IEEESTEC

15th 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
IEEE WIE Affinity Group
IEEE Computational Intelligence Chapter
Serbian Academy of Sciences and Arts (SASA), Branch Niš
Ministry of Education, Science and Technological Development

ISBN: 978-86-6125-257-0

IEEESTEC Student Projects Conference, November 24, 2022, Niš, Serbia
Proceeding of Papers

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

Chairman:
Danijel Danković

Vice Chairman:
Miloš Marjanović

Editors of Proceedings:
Vera Marković
Danijel Danković
Zlatica Marinković
Miloš Marjanović
Biljana Stošić
Emilija Živanović

Secretary:
Jana Vračar
Aleksandra Stojković
Nikola Mitrović
Milan Stojanović
Sandra Veljković
Neda Stanojević
Marija Živković
Anđela Kocić
Nemanja Nedeljković
Anastasija Avramović
Nikola Stojiljković
Strahinja Dimitrijević
Mikica Radenković
Ilija Đorđević Arsić
Maša Đelić

Technical Editors:
Miloš Marjanović
Jana Vračar
Danijel Danković
Emilija Živanović

CIP - Каталогизација у публикацији - Народна библиотека Србије, Београд

621.3(082)
004(082)

IEEESTEC Student Project Conference (15 ; 2022 ; Niš)
IEEESTEC 15th Student Project Conference, [November 24, 2022, Niš] / organized by EESTEC LC, Niš ... [et al.] ; [editors Vera Marković ... [et al.]]. - Niš : Faculty of Electronic Engineering, 2022 (Niš : Unigraf X copy). - 320 str. : ilustr. ; 30 cm

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

ISBN 978-86-6125-257-0

1. Institut inženjera elektrotehnike i elektronike. Studentski ogranak (Niš)

a) Електротехника - Зборници b) Рачунарство - Зборници

COBISS.SR-ID 79823625

Printed by: Unigraf X copy, Niš, 2022.
Circulation: 100 copies

Reviewers:

1. Igor Jovanović
2. Nikola Krstić
3. Vladimir Sibinović
4. Jelena Jovanović
5. Nikola Mitrović
6. Miloš Marjanović
7. Goran Nikolić
8. Ksenija Pešić
9. Biljana Stošić
10. Marko Dimitrijević
11. Aleksandra Stojković
12. Jelena Nedeljković
13. Milan Stojanović
14. Tatjana Nikolić
15. Emilija Živanović
16. Stevica Cvetković
17. Jana Vračar
18. Neda Stanojević
19. Sandra Veljković
20. Vojkan Davidović
21. Nenad Petrović
22. Milena Frtunić-Gligorijević
23. Aleksandar Milenković
24. Milica Jovanović
25. Filip Filipović
26. Vladimir Mitić
27. Nikola Danković
28. Milan Simić
29. Goran Miljković
30. Miljana Milić
31. Dragana Vukić
32. Ana Vučković
33. Teodora Vještica
34. Dejan Mirković
35. Anđelija Đorđević
36. Marija Milijić
37. Igor Stojanović
38. Nikola Stojanović
39. Nikola Vučić
40. Oliver Vojinović
41. Danijel Danković
42. Aleksandar Vučković
43. Marija Živković
44. Aleksandar Atanasković
45. Sandra Đošić
46. Vojkan Kostić
47. Aneta Prijić
48. Vanja Stošić
49. Milan Danković

Table of contents:

Petnaest godina Konferencije studentskih projekata IEEESTEC 1
Izazovi i rešenja u procesu zamene komponente integrisanog kola prekidačkog napajanja... 5
Detekcija i zaobilaženje prepreka pomoću razvojnog sistema Arduino 9
CAN komunikacioni protokol 13
Kontrola brzine okretanja DC motora u zavisnosti od izmerenog rastojanja 19
Solarni sistem za praćenje svetlosti i korišćenje energije solarnih panela pomoću Arduino Mega platforme 23
Projektovanje LED osvetljenja korišćenjem TLC5925 drajvera 27
Izvor jednosmernog napona sa digitalnom kontrolom 31
Projektovanje Power Bank uređaja 35
Život u Švajcarskoj i Master studije na Švajcarskom Federalnom Tehničkom Institutu u Cirihu (ETH Zürich) 39
Linearni izvor napajanja sa promenljivom vrednošću izlaznog napona korišćenjem ATMEGA328P... 47
Kontrola servo motora korišćenjem ARM mikrokontrolera 51
Određivanje statičkog probojnog napona gasnog odvodnika prenapona *EPCOS* 55
Platforma za treniranje neuronskih mreža genetskim algoritmom na jednostavnim video igrama 59
Primena SCADA sistema u procesu proizvodnje sladoleda 63
Robot sa četiri ekstremiteta 67
Primena dekadnog brojača u realizaciji zidnog sata 73
Aplikacija za upoređivanje temperatura zdrave i obolele strane lica u *Matlab* okruženju 77
Bez prepreka - jer život ne može da čeka. Projekat naplatne rampe za vozila hitne službe 81
Pametan automobil i pametna garaža realizovani pomoću Arduino razvojnog sistema 85
My MSA – Web aplikacija za pomoć pri selidbama 89
Projektovanje štampane ploče za MPPT kolo 95
Sistem za praćenje vlažnosti zemljišta i parametara zagađenosti vazduha... 99
Kućna automatizacija bazirana na IoT tehnologijama 103
Kontrola brzine i smera DC motora korišćenjem kola L293D i mikrokontrolera sa ARM arhitekturom 107
Web aplikacija za sortiranje i rangiranje studenata po studijskim grupama 111
Karakterizacija gasnih komponenata proizvođača *Littelfuse* 115
Praktična realizacija elektronskog kola tranzistorskog dvokvadrantnog čopera... 121
UGV – Robot SCORPION 125
Ispitivanje stabilnosti probojnog napona gasnih odvodnika tipa *Littelfuse*-300 V 129
Sistem za prijavljivanje zasnovan na prepoznavanju lica 133
Primena dronova za sejanje drveća 141
Statistička analiza probojnog napona i vremena odziva gasnog odvodnika proizvođača *EPCOS* 350 V 143
Poređenje karakteristika tipova litijum-jonskih baterija 147
Ispitivanje pouzdanosti gasnih odvodnika tipa *Littelfuse*-300 V 151
Generisanje neograničeno dugih klavirskih sintetičkih melodija različitih žanrova... 155
Poređenje tačnosti modela random forest i hibridne neuronske mreže na klasifikaciju teksta... 159
Korišćenje LSTM neuronske mreže za prepoznavanje varanja prilikom ciljanja u video igri CS:GO 163
Kontrola rada DC motora pomoću Arduino okruženja i TTP223-BA6 *touch* senzora 167
Tsp biblioteka 169
Analiza metoda genetskog algoritma u izvođenju formule funkcije sa zadatog grafika 171
Scikit-learn biblioteka i metode mašinskog učenja za obradu neuroslika 175
Sistem za dimovanje RGB diode 179
Projektovanje pametnog sistema za pešačke prelaze 183
Digitalni sat realizovan korišćenjem integrisanog kola DS1302 i ATMEGA328P mikrokontrolera 187
Illumination Indicator Using an Arduino Development Environment 191
Automatizacija parkinga sa PIC mikrokontrolerom 195
Implementacija Linuks operativnog sistema na mikrokontrolerima sa ARM arhitekturom 199
Projektovanje veb aplikacije za grafički prikaz podataka u IoT sistemima 203
Sistem za popunjavanje baze podataka veštački generisanim podacima 207
Testiranje karakteristika Džonsonovog brojača na primeru kola 74HC4017 213
Projektovanje video igre za desktop računare "Remorse" 217
Automatski punjač za LiFePO4 bateriju 221
Ekonomski dispečing termosistema 225
Design and Implementation of a Self-Synchronizing LFSR based Descrambler 229
Redni i paralelni drajver za LE diodu za optičke komunikacije 233
Merenje karakteristika fototranzistora i fotodiode 237
Projektovanje i realizacija samosinhronišućeg skremblera 241
Sistem osvetljenja baziran na fotootporniku i LED diodi 245
Comparison of Capacitive and Resistive Methods for Water Salinity Measurements 247

Primer korišćenja Arduino kontrolera za pravilno sedenje 251
Detekcija pola pomoću konvolucionih neuronskih mreža 253
Ispitivanje karakteristika digitalnih kola 74LS68 i 74LS69 255
Projektovanje višeslojne štampane ploče za komunikaciju sa HC-05 bluetooth modulom 259
GPU Implementation of SVM Algorithm 265
Uređaj za merenje nivoa vode 271
Primena veštačke inteligencije za podešavanje kontrolera laboratorijske stanice za flaširanje 275
Projektovanje i dizajn štampane ploče na primeru LED bedža 279
Razvoj decentralizovanih veb-aplikacija 283
Elektrohemijski procesi odgovorni za nestabilnosti VDMOS tranzistora snage usled NBT naprezanja 287
Određivanje vlažnosti vazduha i temperature korišćenjem senzora DHT11 293
Modeling a Fractal Antenna using Artificial Neural Networks 295
Dozimetar jonizujućeg zračenja zasnovan na Arduino mikrokontroleru i senzoru BG51 299
Detektor vlažnosti zemljišta zasnovan na Arduinu 303
Mašinsko učenje u okviru kursa Informacioni sistemi korišćenjem *Weka* API za Javu 305
MOS tranzistor sa plivajućim gejtom kao dozimetar ultraljubičastog zračenja u fototerapiji 311
U susret jubileju: naša petnaesta godina kroz radionice i druženje 315

Petnaest godina Konferencije studentskih projekata IEEESTEC

Danijel Danković, Emilija Živanović, Miloš Marjanović

Sadržaj – U ovom radu opisan je razvoj IEEESTEC konferencije studentskih projekata od nastanka do danas. Predstavljeni su ciljevi studentske konferencije i načini motivacije studenata i učenika da se bave naučno-istraživačkim radom i pisanjem radova. Predstavljeni su relevantni podaci koji pokazuju razvoj i napredak IEEESTEC 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), IEEE WIE Affinity Group, sekcije IEEE Srbija i Crna Gora [4] i Srpske akademije nauka i umetnosti (SANU) [5]. Od 2018. godine IEEESTEC konferencija podržana je od strane Ministarstva prosvete, nauke i tehnološkog razvoja Republike Srbije [6].

Konferencija je nastala kao rezultat saradnje IEEE studentskog ogranka Univerziteta u Nišu i EESTEC lokalnog komiteta Niš, pa je tako i formirano ime 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 400.000 članova u 160 zemalja sveta, od toga 100.000 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, kao što je EESTEC.

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.

Ogranak SANU u Nišu osnovan je 16. septembra 2016. godine, a konstituisan 04. novembra 2016. godine, čime je prestao sa radom Centar za naučnoistraživački rad SANU i Univerziteta u Nišu, osnovan 1990. godine, i obezbeđen organski prelazak na novi, viši organizacioni nivo i nove oblike rada i standarde predviđene Statutom SANU i Pravilima Ogranka SANU u Nišu. Osnivanjem Ogranka postiže se širenje i snaženje duha i misije SANU u jugoistočnoj Srbiji, i daje neophodan zamah unapređivanju naučnih istraživanja i umetničkoga stvaralaštva od vitalnog interesa za društveni, ekonomski, kulturni razvoj ovoga dela naše zemlje, bez koga nema ravnomernog i uspešnog razvitka Republike Srbije [5]. Od samog osnivanja rukovodstvo Ogranka SANU u Nišu prepoznalo je potencijale IEEESTEC konferencije i podržalo njeno održavanje.

Potencijale IEEESTEC konferencije prepoznale su i druge brojne ustanove, organizacije i kompanije: Gradska opština Pantelej [7], Gradska opština Crveni Krst [8], Centar za promociju nauke (CPN) [9], Zavod za unapređivanje obrazovanja i vaspitanja (ZUOV) [10], IEEE kroz TryEngineering IEEE Pre-University STEM Portal Grant [11], M&G Electronic [12], društvo IEEE Computational Intelligence … Više od 30 tehničkih škola i gimnazija širom Srbije i više od 20 fakulteta iz Srbije i inostranstva dalo je značajan doprinos razvoju i napretku IEEESTEC konferencije.

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:

D. Danković, E. Živanović i M. Marjanović, Katedra za mikroelektroniku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: danijel.dankovic@elfak.ni.ac.rs, emilija.zivanovic@elfak.ni.ac.rs, milos.marjanovic@elfak.ni.ac.rs.

1) Osnovna ideja, na početku, je bila da studenti osnovnih ili master 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.

2) 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 istraživačkih projekata studenata.

3) 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, fizike...).

4) 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.

5) 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. Kao rezultat toga, nekadašnji autori IEEESTEC konferencije sada su predstavnici kompanija i zaduženi su za regrutaciju novih članova iz redova autora.

