nazivom *BMH1401P16A1A*.

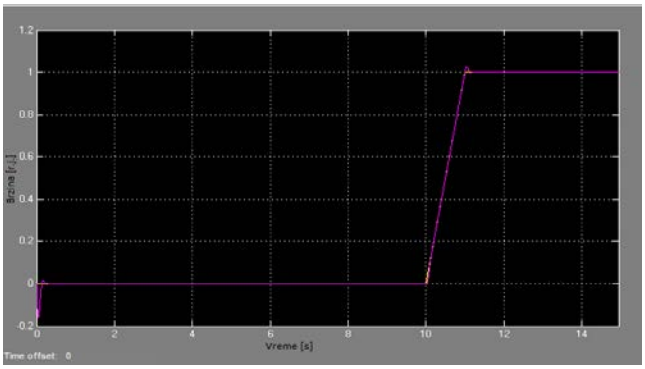
## IV. SIMULACIJA

Za ulazne promenljive simulacija u Matlab/Simulink-u su zadavani signali reference brzine i momenta opterećenja, u zavisnosti od tih uslova, posmatrano je ponašanje motora. Na slikama vidimo upoređene realne signale sa referencama.

*A. Referenca brzine je step signal(step reaguje na 10s), a opterećenje je konstantno(0.5 pu).*

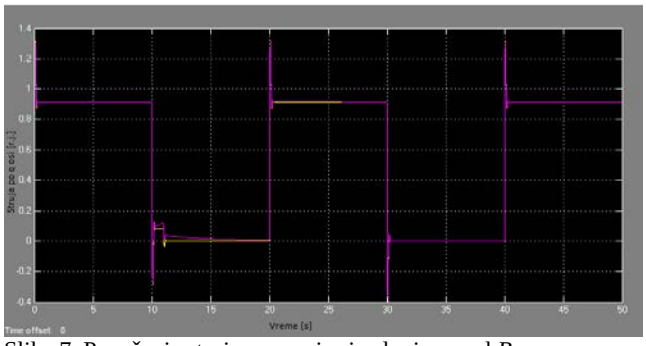



Slika 5. Ponašanje struje po q osi pri uslovima pod *A.*

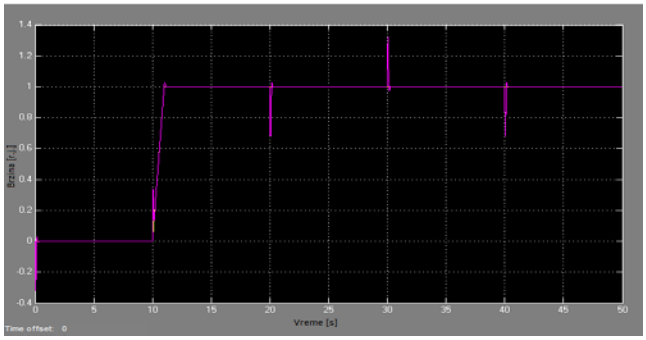



Slika 6. Ponašanje brzine pri uslovima pod *A.*

*B. Referenca brzine je step (step - 10s), a opterećenje pozitivni pravougaoni signal(amplituda - 1pu, perioda - 20s)*

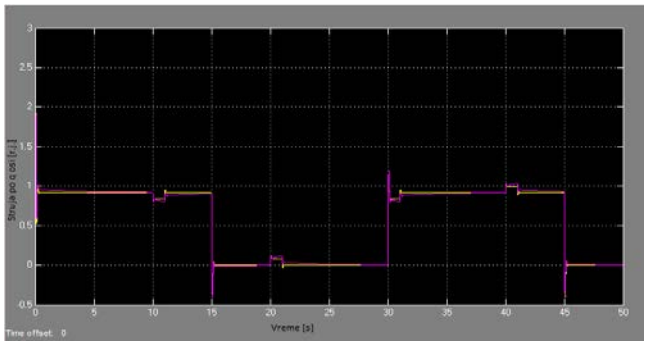



Slika 7. Ponašanje struje po q osi pri uslovima pod *B.*

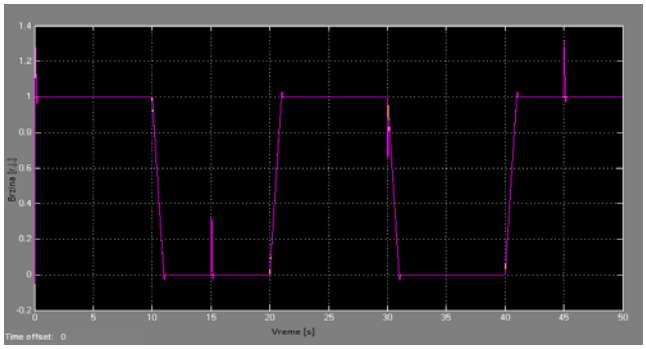



Slika 8. Ponašanje brzine pri uslovima pod *B.*

*C. Referenca brzine je pozitivni pravougaoni signal (amplituda - 1pu, perioda - 20s), kao i opterećenje (amplituda - 1pu, perioda - 30s)*




Slika 9. Ponašanje struje po q osi pri uslovima po *C.*




Slika 10. Ponašanje brzine pri uslovima pod *C.*

## V. EKSPERIMENT

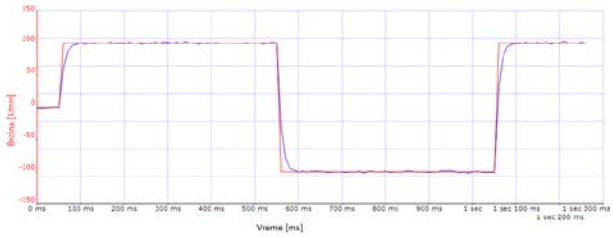
Eksperimentalni deo je primenjen na frekventnom pretvaraču serije Lexium 32 proizvođača Schneider Electric, pod kataloškom oznakom *LXM32CD30N4*. To je sprovedeno softverskim programiranjem pretvarača za potrebe eksperimenta i učitavanjem proračunatih parametara regulatora. Motor je puštan u pogon sa kratkotrajnom povorkom impulsa različitih signala reference brzine iz pretvarača (*expert tuning*), bez ikakvog opterećenja i razmatrano njegovo ponašanje.



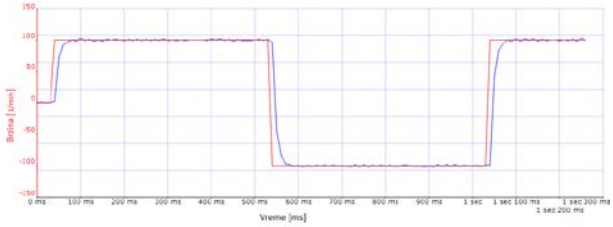
Slika 11. Izgleda laboratorijskog motora i pretvarača.

Rezultati sa mojim parametrima su upoređeni sa standardnim parametrima pretvarača i uočena je razlika.

*A. Proba sa simetričnom pravougaonom referencom brzine (amplituda - 100 obrtaja u minuti, perioda –1 s)*

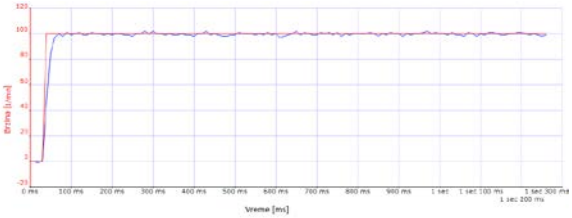


Slika 12. Brzina pod *A.* sa mojim parametrima.

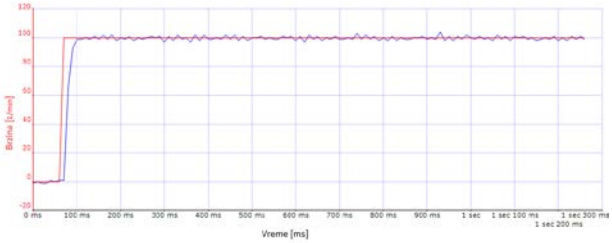


Slika 13. Brzina pod *A.* sa standardnim parametrima.

*B. Proba sa pozitivnom step referencom brzine*



Slika 14. Brzina pod *B.* sa mojim parametrima.



Slika 15. Brzina pod *B.* sa standardnim parametrima.

## VI. ZAKLJUČAK

Ovaj rad je imao za cilj da pokaže sa jedne strane kompleksnost, a sa druge jednostavnost u radu sa ovakvim motorima. Simulacijama i eksperimentalim radom je mahom, dokazana funkcionalnost vektorske upravljačke strukture SMPM-a, kako na računaru, tako i u laboratoriji. Kroz rad sa softverima i energetskom elektronikom se prožima ta kompleksnost rada sa ovim motorima. Proučili smo samo kretanje ovog motora i razmatrali njegove karakteristike. U njegovom ponšanju je uočeno da strujom statora se jednostavno može upravljati razvijenim momentom motora. Tačnije uočeno je da se sa promenom momenta opterećenja, menja i vrednost struje statora, što je očekivano, jer u stacionarnim režimima elektromagnetni moment koji razvija mašina biva jednak momentu opterećenja, pa na taj način struja statora (po q osi) je proporcionalna samom momentu koji razvija radna mašina. Kroz sve ove povratne informacije rada motora i upravljačke olakšice je sa druge strane pokazano i upravo suprotno od kompleksnosti tj. mogućnost da se upravljačka „muka“ razbije i dovede do jednostavnosti.

## LITERATURA


[1] R. Krishnan - „Permanent Magnet Synchronous and Brushless DC Motor Drives” Electrical and Computer Engineering Department Virginia Tech Blacksburg, Virginia, U.S.A. 2010.

[2] Haitham Abu-Rub (Texas A&M University at Qatar, Qatar), Atif Iqbal (Qatar University, Qatar and Aligarh Muslim University, India), Jaroslaw Guzinski (Gdansk University of Technology, Poland) - „HIGH PERFORMANCE CONTROL OF AC DRIVES WITH MATLAB/SIMULINK MODELS”, John Wiley & Sons Ltd, 2012.

[3] Luka Pravica – diplomski rad br. 463 „Strukture upravljanja sinkronim motorom s permanentnim magnetima” Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva, 2012.

[4] Kaushik Jash, Prof. Pradip Kumar Saha, Prof. Goutam Kumar Panda - „Vector Control of Permanent Magnet Synchronous Motor Based On Sinusoidal Pulse Width Modulated Inverter with Proportional Integral Controller” Department of Electrical Engineering, Jalpaiguri Government Engineering College, Jalpaiguri, West Bengal, India.

[5] Katalog: Scheineder Electric „BMH Servo motor – Motor manual 0198441113749, V1.04, 10.2012”

[6] Bratislav Trojić - „Primena Matlab-a za razvoj programa za vektorsko upravljanje motorima sa stalnim magnetima“, diplomski rad, Univerzitet u Nišu, Elektronski fakultet, oktobar, 2017.

# Real-Time Analysis of Sunflower Pollination Levels

Luka Aničin, Vanja Knežević

*Abstract* This paper tackles the problem of detecting and analyzing the pollination of sunflowers using optical data from a low-resolution camera connected to a single-board computer (SBC). It demonstrates an approach which is not computationally intensive and works on low-end hardware with high reliability.

## I. Introduction

As part of our research on unmanned aerial vehicles, we came across the many issues with using expensive components to solve problems which can be delegated onto low-end hardware. The main goal of this project was to find a way to achieve satisfactory results in object recognition using undemanding software, and correspondingly capable computing systems.

The results of this paper should hopefully demonstrate that the resources poured into high-end expensive hardware can be salvaged by figuring out clever software approaches.

## II. Hardware specifications

The SBC used is a Raspberry Pi 3 Model B, running Raspbian (a Debian Linux distribution). It's a general-purpose computer and not a very powerful one, which makes it ideal for our purpose.

The camera is an official module for the Raspberry Pi – Camera Module v2, boasting 8 Megapixels, but lacking in contrast which creates a problem with differentiation of colors, which is a centerpiece of the algorithm.

Parts of the tests were run on virtual machines in laptop computers, using the laptop cameras as sensors. The issues with those cameras are similar to the ones we faced with the Raspberry Pi camera.

The final implementation is meant to work on any reasonably-powered SBC, with a camera upwards of 2 Megapixels.

Our implementation also included a small autonomous vehicle which we designed and built ourselves for which took inputs from the camera system and was primarily used for testing the algorithm's ability to point the camera in the desired direction.

Luka Aničin and Vanja Knežević are students of Computer Science, Singidunum University, Danijelova 32, 11000 Belgrade, Serbia.

E-mails: luka.anicin@gmail.com, kirabo.kv@gmail.com

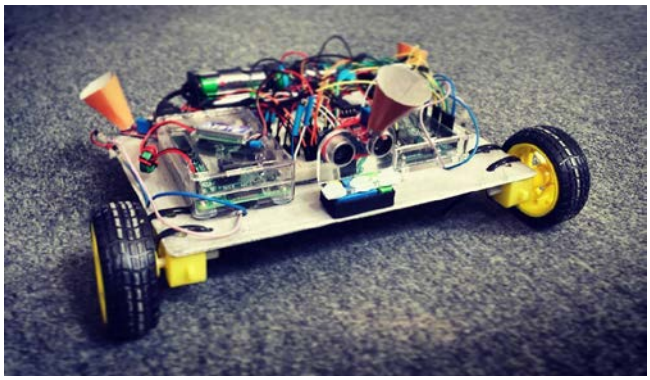


Fig. 1. "Sokownik", our autonomous robot. This configuration has an ultrasonic sensor instead of a camera.



Fig. 2. Raspberry Pi 3 Model B.

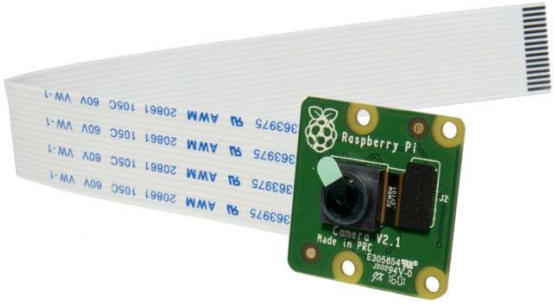



Fig. 3. Raspberry Pi Camera Module v2.

## III. Computer vision pipeline

The currently explored algorithm is based on a Haar cascade classifier[3] and a bit of image manipulation and filtering. The first phase is checking for presence of sunflowers on the image and marking their location in the 2D plane. Second, the perceived nearest flower is shifted to the center of the image by moving the camera on the

recording device. Thirdly, the distance of the centered flower from the camera in physical space is approximated. The final step is approximating the pollination of the observed sunflower.