B. Motivacija učenika i studenata za učešće na IEEESTEC konferenciji

"Trebalo bi da napišete rad za IEEESTEC konferenciju", rečenica koja je izgovorena mnogo puta tokom prethodnih petnaest godina. Međutim, ta rečenica nije dovoljan motiv da učenik ili student zaista napiše rad. Veoma je važno da učenik ili student, videvši naše višegodišnje angažovanje, dobije dodatan motiv i nezavisno donese odluku da napiše rad. Zavisno od njihovog interesovanja, postoji širok opseg različitih motivacija za pisanje rada i učešće na IEEESTEC konferenciji. Prvo, prethodno navedeni ciljevi IEEESTEC konferencije su veoma važan razlog za pisanje radova. Drugo, najbolji radovi se nagrađuju. Na osnovu ocena recenzenata nominuju se tri najbolja rada. Organizacioni odbor konferencije bira i tri rada sa najboljom praktičnom realizacijom, jedan rad za specijalnu nagradu IEEE MTT autorima radova iz oblasti telekomunikacija, kao i jedan rad sa ženskim autorima koji dodeljuje IEEE Women in Engineering (WIE) grupa. Najznačajnija nagrada na konferenciji je nominacija rada koji predstavlja IEEE studentski ogranak Univerziteta u Nišu na svetskom takmičenju IEEE Region 8 Student Paper Contest [13], i ta nagrada od 2021. godine nosi naziv po preminulom akademiku Ninoslavu Stojadinoviću. Treba napomenuti da samo članovi IEEE mogu učestvovati na ovom takmičenju.

Organizatori uz pomoć prijatelja konferencije obezbeđuju veliki broj nagrada za autore najboljih radova (detalji sa dodele nagrada na slici 2). Nagrađeni radovi prezentuju se usmeno u okviru svečanosti otvaranja konferencije (ilustracija slika 3). Svi ostali radovi prezentuju se preko postera, dok autori sa praktičnim projektima demonstriraju svoje radove u holu Elektronskog fakulteta (ilustracija slika 4).

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.



Slika 2. Dodela nagrada nagrađenim autorima.



Slika 3. Usmena prezentacija nagrađenih radova.



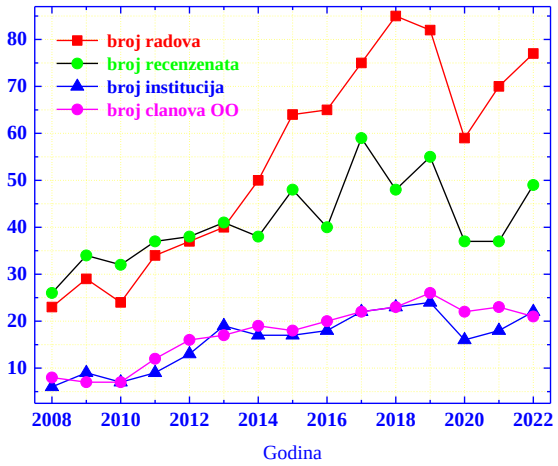
Slika 4. Demonstracija praktično realizovanih projekata.

C. Napredak IEEESTEC konferencije

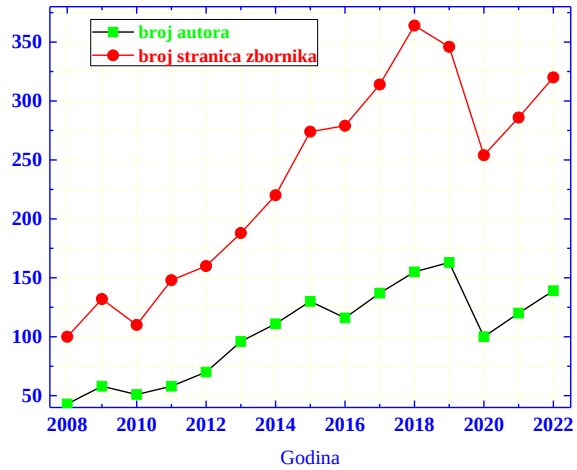
Značajnim angažovanjem organizatora konferencije ona se iz godine u godinu poboljšava i napreduje. Konferencija se organizuje u okviru obeležavanja dana Elektronskog fakulteta, tako da je veliki broj gostiju Elektronskog fakulteta prisutan na konferenciji, a i učesnici konferencije uveličavaju proslavu 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 I. Glavni cilj oranizatora konferencije je da se ovaj trend 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
2018.	85	155	48	364	23	23
2019.	82	163	55	346	24	26
2020.	59	100	37	254	16	22
2021.	70	120	37	286	18	23
2022.	77	139	49	320	22	21



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

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 značajnom događaju, kao što je prikazano u Tabeli II.

TABELA II
IEEESTEC KONFERENCIJA JE POSVEĆENA ZNAČAJNIM JUBILEJIMA

God.	Događaj
2008	Početak saradnje IEEE SB Niš i EESTEC LC Niš
2009	125 godina IEEE-a
2010	50 godina Elektronskog fakulteta u Nišu
2011	40 godina IEEE sekcije Srbija i Crna Gora
2012	5 godina postojanja IEEESTEC konferencije
2013	50 godina IEEE Region 8
2014	50 godina magazina IEEE SPECTRUM
2015	50 godina Univerziteta u Nišu
2016	45 godina IEEE sekcije Srbija i Crna Gora
2017	10 godina postojanja IEEESTEC konferencije
2018	70 godina od osnovananja "Zavodi RR", EI Niš.
2019	135 godina IEEE-a
2020	55 godina Univerziteta u Nišu
2021	50 godina IEEE sekcije Srbija i Crna Gora
2022	15 godina postojanja IEEESTEC konferencije

Ovogodišnja IEEESTEC konferencija poklapa se sa proslavom jednog značajnog jubileja društva IEEE EDS, 75 godina bipolarnog tranzistora [14].



Slika 7. IEEE EDS društvo proslavlja 75 godina postojanja bipolarnog tranzistora [14].

Cilj organizatora IEEESTEC konferencije 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 8, radiće se i dalje na promociji i popularizaciji naučno-istraživačkog rada i pisanja radova.



Slika 8. Atmosfera sa otvaranja IEEESTEC konferencije.

ZAHVALNICA

Ovaj tekst je nastao kao deo regularnih aktivnosti članova IEEE-a 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.ogranaknis.sanu.ac.rs/istorijat/
[6] Ministarstvo prosvete, nauke i tehnološkog razvoja Republike Srbije. Dostupno na: https://www.mpn.gov.rs/
[7] Gradska opština Pantelej. Dostupno na: http://www.pantelej.org.rs/
[8] Gradska opština Crveni Krst. Dostupno na: http://gocrvenikrst.rs/
[9] Centar za promociju nauke. Dostupno na: https://www.cpn.edu.rs/
[10] Zavod za unapređivanje obrazovanja i vaspitanja. Dostupno na: https://zuov.gov.rs/
[11] IEEE TRYEngineering. Dostupno na: https://tryengineering.org/news/announcing-the-2021-ieee-pre-university-stem-portal-grant-recipients/
[12] M&G Electronic. Dostupno na: https://www.mgelectronic.rs/
[13] Student Paper Contest. Dostupno na: http://www.ieeer8.org/category/student-activities/awards-and-contests/student-paper-contest/
[14] 75 godina bipolarnog tranzistora. Dostupno na: https://eds.ieee.org/about-eds/75th-anniversary-of-the-transistor

Izazovi i rešenja u procesu zamene komponente integrisanog kola prekidačkog napajanja u postojećim drajverima za napajanje LED modula

Jana Kotnik

Sadržaj – U ovom radu je prikazana zamena postojećeg integrisanog kola za pobudu gejta za integrisano kolo drugog proizvođača (SS IC- *Second Source Integrated Circuit*) u bloku prekidačkog napajanja (sinhroni spuštač napona). Rezultat istraživanja je adaptacija postojećih uređaja koja je bila neophodna usled velikog nedostatka trenutno korišćenog integrisanog kola na tržištu (*shortage* komponenti).

I. UVOD

Tokom pandemije virusa COVID-19 došlo je do velikih promena na tržištu komponenata, što je za posledicu imalo popunjavanje proizvodnih kapaciteta u Kini. Veliki broj firmi je zbog nedostataka komponenata bilo primorano da vrši adaptaciju svojih uređaja na trenutno dostupne komponente. U ovom radu se mogu videti tehnike ispitivanja kompatibilnosti nove komponente, kao i zahtevi firmi tokom vršenja promene primarnog integrisanog kola (FS IC- *First Source Integrated Circuit*) integrisanog kola u bloku prekidačkog napajanja drajvera, koristeći integrisano kolo drugog proizvođača (SS IC- *Second Source Integrated Circuit*).

II. OPIS PROBLEMA

Prilikom promene proizvođača integrisanog kola u bloku prekidačkog napajanja cilj je bio sačuvati stari dizajn štampane ploče kako se ne bi ulazilo u proces ponovnih testiranja i verifikacije samog dizajna, što je smanjilo izbor proizvođača. Pored toga, novi dizajn bi značio i promenu testnih tačaka, što je dodatni trošak u proizvodnji zbog kupovine nove testne glave. Cilj je bio kreiranje alternativnog BOM-a (BOM- *Bill Of Materials*) uređaja koji će zadržati potpuno iste električne performanse, bez negativnog elektromagnetnog uticaja na uređaj, sa istim termalnim karakteristikama i životnim vekom. Veliki broj drajvera je bio pogođen ovom izmenom- drajveri sa izlaznom snagom od 75-300 W, izlaznog napona do 300 V i izlazne struje od 125-1600 mA.

III. OSNOVNI METODI, TEHNIKE I INSTRUMENTI

Analizom ponašanja sekundarne komponente dobijaju se određeni rezultati koji se upoređuju sa rezultatima koje je davala primarna komponenta. Analiza se svodi na posmatranje signala na izlazu uređaja, temperaturnim merenjima, upoređivanje rezultata elektromagnetnog emitovanja uređaja, kao i treperenja svetla (*flicker*).

Merenja temperature koja se razvija na drajveru se realizuju postavljanjem termoparova na kritične komponente u kolu, kao što su MOSFET tranzistori, elektrolitski kondenzatori i transformatori na čitavom drajveru. Pored toga, drajveri se smeštaju u termalnu kutiju gde se simulira promena ambijentalne temperature do maksimalne specificirane za dati drajver pri čemu se prati njegovo ponašanje.

EMI (*Electro-Magnetic Interference*) testiranje se sprovodi u posebnoj laboratoriji, gde se posmatraju elektromagnetne distorzije na određenim strujama i naponima. Ovo merenje se vrši u okviru firme nakon svake promene na samom dizajnu drajvera.

Što se tiče testiranja treperenja svetla, tj. *flicker*-a, posmatranjem intenziteta osvetljaja potrošača kao i njegovog oscilovanja (potrošačko kolo sačinjeno od LED dioda, tj. LED modul) može se uočiti treperenje ako takvo postoji.

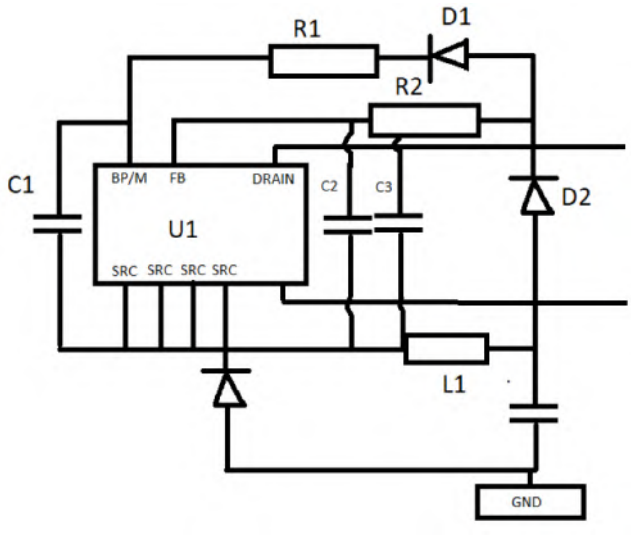
Instrumenti korišćeni za ovo istraživanje su:

- *Ronde&Schwarz* osciloskop,
- *Keyside* DC izvor napajanja,
- *Keithly* multimetar,
- *Chroma* LED opterećenje.

IV. PRIKAZ I ANALIZA REZULTATA ISTRAŽIVANJA

Tokom rada na ovom zadatku mogu se javiti mnogobrojni problemi. Prvenstveno, nakon zamene integrisanog kola, uređaj se nije mogao startovati, a zamenjeno integrisano kolo nije generisalo nikakve signale za pobudu MOSFET-a (*M1*, slika 2) u bloku sinhronog spuštača napona. Na pin V_{dd}, tj. pin za napajanje integrisanog kola U1 doveden je napon sa LVPS (*Low Voltage Power Supply*) bloka koje može koristiti dva različita integrisana kola (postoje dve varijante u slučaju nedostatka neke od komponenti sa istim otiskom na štampanoj ploči). LVPS blok u kolu drajvera ima ulogu da naponski nivo od 400 V spusti na naponski nivo od 12 V (kolo spuštača napona) i time obezbedi potreban signal za napajanje integrisanog kola u bloku sinhronog spuštača napona kao i za napajanje mnogih drugih delova kola. Uprošćena šema bloka LVPS se može videti na slici 1.