*A. Detecting sunflowers in frame*

The current method used for sunflower detection is based on a Haar cascade classifier[1] which is trained on a custom dataset assembled from images on ImageNet[2] from which was built using 1948 negative and 1950 positive samples.

The training phase took 14 hours with a kernel size of 40, with 2 cores of an i5-7600 Intel CPU running on 3.5GHz, and 4GB of dedicated RAM.

On its own, the algorithm gave satisfying results, but in purpose of using the same system on a camera mounted on a flying vehicle, redundancy is necessary. To increase accuracy, a mask which eliminates all false positives by checking for matching colors is applied.



Fig. 4. From top to bottom: Classification without color filter, Classification with too broad a filter, Classification with tweaked color filter.

*B. Centering the recording device onto the perceived nearest sunflower*

This is done by calculating the pixel width of the bounding boxes around all detected flowers, selecting the one with the highest pixel width and steering the carrier vehicle in such a way that the largest flower is within a certain boundary around the center of the frame.

$$d = \left| \frac{w}{2} - \frac{p}{2} \right| \tag{1}$$

d – calculated distance from center of frame, w – frame width (in px), p – boundary box around flower (in px)

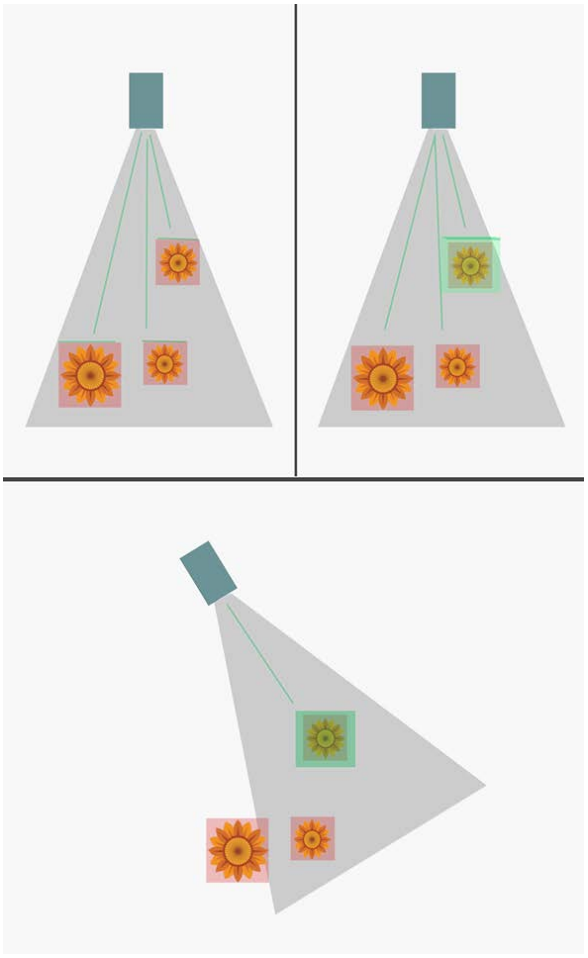


Fig. 5. Sketch of nearest-flower-centering process in action.

*C. Approximating distance of centered flower from the camera in physical space*

Having centered the perceived nearest flower, triangle similarity is used to approximate the distance from camera to flower in centimeters.

$$D = \frac{W \cdot f}{p} \tag{2}$$

$$f = \frac{p \cdot D}{W \cdot \alpha} \tag{3}$$

D – calculated distance from camera, W – average radius of a sunflower, f – focal distance, $\alpha$ - hyperparameter (magic number)

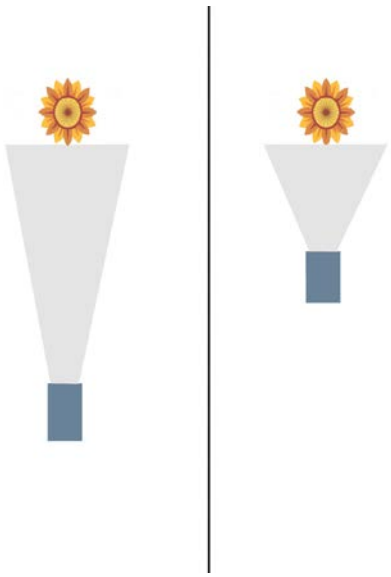


Fig. 6. Sketch demonstrating the measurement of camera distance to observed object, and subsequent moving of the camera closer to the object.

The sunflowers tested on have an average radius of 19cm, which is adopted as W. The issues come with calculating focal length. The first approach is to calculate focal distance at the moment of detection, but the results are poor. The current approach is simulating a situation in which the sunflower is 1cm away from the lens of the camera. In this version, parameter alpha (Eq. 3) is introduced, which in this case equals 10. This parameter makes up for part of the error in the analytical approximation.

The camera carrier vehicle then starts moving towards the sunflower steadily, until the flower is contained in most of the frame.

### *D. Pollination analysis*

The pollinated seeds have a very distinctive bright yellow bulge, compared to the dark brown color of the (yet) unpollinated seeds. Now that the flower is in focus, the amount of pollination can be measured. This is done in two phases.

Firstly, the seed-bearing inner circle of the sunflower is isolated and stored as a separate image after cropping out the bright yellow petals of the flower.

After that's done, the amount of yellow against brown in the isolated circle is analyzed and the ratio is returned as a percentage. Yellow represents the pollinated seeds, while brown are yet to be pollinated.

## IV. Further work

Future improvements to the current implementation would be switching part of the architecture onto a convolutional neural network [5] (CNN) model, which would improve accuracy and speed of primarily the pollination detection step, but also potentially the detection of sunflowers in an image as well. The potential downsides of this are that trained neural nets are a lot larger in size, and more computationally expensive than a trained Haar classifier.

A more affordable solution could be to train the Haar classifier on more stages. This approach will increase the number of features that the classifier uses to detect objects in the frame, which will reduce false positives and in turn increase accuracy, while remaining at the same computational cost and speed.

There is also the option of using either a stereo optical system or sonar sensors to accurately measure the distance of objects.

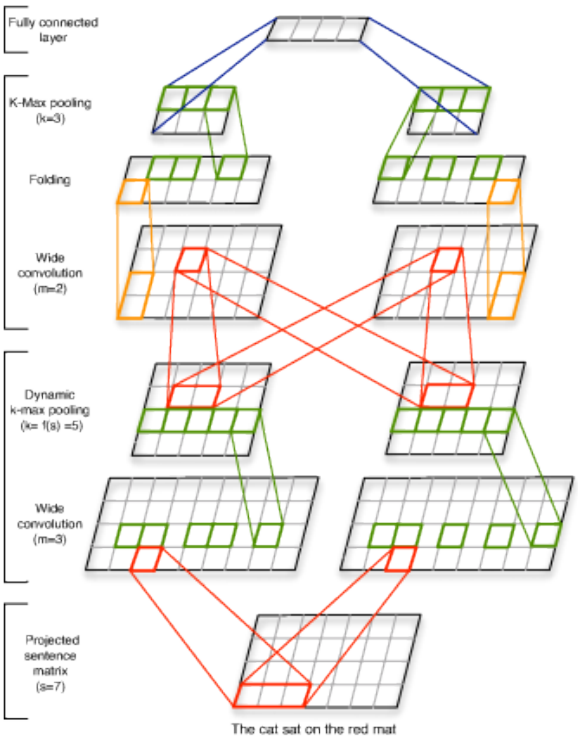



Fig. 7. A depiction of a convolutional neural network.

## V. Conclusion

Considering the limited capabilities of a low-end single camera system without any additional sensors, a logical conclusion arises; in order to create and operate a truly useful computer vision system, sensor fusion and relying on hardware instead of software are still unavoidable even with the stellar software engineering advancements in the field.

We will chase after state-of-the-art computer vision results, and try to work our way around the limitations of low-end hardware accordingly.

## References

[1] https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

[2] https://www.raspberrypi.org/products/camera-module-v2/
[3] P. Viola, and M. Jones, “Rapid Object Detection using a Boosted Cascade of Simple Features”, in Conference on Computer Vision and Pattern Recognition 2001, Kauai, Hawaii, USA, 2001, pp. 511-518.
[4] http://www.image-net.org
[5] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner: Gradient-Based Learning Applied to Document Recognition, Proceedings of the IEEE, November 1998, vol. 86, no. 11, pp. 2278-232.

# Prošireno realna i virtuelno realna 10. IEEESTEC konferencija

Miša Ković, Darko Đorđević, Aleksandar Vukašinović, Slavimir Stošović

*Sadržaj* – U ovom radu je predloženo rešenje „Virtuelna konferencija“ koje omogućava učesnicima konferencije pripremu za konferenciju, glasanje i odabir najboljih radova korišćenjem Android mobilnog uređaja, naočara za virtuelnu realnost (eng. Gear VR) i WEB servera sa bazom u kojoj se čuvaju radovi, kao i glasovi učesnika. Komunikacija između Android aplikacije i baze podataka se odvija korišćenjem PHP tehnologije. U susret modernim tehnologijama, aplikacija sadrži nekoliko modova koji su realizovani tehnologijama za proširenu i virtuelnu realnost, što treba da pruži novu dimenziju svim budućim IEEESTEC konferencijama ali je primenjivo i na sve druge konferencije.

## I. UVOD

Prema proceni stručnjaka proširena realnost (eng. *AR – Augmented Reality*) [1] i virtuelna realnost (eng. *VR - Virtual Reality*) [2] predstavljaju skup tehnologije koje će imati značajan uticaj na budući razvoj čovečanstva.

VR tehnologije se danas koriste u dva smera. U prvom smeru vrši se spajanje jedinstvene strukture vizuelnih, zvučnih a vrlo često i drugih čulnih iskustava (čula mirisa i ukusa), kako bi stvorile iluziju da neopipljive stvari smeštene u računaru mogu da se vide, čuju i osete na drugačiji, poseban način. U drugom, ove se tehnologije koriste da bi registrovale ljudske pokrete, zvuke i druge ulazne podatke na način koji ih je moguće obraditi na računaru. Dvosmerno korišćenje ovakvih tehnologija, od računara ka čoveku i od čoveka ka računaru, stvara interakciju između virtuelnih aplikacija i ljudi. Dinamičku karakteristiku ovom doživljaju omogućava komunikacija sa serverom i bazom podataka koja pruža mogućnost da urednici aplikacije na jednostavan način manipulišu sadržajem dostupnim korisnicima.

Jedan od načina za prikaz i doživljaj virtuelne realnosti su specijalne naočare sa elastičnim kaiševima za lakše i udobnije korišćenje, u njima se nalaze sočiva koja stvaraju doživljaj prisutnosti u virtuelnoj okolini koju korisnik gleda. Postoje dve osnovne grupe VR naočara. Prve predstavljaju naočare koje su žičanim putem povezane sa računarom i imaju ugrađeni displej za emitovanje sadržaja. Osim toga, prate ih i senzori koji omogućavaju precizno određivanje položaja korisnika u prostoru, kao i kontroleri [3] koji služe za upravljanje objektima na sceni.

Najpoznatiji modeli ove grupe su HTC Vive (slika 1a) i Oculus Rift. Drugu grupu čine modeli virtuelnih naočara, kao što je VR Box (Slika 1. b) koji su kompatibilni sa različitim modelima pametnih telefona (eng. *Smartphone*). Pametni telefon je zapravo neophodan za doživljaj virtuelne realnosti jer njegov displej služi za emitovanje sadržaja. U projektu koji je opisan u ovom radu koriste se naočare Samsung Gear VR (slika 1c) [4] koje su kompatibilne sa svim telefonima novije generacije Samsung Galaxy S serije, počev od Galaxy S6 pametnog telefona. Pored Samsungovog Gear VR-a, među najpoznatijim naočarama na tržištu su i Google cardboard naočare.




a) b) c)

Slika 1. VR naočare
a) HTC Vive, b) VR Box, c) Samsung Gear VR

Proširena stvarnost predstavlja stvaran svet proširen sa kompjuterski generisanim podacima. Primenjuje se u raznim oblastima kao što su: edukacija, arhitektura, turizam, vojska i mnoge druge. AR aplikacije se pišu u specijalnim 3D programima koji omogućavaju programerima da animiraju digitalnu informaciju pomoću AR „markera“ u realnom svetu. Sve što je potrebno za realizaciju proširene stvarnosti je mobilni telefon sa kamerom, marker, koji može biti u formi QR koda ili bilo koje druge slike, i aplikacija sa podrškom AR tehnologije.

U ovom radu opisana je mobilna aplikacija koja je kreirana u Unity [5] razvojnom okruženju, a može biti pokrenuta i na Android [6] i na iOS [7] mobilnim uredjajima. Aplikacija sadrži nekoliko modova koji su realizovani VR i AR tehnologijama. Pored moda za glasanje za pobednički rad, nalazi se i mod za prenos konferencije uživo kamerom koja snima u modu 360 stepeni, pregled 3D modela, zatim pregled promo videa timova koji su prethodno naveli takmičari prilikom prijavljivanja na takmičenje, kao i virtualnu turu Elektronskog fakulteta u Nišu.

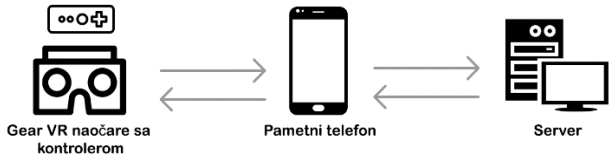
## II. MODEL KOMUNIKACIJE UREĐAJA

Princip funkcionisanja komunikacije između VR naočara, telefona i servera je grafički prikazan na Slici 2.

Miša Ković, Darko Đorđević i Aleksandar Vukašinović su studenti na smeru Savremene Računarske Tehnologije i članovi VTŠ Apps Tima Visoke Tehničke Škole Strukovnih Studija u Nišu, Aleksandra Medvedeva 20, e-mail: appsetam@vtsnis.edu.rs

Slavimir Stošović je koordinator VTŠ Apps Tima Visoke Tehničke Škole Strukovnih Studija u Nišu, Aleksandra Medvedeva 20, e-mail: slavimir.stosovic@vtsnis.edu.rs

Na serveru se nalazi baza podataka u kojoj se čuvaju radovi učesnika konferencije. Radove prijavljuju i snimaju u bazu sami autori pomoću portala IEEESTEC konferencije [8], koji je unapređen za potrebe ovog rada. Zbog specifične IEEESTEC konferencije i u nedostatku gotovog CMS rešenja, autori prvobitnog rada su napravili svoje rešenje prilagođena potrebama konferencije. Portal se nalazi na serveru koji prosleđuje podatke aplikaciji koja se nalazi na pametnom telefonu, a ona prevodi te podatke u 3D prikaz. Komunikacija između mobilnog uređaja i servera se odvija putem HTTP protokola [9]. Mobilne aplikacije šalju određeni zahtev serveru na osnovu koga on priprema tražene podatke i vraća ih aplikaciji u JSON (*JavaScript Object Notation*) [10] formatu.