J. Kotnik je student master studija na modulu Elektronika i mikrosistemi Elektronskog fakulteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: jana.kotnik@elfak.rs.



Slika 1. Šema LVPS bloka.

Drajver je funkcionisao ispravno kada je bila korišćena prva konfiguracija LVPS bloka sa integrisanim kolom označenim U1, ali ne i kad je sekundarna komponenta U1A povezana. Zbog sumnje da je napon koji se dobija iz LVPS bloka nedovoljno stabilan za rad zamenskog integrisanog kola dovodi se napon od 12 V sa *Keyside* DC izvora napajanja. Pošto uređaj i dalje nije davao izlazne signale, povećavanjem ulaznog napona za korak od 0.1 V, na naponskom nivou od 12.5 V uređaj je počeo sa stabilnim radom, čime se zaključuje da je potreban veći naponski nivo za napajanje sekundarnog integrisanog kola. Istraživanjem razloga zašto je drajver radio sa U1 ali ne i sa U1A, došlo se do zaključka da je talasnost napona na U1 mnogo veća (*ripple*), te da je signal upravo zbog talasnosti dostizao naponske nivoe od 12.5 V što je bilo dovoljno za napajanje primarnog integrisanog kola, dok to kod U1A nije slučaj jer ova komponenta ima stabilniji izlazni napon. Naponski nivo LVPS bloka je iz ovog razloga podignut na 13 V što je obezbedilo nesmetani rad sekundarnog integrisanog kola.

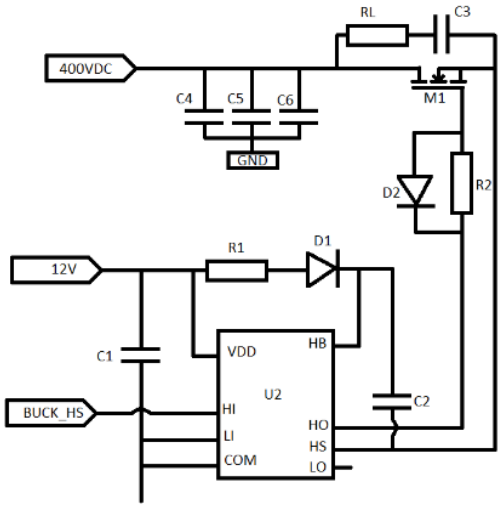
Povećanje izlaznog napona LVPS bloka je postignuto promenom otpornika u samom bloku, tačnije promenom odnosa otpornosti unutar bloka. Otpornik *R1* je povećan sa 10 kΩ na 11 kΩ, dok je otpornost otpornika *R2* povećana sa 42 kΩ na 51 kΩ. Uz pomoć drugačijeg odnosa u razdelniku napona, izlazni napon je podignut na naponski nivo od 13 V. Do optimalne vrednosti otpornika se došlo simulacijom u programu *LT Spice*.

Sledeći problem koji se javio tokom zamene integrisanog kola je bio *bootstrap* kondenzator *C2* čija se vrednost morala promeniti (slika 2).

Za svaku promenu integrisanog kola koje za ulogu ima pobudu MOSFET-a kao i promenu samog tranzistora, mora se prilagoditi *bootstrap* kondenzator zbog toga što on većinski zavisi od gejt-sors naelektrisanja koje je potrebno da se tranzistor dovede u provodno stanje [1]. Iz tehničke dokumentacije za MOSFET LNK3202/3204-6 tranzistor dobijaju se informacije o količini naelektrisanja gejta [2] (Tabela I). Iz tabele I se može videti da je ukupno naelektrisanje gejta u MOSFET-u jednako 9.4 nC, pa je jednačinama (1) i (2) izračunata ukupna količina naelektrisanja potrebna *bootstrap* kondenzatoru:

$$Q_{total} = Q_g + \frac{I_{qbs}}{f_sw_min} \tag{1}$$

$$Q_{total} = 9.4nC + \frac{300\mu A}{50kHz} = 15.4nC \tag{2}$$



Slika 2. Deo uprošćene šeme prekidačkog napajanja.

Kada se izračuna ukupna količina naelektrisanja, i uz izmerenu vrednost $\Delta Vboot$ (talasnost na *bootstrap* kondenzatoru), jednačinom (3) se može izračunati vrednost kondenzatora potrebna da MOSFET provodi onda kada je to potrebno:

$$C_{bootstrapMIN} = \frac{Q_{total}}{\Delta V_{boot}} = \frac{15.4nC}{250mV} = 61.6nF \tag{3}$$

Da bi bili sigurni da će *bootstrap* kondenzator isporučivati dovoljnu količinu naelektrisanja, koristi se faktor sigurnosti 5 (jednačina (4)):

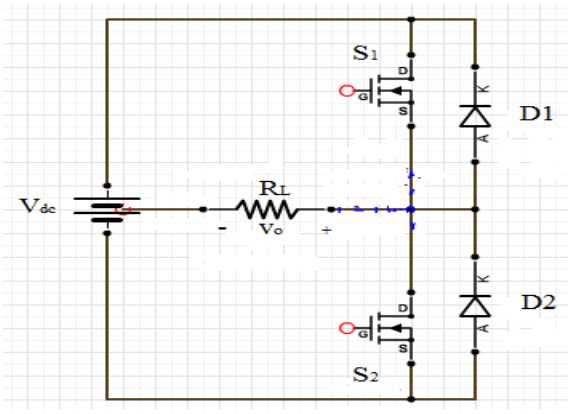
$$C_{bootstrap} = 5 * C_{bootstrapMIN} = 308nF \tag{4}$$

Izabrana komponenta je *bootstrap* kondenzator vrednosti 330 nF.

Sledeći problem koji se javio tokom korišćenja sekundarne komponente bio je javljanje pojave *hard-switching* [3]. Prethodno dizajnirano kolo sinhronog spuštača napona je bilo projektovano tako da se u ovom polumostnom naponskom inverteru (slika 3) MOSFET-ovi uključuju i isključuju naizmenično tj. kada napon drejn-sors jednog MOSFET-a dođe na nulu, tek tada počinje da raste napon drejn-sors na drugom tranzistoru. Tačnije, ako je jedan MOSFET u provodnom stanju, drugi mora biti u zakočenju. Ovo je pojava koja se naziva ZVS (*Zero Voltage Switching*) [4].

TABELA I
INFORMACIJE O NAELEKTRISANJIMA GEJTA MOSFET-A LNK3202/3204-6 IZ TEHNIČKE DOKUMENTACIJE

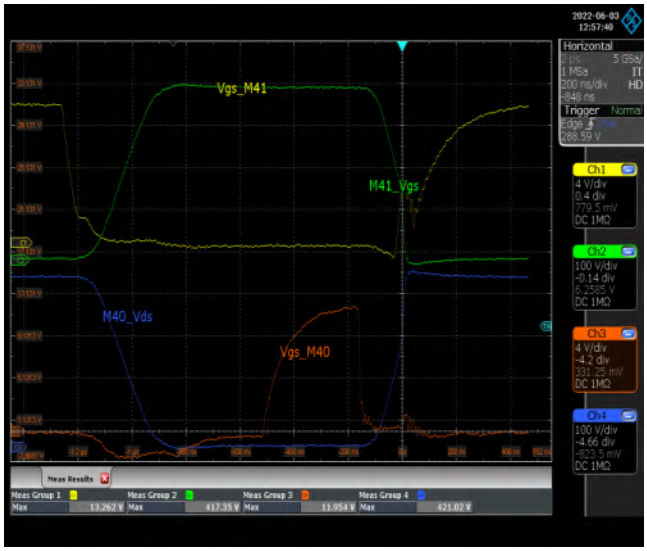
Parametar	Sim-bol	Tipična vrednost	Jedi-nica	Uslovi testiranja
Gejt-sors količina naelektrisanja	Q_{gs}	1.1	nC	V_{DD}=480 V I_D=1.4 A V_{GS}=0 do 10 V
Gejt-drejn količina naelektrisanja	Q_{gd}	5	nC	V_{DD}=480 V I_D=1.4 A, V_{GS}=0 do 10 V
Ukupna količina naelektrisanja gejta	Q_g	9.4	nC	V_{DD}=480 V I_D=1.4 A V_{GS}=0 do 10 V
Napon Milerovog platoa	$V_{plateau}$	5.4	V	V_{DD}=480 V I_D=1.4 A V_{GS}=0 do 10 V



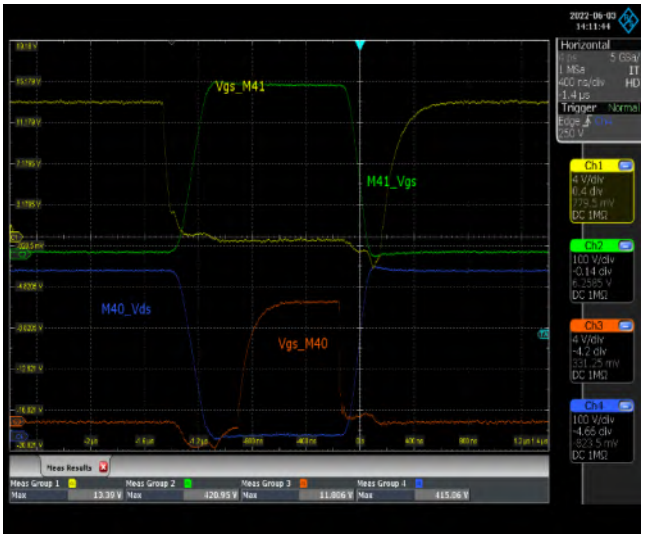
Slika 3. Šematski prikaz polumostnog naponskog invertera.

U *hard-switching*-u postoji određeni vremenski period kada su oba MOSFET-a u provodnom stanju, što dovodi do velikih energetskih gubitaka i pojave šuma (slika 4). Pored toga, može se desiti i kratak spoj između naponskog nivoa od 400 V i mase. Samim tim što je primarna komponenta imala mnogo manja propagaciona kašnjenja, i vreme kada su oba MOSFET-a u zakočenom stanju je bilo manje (*dead-time*), tj. u ovom slučaju nije ni postojalo. Sa većim propagacionim kašnjenjem novog IC-a, *dead-time* softverski definisan za primarno integrisano kolo nije bio dovoljan. Podešavanjem *dead-time*-a kroz odgovarajući softver mikrokontrolera obezbeđen je ZVS (slika 5).

Na kraju je primećeno da se sekundarna komponenta više greje od prvobitnog integrisanog kola zbog integrisane *bootstrap* diode, što je rešeno limitiranjem struje kroz istu uz pomoć manjeg *bootstrap* kondenzatora. Pored toga, preko bloka sinhronog spuštača napona je stavljen *gap-pad* čime je omogućena ravnomernija i brža disipacija toplote na okolinu i metalno kućište drajvera [5]. *Gap-pad* predstavlja tanki materijal od silikonskog polimera sa dodacima keramike koji služi za odvođenje toplote sa dela štampane ploče na koji je zalepljen (slika 6).



Slika 4. Žuti signal predstavlja signal gejt-sors na *high-side* MOSFET-u (*M2*), zeleni predstavlja drejn-sors napon *M2*. Ispod se može videti plavi signal (drejn-sors *M1*) i narandžasti signal (gejt-sors *M1*). Na slici su signali pre podešavanja *dead-time*-a, tačnije, jako je dugačak period između vođenja *M1* i *M2* tj. između žutog i narandžastog signala (kada tranzistor vodi, napon gejt-sors ima pozitivni naponski nivo, čime dolazi do otvaranja provodnog kanala). Uz pomoć žutog signala se može detektovati pojava *hard-switching*-a (nije stabilan). Na ovoj slici se može primetiti da napon drejn-sors na *M1* još uvek nije pao na nulu, a napon gejt-sors je već počeo da raste.



Slika 5. Slika signala iz osciloskopa nakon podešavanja *dead-time*-a. Kao što se može videti, sada je žuti signal stabilan. Period vremena u kome su oba MOSFET-a u zakočenju je znatno manji nego na slici 4.



Slika 6. Fotografija termalnog *gap-pad*-a.

V. ZAKLJUČAK

Prilikom zamene bilo koje komponente na postojećem uređaju, cilj je pronaći komponentu sa boljim tehničkim karakteristikama. U ovom radu je pisano o zameni integrisanog kola koja za ulogu ima pobudu gejta, a prilikom izbora zamene traži se komponenta sa manjim propagacionim kašnjenjima, manjom disipacijom energije i ostalim boljim performansama. Poželjno je iskoristiti komponentu čija je cena povoljnija, a da ispunjava sve zahteve po pitanju performansi.