Slika 2. Principijelna šema komunikacije sistema.

## III. PORTAL IEEESTEC KONFERENCIJE

Portal IEEESTEC konferencije predstavlja PHP aplikaciju koja je razvijena sa ciljem da olakša aktivnosti vezane za konferenciju svim akterima – autorima slanje radova i praćenje recenzija, recenzentima ocenjivanje radova, uredniku organizaciju same konferencije, a posetiocima daje sasvim novo iskustvo i novi način praćenja radova preko VR mobilne aplikacije.

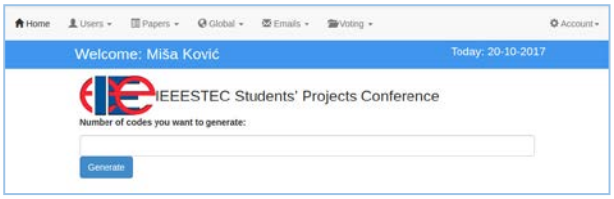
Trenutna verzija aplikacije se zasniva na već razvijenoj aplikaciji za organizovanje IEEESTEC konferencije 2013[7]. Nova platforma je obogaćena modulom za komunikaciju sa aplikacijom za VR, sistemom zasnovanom na API (*ApplicationProgrammableInterface*) tehnologiji [11], koji pruža određene servise.

Portalu se pristupa odlaskom na adresu *www.ieeeprijava.elfak.ni.ac.rs*, čime se pristupa stranici za prijavljivanje na sistem, odnosno login formi. Login forma ima zadatak da izvrši autentifikaciju i autorizaciju korisnika kako bi mu omogućila pristup stranicama i operacijama u skladu sa privilegijama koje odgovaraju toj grupi korisnika. Svi korisnici su podeljeni u tri grupe:

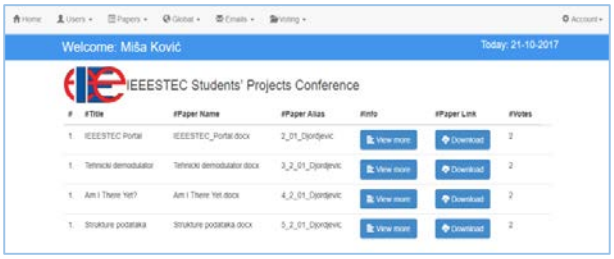
- urednici (ujedno imaju i administratorske privilegije),
- recenzenti (ocenjuju radove popunjavajući recenzentski formular)
- učesnici konferencije (šalju radove i prate recenzije za svoj rad).

Zbog projekta *VR Conference*, nova verzija portala IEEESTEC konferencije poseduje dodatne funkcionalnosti:

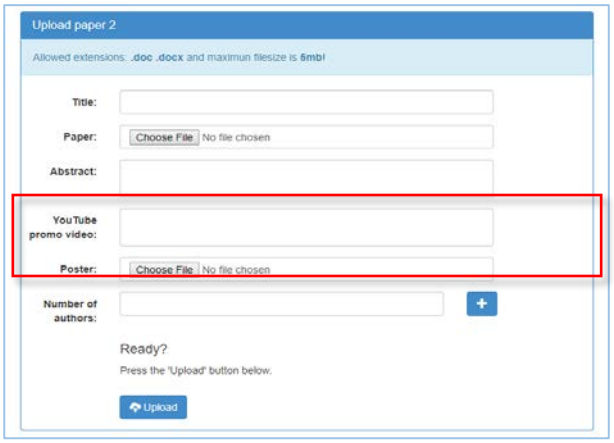
- **Generisanje QR kodova koji se koriste za glasanje** (*Generatecodesforvoting*) – glasanje za radove se autorizuje tako što korisnik koji ima pravo da glasa dobija QR kod koji mu to omogućava. Na ovoj stranici urednik može kreirati proizvoljan broj novih QR kodova za glasanje. Jednim QR kodom korisnik može glasati za dva rada (slika 3).
- **Pregled generisanih kodova** (*View all codes*) – na ovoj stranici su prikazani svi QR kodovi koji se koriste za glasanje, kao i prikaz da li su iskorišćeni ili ne.
- **Pregled rezultata glasanja** (*View voted papers*) - stranica koja služi za pregled glasova. Sadrži listu radova sortiranu po broju glasova u opadajućem redosledu, dok su prva tri mesta izdvojena na vrhu liste (slika 4).
- **Dodavanje promotivnog materijala** (*Upload paper*) – stranica na kojoj se vrši prijava rada za konferenciju je obogaćena poljima za dodavanje linka do YouTube promotivnog videa i postera rada (slika 5). Poster rada se u formatu slike (jpg, jpeg ili png) čuva na serveru, u folderu namenjenom za uploadovane fajlove. Taj folder sadrži podfoldere za radove i postere. Video materijal i poster se mogu pregledati kroz mobilnu aplikaciju opisanu u narednom poglavlju.




Slika 3. Forma za generisanje novih QR kodova.




Slika 4. Pregled radova sa brojem glasova.




Slika 5. Nova forma za dodavanje rada sa novim poljima za unos.

U pozadini generisanja QR kodova se nalaze metode *makeCodes* koja se koristi za generisanje novih kodova i *codeToImage* koja pretvara QR kod u sliku. Metoda prihvata jedan parametar pod imenom *codeNum* tipa integer i on neposredno određuje broj generisanih kodova. Unutar *for* petlje se poziva *helper* funkcija *CodeIgniter* frameworka koja se zove *random_string* i generiše određeni broj nasumičnih karaktera, odnosno nasumičan string. Kao prvi parametar je prosleđen string vrednosti "*alnum*" koji određuje da će generisani string biti alfanumeričkog tipa, a drugi parametar je integer vrednosti 12 i određuje da će generisani string biti dug 12 karaktera. Metoda vraća niz generisanih kodova koji se kasnije čuvaju u bazi.

```
public function makeCodes(int $codeNum) {
    $codes = [];
    $this->load->helper('string');
    for ($i = 0; $i < $codeNum; $i++) {
        $codes[] = ['code' => random_string('alnum', 12)];
    }
    return $codes;
}
```

Slika 6. Primer metode *makeCodes.*

Za pretvaranje koda u QR sliku se koristi metoda *codeToImage* klase *Votecode*. Ona prihvata jedan parametar tipa string koji predstavlja kod koji treba biti pretvoren u QR kod korišćenjem Google API-ja. Njemu se pristupa putem GET zahteva i svi potrebni parametri se prosleđuju kao GET vrednosti. Parametru *chs*, koji predstavlja veličinu tražene slike, je prosleđen string "300x300". Parametar *cht* predstavlja tip konverzije a prosleđena vrednost je "*qr*". Najbitniji parametar jeste *chl* i on sadrži vrednost koja se pretvara u QR kod. Za njega je prosleđena vrednost koja je dobijena kao parametar *codeToImage* metodi, a koju treba pretvoriti u QR kod. Metoda vraća URL kojim se pristupa API servisu.

```
public static function codeToImage($code) {
    $url = 'https://chart.googleapis.com/chart?';
    $get = [
        'chs' => '300x300',
        'cht' => 'qr',
        'choe' => 'UTF-8'
    ];
    $get['chl'] = $code;
    foreach ($get as $key => $value) {
        $appends = $key . '=' . $value . '&';
        $url .= $appends;
    }
    return $url;
}
```

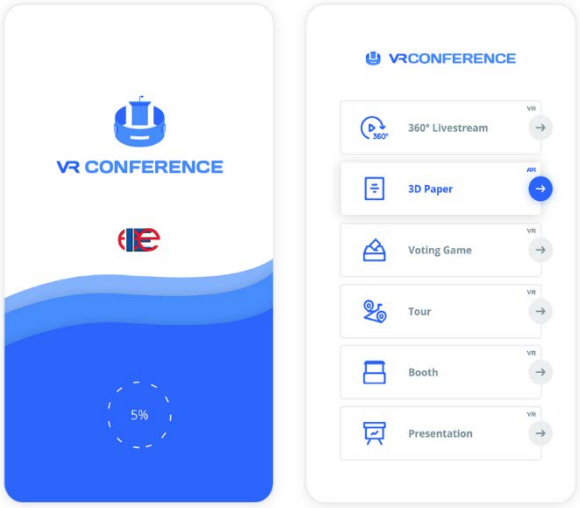
Slika 7. Primer metode *codeToImage.*

## IV. VR Conference mobilna aplikacija

Mobilna aplikacija je deo sistema kojim korisnici pristupaju celokupnom sadržaju konferencije. Ona se sastoji iz nekoliko modula koji su realizovani VR i AR tehnologijama. Svaki modul ima zadatak da na svoj način prikaže nešto od dostupnog sadržaja ili pruži korisniku neki vid interakcije sa serverom, kao što je to slučaj kod modula za glasanje. Kako bi aplikaciju učinili još zanimljivijom pripremljen je i celokupni vizuelni identitet predloženog rešenja (slike 8 i 9)



Slika 8. Izgled ikonice aplikacije VR Conference.




Slika 9. Izgled početnog ekrana i ekrana sa menijem.

Moduli VR Conference mobilne aplikacije koji se mogu birati na početnom ekranu (slika 9) su:

*A. VR 360 ?Livestream*
*B. AR 3D Paper*
*C. VR Voting Game*
*D. VR Tour*
*E. VR Booth*
*F. VR Presentation*

Neki delovi aplikacije su dostupni svim korisnicima, kao što je VR 360 ?Livestream, a za neke je potreban neki vid autorizacije, što je slučaj sa glasanjem za koje korisnik mora skenirati QR kako bi pristupio sistemu. Svi moduli su detaljno objašnjeni u nastavku.

### *A. VR 360 Livestream*

VR 360 Livestream je modul preko koga se može pratiti konferencija uživo preko interneta. Konferencija se prenosi kao video snimak u modu 360 koji se nalazi na Youtube kanalu VTŠ Apps Tima. Snimanje se vrši

korišćenjem Samsung Gear 360 kamere koja snima u realnom vremenu i u svim pravcima. Mobilni telefon deli snimak na dva dela ekrana za prikazna VR naočarima, što zapravo omogućava gledaocu da okretanjem oko svoje ose sam bira ono što želi da prati.

*B. AR 3D Paper*

Ovaj modul omogućava korisnicima da skeniranjem QR koda putem HTTP zahteva od API-a na serveru dobiju podatke o radu, preciznije poster i link do promotivnog videa. Nakon prikupljanja tih podataka, pojavljuju se opcije da korisnik otvori ili poster ili promotivni video (slika 10.) Autori rada pri prijavljivanju prilažu poster u formatu slike i link do promotivnog video klipa na YouTubu. Nije obavezno da autor priloži ove sadržaje te korisnik ima pristup samo onome što je priloženo. QR kod, koji generiše IEEESTEC portal aplikacija za svaki prihvaćeni rad, organizator konferencije može štampati u zborniku radova pored odgovarajućeg rada. Na taj način zbornik dobija novu dimenziju korišćenjem AR tehnologije. Prikaz sa opcijama razvijen je u Unity okruženju.

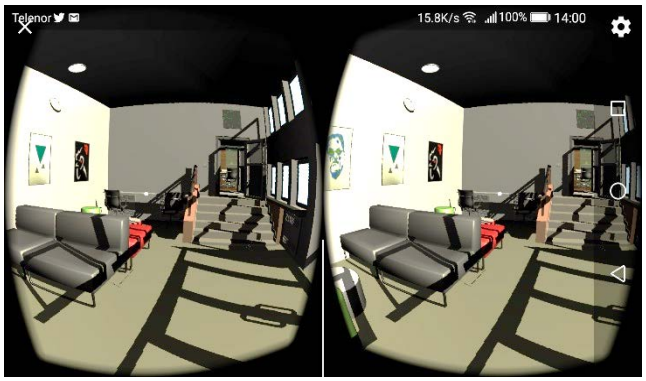



Slika 10. Izgled početnog ekrana i ekrana sa menijem.

*C. VR Voting Game*

VR Voting Game je modul koji korisnicima omogućava glasanje za radove koji učestvuju na konferenciji na atraktivan način u virtuelnom prostoru. Pošto glasanje nije dostupno svima već samo učesnicima na konferenciji, neophodno je da se korisnik autorizuje pre glasanja. Autorizacija se vrši pomoću QR kodova koji se dele svakom učesniku po jedan. Pomoću jednog QR koda korisnik može da izabere dva rada za koje želi da glasa. Sistem je modularno napravljen tako da administrator može promeniti broj radova za koje korisnik može glasati na osnovu jednog QR koda. Autorizacija funkcioniše tako što aplikacija od skeniranog QR koda dobija tekstualni kod, pomoću HTTP-a sa serverom proverava da li je taj kod validan i na osnovu odgovora dozvoljava korisniku ili ne dozvoljava da glasa.

Interaktivnost glasanja se ogleda u tome što korisnik mora da odigra igricu kako bi glasao. Igrica se, kao i samo glasanje, igra u virtualnoj realnosti. Igrač, preciznije virtualni avatar igrača, se nalazi u virtualnoj prostoriji (slika 11). Slika je na displeju telefona podeljena na dva dela (za levo i desno oko) a VR naočare spajanjem ove dve slike stvaraju dubinu virtuelnog prostora. Igrač prostoriju vidi iz prvog lica, a kada pogleda nadole može da vidi i svoj avatar, a može i da se pomera pomoću kontrolera. U prostoriji se nalazi pet skrivenih objekata i cilj je da ih igrač pronađe i prenese na odgovarajuće mesto. Igraču se meri vreme za koje obavlja ovaj zadatak i to vreme se pamti u bazi.




Slika 11. Izgled VR prostora za igru podeljenog na dva dela za prikazivanje na VR naočarima.

Po završetku igre, igrač, tj. njegov avatar, je prebačen u drugu virtualnu prostoriju sa glasačkom kutijom (slika 12). Glasanje se vrši tako što igrač ima pristup izlistanim radovima i može da izabere dva rada koja želi da ubaci u kutiju. Time se njegovi glasovi čuvaju u bazi podataka i završava igra, tj. glasanje.