LITERATURA

[1] J. K. Application Note AN-1123, Bootstrap Network Analysis: Focusing on the Integrated Bootstrap Functionality, By Andrea Merello

[2] Tehnička dokumentacija. https://eu.mouser.com/datasheet/2/328/LinkSwitch-TN2_family_datasheet-1036065.pdf

[3] Cittanti, D.; Gammeter, C.; Huber, J.; Bojoi, R.; Kolar, J.W. A Simplified Hard-Switching Loss Model for Fast-Switching Three-Level T-Type SiC Bridge-Legs. Electronics 2022, 11, 1686. https://doi.org/10.3390/ electronics11111686

[4] Phan, N.-T.; Nguyen, A.-D.; Liu, Y.-C.; Chiu, H.-J. An Investigation of Zero-Voltage-Switching Condition in a High-Voltage-Gain Bidirectional DC–DC Converter. Electronics 2021, 10, 1940. https://doi.org/10.3390/ electronics10161940

[5] Górecki, K.; Ptak, P.; Torzewicz, T.; Janicki, M. Influence of a Thermal Pad on Selected Parameters of Power LEDs. *Energies* 2020, *13*, 3732. https://doi.org/10.3390/en13143732

Detekcija i zaobilaženje prepreka pomoću razvojnog sistema Arduino

Jelena Nedeljković

Sadržaj – Savremen način života za sobom nosi povećano korišćenje digitalnih uređaja koji danas preuzimaju svaki aspekt našeg života. Kao posledica toga, na polju nauke se razvijaju tehnologije i uređaji koji za cilj imaju da nam olakšaju svakodnevni život. Jedna od aktivnosti koje mogu doprineti olakšavanju života ljudi jeste zaobilaženje prepreka. Mašine koje su programirane da zaobilaze prepreke mogu se koristiti u raznim industrijama. Cilj ovog rada jeste prikaz moguće primene sistema za detekciju i zaobilaženje prepreka. Praktičan doprinos ovog rada jeste detaljan opis jedne od mogućih implementacija ovog sistema.

I. UVOD

Kod samostalnih sistema, mogućnost detekcije i zaobilaženja prepreka je ključna osobina koja sprečava sistem da udari u prepreku. Dobar sistem će uvek minimizirati grešku i sprečiti uređaj da napravi štetu svom potencijalno jako skupom okruženju. Iz ovih razloga je dobar sistem za detekciju i zaobilaženje prepreka jako bitan.

Osnovna svrha ovakvog sistema je mogućnost realizacije autonomnih mašina.

Mašine koje se samostalno kreću su korisne pri izvršavanju fizičkih poslova koji bi ljudima oduzeli vreme i energiju. Primeri takvih poslova jesu premeštanje objekata u prodavnicama, fabrikama, usisavanje, itd. Detekcija i zaobilaženje prepreka se koriste u svim uređajima koji su autonomni, kao što su autonomna vozila ili uređaji za nošenje lekova u bolnicama.

Mehanizam opisan u ovom radu je baziran na razvojnom sistemu *Arduino*. Sam razvojni sistem sa svojim periferijama je postavljen na pokretnu platformu kojom upravlja.

Ovaj rad se sastoji iz pet poglavlja. Prvo poglavlje predstavlja uvod, drugo poglavlje sadrži opis komponenata koje su korišćene u ovom radu, u trećem poglavlju se nalazi opis realizovanog sistema, peto poglavlje daje opis makete izrađene za ovaj rad i poslednje poglavlje predstavlja zaključak.

II. IZBOR KOMPONENATA

A. Arduino

Arduino je mikrokontrolerska platforma razvijena 2005. godine u Italiji na Institutu za interaktivni dizajn u Ivreji. Projektovan je kao harver otvorenog koda (engl. *open source*). To znači da svako može da pravi pločice kompatibilne sa *Arduinom*. *Arduino* ploču čine 8-bitni Atmel AVR mikrokontroler sa komponentama koje omogućavaju programiranje i povezivanje sa drugom elektronikom [1]. Postoji veliki broj različitih modela ovih pločica. U ovom projektu je korišćen *Arduino Uno R3* koji je prikazan na slici 1.



Slika 1. Arduino Uno R3.

B. Arduino Motor Shield

Ukoliko korisnik želi da kontroliše motor pomoću *Arduino* razvojne ploče potrebna je neka vrsta kontrolera motora. Motor se ne može kontrolisati direktno putem pinova zato što motori zahtevaju visoku struju koju izlazni pinovi sa *Arduina* ne mogu da pruže. To je razlog korišćenja kontrolera motora. Jedan od kontrolera motora jeste i *Arduino Motor Shield* koji ima mogućnost da kontroliše različite tipove DC motora. Pošto je realizovan kao *Shield* za *Arduino* montira se tako što se odgovarajući pinovi povežu. *Arduino* pločica ima standardizovan raspored konektora koji omogućava lako povezivanje sa dodatnim modulima [1]. *Arduino Motor Shield* je baziran na kontroleru L293 koji je dizajniran da pokreće releje, DC motore, i slično [2]. Unutar *Arduino Motor Shield-a* se nalaze integrisana dva L293D kontrolera i jedan 74HC595 pomerački registar. Takođe, postoje i konektori za konekciju sa eksternim napajanjem.

U ovom projektu *Shield* je korišćen za kontrolu jednostavnog DC motora. Tačnije, korišćen je za istovremenu kontrolu dva DC motora. *Shield* pruža

J. Nedeljković je sa Katedre za elektroniku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: jelena.nedeljkovic@elfak.ni.ac.rs.

mogućnost manipulacije smera u kome se kreće motor kao i manipulacije brzine kojom se kreće.



Slika 2. Arduino Motor Shield.

C. Arduino DC motor

Elektromotori su uređaji koji pretvaraju električnu energiju u mehaničku. Elektromotori uglavnom rade na principu elektromagnetne indukcije ali postoje i motori koji koriste druge elektromehaničke fenomene, kao što su elektrostatička sila i piezoelektrični efekat [3]. Kada se kaže elektromotor uglavnom se misli na rotacioni elektromotor, tj. motor koji stvara obrtno kretanje.

U ovom projektu je korišćen jednosmerni motor. Glavna prednost ovih motora u odnosu na one koji se napajaju naizmeničnim veličinama je lakše upravljanje brzinom, momentom i smerom obtanja. DC motori se uglavnom prave za manje snage jer bi za veće snage bilo neophodno da imaju veće i jače izvore jednosmerne struje.

U ovom projektu su korišćena dva DC motora na kojima se nalaze točkovi kako bi bilo omogućeno pokretanje platforme.



Slika 3. DC motori sa točkovima.

D. Ultrazvučni senzor

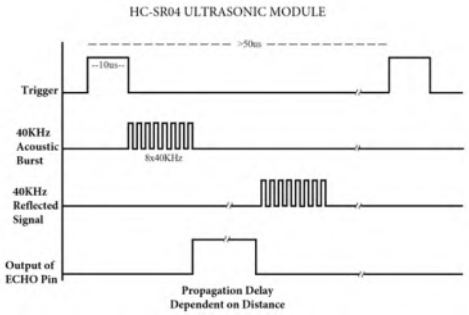
Ultrazvučni senzor se koristi za merenje rastojanja između senzora i objekta koji se nalazi ispred njega [4]. Ovaj senzor je jednostavan za upotrebu i nudi zadovoljavajuću preciznost. U ovom radu korišćen je senzor HC-SR04 koji je prikazan na slici 4. Rezolucija senzora je 0.3cm. Opseg detektovanja objekata je od 2 do 500 cm.



Slika 4. Ultrazvučni senzor HC-SR04.

Sam senzorski modul ima četiri pina: *VCC* (napajanje od 5V), *TRIG* (ulazni pin koji je zadužen za emitovanje frekvencije), *ECHO* (izlazni pin koji daje informaciju o povratnom talasu), *GND* (uzemljenje).

Senzor radi prema sledećem principu: prvo je potrebno dovesti senzorski impuls u trajanju od 10 μs na *TRIG* pin. Po završetku tog impulsa senzor šalje 8 ultrazvučnih impulsa frekvencije 40 kHz. U tom trenutku pin *ECHO* prelazi sa stanja logičke nule na stanje logičke jedinice i to stanje zadržava sve dok se reflektovani impulsi ne vrate nazad do senzora. Merenjem trajanja *ECHO* impulsa i poznavanjem brzine zvuka može se izračunati rastojanje između senzora i predmeta. Princip rada senzora je prikazan na slici 5.



Slika 5. Princip rada ultrazvučnog senzora.

Vreme trajanja *ECHO* impulsa je vreme koje je potrebno da ultrazvučni talas ode do predmeta čije se rastojanje meri i da se vrati nazad do senzora. Ukupno rastojanje koje se dobije potrebno je podeliti sa 2 kako bi se dobilo rastojanje između senzora i predmeta, kao što je prikazano u jednačini 1:

$$Distance = (time\ taken * speed\ of\ sound) / 2 \quad (1)$$

III. PRINCIP RADA

Sistem opisan u radu se sastoji iz sledećih elemenata:

A. Podsistem za detekciju prepreka pomoću ultrazvučnog senzora;
B. Podsistem za kontrolu motora i smera kretanja;
C. Podsistem za uključivanje LED i zujalice.

A. Podsistem za detekciju prepreka pomoću ultrazvučnog senzora

Funkcija ovog podsistema je detekcija prepreka pomoću ultrazvučnog senzora. Ultrazvučni senzor se koristi za merenje rastojanja između senzora i predmeta koji se nalazi ispred njega. Senzor koji je korišćen pri izradi ovog rada jeste HC – SR04 rezolucije 0.3 cm i opsega detektovanja objekata od 2 do 500 cm koji je detaljno objašnjen u prethodnom poglavlju. Nakon detekcije prepreke pokreće se niz aktivnosti u cilju zaobilaska detektovanog objekta.

B. Podsistem za kontrolu motora i smera kretanja

U cilju pokretanja platforme u željenom smeru relizovan je podsistem za kontrolu motora i smera kretanja. Ovaj podsistem ima ulogu pokretanja platforme u pravcu koji zavisi od činjenice da li je neka prepreka detektovana ispred sistema. Ukoliko je objekat detektovan proverava se da li u okolini ima još objekata. Ta aktivnost se obavlja pomoću *Servo* motora i ultrazvučnog senzora koji je fiksiran na *Servo* motor. Time je omogućeno da se proverava stanje udaljenosti prepreka u okolini. Kada se utvrdi na kojoj strani se nalazi najudaljeniji objekat, podsistem za kontrolu motora i smera kretanja će usmeriti sistem ka toj strani. Ovaj proces se nastavlja sve vreme dok postoji napajanje.

C. Podsistem za uključivanje LED i uključivanje zujalice

U cilju signalizacije kada je neka prepreka blizu a kada nije koriste se dve LED, zelena i crvena, i jedna zujalica. Ukoliko nije detektovan nijedan objekat u blizini uključena je zelena LED, a ukoliko je detektovan neki objekat uključene su crvena LED i zujalica.

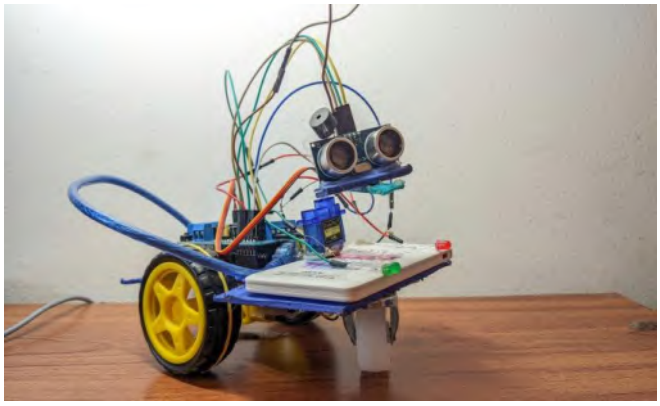
IV. MAKETA I ALGORITAM

Maketa koja je realizovana za ovaj rad ima cilj da dočara primenu sistema za detekciju i izbegavanje prepreka. U nastavku teksta, na slici 6, je prikazana fotografija makete koja je izrađena za ovaj rad.

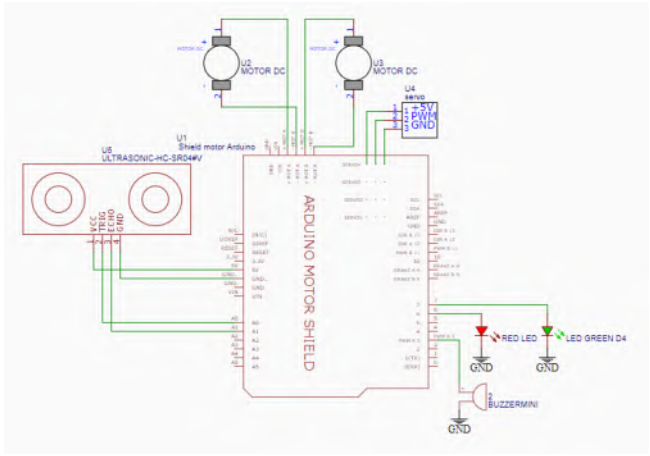
U nastavku teksta, na slici 7, dat je prikaz električne šeme realizovanog sistema. Na slici 8 je prikazan algoritam.