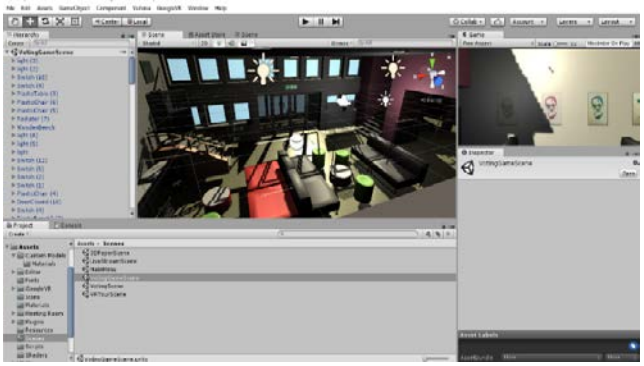


Slika 12. Izgled VR prostora za glasanje.

Sve VR funkcionalnosti su razvijene korišćenjem Unity više-platformskog razvojnog i izvršnog okruženja koje omogućava razvoj 2D i 3D aplikacija (slika 13). Programski jezik koji se koristi prilikom razvoja aplikacije je C# a može se koristiti i JavaScript. Pojam više-platformsko okruženje predstavlja mogućnost pisanja

jednog koda i eksportovanja aplikacije za željenu platformu (Windows, Android, iOS, Windows phone, Linux, Google Cardboard, Samsung Gear VR i dr.).

Sa novijim verzijama Unity3D razvojnog okruženja ostvaruje se mogućnost kreiranja aplikacija za virtualnu relanost. Da bi se započeo razvoj aplikacija sa podrškom virtualne realnosti potrebno je učitati VR dodatak okruženju (eng. *SDK*). Najčešće korišćeni dodaci prilikom razvoja VR aplikacija su: Google VR SDK [12] kao i Oculus VR SDK [13]. Od ovog dodatka zavisi sa kojim VR sistemom će aplikacija biti kompatibilna. Unity se sastoji od scena, objekata i skript fajlova. Scena predstavlja stanje aplikacije, objekat je 2D ili 3D model a skripta predstavlja funkciju (akciju) koja može da se implementira na modelu ili sceni.



Slika 13. Izgled Unity radnog okruženja.

Isečak koda koji je prikazan u nastavku (slika 14) predstavlja deo funkcije koja omogućava korisniku da prenosi predmete u igri. Konkretno predstavljen je uslov koji proverava da li je *back* dugme na telefonu pritisnuto, ako jeste selektovanom predmetu se transformiše pozicija u prostoru da bi korisniku koji igra igru stvorilo doživljaj da taj predmet nosi. Pored transformacije pozicije menja se i stanje predmeta u *grabbed*, odnosno u stanje nošenja.

```
void Update()
{
    RaycastHit hit;
    if (Physics.Raycast(transform.position, transform.forward, out hit))
    {
        if (hitting)
        {
            hitObject.SendMessage("OnMitExit");
            hitting = false;
            hitObject = null;
        }

        //DESNI KLIK
        if (Input.GetKeyDown(KeyCode.Escape) || Input.GetMouseButtonDown(1))
        {
            if (hit.transform.tag == "GrabObject" && grabed == false)
            {
                grabedObject = hit.collider.gameObject;
                hit.collider.GetComponent<Rigidbody>().isKinematic = true;
                hit.collider.enabled = false;

                grabedObject.transform.parent = transform;
                grabedObject.transform.position = new Vector3(
                    gameObject.transform.position.x,
                    gameObject.transform.position.y - 0.2f,
                    gameObject.transform.position.z
                ) + gameObject.transform.forward * distance;
                grabed = true;
            }
            else if (grabed)
            {
                grabedObject.transform.parent = null;
                grabedObject.GetComponent<Rigidbody>().isKinematic = false;
                grabedObject.GetComponent<BoxCollider>().enabled = true;
```

Slika 14. Deo VR koda za kreiranje funkcionalnosti prenošenja objekata.

*D. VR Tour*

VR Tour je virtualna tura Elektronskog fakulteta u Nišu, na kome se održava IEEESTEC konferencija. Cilj ovog modula jeste da posetiocima koji do sada nisu bili na fakultetu prošetaju korišćenjem VR tehnologije kroz fakultet i upoznaju se sa ključnim tačkama na kojima se održava konferencija. Ovaj sistem funkcioniše tako što se specijalnom kamerom, istom koja se koristi za VR 360 ?Livestream modul, naprave fotografije u modu 360 ?čitave ture pa se posle uz pomoć Unity okruženja spoje i napravi sistem za kretanje kroz prostor.

*E. VR Booth*

U ovom modulu korisnik u virtualnoj realnosti pred sobom ima prazan štand i paletu elemenata koje može koristiti za konfiguraciju svog štanda. Kako bi dobio pravi utisak izgleda štanda, korisnik ima mogućnost da u aplikaciju učita svoj poster, model svog projekta i promotivni video klip. Korisnik može da odluči na koje će mesto na svom VR štandu smestiti ove elemente, a učitani video klip se reprodukuje na virtualnom televizoru, koga korisnik takođe može proizvoljno da pozicionira.



Slika 15.VR štand VTŠ Apps Teama.

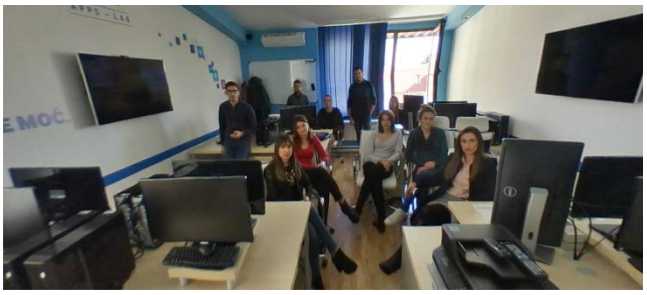
*F. VR Presentation*

VR Presentation je modul koji korisniku omogućava da učita PowerPoint prezentaciju u mobilnu aplikaciju (slika 16) i u virtualnoj realnosti se nađe u prostoriji u kojoj se radovi prezentuju na konferenciji.




Slika 16. Prezentacija u VR Presentation modulu.

VR prostorija u kojoj se nalazi korisnik je snimljena Samsung Gear 360 kamerom u modu 360° i pruža korisniku realan osećaj koji bi imao dok prezentuje. Na mestu gde je platno na koje se projektuje slika je učitana prezentacija koju je korisnik odredio, a u prostoriji se takođe nalazi simulirana publika (slika 17).



Slika 17. Publika u VR Presentation modulu.

## V. ZAKLJUČAK

Sa vrtoglavim razvojem tehnologija i distribucijom podataka koju taj razvoj nosi sa sobom, pojavila se potreba za novim načinima prezentovanja sadržaja. VR tehnologija pruža mogućnost sasvim drugačijeg pristupa prikazu sadržaja, a u saradnji sa serverskim tehnologijama (PHP i MySQL) imamo mogućnost i da taj sadržaj bude dinamičan.

Jasno je da je ovakav sistem vrlo koristan u industriji zabave, ali je cilj autora ovog rada bio da se VR iskoristi u obrazovne svrhe i time učešće na konferenciji učinimo jednostavnijim. S obzirom da se radi o 10. jubilarnoj konferenciji cilj autora je takođe bio da na ovaj način predloži korišćenje novih tehnologija na budućim konferencijama u cilju povećanja atraktivnosti same konferencije.

Koristeći ovaj sistem ne samo da prijavljivanje na konferenciju i ocenjivanje radova postaje lakše, već i glasanje postaje interesantno i interaktivno.

## ZAHVALNICA

Aplikaciju su realizovali članovi VTŠ Apps Tima u okviru Samsung Apps Laboratorije na Visokoj tehničkoj školi strukovnih studija u Nišu. Autori rada se zahvaljuju svim članovima tima koji su aktivno učestvovali na realizaciji projekta opisanog u ovom radu. Posebnu zahvalnost autori duguju beogradskoj kancelariji kompanije Samsung (*Samsung Electronics Adriatic, Belgrade Branch*) na podršci u radu od 2011 godine do danas.

## LITERATURA

[1] Woodrow Barfield, Fundamentals of Wearable Computers and Augmented Reality (2nd Ed), CRC Press, jun 2015

[2] Craig, Alan B.; Sherman, William R.; Will, Jeffrey D. (2009). Developing Virtual Reality Applications: Foundations of Effective Design. str. 1—2,281—292,336.

[3] Nicolas Nova, Laurent Bolli, Joypads!: The Design of Game Controllers, CreateSpace Free Publishing Platform, 2014.

[4] Samsung Gear VR zvanični sajt, http://www.samsung.com/global/galaxy/gear-vr/, septembar 2017.

[5] Joseph Hocking, Unity in Action
Multiplatform game development in C# with Unity 5, Jun 2015.

[6] Ed Burnette, Hello, Android (4th edition) Introducing Google's Mobile Development Platform, Pragmatic Bookshelf, 2015.

[7] Aaron Hillegass, Christian Keur, iOS Programming: The Big Nerd Ranch Guide (4th Edition), Big Nerd Ranch Guides, 2014

[8] StefanCvetković, Miloš Milojević, Slavimir Stošović "PHP Web portal IEEESTEC konferencije," Zbornik radova 6 IEEESTEC Konferencija, Niš, Serbia, Nov. 2013., pp. 193-196

[9] Brian Totty i David Gourley, HTTP: The Definitive Guide, O'Reilly, Jun 2009.

[10] JSON zvanični sajt, http://www.json.org/, septembar 2017.

[11] Leonard Richardson, RESTful Web APIs: Services for a Changing World, O'Reilly, 2013.

[12] Google VR SDK zvanični sajt za developere, https://developers.google.com/vr/unity/, septembar 2017.

[13] Oculus VR SDK, zvanični sajt za developere, https://developer.oculus.com/mobile, septembar 2017.

# All Sky Cloudiness Estimator

Vladimir Nikolić, Leander Schröder, Aleksa Tešić

*Abstract* We have developed a software package that can detect clouds in all sky camera images and give estimate of how cloudy a particular area of the sky is. In practice, it can be used with a set of images to get an overview of cloudiness for the whole night(s). We use this software with an all sky camera that works in conjunction with a narrow field camera to give us relatively accurate cloudiness information for its field of view..

## I. Introduction

The project started as a learning project during Petnica Summer Institute: Machine Learning 2017 seminar. In an attempt to make something useful from the learning process, we contacted the Petnica Meteor Group to inquire about some of the problems they are having that could be solved with machine learning. The problem we were given was this.

They use two cameras for meteor observation - a narrow field video camera that is triggered by meteors, and an all sky camera that they use to estimate cloudiness for the video camera. The latter camera is necessary, as the former will not make any recordings during cloudy periods. This means that if they get no recordings at all, they do not know whether that is because it is cloudy or because the shower is not active. Before this project was started, whoever was in charge of the observation had to manually go through the all sky images and determine cloudiness. This, of course takes time, and can be automated.

## II. Existing Solutions

Before starting the project, we examined the existing solutions. There are, in fact, plenty that detect clouds in images, both day and night, however, each one of them had some problems:

- None of the solutions take into account clouds that are not obvious, such as those where you cannot see them, because the sky is just black. These clouds are high and can be detected by the lack of stars in that area of the sky.
- Most of them were fairly simple procedural solutions. They are prone to inaccuracies if the conditions change, such as lighting. A machine learning approach is able to take into account various situations and handle them properly, if given proper data.
- Less importantly, they did not estimate cloudiness for a given field of view, which is what we needed.

Vladimir Nikolić, Leander Schröder and Aleksa Tešić are with the Petnica Science Center, Petnica 14104 Valjevo, Serbia, email: vladimir.nik94@gmail.com

## III. Our Solution

To solve the problem, We attempted a machine learning approach. The idea is to create a neural network that takes a night sky image as its input and have it output a cloud map. In this context, a cloud map would be a black and white image, where pixel intensity represents likelihood of that pixel belonging to a cloud. So, for example, a white pixel is definitely a cloud, a gray pixel is probably part of a cloud, and a black one is not. The output is intuitive, as white areas represent actual clouds, as seen on (Fig. 1)

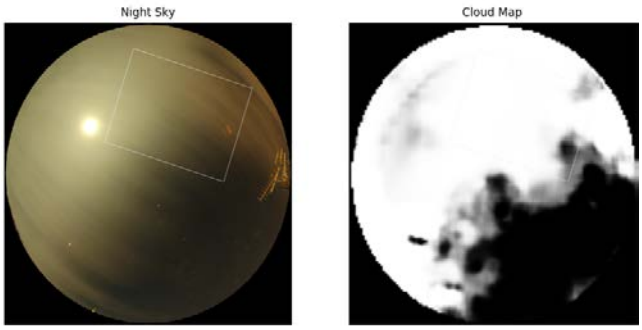



Fig. 1. Example of the neural network output, showing where the clouds are for the given input image.

### *A. Neural Network*

We used a Convolutional Neural Network (CNN) [1] which are standardly used when processing images. As a matter of fact, they process matrices (images in this case), so they can be applied to any problem where locality is relevant in given matrices. If the reader is not familiar with artificial neural networks, it is only important to know that they are very flexible and can be trained to output any kind of data as long as they are provided data to learn from. In this instance, we want to output a cloud image from a given sky image, so we need to give sky images and corresponding cloud images that we have labeled so that the network can learn to do the task by itself. (Fig. 2) shows an example of a CNN architecture, but they will not be explained in more detail as they are fairly complex and too long for this paper.

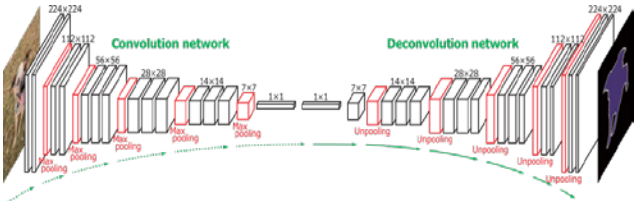



Fig. 2. Example of a Convolutional Neural Network.

*B. Data*

The neural network needs data to learn from and that is, in our case, all sky images and their labels. As we have found no existing solution that solves this problem properly, we also could not find a fitting dataset, so we did the labeling ourselves. This process is slow, so only around 200 images have been labeled so far. An example of an image and its label can be seen on (Fig. 3). To mitigate this problem, we have synthesized data from our dataset. Data synthesis, in this context, means slightly altering existing data in such a way that we get new useful data. Concretely, because of circular nature of all sky images, we are able to rotate the images and their labels several times, where each rotation would be a new data point, as seen on (Fig. 4).