V. ZAKJUČAK

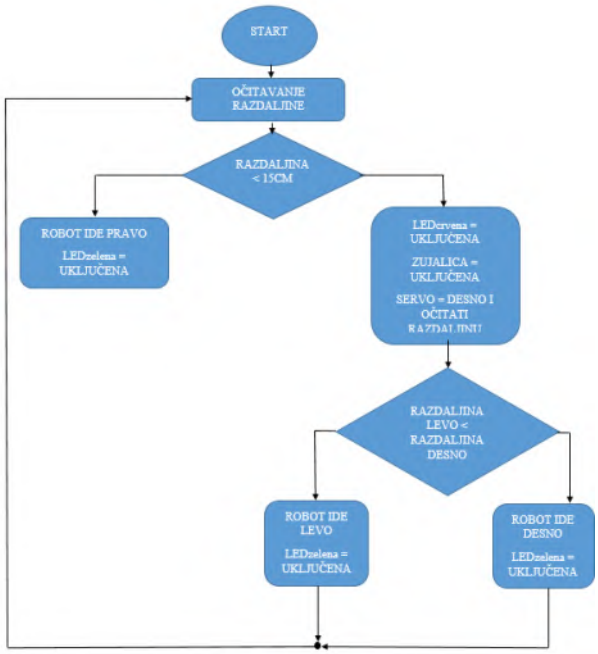
U modernom dobu se teži ka olakšavanju ljudskih života pomoću moderne tehnologije. Iz tog razloga se aktivno radi na realizaciji novih uređaja koji mogu olakšati živote ljudi. Tehnologija za izbegavanje prepreka je tome dosta doprinela. Pomoću ove tehnologije realizovani su pametni usisivači, pametne mašine za prenošenje stvari, itd. Ovakvi uređaji su doprineli u olakšavanju svakodnevnih života velikom broju ljudi.



Slika 6. Maketa.



Slika 7. Električna šema.



Slika 8. Algoritam.

ZAHVALNICA

Zahvaljujem se prijateljima i kolegama koji su mi ukazali na određene nepravilnosti prilikom pisanja ovog rada.

LITERATURA

[1] Arduino. Dostupno na: https://sr.m.wikipedia.org/sr/Arduino
[2] Motor Shield Rev3. Dostupno na: https://docs.arduino.cc/hardware/motor-shield-rev3
[3] Elektromotor. Dostupno na: https://sr.m.wikipedia.org/sr-ec/%D0%95%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BC%D0%BE%D1%82%D0%BE%D1%80
[4] Tehnička dokumentacija za ultrazvučni sensor HCSR04. Dostupno na: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf
[5] S.Fitzgerald and M.Ahiloh, *The Arduino project book*, Septembar 2012.
[6] M. Margolis, *Arduino Cookbook*, March 2011.
[7] W.Brand, *Arduino for dummies*
[8] D. Molloy, *Raspberry Pi*, New York, 2016.

CAN komunikacioni protokol

Đorđe Lazarević, Marko Živković, Đorđe Kocić, Jovan Ćirić

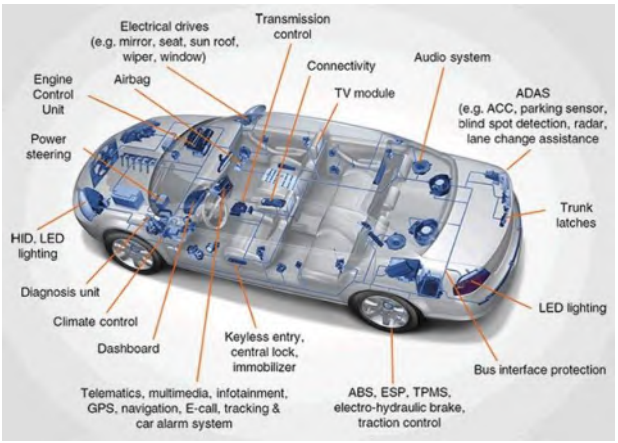
Sadržaj – U ovom radu je prikazana funkcionalnost CAN komunikacionog protokola i princip rada istog uz praktičnu realizaciju sistema za rad sa CAN komunikacionim protokolom. Takođe su prikazani i paketi podataka poslati sa čvorova na logičkom analizatoru sa dekodiranim frejmovima između primopredajnika kao i diferencijalni signali na osciloskopu.

I. UVOD

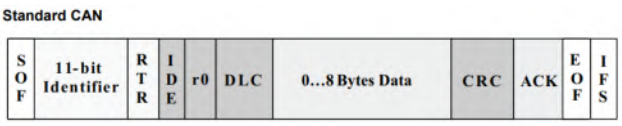
CAN (*Control Area Network*) magistrala podataka je robusna magistrala, razvijena 1983. godine u kompaniji *Bosch*, za prenos podataka. Standardno je dizajnirana da omogući međusobnu komunikaciju mikrokontrolera i drugih komponenti ili delova sistema bez *host* računara. To je protokol baziran na porukama projektovan originalno za multipleksiranje provodnika u sklopu automobila i glavna svrha mu je ušteda bakra, ali može biti korišćen i u drugom kontekstu. Za svaku komponentu odnosno uređaj frejm podatka je poslat sekvencijalno, ali na takav način da ukoliko više od jednog uređaja šalje podatke u isto vreme uređaj sa najvećim prioritetom može da nastavi dok se drugi isključuju. Frejmovi se šalju i primaju na svim uređajima u okviru mreže. Primena CAN protokola je najzastupljenija u automobilima odnosno u automobilskoj industriji, dok se takođe primenjuje u industrijskoj automatici, liftovima, medicinskoj instrumentaciji, itd. Moderni automobili imaju više od 70 kontrolnih jedinica za različite podsisteme. Tipično je testni čvor jedinica za kontrolu motora. Dok su ostali čvorovi transmisija, *airbag*-ovi, servo, kontrola protoka goriva, ABS i ostali [1]. Na slici 1 je prikazan skup senzora i kontrolnih jedinica koji komuniciraju preko CAN magistrale podataka.

II. CAN PROTOKOL

CAN magistrala je *multimaster*, *message-broadcast* sistem koji podržava brzinu prenosa podataka od 1Mbit/s. Za razliku od USB ili *Ethernet*-a, CAN ne omogućava slanje velikih blokova podataka od tačke do tačke (*point-to-point*) od čvora A do čvora B pod nadzorom centralnog mastera. CAN komunikacioni protokol je osluškivač nosioca (*carrier-sense*), protokol sa višestrukim pristupom sa detekcijom kolizije i odlučivanjem o prioritetu poruke (CSMA/CD+AMP). CSMA znači da svaki čvor na magistrali mora da čeka propisano vreme neaktivnosti pre nego što pokuša da pošalje poruku. CD+AMP znači da su kolizije rešene kroz inteligentno definisanje bita i bazirana je na preprogramiranom prioritetu svake poruke u polju identifikatora poruke. Identifikator sa najvišim prioritetom uvek prvi pristupa magistrali. To je poslednja logička 1 u identifikatoru sadržana u prenosu koja je na najvišem prioritetu. Svaki čvor na magistrali učestvuje u upisivanju svakog bita po principu „kao što je napisano“ a odlučujći čvor prepoznaje ako postoji bit logičke 1 na magistrali. Standard sa 11-bitnim identifikatorom, omogućava brzine od 125 kb/s do 1 Mb/s. Standard je kasnije dopunjen sa produženim 29-bitnim identifikatorom. Standardni 11-bitni identifikator obezbeđuje 2^{11} ili 2048 različitih identifikatora poruka, dok 29-bitni identifikator obezbeđuje 2^{29} ili 537000000 različitih identifikatora. Na slici 2 je prikazan *standard* CAN sa 11-bitnim identifikatorm dok je na slici 3 prikazan *standard extended* CAN sa 29-to bitnim identifikatorom.



Slika 1. Kontrolne jedinice automobila povezane preko CAN magistrale.

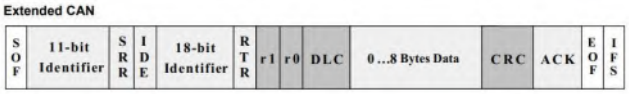


Slika 2. Standard CAN sa 11-bitnim identifikatorom.

Đ. Lazarević, student treće godine doktorskih studija na modulu Nanotehnologije i mikrosistemi, Elektronskog fakulteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: djordje.lazarevic@elfak.ni.ac.rs.

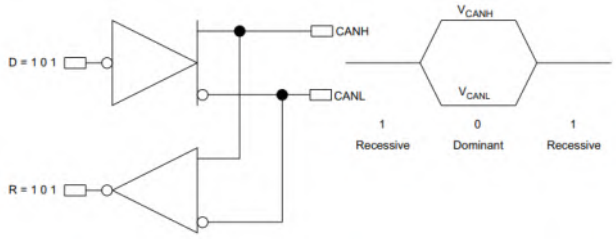
M. Živković, Đ. Kocić, studenti treće godine doktorskih studija na modulu Elektroenergetika, Elektronskog fakulteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: marko.m.zivkovic@elfak.ni.ac.rs, seriousdjoka@gmail.com.

Dr J. Ćirić, dipl. inž. tehnologije, naučni saradnik, Toplička akademija strukovnih studija, Odsek za poljoprivredno-prehrambene studije, Ćirila i Metodija 1, 18400 Prokuplje, Srbija, E-mail: joca.ciric@gmail.com.



Slika 3. Extended CAN standard sa 29-bitnim identifikatorom.

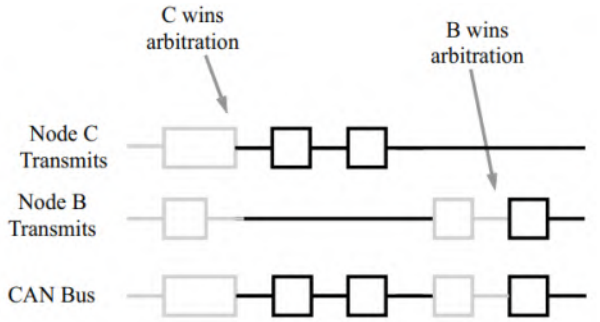
Fundamentalna karakteristika CAN-a je inverzno logičko stanje na magistrali, i ulaza drajvera i izlaza prijemnika. U normalnim okolnostima, logička 1 je pridružena logičkoj 1 i logička 0 je pridružena logičkoj 0, ali nije to nije slučaj na CAN magistrali. Razlog zašto CAN primopredajnici imaju pinove ulaza drajvera i izlaza prijemnika pasivno povezane na nivo logičke 1 je taj da u slučaju odsustva bilo kog ulaza, komponenta automatski vraća stanje na podrazumevanu vrednost kod svih ulaznih i izlaznih pinova. Na slici 4 je prikazana inverzna logika CAN magistrale.



Slika 4. Inverzna logika CAN magistrale.

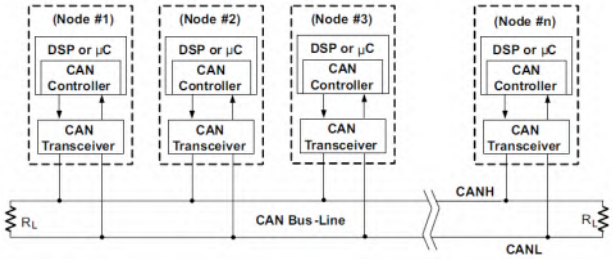
Dodela prioriteta poruka u identifikatoru je prednost CAN-a koja ga čini pogodnim za korišćenje u sklopu *real-time* kontrolnih okruženja za rad u realnom vremenu. Što je niži broj identifikatora binarne poruke, viši je prioritet. Identifikator koji se sastoji u potpunosti od nula je najviši prioritet poruke u mreži zato što je on dominantan na magistrali najduže. Stoga ako dva čvora počnu slanje istovremeno, čvor koji pošalje zadnji bit identifikatora 0 je dominantan dok ostali čvorovi koji šalju 1 su recesivni. Čvor sa identifikatorom sa zadnjim bitom 0 zadržava kontrolu na CAN magistrali i završava svoju poruku. Dominantni bit uvek obori recesivni bit na CAN magistrali. Čvor koji šalje ujedno nadgleda svaki bit sopstvenog prenosa. Ovo je razlog zašto se CANH i CANL izlazni pinovi drajvera interno povezuju na ulaze prijemnika. Propagaciono kašnjenje signala u internoj petlji od ulaza drajvera do izlaza prijemnika je tipično korišćeno za kvalitativno merenje CAN primopredajnika. Ovo propagaciono kašnjenje se odnosi na vreme petlje (t_{LOOP}).

CAN arbitraža procesa je upravljana automatski od strane CAN kontrolera. Zato što svaki čvor kontinualno nadgleda svoji sopstveni prenos, ako je recesivni bit čvora B oboren dominantnim bitom čvora C, B detektuje da se stanje magistrale ne poklapa sa bitom koji je poslat. S toga čvor B prekida predaju dok čvor C nastavlja sa slanjem svoje poruke bez prekidanja. Još jedan pokušaj slanja poruke je načinjen od strane čvora B jednom kada magistrala bude oslobođena od strane čvora C. Ova funkcionalnost je sadržana u potpunosti u CAN kontroleru i kompletno dostupna korisniku CAN-a. Na slici 5 je prikazana arbitraža na CAN magistrali.