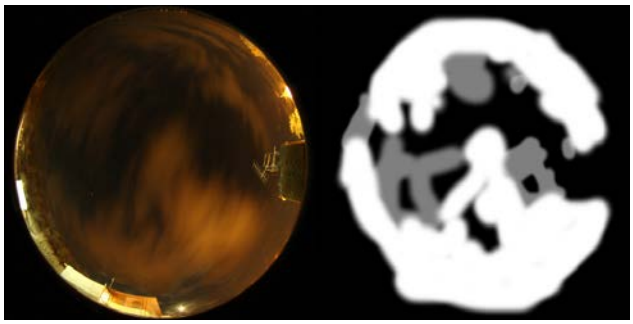


Fig. 3. Example image and its label.

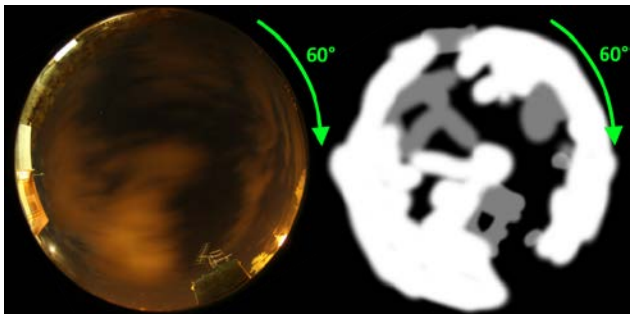



Fig. 4. Example of a rotation.

This method multiplies our data by the number of rotations we make to each image, but it is not perfect. The data we get is not qualitatively too different, so this only improves our network to some extent. Despite this, we have gotten fairly good results.

*C. Field of View*

The end goal is to get cloudiness percentage for a narrow field camera, so from a cloud map output, we need to extract area of interest. The narrow field camera has the following parameters for its field of view: azimuth and elevation of the center, width of view, aspect ratio, and rotation from the horizon.

With these parameters, the program must calculate the vertices of the rectangle that forms the field of view. In order to do that, we needed a method to convert coordinates between two systems - horizontal and cartesian. This means we need to be able to convert azimuth and altitude to pixel coordinates and vice versa.\\

Azimuth is easy to calculate, as the all sky images are circular and azimuth is just angular offset on the image center. Altitude, however, is not as trivial, as all sky images are distorted - the altitude does not decrease linearly as you move away from the zenith. To get around this, we took several cloudless all sky images and catalogued the stars we found (Fig. 5). Because we identified the stars, we knew their horizontal coordinates and we also knew their pixel coordinates, since we found them on actual images. With this catalogue, we could approximate mapping between horizontal and cartesian system. Altitude, in the context of an image, is distance from the center of the image, as that is where the zenith is. Given an altitude, we can determine how far it is from the center by finding two stars in the catalogue where our given altitude lies between the altitudes of these two stars and linearly interpolate the distance.

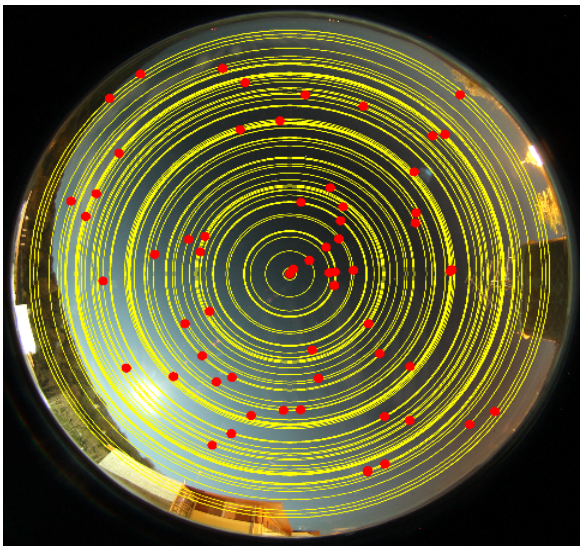


Fig. 5. The dots are the stars that we have catalogued and the concentric circles represent distances from the center with equal altitude and each one passes through a star of that altitude. The areas in between are linearly interpolated.

## IV. System in Practice

The whole pipeline works like this: The user inputs all the necessary parameters, including field of view, date and time interval of interest, and night sky images. The user is then, optionally, presented with each individual sky image, its cloud map and cloudiness percentage for correctness check. After this finishes, a table with date/time and cloudiness columns is saved to a file as the end result.

(Fig. 6) and (Fig. 7) show the optional correctness check phase which displays the relevant parts of the whole process. The code for the project can be found online [2].

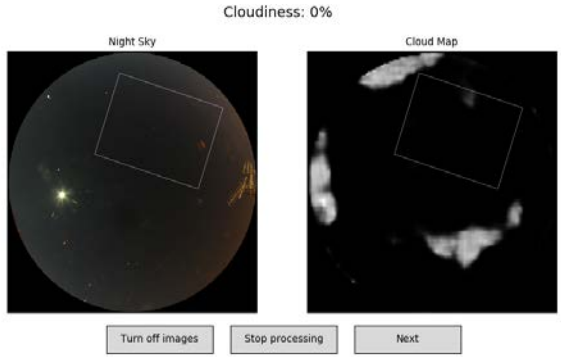



Fig. 6. System in practice example 1.

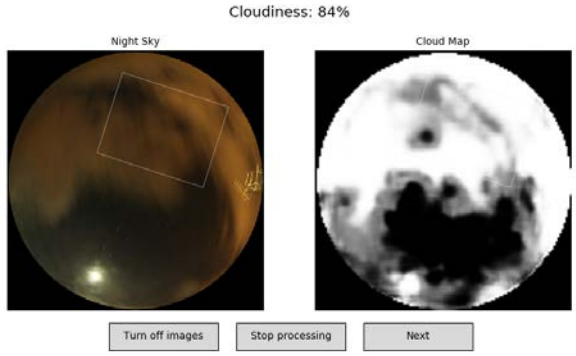



Fig. 7. System in practice example 2.

## V. Future Work

The main point that needs to be addressed is the variability (or lack of thereof) of accepted input images. Right now, it only works well with all sky night images taken from Petnica camera with its current settings. This is because the dataset the network was trained on only has these kinds of images. To improve the model, we need to label images taken with different cameras and settings such as brightness, sensitivity, exposure, etc. It is also possible to label daytime images to generalize cloud detection. All in all, the model improves with every new labeled image that the network can learn from.

Another possible improvement is a different neural network architecture. The current one is not too complex, but works well on everyday computers. If a more accurate estimation is needed, a more complex architecture could be made, although, to anyone familiar with machine learning, a more complex network does not necessarily perform better due to overfitting.

## Acknowledgements

We would like to thank Branislav Savić from Petnica Meteor Group. He has provided us advice and answered all the questions related to the astronomy part of this project.

## References

[1] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner: Gradient-Based Learning Applied to Document Recognition, Proceedings of the IEEE, November 1998, 86(11):2278-2324
[2] https://github.com/schutzekatze/allsky-cloudiness-estimator

# Sistem za udaljeni nadzor pacijenata pomoću pametnog sata

Darko Kostić, Zoran Kuštrimović, Nikola Jovanović,
Danilo Cvetković, Miloš Kosanović

*Sadržaj* - U ovom radu predloženo je rešenje za udaljenu kontrolu zdravlja pacijenata i starijih osoba. Ideja je da se doktoru omogući udaljeni uvid u statistiku aktivnosti pacijenta, njegovog pulsa, kretanja, kao i mogućnost da u bilo kom trenutku dobije podatke izmerene pomoću pametnog sata. Implementirana je i mogućnost slanja obaveštenja i saveta pacijentu od strane medicinskog osoblja. Zdravlje i aktivnosti pacijenta se mere pomoću Samsung Gear S3 sata dok za statistiku, notifikacije i pokretanje merenja doktor može koristiti Android ili veb aplikaciju. Na taj način se sa udaljenog mesta vrši kontrola zdravstvenog stanja pacijenta. Komunikacija između aplikacije na satu i Android aplikacije se odvija u realnom vremenu, dok se podaci čuvaju u bazi na serveru.

## I. UVOD

Sveprisutno računarstvo je koncept softverskog inženjerstva i računarstva, u kome se obrada informacija može desiti bilo kada i bilo gde. Termin je prvi put upotrebio Mark Weiser još 1988, a objasnio ga u radu [1].

Za razliku od desktop računarstva, sveprisutno računarstvo se može desiti na bilo kom uređaju, na bilo kojoj lokaciji i u bilo kom formatu. Interakcija sa korisnikom se može obaviti preko laptopa, tableta ili mobilnog telefona, ali i preko bilo kog uređaja koji čovek dnevno koristi kao što je sat, frižider ili veš mašina. Tehnologije koje omogućavaju sveprisutno računarstvo su internet, napredni softver srednjeg sloja (*middleware*), različiti senzori, RFID, mikroprocesori, mobilni komunikacioni protokoli, distribuirani sistemi, novi materijali i druge. Ovakvi sistemi su deo okruženja koje detektuje prisustvo ljudi i interaguje sa njima. Svi uređaji su povezani, inteligentni i pomažu čoveku u obavljanju njegovih svakodnevnih aktivnosti. Za ovakve sisteme obično kažemo da su:

- Integrisani (*embedded*): mnogo uređaja međusobno komunicira i integrisano je u jedinstven sistem,
- Svesni konteksta (*context aware*): uređaji mogu da prepoznaju kontekst u kome se korisnik nalazi,
- Personalizovani - (*personalized*): prilagođeni su svakom korisniku posebno.
- Prilagodljivi (*adaptive*): mogu se prilagoditi korisniku metodama mašinskog učenja,
- Prediktivni (*anticipatory*): mogu predvideti ponašanje korisnika.

Nikola Jovanović, Darko Kostić, Danilo Cvetković i Zoran Kuštrimović su studenti na smeru Savremene Računarske Tehnologije i članovi VTŠ Apps Tima.

Miloš Kosanović je koordinator VTŠ Apps Tima Visoke Tehničke Škole Strukovnih Studija u Nišu, Aleksandra Medvedeva 20, e-mail: milos.kosanovic@vtsnis.edu.rs

Razvoj ovakvih sistema značajno je olakšan u poslednje vreme usled činjenice da je cena hardverskih uređaja značajno smanjena. Razvojem pametnih nosivih uređaja kao što su narukvice, satovi, majice, obuća moguće je nesmetano snimati stanje korisnika bez većeg ometanja njegovih svakodnevnih aktivnosti. Očekivano je da ćemo u budućnosti svi nositi po nekoliko pametnih uređaja na sebi. Svi ovi uređaji će snimati podatke, komunicirati međusobno i slati podatke u centralizovani sistem koji će biti zadužen za nadgledanje našeg zdravlja kao što je opisanu u [2 - 4].

Sistem opisan u ovom radu služi za udaljeni nadzor pacijenata i nazvan je zdravstveni pomoćnik (*Health Assistant*). Kao hardverski uređaj koji pomoću različitih senzora prati stanje pacijenta korišćen je pametni sat Samsung Gear S3. Sat snima različite podatke i šalje ih telefonu na dalju obradu kako bi se utvrdilo zdravstveno stanje osobe. Sam sat prepoznaje određene aktivnosti, kao i promene stanja osobe.

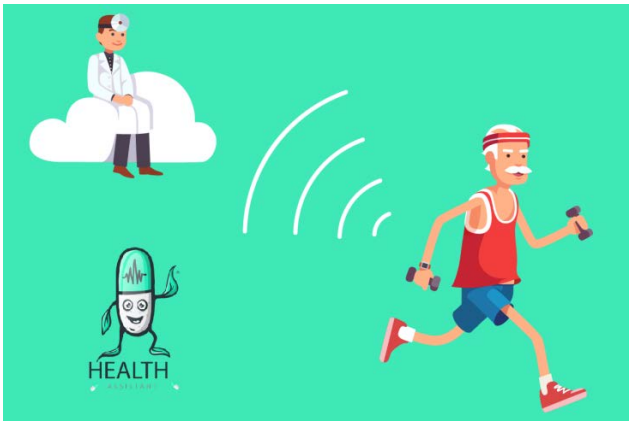
Posebna pažnja u radu posvećena je komunikaciji i prenosu i prikupljanju podataka sa senzora na satu do telefona, kao i udaljenoj kontroli senzora na satu pomoću telefona ili veb aplikacije. Dalja obrada ovih podataka nije detaljno razmatrana i biće predmet nekog drugog rada u budućnosti.

Sistemi koji koriste samo mobilni telefon bez pametnog sata su opisani u [3] [4]. Mobilni telefon i pametni sat korišćeni su za prepoznavanje aktivnosti, ali ne na Tizen platformi [2]. Pametni sat kao pomoćno sredstvo za reanimaciju korišćeno je u [5]. Koliko je autorima poznato, sistem opisan u ovom radu je prvi sistem rađen za Tizen platformu, prvi sistem za udaljenu kontrolu zdravlja u Srbiji i sa fokusom na zdravstvenu negu starijih osoba.

## II. SVEPRISUTNO ZDRAVSTVO

Sveprisutno zdravstvo (*pervasive health*) se može definisati kao zdravstvena usluga većeg kvaliteta pružena bilo kome, bilo kada, i bilo gde. Demografske promene stanovništva, sociološke promene, povećanje troškova zdravstvenog sistema i povećanje populacije starijih ljudi u razvijenim zemljama je dovelo do značajnih problema u organizovanju zdravstvenog sistema. Zbog toga je došlo do

potrebe za promenom strategija i efikasnijom organizacijom pružanja zdravstvenih usluga, a sveprisutno zdravstvo i personalizovano zdravstvo čine značajne iskorake u tom pravcu. To podrazumeva individualno praćenje zdravstvenog stanja osobe kako bi se predvidela individualna sklonost ka nekom oboljenju, bazirana na genetskim ili drugim faktorima, i omogućilo preventivno delovanje i na taj način pružila kvalitetnija i efikasnija medicinska usluga.




Slika 1. Izgled modernog zdravstva.

Transformacija standardnog zdravstvenog sistema zahteva znatna novčana ulaganja, edukaciju ljudi, promenu navika i celokupnog pogleda na zdravstveni sistem i sve njegove prateće procese i procedure. Reorganizacija celokupnog poslovanja zdravstvenih ustanova je neophodna i nije čudno što mnoge velike kompanije tu vide svoju šansu.

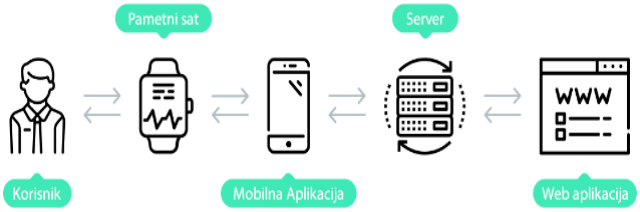
Mogućnosti primene kontekstno svesnih sistema i sistema za prepoznavanje aktivnosti su verovatno najveće u domenu zdravstvene nege, gde se javlja sve veća potreba za sistemima koji detektuju ljudske aktivnosti, naročito osoba starijih od 60 godina. Po izveštaju Svetske zdravstvene organizacije broj ljudi ove starosne dobi se značajno povećao u prošloj deceniji. Ljudi starijih godina imaju specifične potrebe kada se radi o njihovoj zdravstvenoj nezi kao što je opisano u [6].

Značajni procenat ovih ljudi može imati neki oblik demencije, Parkinsonove bolesti, dijabetesa, kardio-vaskularnih bolesti, kao i neke druge hronične bolesti kao što je artritis ili mogu imati smetnje u fizičkim aktivnostima. Tehnologije koje se primenjuju u sveprisutnom računarstvu mogu značajno pomoći da se ovim ljudima poboljša kvalitet života, smanji vreme koje je neophodno da neko provede sa njima, odnosno smanji cenu te brige generalno. To znači da ove tehnologije nisu namenjene samo pacijentima, već i njihovim porodicama, zdravstvenim radnicima kao i celokupnom društvu.

## III. ARHITEKTURA SISTEMA

Tipična arhitektura sistema za prepoznavanje ljudskih aktivnosti generalno sadrži tri različite faze. Prva faza obuhvata pribavljanje sirovih podataka sa senzora, druga pamćenje i obradu tih podataka kako bi se dobile informacije o kontekstu i treća koja predstavlja primenu metoda za učenje ili zaključivanje kako bi se odredile proste ili složene aktivnosti i pružile korisne informacije. U ovom radu je opisana arhitektura jednog takvog sistema koja pre svega treba da služi za pribavljanje sirovih podataka sa pametnog sata i njihov prikaz na mobilnom telefonu i veb čitaču.

Princip funkcionisanja celokupnog sistema je grafički prikazan na Slici 2.




Slika 2. Koncept funkcionisanja Sistema.

Svi korisnici sistema, moraju biti registrovani u bazi. Prilikom kreiranja naloga, PHP aplikacija proverava da li je uneto korisničko ime za novo-kreiranog korisnika slobodno. Istovremeno se radi i validacija forme u pogledu provere obaveznih polja i da li su podaci uneti u pravilnom formatu. Ukoliko kreirani nalog zadovoljava sve uslove i sigurnosne provere, PHP aplikacija skladišti nalog sa detaljnim informacijama o korisniku u MySQL bazu.

Zahtev za merenje ili prikaz podataka se može poslati sa veb ili mobilne aplikacije. Podaci se nakon zahteva koji mobilni uređaj poslao vraćaju sa pametnog sata na mobilni uređaj koji te podatke prosleđuje na server. Podaci se dalje obrađuju i čuvaju na serveru, a kasnije mogu koristiti za prikaz i izračunavanje dnevnih, nedeljnih i mesečnih statistika. Sat je povezan sa telefonom *Bluetooth* vezom. Pošto sat za konstantnu komunikaciju sa telefonom već koristi tu vezu, neće doći do prekomerne potrošnje baterije.

Senzori na pametnom satu koje smo koristili za određivanje zdravstvenog stanja osobe su: senzor za merenje pulsa, akcelerometar, žiroskop, magnetometer i senzor osvetljenja.

Ljudske aktivnosti koje pametni sat sam prepoznaje i meri obuhvataju mirovanje, hodanje i trčanje. Za prepoznavanje ostalih aktivnosti neophodno je primeniti metode mašinskog učenja što prevazilazi obim ovog rada, ali je svakako tema kojom će se autori baviti u budućnosti.

Pomoću navedenih senzora i prepoznatih aktivnosti prikazuje se statistika za određeni vremenski period. Statistika obuhvata koliko je dugo osoba provela u

mirovanju, koliko se dugo kretala, i koliko je vremena provela trčeći.

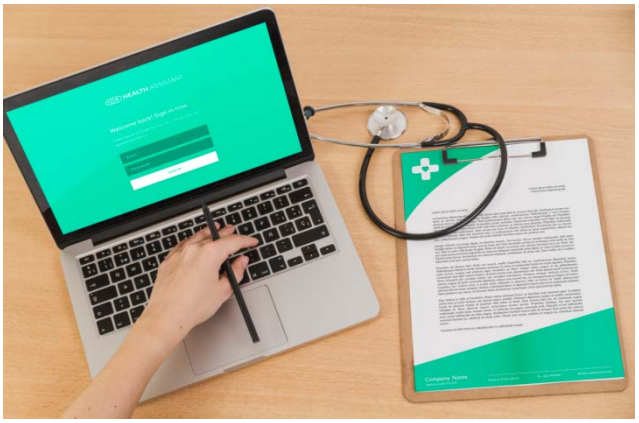
Sistem će takođe meriti i socijalnu aktivnost korisnika tako što će računati broj poslatih poruka, minute razgovora, kao i broj različitih ljudi sa kojima se korisnik dopisivao.

## IV. REALIZACIJA VEB APLIKACIJE

Klijentska strana aplikacije implementirana je pomoću HTML i CSS jezika. Sva funkcionalnost aplikacije kreirana je pomoću JavaScript jezika i jQuery biblioteke. JQuery olakšava i skraćuje pisanje koda, tako da se uz samo par linija može implementirati bilo koja potrebna funkcionalnost veb aplikacije. Za generisanje strukture se koristi i Bootstrap, open-source JavaScript framework, razvijen sa ciljem da omogući i olakša razvoj veb formi kao i razvoj naprednih veb komponenti.

Serverska aplikacija je bazirana na PHP tehnologijama. PHP je specijalizovan skript jezik prvenstveno namenjen za izradu dinamičkog veb sadržaja i izvodi se na strani servera. [7]. Ona omogućava doktoru da pošalje obaveštenje pacijentu.

Dodatna mogućnost veb aplikacije jeste preuzimanje određenih informacija koje su bitne za zdravlje pacijenta. Informacije su detaljnije opisane u poglavlju "Realizacija Tizen aplikacije". Doktor šalje zahtev za merenje svih ili samo određenih podataka, zatim dobija odgovor koji se nakon toga upisuje u MySQL bazu ili .csv fajl u zavisnosti od podataka koji su vraćeni. Doktor u bilo kom trenutku može da aktivira senzore na satu i izvrši njihovo očitavanje, kao i da pristupi prethodno izmerenim podacima koji mogu biti prikazani za određeni vremenski period (dnevna, nedeljna i mesečna statistika).



Slika 3. Koncept funkcionisanja sistema.

## V. REALIZACIJA TIZEN APLIKACIJE

Aplikacija za "Zdravstvenog pomoćnika" u ovom radu je razvijena za Samsungov model Gear S3, koji radi na Tizen operativnom sistemu. Tizen je otvoren i fleksibilan operativni sistem napravljen da omogući rad različitih uređaja (telefona, satova, igračkih konzola, televizora). Razvijen od strane zajednice programera, pod Open Source licencom i otvoren je za sve članove koji žele da učestvuju. Operativni sistem Tizen dolazi u više profila kako bi ispunio sve zahteve industrije. Sadašnji profili su Tizen Mobile, Tizen TV i Tizen za nosive (*wearable*) uređaje.

Za izradu aplikacije korišćeno je Tizen SDK razvojno okruženje [8] korišćenjem HTML5, CSS, JavaScript tehnologija. Struktura aplikacije, odnosno kontrola koje čine korisnički interfejs, kreira se korišćenjem HTML5 jezika. Nakon kreiranja strukture, raspored i dizajn HTML elemenata se podešava pomoću CSS-a. Sva funkcionalnost aplikacije kreira se pomoću JavaScript jezika.




Slika 4. Izgled aplikacije na pametnom satu.

Aplikacija na satu komunicira sa aplikacijom na telefonu korišćenjem SASocket biblioteke iz Tizen SDK razvojnog okruženja, koja omogućava uspostavljanje konekcije, slanje i primanje podataka [8]. U slučaju da mobilni telefon izgubi konekciju sa satom, sat ima mogućnost čuvanja podataka u svojoj lokalnoj bazi (*local storage*) kako bi poslao podatke kada se konekcija ponovo uspostavi.

```
localStorage.setItem("Hearth_rate",
  "hrmInfo.heartRate");
```

Kada je konekcija uspostavljena, startuje se osluškivač podataka koji se aktivira kada sa klijenta stigne zahtev za razmenu podataka. Nakon što stigne zahtev za merenje i slanje podataka, Tizen aplikacija pokreće senzore koji su zatraženi od strane Android aplikacije i koji vrše merenje podataka u realnom vremenu što je prikazano u nastavku.

```
tizen.humanactivitymonitor.start('HRM',
function(hrmInfo){ . . . });
```

Nakon završetka merenja, dobijeni podaci iz senzora šalju se Android aplikaciji na dalje korišćenje.

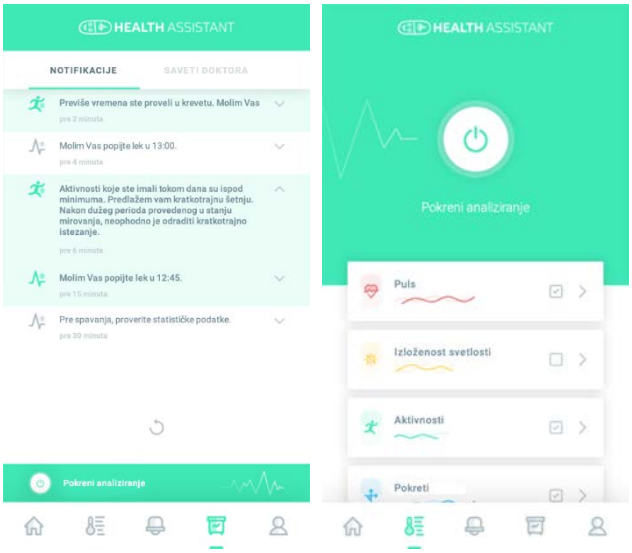
```
SASocket.sendData(SAAgent.channelids[0],  „HRM“  +
hrmInfo.hearthRate );
```

Tizen SDK sadrži Sensor API [9] koji nudi veliki broj interfejsa i metoda za upravljanje podacima iz senzora. Da bi podaci kao što su puls, ili aktivnost pacijenta, bili pročitani, neophodno je da aplikaciji budu dodeljene odgovarajuće privilegije (*permission*). Privilegija koja dozvoljava pristup podacima o zdravlju se zove *health info*. Tipovi senzora koji su takođe realizovani su: senzori za pokrete (žiroskop i akcelerometar), senzor za svetlost (nudi podatke o stepenu osvetljenja). Na satu se nakon svakog merenja prikazuju dobijeni rezultati.

## VI. Realizacija Android aplikacije

Android aplikacija omogućava praćenje podataka sa senzora sata. Aplikacija ima dva nivoa pristupa, za pacijenta i doktora. Pacijent direktno komunicira sa satom, dok aplikacija omogućava doktoru da prati statistiku o zdravstvenom stanju pacijenta, da pokrene zahtev za merenje, kao i da pošalje obaveštenja pacijentu. Korisnik može izabrati da pokrene sve ili samo određene senzore. Da bi korisnik koristio aplikaciju potrebno je da se prijavi na sistem kao pacijent ili doktor.

Ukoliko se korisnik prijavi kao pacijent, neophodno je da se aplikacija na satu poveže sa Android aplikacijom. Aplikacija šalje zahtev za konekciju aplikaciji na satu, i kada dobije nazad odgovor o uspešno ostvarenoj konekciji, korisnik ima mogućnost da izvrši merenje.




Slika 5. Izgled aplikacije na Android uređaju.

Kada korisnik odabere koje senzore želi da meri šalje se zahtev za pokretanje senzora aplikaciji na satu putem servera ukoliko je zahtev zatražio doktor, ili putem direktne konekcije sa satom ukoliko je zahtev zatražio pacijent.

```
mServiceConnection.send(getServiceChannelId(0),
requestDataMessage.toByteArray())
```

Zatim se aktivira osluškivač koji očekuje odgovor od aplikacije na satu sa podacima dobijenih sa senzora. Kada odgovor stigne, aplikacija podatke šalje na server i prikazuje rezultate merenja korisniku.

Za izradu Android aplikacije korišćeno je Android SDK [10] razvojno okruženje. Aplikacija u ovom projektu je razvijena pomoću Kotlin jezika [11]. Za komunikaciju sa satom korišćen je Samsung Accessory SDK [12] razvojno okruženje i Samsung Accessory protokol SAP.

Kotlin kao programski jezik za Android razvoj omogućava korišćenje biblioteke pisanih u Java programskom jeziku, što dalje omogućava korišćenje svih već pomenutih biblioteka. Aplikacija je izrađena tako da ima ulogu propratne (*companion*) aplikacije zajedno sa aplikacijom za pametni sat. Kažemo da je aplikacija koja se izvršava na satu zapravo servis (*provider*) aplikacija i ona omogućava korišćenje resursa sata, dok je Android aplikacija klijent (*consumer)*, i koristi servise aplikacije sata.

Da bi aplikacija na mobilnom telefonu i satu uspostavile vezu, potrebno je da ostvare konekciju pomoću *Accessory Service* profila koji je implementiran na obe strane. Za komunikaciju preko ovog profila izabrana je *Bluetooth* veza, a moguće je izabrati i *Wifi* vezu. Kada se aplikacija na satu pronađe, otvara se veza između klijent Android i servis aplikacije na satu uz pomoć SASocket biblioteke.

```
override fun onReceive(channelId: Int, data:
ByteArray?) {
   if(data != null) {
   val message = String(data)
   updateTextView(message)
   }}
```

Za HTTP pozive i razmenu podataka sa serverom korišćena je *Volley* [13] biblioteka.

```
val request =
 JsonObjectRequest(Request.Method.POST,
   Constants.LOGIN_API_URL, params,
   Response.Listener {res -> updateUserData()},
Response.ErrorListener { error ->toastMessage()})
que.add(request)
```

## VII. Zaključak

Sistem "zdravstveni pomoćnik" za udaljeni nadzor pacijenata omogućava udaljeni nadzor zdravstvenog stanja pacijenata i samim tim poboljšanje zdravstvenih usluga. Ciljna grupa sistema jeste populacija starijih ljudi, kao i zdravstvena nega osoba sa posebnim potrebama. U ovom radu opisana je implementacija celog sistema. Takođe, opisan je i način funkcionisanja delova sistema, ali i tehnologije koje su korišćene za njihov razvoj. Sistem omogućava medicinskom osoblju da pošalje poruku pacijentu, izvrši udaljeno merenje pulsa pacijenta, akcelerometarskih i drugih podataka, i prepozna njegovu trenutnu aktivnost (mirovanje, hodanje, trčanje).