Slika 5. Arbitraža na CAN magistrali.

Veza podataka i fizički signalni slojevi prikazani na slici 6. su normalno vidljivi operatoru sistema i uključeni su u mnogim kontrolerima koji implementiraju CAN protokol. Na slici 6 je prikazan način povezivanja više čvorova za komunikaciju preko CAN magistrale.

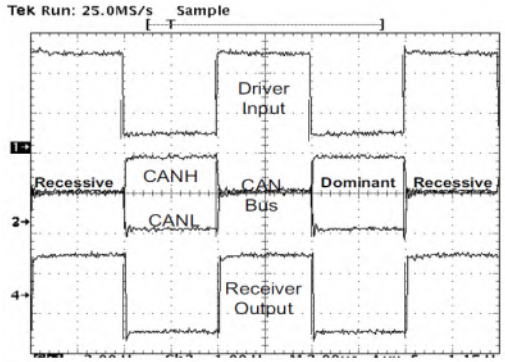


Slika 6. Povezivanje više čvorova na CAN magistralu podataka.

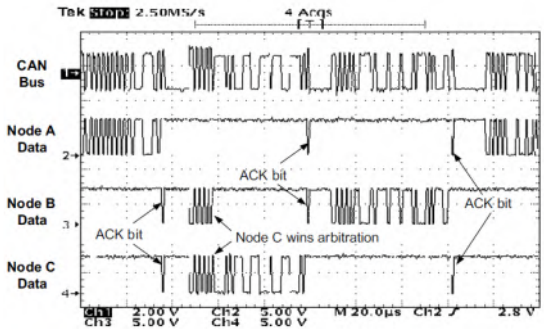
Signali na magistrali su diferencijalni zbog čega CAN magistrala poseduje veliku imunost na smetnje i toleranciju grešaka. Balansirani diferencijalni signali na obe linije redukuju superponiranje šuma i dozvoljavaju velike brzine signala preko upredene parice (kabla sa upredenom paricom). Balansiranje diferencijalnim signalima podrazumeva da je struja svakog signala odnosno obe signalne linije identična samo u suprotnom smeru i samim tim su smetnje automatski poništene što je i ključ malog emitovanja smetnji. Korišćenje balansiranih diferencijalnih prijemnika i upredene parice poboljšavaju potiskivanje zajedničkog signala i veliku imunost na smetnje na CAN magistrali.

Dve signalne linije na magistrali , CANH i CANL u mirnom recesivnom stanju su pasivno prespojene na 2.5 V. Dominantno stanje na magistrali postavlja CANH~1V na nivo od 3.5 V, postavlja CANL~1V na nivo od 1.5 V, praveći tipično 2 V razlike signala [2]. Razlika CANH i CANL signala kao i sami signali su prikazani na slici 7.

Na slici 8. je prikazana međusobna komunikacija čvorova.



Slika 7. CANH i CANL linija – recesivno i dominantno stanje.



Slika 8. Međusobna komunikacija čvorova na CAN magistrali.

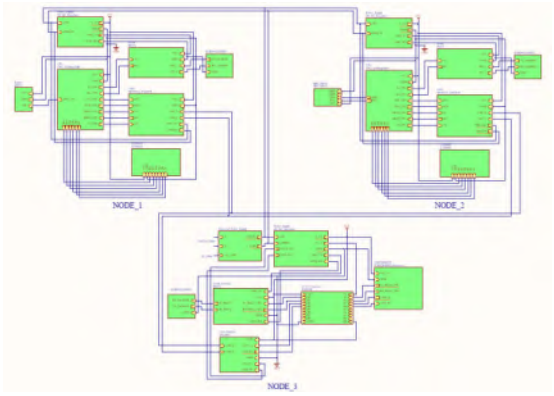
III. PRAKTIČNA REALIZACIJA SISTEMA ZA ANALIZU CAN PROTOKOLA

Sistem za analizu CAN komunikacionog protokola je projektovan u softveru Altium Designer [3] tj. električna šema i štampana ploča sistema, dok je simulacija rada pojedinih delova sistema izvršena u softveru Proteus [4]. Razvoj softvera za mikrokontroler ATMega328P [5] je izvršen u softverskom okruženju Microchip Studio [6] dok je razvoj softvera za STM32F429ZI Discovery [7] razvojno okruženje realizovan u STM32CubeIDE [8] okruženju. Sistem se sastoji iz 3 CAN čvora pri čemu čvor 1 i čvor 2 vrše merenje i slanje podataka preko CAN protokola dok 3. čvor predstvalja čvor za nadgledanje ili test čvor koji prati događaje na magistrali i vrši prikaz izmerenih veličina na grafičkom LCD displeju. Mikrokontrolerski sistem koji predstavlja čvor 1 u ovom slučaju vrši merenje temperature i vlažnosti vazduha pomoću DHT11 [9] senzora dok se ti podaci sa senzora preuzimaju preko mikrokontrolera ATMega328P i dalja obrada i slanje podataka se vrše preko CAN kontrolera MCP2515 [10] i primopredajnika TJA1050 [11].

Čvor 2 predstavlja identičan sistem sa mikrokontrolerom ATMega328P koji vrši merenje ambijentalnog osvetljenja pomoću senzora BHT1750 [12] i dalje preko MCP2515 i TJA1050 šalje podatke na CAN magistralu.

Testni čvor kontroler odnosno razvojno okruženje 32F429IDISCOVERY sa 32-bitnim ARM mikrokontrolerom STM32F429ZIT6 [13] predstavlja čvor 3 na kome se vrši obrada podataka primljenih sa CAN magistrale i nadgledanje svih čvorova koji su povezani na CAN magistralu. Za prijem podataka je isključivo korišćen modul sa CAN primopredajnikom TJA1050 iz razloga što ARM mikrokontroler poseduje integrisani CAN kontroler tako da MCP2515 na strani prijema nije neophodan. Analiza CAN protokola odnosno signala sa Tx i Rx pinova integrisanog kola TJA1050 je vršena uz pomoć logičkog analizatora u vidu minijaturne akvizijske kartice. Fizički signali na CANH i CANL liniji su analizirani pomoću digitialnog osciloskopa. Napajanje sistema je realizovano uz pomoć industrijskog napajanja 12VDC dok je na ploči realizovan DC - DC regulator koji reguliše napon na 5 V i omogućava napajanje svih delova sistema.

Sistem za komunikaciju mikrokontrolera sa periferijama primenom CAN protokola se sastoji iz 3 čvora. Čvor 1 i čvor 2 su sistemi bazirani na 8-bitnom mikrokontroleru ATMega328P dok je test čvor kontroler 32-bitni mikrokontroler STM32F429ZIT. Kompletan blok dijagram povezivanja čvorova je prikazan na slici 9.

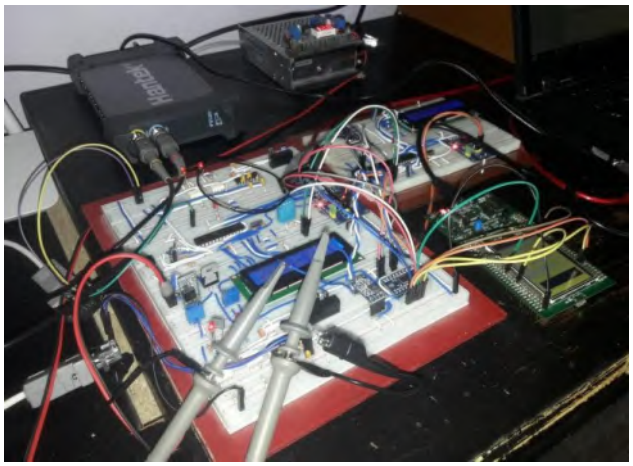


Slika 9. Blok dijagram povezivanja čvorova na CAN magistrali.

IV. IZRADA PROTOTIPA I PROVERA RADA SISTEMA

Za izradu prototipa sistema koji se koristi za analizu i prikaz rada CAN komunikacionog protokola su korišćene protoploče. Za povezivanje razvojnog okruženja sa protopločama su korišćeni izolovani kablovi namenjeni za ispitivanje elektronskih kola na protoploči. Snimanje komunikacionih linija Tx i Rx na CAN primopredajnicima je izvršeno korišćenjem 8-kanalnog Saleae [14] logičkog analizatora. Za analizu analognih vrednosti napona na CANH i CANL liniji je korišćen digitalni osciloskop proizvođača Hantek u vidu akvizicijske kartice sa sondama povezanim direktno na izlazima kola TJA1050. Izlazni naponi na CANH i CANL liniji su u opsegu do 5 V, tačnije 3.5 V na CANH izlazu i oko 1.5 V na CANL izlazu, za dominantni bit, koji su diferencijalni i u mnogo većem procentu imuniji na smetnje u odnosu na tipičan serijski prenos preko RS232 interefejsa. Softver koji se koristi za prikupljanje i analizu podataka sa osciloskopa je Hantek 6022BE [15], dok je za akvizicjsku karticu analizatora

korišćen softver Logic 2.3.3 [16]. Brzina CAN komunikacije je podešena na maksimalnu vrednost odnosno 1 MBaud tj približno 1 Mbit/s. Identifikatori poruka koje se šalju na CAN magistrali su 0x036 sa čvora 1 koji vrši slanje vrednosti temperature i vlažnosti vazduha i 0x037 sa čvora 2 koji šalje vrednost ambijentalnog osvetljenja direktno u jedinici lux. Dok testni čvor kontroler odnosno čvor 3 ima identifikator poruke 0x030 i on služi za prijem i obradu podataka sa magistrale. Prioritet slanja poruke na magistrali se određuje nakon arbitraže i čvor sa identifikatorom 0x036 uvek šalje prvi jer čvor sa 0x030 uglavnom služi kao prijemnik mada će on uvek imati prioritet ukoliko je potrebno poslati neki kontrolni frejm ili poruku. Na slici 10 je prikazano povezivanje sistema za analizu CAN protokola.



Slika 10. Povezivanje sistema za analizu CAN komunikacionog protokola.

Slanje poruka na CAN magistrali počinje spuštanjem Tx linije na LOW pri čemu nastupa arbitraža i određivanje dominantnog i recesivnog čvora na mreži. S obzirom da u ovom slučaju imamo dva čvora koji vrše slanje podataka na magistrali uvek dobija dominantno stanje onaj sa nižim binarnim identifikatorom poruke. Na slici 11 je prikazan identifikator poruke 0x036 dok je na slici 12 poruka sa identifikatorom 0x036 odnosno poruka koju šalje čvor 1.



Slika 11. Identifikator 0x036 u poruci sa čvora 1.



Slika 12. Poruka sa čvora 1 na CAN magistrali.

Čvor 1 je u arbitraži prešao u dominantno stanje i prvi slao poruku na magistrali dok je čvor 2 postao recesivan. Nakon završenog slanja poruke sa čvora 1 nastupa slanje poruke sa čvora 2. Identično kao i kod čvora 1 slanje kreće sa SOF bitom odnosno promenom logičkog stanja magistrale sa HIGH na LOW. Binarni identifikator čvora 2 ima hex vrednost 0x037. Na slici 13 je prikazan binarni identifikator 0x037 poruke sa čvora 2 dok je na slici 14 prikazana poruka sa čvora 2.

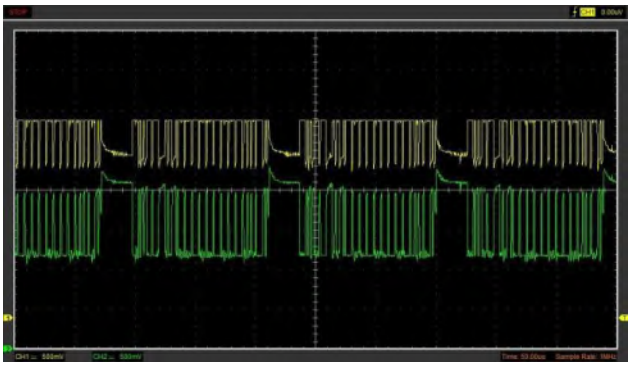


Slika 13. Indentifikator 0x037 u poruci sa čvora 2.



Slika 14. Poruka sa čvora 2 na CAN magistrali.

Za analizu analognih vrednosti napona na izlazu primopredajnih modula se koristi digitalni osciloskop. Na slici 15 je prikazan diferencijalni signal na CAN magistrali.



Slika 15. Diferencijalni signal na CAN magistrali.

V. Zaključak

Primena CAN protokola je velika u medicinskoj instrumentaciji i industrijskoj elektronici. Svuda gde je potreban komunikacioni protokol koji omogućava smanjivanje nivoa smetnji, robusnost, veliku brzinu i fleksibilnost je moguće koristiti CAN protokol. U ovom radu su prikazani osnovni principi CAN protokola i uspešno slanje i prijem podataka. Moguće je dalje unapređenje sistema i povezivanje sa drugim vrstama žičanih interfejsa kao što je i slučaj u realnim sistemima.

Zahvalnica

Autori se zahvaljuju dipl. inž. Milošu Marjanoviću na korisnim informacijama i sugestijama prilikom izrade ovog rada.

Literatura

[1] History of CAN technology. Dostupno na: https://www.can-cia.org/can-knowledge/can/can-history/.

[2] Introduction to the Controller Area Network. Dostupno na: https://www.ti.com/lit/an/sloa101b/sloa101b.pdf?ts=1592815338549&ref_url=https%253A%252.
[3] Altium Designer. Dostupno na: https://www.altium.com/altium-designer
[4] Proteus, PCB Design & Simulation. Dostupno na: https://www.labcenter.com/
[5] Microcontroller ATMega328P, datasheet. Dostupno na: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf.
[6] Microchip Studio . Dostupno na: https://www.microchip.com/en-us/tools-resources/develop/microchip-studio
[7] STM32F429ZI Discovery, Discovery kit with STM32F429ZI MCU. Dostupno na: https://www.st.com/en/evaluation-tools/32f429idiscovery.html
[8] STM32CubeIDE, Integrated Development Environment for STM32. Dostupno na: https://www.st.com/en/development-tools/stm32cubeide.html
[9] DHT11 datasheet. Dostupno na: https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf.
[10] MCP 25115, Stand-Alone CAN Controller with SPI Interface. Dostupno na: https://ww1.microchip.com/downloads/en/DeviceDoc/MCP2515-Stand-Alone-CAN-Controller-with-SPI-20001801J.pdf.
[11] TJA1050 datasheet. Dostupno na : https://www.nxp.com/docs/en/data-sheet/TJA1050.pdf.
[12] BHT1750 datasheet. Dostupno na : https://www.mouser.com/datasheet/2/348/bh1750fvi-e-186247.pdf.
[13] STM32F429ZIT6 microcontroller datasheet. Dostupno na: https://www.st.com/resource/en/datasheet/dm00071990.pdf.
[14] Saleae, Logic analyzers. Dostupno na: https://www.saleae.com/?gclid=EAIaIQobChMIn5eC2Jr5-gIVUAGLCh3ggQDtEAAYASAAEgKlwvD_BwE
[15] Hantek 6002BE Series. Dostupno na: http://www.hantek.com/products/detail/31
[16] Logic 2.3.3. Dostupno na: https://discuss.saleae.com/t/logic-2-3-3-now-with-protocol-trigger/634

Kontrola brzine okretanja DC motora u zavisnosti od izmerenog rastojanja

Anastasija Avramović, Strahinja Dimitrijević

Sadržaj – U radu je realizovan sistem koji uz pomoć *Nucleo* razvojnog okruženja kontroliše brzinu okretanja DC motora u zavisnosti od rastojanja koje se meri ultrazvučnim senzorom. Pri rastojanju većem od 100 cm, motor se okreće maksimalnom brzinom. Brzina motora se smanjuje proporcionalno rastojanju u opsegu od 100 cm do 50 cm. Ukoliko je rastojanje manje od 50 cm, motor treba da se zaustavi i uključi svetlosno upozorenje. Kada se rastojanje poveća motor se ponovo pokreće.

I. Uvod u komponente snage

Svaka elektronska komponenta koja može pouzdano da vodi struju jaču od 1 A može se smatrati snažnom komponentom ili komponentom snage. Bipolarni tranzistor otkriven je 1947. godine, a prve komercijalne bipolarne komponente, diode i tiristori, sposobne da rade na visokim nivoima struje i napona, pojavile su se na tržištu ranih 1950-tih godina. Tehnologija bipolarnih komponenata snage usavršena je i dominirala tokom narednih tridesetak godina dvadesetog veka, pri čemu su razvijene i veoma snažne komponente koje su u stanju da vode struju od preko 10 kA i izdrže napone iznad 4 kV. U međuvremenu, razvijane su i nove familije diskretnih komponenata snage, od kojih je najznačajniji MOS tranzistor snage, razvijen 1970-tih godina. Za razliku od bipolarnih komponenata, MOS tranzistor je naponski kontrolisana komponenta visoke ulazne impedanse, koja omogućava da upravljačka kola budu mnogo jednostavnija, što u velikoj meri smanjuje cenu snažnih elektronskih sistema, čineći ih atraktivnim za brojne nove aplikacije, kao što su uređaji u domaćinstvu, automobilska elektronika, itd [1].

Osim MOS (*Metal Oxide Semiconductor*) tranzistora, razvijane su i druge komponente snage, kao što su statički indukcioni tranzistor i tiristor (SIT - *static induction* transistor i SITh - *static induction thyristor*), JFET (*junction field-effect transistor*), bipolarni tranzistor sa izolovanim gejtom IGBT (*Insulated Gate Bipolar Transistor*), MOS kontrolisani tiristor MCT (MOS-*controlled thyristor*) itd. Od ovih komponenata, najznačajniji je IGBT, u kome su sa uspehom objedinjene dobre osobine tehnologija MOS i bipolarnih tranzistora (visoka impedansa i naponska kontrola na ulazu, kao i velika brzina MOS tranzistora sa visokom provodnošću bipolarnog tranzistora).

A. Avramović i S. Dimitrijević su studenti osnovnih studija na modulu Elektronske komponente i mikrosistemi, Elektronski fakultet, Univerzitet u Nišu. Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: anastasijaavram@elfak.rs, strax.dimitrijevic@elfak.rs.

U ovom radu biće predstavljen sistem za kontrolu brzine okretanja DC motora koji se upravlja integrisanim snažnim kolom, drajverom za induktivna opterećenja L293D. DC motori su važni u procesu konverzije energije. Njihov osnovni zadatak je da pretvaraju električnu energiju u mehaničku energiju. Brzina DC motora može se kontrolisati pomoću napona napajanja, promenom fluksa ili korišćenjem tiristora. Pored toga, DC motori nude preciznost koja se zahteva u industriji.

II. Korišćene komponente za realizaciju projekta

A. Nucleo STM32F103RB

Nucleo je razvojno okruženje koje omogućava korisnicima izradu prototipova uz pomoć STM32 mikrokontrolera [2]. *Nucleo* okruženje prikazano je na slici 1.



Slika 1. Nucleo okruženje.

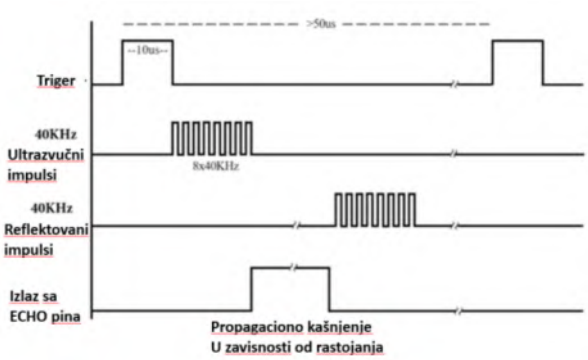
B. Ultrazvučni senzor

Ultrazvučni senzor daljine se koristi za merenje rastojanja između senzora i objekta koji se nalazi ispred njega [3]. Ovaj senzor nudi dobru preciznost i jednostavan je za upotrebu. U ovom radu koristiće se senzor HC-SR04 prikazan na slici 2. Rezolucija senzora je 0.3 cm, a opseg detektovanja objekta od 2 do 500 cm. Pored ovih komponenata korišćene su: DC motor, OLED (*organic light-emitting diodes*) displej, otpornik od 220 Ω, LED diode i baterija od 9 V.



Slika 2. Ultrazvučni senzor HC-SR04.

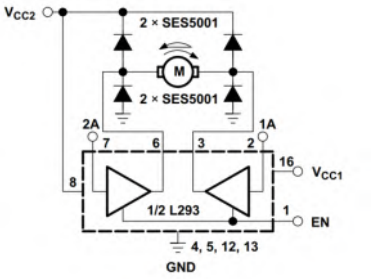
Princip rada senzora je sledeći: najpre je potrebno dovesti na TRIGGER pin senzora naponski impuls u trajanju od 10 µs. Nakon završetka tog impulsa, senzor šalje 8 ultrazvučnih impulsa frekvencije 40 kHz. U tom trenutku, pin ECHO menja vrednost na logičku jedinicu, i takav ostaje sve dok se reflektovani impulsi ne vrate nazad do senzora. Merenjem trajanja ECHO impulsa može se izračunati rastojanje, uz poznavanje brzine zvuka. Princip rada prikazan je na slici 3. Ukupno rastojanje se dobija kao (*vreme·brzina zvuka*)/2.



Slika 3. Princip rada ultrazvučnog senzora.

C. L293D

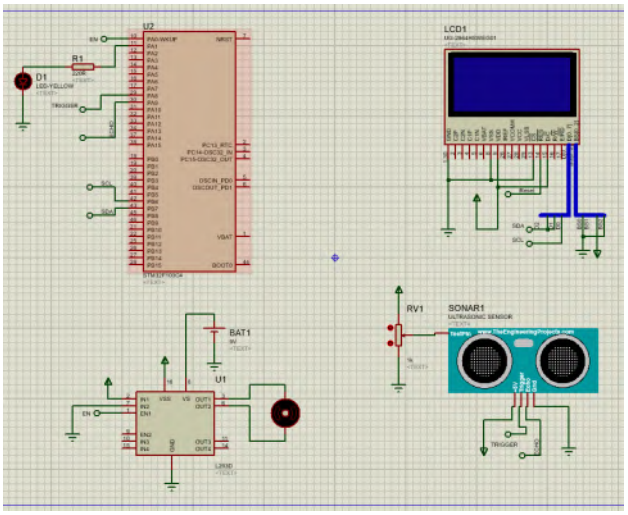
Integrisano kolo L293D predstavlja drajver za induktivna opterećenja. Na njemu se mogu povezati dva DC motora, ili jedan, uz mogućnost promene smera okretanja. Za potrebe ovog projekta, iskorišćenja je bidirekciona šema povezivanja koja je izvedena iz njegove tehniče dokumentacije [4] prikazana na slici 4. Kontrola smera se vrši dovođenjem logičke nule ili jedinice na pinove 2A i 1A. Ako je pin 2A HIGH, a 1A LOW, motor se okreće u jednom smeru, a zamena vrednosti uzrokuje okretanje u drugom smeru.



Slika 4. Šema za povezivanje IK L293D sa DC motorom.

III. Opis šeme sistema

Za projektovanje električne šeme korišćen je softverski paket *Proteus*. Električna šema prikzana je na slici 5. Centralni deo sistema je razvojno okruženje *Nucleo* sa STM32F103 mikrokontrolerom [2]. OLED [5] se fizički povezuje na razvojno okruženje, I2C protokol služi za komunikaciju. Na šemi su prikazane oznake linija ove veze (SDA i SCL). Displej služi za prikazivanje rastojanja. Na razvojno okruženje je povezan i ultrazvučni senzor [3] (linije TRIGGER i ECHO). Kao indikator da je rastojanje manje od 50 cm, koristi se LE dioda, povezana na pin PA1 preko otpornika od 220 Ω, koji će ograničiti struju kroz diodu. Za pokretanje motora motora koristiće se kolo L293D [4], prvi pin (EN) se povezuje na PA0, tu se dovodi PWM signal. Drugi i četvrti pin zaduženi su za definisanje smera okretanja motora. Pin 2 je povezan na napajanje, a pin 4 na masu, što znači da će se motor okretati u jednom smeru. Osmi pin je povezan na bateriju od 9 V, dok je 16. pin povezan na VCC. Napon na 8. pinu obezbeđuje potrebnu struju za pokretanje DC motora. Zbog toga je na taj pin povezana baterija od 9 V. Napon na pinu 16 služi za interno napajanje integrisanog kola, pa je tu dovedeno stabilnih 5 V sa razvojnog okruženja.



Slika 5. Električna šema sistema.

IV. Podešavanja u CubeMx

CubeMx je grafički alat koji omogućava vrlo lako konfigurisanje STM32 mikrokontrolera i generisanje kompletnog C projekta za: IAR, KEIL, AC6, ATOLLIC, itd. Pošto ultrazvučni senzor šalje podatke u µs, potrebno je napraviti funkciju za mikrosekunde. O funkciji će biti reči u daljem tekstu. Koristi se Timer 4, u *CubeMX*. Potrebno je dozvoliti *interni clock* koji je podešen na 64 MHz. Zatim podesiti *prescaler* na 64, da bi takt *timer-a* bio 1 MHz. To znači da će tajmer inkrementirati na svakih 1 µs. ARR je podešen na maksimalnu moguću vrednost, ova vrednost

predstavlja granicu do koje se može računati u mikrosekundama. Pinovi PA8 (TRIGGER) i PA9 (ECHO) su povezani na senzor. Razlog ovakvog podešavanja tajmera biće objašnjen u nastavku.

Sledeća podešavanja odnose se na tajmer *TIM2*, koji će generisati PWM potreban da bi se menjala brzina okretanja motora. S obzirom da se koristi interni oscilator frekvencije 64 MHz, potrebno je tu vrednost spustiti na 10 kHz, što je optimalna frekvencija za DC motor. Za postizanje te vrednosti potrebno je uneti u prescaler vrednost 64 - 1, a u ARR uneti vrednost 100. Na taj način, konfigurisan je tajmer čija je frekvencija 10 kHz. Takođe, potrebno je podesiti *Channel 1* na vrednost *PWM Generator CH1*. U polju *Mode* potrebno je izabrati vrednost *PWM mode 1*. Postavljen je inicijalni faktor ispune u polju *Pulse* na 25, ali ovo nije od ključnog značaja, jer će se u toku izvršenja programa menjati vrednost faktora ispune.