Zdravstveni pomoćnik je sistem koji zahteva pristup privatnim podacima koji su osetljive prirode. Pristup može

predstavljati izazov i problem ukoliko nije pravno i etički regulisan. Dodatni problem predstavlja i cena pametnog sata, kao i opasnost od skidanja sata osoba sa demencijom. Navedeni problemi bi mogli biti prevaziđeni razvojem hardverskog uređaja koji bi bio jeftiniji i pogodniji za nošenje.

U budućnosti je planirano da se grupa senzora proširi i na one koji se ne nalaze na pametnom satu. Više pažnje će biti posvećeno obradi podataka i prepoznavanju dodatnih ljudskih aktivnosti primenom metoda mašinskog učenja, kao i prepoznavanje promene položaja tela.

## ZAHVALNICA

Aplikaciju su realizovali članovi VTŠ Apps Tima u okviru Samsung Apps Laboratorije na Visokoj tehničkoj školi strukovnih studija u Nišu. Autori rada se zahvaljuju svim članovima tima koji su aktivno učestvovali na realizaciji projekta opisanog u ovom radu. Autori se zahvaljuju i prof. dr Stojanović Draganu na korisnim sugestijama.

## LITERATURA

[1] Weiser, M., The computer for the 21st century, Acientific American 265(3): 94–104, 1991.

[2] Avgoustinos Filippoupolitis, Babak Takand, George Loukas, Activity Recognition in a Home Setting using Off the Shelf Smart Watch Technology, International Conference on Ubiquitous Computing and Communications, 2016

[3] Stefan Forsström, Victor Kardeby, Estimating Contextual Situations using Indicators from Smartphone Sensor Values, IEEE International Conference on Internet of Things (iThings 2014), 2014

[4] Nicholas D. Lane, Tanzeem Choudhury, Andrew Campbell, Mashfiqui Mohammod, Mu Lin, Xiaochao Yang, Afsaneh Doryab, Hong Lu, Shahid Ali and Ethan Berke, BeWell: A Smartphone Application to Monitor, Model and Promote Wellbeing, Pervasive Health 2011-- 5th International ICST Conference on Pervasive Computing Technologies for Healthcare, Dublin, 23-26 May 2011

[5] Agnes Gruenerbl, Gerald Pirkl, Eloise Monger, Mary Gobbi, Paul Lukowicz, Smart-Watch Life Saver: Smart-Watch Interactive-Feedback System for Improving Bystander CPR, ISWC '15, Osaka, Japan, 2015

[6] Qin Ni, Ana Belén García Hernando, Iván Pau de la Cruz, The Elderly's Independent Living in Smart Homes: A Characterization of Activities and Sensing Infrastructure Survey to Facilitate Services Development, Sensors, 2015

[7] Hugh E. Williams, David Lane, Veb aplikacije i baze podataka – PHP i MYSQL, Mikro knjiga 2003

[8] https://developer.tizen.org/tizen, posećeno 23.10.2017

[9] https://developer.tizen.org/development/apireferences/web-application?redirect=https://developer.tizen.org/dev-guide /3.0.0/org.tizen.web.apireference/html/device_api/wearable/ tizen/sensor.html, posećeno 23.10.2017

[10] Wei-Meng Lee, Beginning Android 4 Application Development, John Wiley and Sons, Inc. 2012

[11] https://kotlinlang.org/, posećeno 23.10.2017

[12] http://developer.samsung.com/galaxy/accessory, posećeno, 23.10.2017

[13] https://developer.android.com/training/volley/index.html, posećeno 23.10.2017.

# Erasmus+ student mobility between the University of Niš and the Technical University of Sofia

Nikola Popović, Aleksandra Đorić, Zornitsa Krasteva, Ivaylo Vladimirov

*Abstract* - This paper describes experiences of Serbian students who resided in Bulgaria at Technical University in Sofia (TUS) as well as Bulgarian students who stayed at the Faculty of Electronic Engineering at University in Niš within the ERASMUS + International Credit Mobility (ICM) student exchange program. Similarities and differences in the way of student's life and academic activities at these two universities are presented from the student's perspective of basic, master and doctoral academic studies.

## I. INTRODUCTION

The University of Niš was incorporated as an independent degree-granting institution on June 15, 1965 and consists of 13 faculties. Most of the faculties have a structure comprising departments, divisions or majors, offering study and research opportunities at the undergraduate and graduate levels, including opportunities to obtain Ph.D. degrees. The university main offices are in one of the most representative city structures (Fig. 1.), an edifice built in the late 19th century, which was formally recognized as a cultural treasure of major significance in 1979. The building was constructed in 1887 in the neo-Renaissance style, designed by an unknown Viennese architect.



Fig. 1. University of Niš, Niš, Serbia.

Nikola Popović and Aleksandra Đorić are with the Department of Telecommunications, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, e-mail: nikolap85@gmail.com, alexdjoric@yahoo.com

Zornitsa Krasteva and Ivaylo Vladimirov are with the Faculty of Telecommunications, Technical University of Sofia, St. Kliment Ohridski 8, 1756 Sofia, Bulgaria, e-mail:

The Technical University of Sofia (Fig. 2) is the largest educational and scientific complex in Bulgaria in the field of technical and applied science with an institutional accreditation grade of 9.5 (on the scale of 10) for the period 2012 - 2018. It is an independent institution since 1953, when the Polytechnic was divided into four separate technical institutes. It has had its present name and university status since 21 July 1995 and has 14 main faculties based in Sofia, Plovdiv and Sliven, as well as 3 additional ones with education only in foreign languages - German, English and French.

There is an interesting fact that only Sofia has around 15 different universities while Serbia has 8 state and 10 private universities in the whole country. It's a custom that each bigger city in Bulgaria has more than one state university which is not the case for the Serbian university cities.




Fig. 2. Technical University of Sofia, Sofia, Bulgaria.

This was the first year of successful collaboration between the University of Niš and the Technical University of Sofia through the ERASMUS+ International Credit Mobility (ICM) programme. Two students from the University of Niš on master and PhD courses were 5 months at the Faculty of Telecommunications in Sofia and two students on basic studies from TUS have spent 4 months at the Faculty of Electronic Engineering in Niš. Considering that Serbia and Bulgaria are neighbor countries and have similar culture, language and lifestyle, it was not difficult for the students to adapt to the new environment.

The Faculty of Electronic Engineering in Niš was established in 1960 and proudly celebrate half a century of existence and successful work. The Faculty of Telecommunications in Sofia exists from 1989 after a few splitting with other departments at the TU.

## II. City Guide

A. *Places you should visit when you're in Niš*

Niš is the third largest city of Serbia and as a modern university city at the same time is natural, social, economic, educational, health, cultural and sport centre of southeast Serbia. Due to its geographical position, since ancient times has been considered as the gateway between the East and West. It is also the birthplace of Constantine the Great, the first Christian Emperor and the founder of Constantinople.

As one of the oldest cities in the Balkans and Europe, Niš has a rich historical heritage. Some of the main tourists attractions are: *Niš Fortress* - built by the Turks in the first decades of the 18th century (1719–23); *Skull Tower* - a tower made out of skulls of Serbian uprisers, killed and decapitated by the Ottomans in period (1804–13); *Concentration camp* "*Crveni Krst*" - one of the few preserved Nazi concentration camps in Europe; *Mediana* - archeological site, an Imperial villa, from the late Roman period located on the road leading to Sofia, Bulgaria; *Čegar Hill* - the place where Battle on Čegar Hill took place on 19 May 1809; *Memorial Park Bubanj* - World War II memorial complex, etc.



Fig. 3. Niš Fortress, Niš, Serbia.

Niš Fortress in Fig. 3 is a complex and important cultural and historical monument. It rises on the right bank of the Nišava River, overlooking the area inhabited for longer than two millennia. The extant fortification is of Turkish origin, dating from the first decades of the 18th century. The Fortress was erected on the site of earlier fortifications – the ancient Roman, Byzantine, and later yet Medieval forts. In the period when it belonged to Turkish people, the fort gained its today's look.



Fig. 4. Bubanj Memorial Park, Niš, Serbia.

Memorial Park Bubanj (Fig. 4) is a World War II memorial complex built in 1963 to commemorate the shooting and execution of more than 10,000 citizens from Niš and other parts of the country.

*B. Places you should visit when you're in Sofia*

Sofia is the capital and largest city of Bulgaria with 1.68 million people residing in its metropolitan area. Being in the centre of the Balkan Peninsula, it is midway between the Black Sea and the Adriatic Sea, and closest to the Aegean Sea.

Sofia was founded 2,500 years ago and over the centuries it has been given several names - *Serdica*, *Sredetz* and the remains of the old cities can still be viewed today.

One of Europe's oldest cities, Sofia is where the Middle East and Asia meet Central and Western Europe. Red Army monuments, domed cathedrals and Ottoman mosques share the city with glitzy shopping malls, packed clubs and luxurious hotels. The attractive city is also home to a number of intriguing museums, theaters, art galleries and fine restaurants. Although the city is ancient, it has a dynamic young vibe that attracts visitors.

There are 47 churches within Sofia, and the *Saint Alexander Nevsky Cathedral* is one of the largest Eastern Orthodox cathedrals in the world, as well as one of Sofia's symbols and primary tourist attraction (Fig. 5). Built in Neo-Byzantine style in honor of the hundreds of thousands of Ukrainian, Bulgarian, Russian and Belorussian soldiers killed in the Russo-Turkish War, the church can hold over 10,000 worshippers in its 3,170 square meters. It is the second-largest cathedral located on the Balkan Peninsula, after the Cathedral of Saint Sava in Belgrade.



Fig. 5. The St. "Alexander Nevsky" Cathedral, Sofia, Bulgaria.

Other impressive religious institutions in the capital include *the Boyana church* - dating from the 10th-11th century, protected by UNESCO; *the Saint Sophia Basilica* - the second oldest church in Sofia, dating from the 4th-6th century; *the St. Nikolai Russian Church* - built in 1914; *the St. Nedelya Church* - dating from the 10the century; *the Sofia Synagogue* - the largest synagogue in Southeastern Europe, built in 1909; *the Banya Bashi Mosque* - Ottoman architecture, completed in 1566, etc.

The statue of St. Sofia in Fig. 6 is a symbol of the city because St. Sofia is the city's protector. It's a 24 meter high bronze statue that replaced the statue of Vladimir Ilich Lenin that was on the same spot. She stands on a place where the old culture meets the modern lifestyle. In her left hand she holds an owl that is a symbol of wisdom because Sofia means "wisdom" in Greek. In her right hand she holds a laurel wreath – a powerfull symbol for greeks and romans meaning peace, success, reputation etc.



Fig. 6. The Statue of Saint Sophia, Sofia, Bulgaria.

## III. STUDENT LIFE

During the Erasmus+ program students from foreign countries in Niš don't have the oppurtunity to live in students dormitories. They have to rent apartments. Big role in finding an apartment takes the ESN organization – Erasmus Student Network, which is a non-profit international student organisation. Their mission is to represent international students, to provide opportunities for cultural understanding and self-development under the principle of Students Helping Students. Their logo is presented in Fig.7.




Fig. 7. ESN Niš.



Fig. 8. First acquaintance between the Serbian and Bulgarian participants in the Erasmus+ exchange programme.

On the other hand, Erasmus+ students from foreign countries in Sofia have the oppurtunity to live in students dormitories in the student city. Student city ("Studentski grad") is the student campus area for most universities in Sofia and also one of the 24 districts of Sofia. Student city quarter is located some 7km southeast of the center of Sofia. It was created in the 1960s and over 40 000 Bulgarian and foreigner students live there. The real number of people living temporarily there can hardly be estimated. Unlike most campus areas in Western Europe and Northern America, Student city is a common living place for most of the students of Sofia's numerous universities rather than being in the vicinity of one particular university. This helps students from different higher education institutions to meet and interact. It is one of the most diversified areas in Sofia, with old dwellings from the communist era and flashy new bars, clubs and fast-food joints springing up almost daily. Figure 9 shows block number 54, which is a typical representative of the dormitory buildings that are offered to the domestic and international students by TU Sofia.



Fig. 9. Student city, Sofia, Bulgaria.

Whether the students choose to live in the dormitory or in the apartment the scholarship covers all expenses regarding accommodation, food, books, travel tickets and pocket money.



Fig. 10. Lunch with international friends.

## IV. ACADEMIC ACTIVITIES

Taking part in the Erasmus+ program didn't include only traveling, sightseeing and meeting students from all

parts of the world, but also studying and making projects. Students didn't have classical exams and each student had a different obligations and tasks to perform with supervision of assigned mentor. Gained results were put in the final reports and were also presented in front of the advisors and other Erasmus+ mobility participients. Based on that final marks were gained. The most interesting thing was the difference in the grading system between the Serbian and Bulgarian universities. In Serbia the mark scale starts from 6 and goes to 10 whereas in Bulgaria the lowest passing mark is 2 and higest is 6.

Bulgarian part in Erasmus+ program was working with Advanced Design System (ADS) software for design and simulation of the RF and microwave circuits. One of the projects was designing a microstrip low-pass filter with open-stubs and comparing obtained results for the fabricated filter and the designed one (Fig 11). For fabrication we used Board Master and ProMat S100 and for measurements - Hewlett Packard 8753C Network Analyzer and Hewlett Packard 85047A S Parameter Test Set.

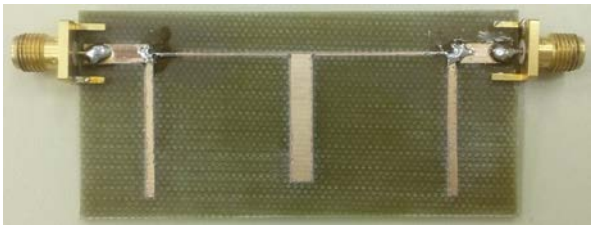


Fig 11. A photo of the fabricated filter.

Bulgarian students also had the oppurtunity to help in the organization of ICEST 2017 - 52nd International Scientific Conference on Information, Communication and Energy Systems and Technologies that was held at the Faculty of Eletronic Engineering (Fig. 12).




Fig 12. ICEST 2017, Niš, Serbia.

During the Erasmus+ program Serbian students participated in making web site about cooperation between TU Sofia and Universities of Niš and Belgrade. All their academic and social activities are presented on the following link: https://nemanja172.wixsite.com/erasmus.

Master student had opportunity to learn to use some new softwares for designing antennas and transformators. Also, Ph.D student gained basic knowledge on VHDL (hardware description language used to describe digital and mixed-signal systems such as field-programmable gate arrays and integrated circuits) and try to conduct thesis study related to the development of the baseband digital linearization method by using FPGA board (Fig. 13) and VHDL description for digital signals formation and their application in linearization of RF power amplifiers.



Fig.13. ZED FPGA board.



Fig 14. Workshop activities with our mentor and Erasmus+ students from other countries and universities.

## V. Conclusion

Despite the similarities between the neigbor countries, this exchange program was exquisite opportunity for all participants to learn something new regarding the professional life, to adjust on work and collaboration in foreign language and with meeting other ERASMUS+ students to widen their social circles. Life and work in the international environment allows everyone to expand new horizons, get to know different cultures and ways of thinking and that is an invaluable experience.

## References

[1] https://en.wikipedia.org/wiki/Technical_University,_Sofia
[2] https://www.ni.ac.rs/en/- web site of University of Niš
[3] http://www.tu-sofia.bg/ - web site of TU Sofia
[4] https://nemanja172.wixsite.com/erasmus- web site made by Serbian students
[5] http://www.serbia.com/visit-serbia/cities/nis/
[6] http://www.visitcapitalcity.com/europe/sofia-bulgaria
[7] https://wikitravel.org/en/Sofia
[8] https://en.wikipedia.org/wiki/Sofia

# Multimedijalni sistem informisanja učenika škole

Luka Zdravković, Nikola Krstić, Milan Stanković, Igor Arsić

*Sadržaj* - U ovom radu opisan je multimedijalni sistem informisanja učenika i nastavnika škole. Ovaj projekat sadrži neke od bitnih informacija kao što su vremenska prognoza, tekući čas/odmor, vreme do kraja istog, obaveštenja namenjena učenicima i zaposlenima škole, kao i zabava u vidu muzike za vreme odmora. Cilj rada je da uz pomoć ovog sistema roditelji, nastavnici i učenici na jedinstven, kreativan i efikasan način dolaze do bitnih informacija.

## I. Uvod

Od davnina u školama su se koristile razne metode za obeležavanje početka i kraja časa. Velika mana svih tih metoda je bila njihovo ručno upravljanje. Naime, nijedan od tih sistema nije bio automatizovan, što je donosilo neprecizno merenje trajanja časova. Ovaj sistem nudi rešenje za prethodno navedeni problem jedinstvenom implementacijom najsavremenije tehnologije računarskih nauka, što omogućava precizno merenje vremena do sledećeg zvona kao i digitalno informisanje učenika i zaposlenih u školi.

Drugim rečima: digitalni časovnik menja tradicionalno školsko zvono, a digitalna obaveštenja oglasnu tablu.

## II. Razvoj uređaja

### *A. Princip rada i izrada sistema*

Sistem se sastoji od jedne klijentske dinamične ASP.NET veb stranice i Windows Forms aplikacije koja njom upravlja. Jedan PC računar pokreće stranicu i aplikaciju na lokalnom serveru. Stranica se zatim putem HDMI kabla prezentuje na televizore u školskim hodnicima. Desktop aplikacija pokrenuta na računaru omogućava razne mogućnosti za upravljanje stranicom. Njihova međusobna komunikacija se realizuje lokalnom SQL bazom podataka.

Izgled grafičkog interfejsa aplikacije prikazan je na slikama 1 i 2.

Luka Zdravković, učenik srednje tehničke škole „Rade Metalac“ u Leskovcu, e-mail: lukazd99@outlook.com

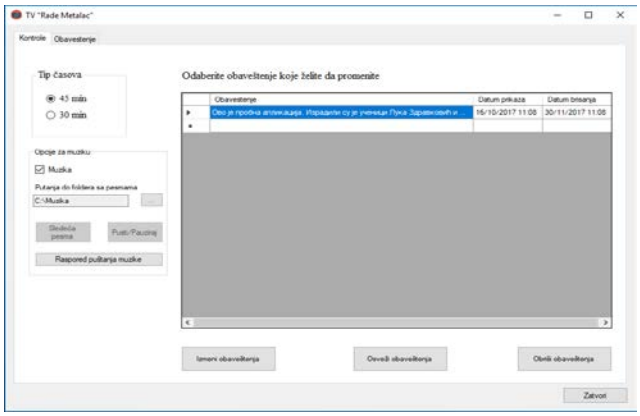
Nikola Krstić, učenik srednje tehničke škole „Rade Metalac“ u Leskovcu, e-mail: krsticnikola1999@gmail.com

Milan Stanković, profesor elektrotehničke grupe predmeta u TŠ "Rade Metalac", Leskovac e-mail: mist85@gmail.com

Igor Arsić, profesor elektrotehničke grupe predmeta u TŠ "Rade Metalac", Leskovac e-mail: igorarsic78@gmail.com



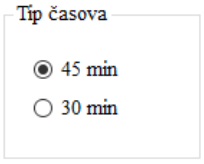

Slika 1. Izgled grafičkog interfejsa aplikacije.




Slika 2. Izgled grafičkog interfejsa aplikacije.

### *B. Funkcionalnosti aplikacije*

- Režim časova ( 45min / 30min )




Slika 3. Režim rada časova

Na slici 3 su prikazana dva kružića koja omogućavaju jednostavnu promenu režima trajanja časova. Pri promeni selekcije aplikacija šalje informaciju stranici o promeni i aplikacija se adaptira odabranom režimu.

- Obaveštenja

Na slici 4 prikazan je prozor za unos novih obaveštenja koji sadrži:

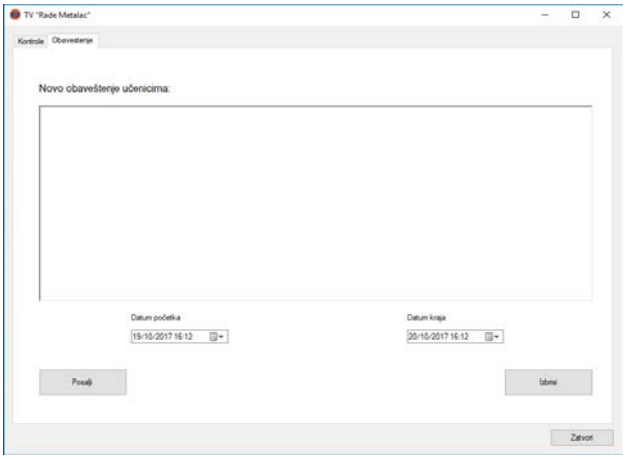
- Polje za unos teksta obaveštenja.

- Polje za odabir datuma i vremena početka prikaza obaveštenja (ukoliko nije navedeno, obaveštenje se odmah prikazuje).

- Polje za odabir datuma i vremena brisanja obaveštenja.

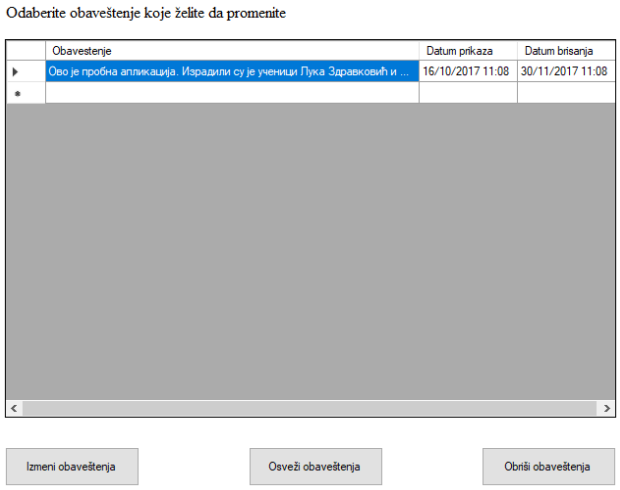
- Dugme za slanje obaveštenja.

- Dugme za poništavanje.




Slika 4. Prozor za unos novih obaveštenja.

Ukoliko obaveštenje koje se šalje prođe svu neophodnu validaciju, biće prikazana poruka o uspešnom slanju. Sva obaveštenja koja se nalaze u bazi podataka su prikazan u vidu tabele na glavnom prozoru aplikacije (slika 3.2).




Slika 5. Prozor za promenu obaveštenja.

Pored toga što sistem vrši automatsko ažuriranje, ovaj prozor omogućava ručno brisanje i izmenu obaveštenja.

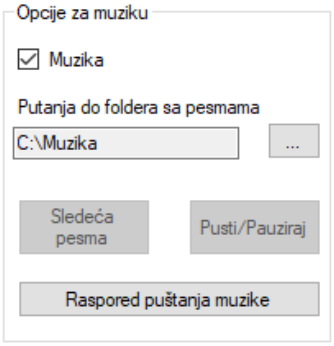
Klikom na dugme „izmeni“, ukoliko je korisnik odabrao barem jedan red u tabeli, prikazaće mu se novi prozor za izmenu teksta, datuma početka i kraja obaveštenja.

Klikom na dugme „izbriši“ odabrano obaveštenje iz tabele se briše iz baze.

Postoji i dugme „osveži“ čija je namena osvežavanje prozora u kome se prikazuju obaveštenja.

- Muzički prozor

Aplikacija sadrži prozor sa kontrolama za upravljanje muzikom, koja se automatski pušta svakog odmora.



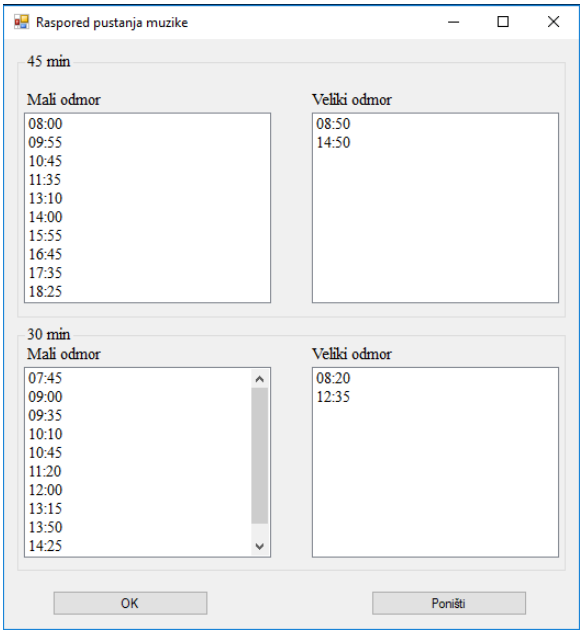

Slika 6. Opcije za muziku.

Dugme za čekiranje određuje da li će se muzika puštati ili ne.

Sledeća kontrola predstavlja putanju do lokalnog foldera iz kog će se puštati muzika, korisniku je predstavljeno dugme pomoću kog može na veoma lak način izabrati putanju.

Korisnik ima mogućnost puštanja nove pesme kao i pauziranje trenutne.

Na kraju korisnik ima mogućnost predefinisanja rasporeda puštanja muzike.




Slika 7. Raspored puštanja muzike.

*C. Definicija baze podataka*

| IME | TIP PODATKA | OGRANIČENJE |
|---|---|---|
| ID | INT | PRIMARNI KLJUČ |
| TEKST | NVARCHAR | NOT NULL |
| DATUM_PRIKAZA | DATETIME | NOT NULL |
| DATUM_BRISANJA | DATETIME | NOT NULL |

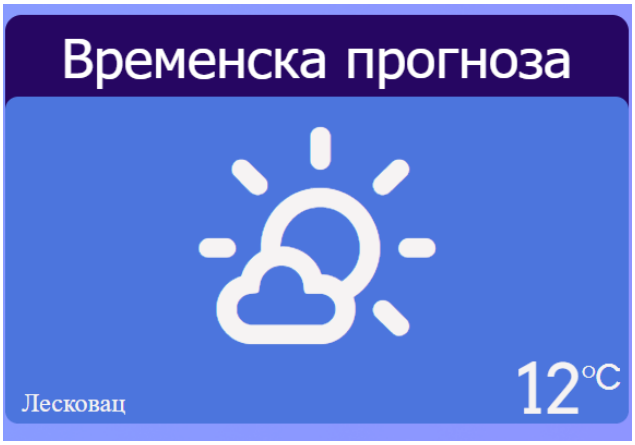
Izgled grafičkog interfejsa ASP.NETstranice prikazan je na slici 8.



Slika 8. Izgled grafičko interfejsa stranice.

Stranica sadrži sledeće prozore:

- Vremenska prognoza



Slika 9. Vremenska prognoza.

- Logo škole, trenutni datum i vreme



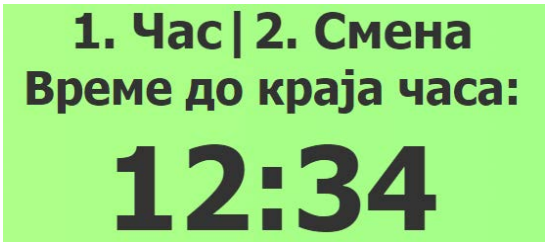
Slika 10. Logo škole.

- Animirani prozori za prijatelje škole



Slika 11. Prijatelji škole.

- Tekstualna labela koja pokazuje trenutni čas ili odmor, zajedno sa otkucavajućim časovnikom do sledećeg zvona



Slika 12. Trenutni čas.

- Traka sa obaveštenjima koja je uvek u pokretu



Slika 13. Traka sa obaveštenjima.

Projekat je razvijen pomoću sledećih tehnologija:
Microsoft Visual Studio
Microsoft .NET Framework 4.0
Microsoft SQL Server
Microsoft SQL Server Management Studio
Microsoft ASP.NET, ADO.NET, Windows Forms

## III. ZAKLJUČAK

U praktičnoj primeni projekta se pokazalo da je ovakav sistem izuzetno koristan i inovativan način informisanja učenika i zaposlenih. U zavisnosti od arhitekture objekta se određuje broj monitora koji prikazuju klijentsku aplikaciju. Desktop aplikacija zahteva samo jedan računar. Preporučuje se postavljanje uređaja u holu na ulazu škole jer će na tom mestu sistem najefikasnije obavljati svoju funkciju tj. biće od najveće koristi.

Prednost ovog uređaja ogleda se u tome što prisutne u školi u svakom trenutku obaveštava o aktivnostima u školi. U daljem razvoju planira se uvođenje i zvučne signalizacije za početak i kraj časova čime će se objediniti funkcije sa klasičnim školskim zvonom a samim tim će u potpunosti zameniti postojeće zvono u školama.

## LITERATURA

[1] https://docs.microsoft.com/enus/dotnet/framework/winforms/windows-forms-overview
[2] https://msdn.microsoft.com/en-us/library/ms973868.aspx
[3] https://docs.microsoft.com/en-us/sql/sql-server/sql-server-technical-documentation