Na kraju, za svetlosnu signalizaciju, postavljen je pin PA1 kao izlazni. Na ovom pinu biće prosleđen visok naponski nivo onog trenutka kad se okretanje motora zaustavi. Na pin PA1 povezana je dioda sa otpornikom od 220 Ω.

V. KOD U *KEIL-U*

Keil MDK je kompletno okruženje za razvoj softvera za niz ARM Cortex-M baziranih mikrokontrolerskih uređaja. U ovom projektu jedan deo koda odnosi se na deklaraciju promenljivih i inicijalizaciju tajmera. Drugi deo koda namenjen je ultrazvučnom senzoru. U okviru toga definisana je funkcija koja će praviti *delay* u mikrosekundama (pošto ne postoji predefinisana fukncija za kašnjenje u mikrosekundama; funkcija *HAL_DELAY()* za argumente uzima milisekunde), a unutar *USER CODE 3* opisan je rad senzora. Treći deo koda namenjen je uspostavljanju veze između promenljive koja čuva rastojanje i PWM-a koji se šalje DC motoru. Kompletno objašnjen kod može se videti na sledećem linku https://github.com/Pcelica01/Kod.git.

Funkcija koja pravi kašnjenje u mikrosekundama prikazana je na slici 6.

```
void delay_us (uint16_t us) //Deklaracija funkcije lus
{
  __HAL_TIM_SET_COUNTER(&htim4,0);//Brojac u TIM4 postavlja na nulu
  while (__HAL_TIM_GET_COUNTER(&htim4) < us);
  // Brojac broji sve dok ne dostigne odgovarajucu vrednost
  //Svako brojanje traje 1us
```

Slika 6. Funkcija koja pravi delay u μs.

Funkcija kao argument uzima celobrojnu vrednost koja predstavlja broj mikrosekundi. U telu funkcije, najpre se tajmer pokreće uz postavljanje inicijalne vrednosti na nulu. *While* petlja će ostati u stanju *true* sve dok tajmer ne dostigne željenu vrednost mikrosekundi. Nakon dostizanja vrednosti, uslov petlje postaje *false* i izlazi se iz tela petlje.

Unutar USER CODE 3 (slika 7) nalaze se naredbe koje se odnose na setovanje ultrazvučnog senzora, konkretno: slanje intervala od 10 μs pinu TRIGGER i merenje vremena tokom dok je pin ECHO na logičkoj jedinici. U nastavku će biti objašnjen način na koji se meri vreme trajanja impulsa sa pina ECHO. Naime, kada dođe do rastuće ivice signala ECHO, promenljiva *numTicks* kreće da broji, odnosno, počevši od nule, na svake 2 μs inkrementira svoju vrednost. Na taj način, nakon nailaska opadajuće ivice ECHO signala, u promenljivoj se nalazi broj intervala u trajanju od 2 μs. Kada se taj broj intervala pomnoži sa 2 μs, dobija se vreme trajanja logičke jedinice signala ECHO.

```
    HAL_GPIO_WritePin(TRIGGER_GPIO_Port, TRIGGER_Pin, GPIO_PIN_RESET);
    //Triger pin postavljamo na nulu da bi postigli stabilno stanje
    delay_us(2); //Pozivamo funkciju delay_ms i saljemo joj argument 2us

    // ****Pocetak rutine merenja ultrazvucnog senzora***//
    //1. Postavljamo triger pin  na 1, zatim cekamo 10us
    HAL_GPIO_WritePin(TRIGGER_GPIO_Port, TRIGGER_Pin, GPIO_PIN_SET);
    delay_us(10); //Pozivanje funkcije
    HAL_GPIO_WritePin(TRIGGER_GPIO_Port, TRIGGER_Pin, GPIO_PIN_RESET);
    //Postavljanje triger pina na nulu

    //2. Cekamo rastucu ivicu sa echo pina

    while(HAL_GPIO_ReadPin(ECHO_GPIO_Port, ECHO_Pin) == GPIO_PIN_RESET);

    //3. Merenje sirine impulsa sa ECHO pina
    //Dok je eho setovan numTicks broji
    numTicks=0;
    while(HAL_GPIO_ReadPin(ECHO_GPIO_Port, ECHO_Pin) == GPIO_PIN_SET)
{
numTicks++;
delay_us(2); //Korisnik proverio da posle 2us nailazi opadajuca ivica
}

//4. Procena rastojanja u cm
distance=(numTicks + 0.0f)*2.8*speedOfSound;
```

Slika 7. Naredbe koje se odnose na ultrazvučni senzor.

Međutim, zbog internih kašnjenja prouzrokovanih vremenom inicijalizacije samog tajmera i ostalih faktora, tajmer neće praviti delay od 2 μs, već nešto veći, što se merenjem pokazuje da iznosi 2.8 μs. Zbog toga se broj intervala na kraju množi sa 2.8, čime se dobija vrlo precizna aproksimacija potrebnog vremena. Kada se vreme trajanja impulsa sa ECHO pina pomnoži brzinom zvuka, tačnije sa promenljivom *speedOfSound* (koja je inicijalno postavljena na polovinu stvarne brzine zvuka), dolazi se do vrednosti rastojanja u centimetrima. Promenljiva koja čuva vrednost rastojanja u cm jeste *distance*. Na taj način, uspešno je izmereno rastojanje od senzora do posmatranog objekta, a ta vrednost sačuvana u promenljivoj, što omogućava da se kasnije njom manipuliše kako bi se kontrolisala brzina motora.

Na kraju sledi pokretanje DC motora u zavisnosti od rastojanja. Unutar USER CODE 3 opisan je način na koji se dovode u vezu rastojanje i brzina motora (slika 8).

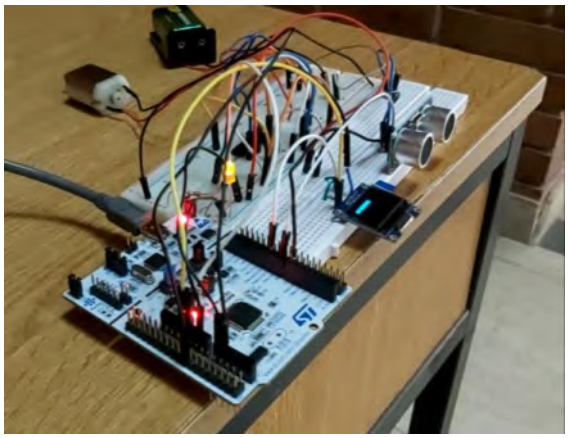
VI. PRAKTIČNA REALIZACIJA

Praktična realizacija ovog sistema prikazana je na slici 9. Ukoliko je rastojanje veće od 100 cm, u registar CCR1 upisuje se vrednost 99. S obzirom da je prilikom inicijalizacije tajmera TIM2 uneta vrednost ARR = 100-1,

vrednost 99 predstavljaće maksimalni faktor ispune, što za posledicu ima maksimalnu brzinu okretanja motora. Sve vreme, dok se ne dostigne rastojanje manje od 50cm, na pin PA1 (gde se nalazi dioda) šalje se nizak naponski nivo. Tek kada rastojanje bude manje od 50 cm, na PA1 šalje se visok naponski nivo, i dioda signalizira da je rastojanje manje od 50 cm.

```
if(distance > 100){
        htim2.Instance -> CCR1 = 99;
        HAL_GPIO_WritePin(Diode_GPIO_Port, Diode_Pin, 0);
}
else if (distance >= 50) {
        HAL_GPIO_WritePin(Diode_GPIO_Port, Diode_Pin, 0);
        htim2.Instance -> CCR1 = (0.78 * distance +21);
}
else {
     HAL_GPIO_WritePin(Diode_GPIO_Port, Diode_Pin, 1);
     htim2.Instance -> CCR1 = 0;
}
```

Slika 8. Pokretanje motora zavisno od rastojanja.



Slika 9. Praktična realizacija sistema.

Ukoliko je rastojanje između 50 cm i 100 cm, neophodno je da se brzina motora smanjuje proporcionalno rastojanju. S obzirom da je bilo problema oko pokretanja motora sa PWM-om manjim od 60 %, brzina motora se proporcionalno smanjuje sa PWM-om ne od 99 do 0 (što bi bilo idealno), već od 99 do 60. To znači da je potrebno interval razdaljine od 100 cm do 50 cm skalirati na vrednost PWM-a od 99 do 60. To se postiže relacijom: *pwm = distance*0.78 + 21*. Ova vrednost se upisuje u registar CCR1, zatim se motoru prosleđuje PWM signal čiji je faktor ispune proporcionalan rastojanju merenom pomoću senzora. Kada vrednost rastojanja bude manja od 50 cm, u registar CCR1 šalje se vrednost 0, što predstavlja nulti faktor ispune, pa se motor odmah zaustavlja.

VII. Zaključak

Ovaj projekat može se unaprediti na mnogo načina, što zavisi od volje i kreativnosti pojedinca. Umesto OLED-a može se iskoristiti bilo koji displej, a na njemu se pored rastojanja mogu ispisivati i trenutna brzina motora, faktor ispune, smer okretanja itd… DC motor u ovom projektu se okreće samo u jednom smeru, što mu nije jedina opcija. Integrisano kolo L293D nudi mogućnost promene smera obrtanja motora. Ako eksploatišemo tu činjenicu, moguće je unaprediti projekat time da se, na primer, menja smer okretanja motora u zavisnosti od toga da li se rastojanje povećava ili smanjuje. Izmene u kodu bile bi minimalne, a postigao bi se zanimljiv efekat.

Zahvalnica

Rad je realizovan kao deo projektnog zadatka iz predmeta Komponente i kola snage, na modulu Elektronske komponente i mikrosistemi. Autori se zahvaljuju mentorima Milošu Marjanoviću i Milanu Stojanoviću na pruženom znanju i pomoći.

Literatura

[1] I. Manić „*Komponente i kola snage*" Niš. Dostupno na: http://mikro.elfak.ni.ac.rs/wp-content/uploads/Komponente-i-kola-snage_predavanja.pdf
[2] Tehnička specifikacija za NucleoF103. Dostupna na: https://os.mbed.com/platforms/ST-Nucleo-F103RB/
[3] Tehnička dokumentacija za ultrazvučni sensor HCSR04 Dostupno na: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf
[4] Tehnička dokumentacija za L293D. Dostupno na: https://www.ti.com/lit/ds/symlink/l293.pdf
[5] Tehnička dokumentacija za SSD1306 kontroler. Dostupna na: https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf

Solarni sistem za praćenje svetlosti i korišćenje energije solarnih panela pomoću Arduino Mega platforme

Đorđe Pavlović, Petar Tanasković, Nemanja Plazinić, Miloš Šekularac

Sadržaj − Rad podrazumeva automatizovanu maketu solarnog pratioca pomoću Arduino Mega platforme. Sistem bi se trebao okretati prema najvećem izvoru svetlosti uz pomoć 2 servo motora koji ga rotiraju po X i Y osi. Jačinu i smer svetala očitavaju 4 LDR-a (Light Dependent Resistor), odnosno 4 fotootpornika. Po dva su postavljena na svaku stranu (gore, dole, levo, desno). Takođe je bitno napraviti i odgovarajuću maketu kako bi se ceo sistem pravilno okretao, a da fotootpornici ne bi dobijali pogrešnu informaciju potrebno ih je propisno odvojiti. Za ispravno funkcionisanje celog sistema potrebno je napisati kod u prilagođenom C++ programskom jeziku, te isprogramirati Arduino Uno tim istim kodom.

I. Uvod

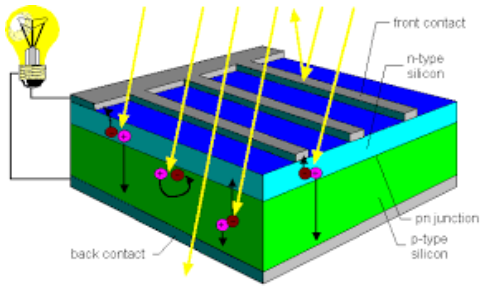
Ovakav solarni pratilac ima smisla na objektima koji su u konstantnom pokretu i često menjaju svoj položaj (npr. sateliti u svemiru). Na stacioniranim objektima ili kada maketa stoji individualno na zemlji nema nekog smisla, jer poznajemo kretanje Sunca i u takvim slučajevima nema potrebe za korišćenjem dve ose nego je dovoljna samo jedna. Kodom su definisane sve pojedinosti sistema, kao što je izračunavanje razlike u pokazivanju na fotootpornicima, ili određivanje za koliko stepeni je potrebno da se okreću servo motori (ukoliko je razlika van granica tolerancije).

II. Princip rada

Eksperimentalna analiza sistema se vrši preko monitora. Na monitoru se mogu videti kakve vrednosti LDR otpornici primaju, pa se tako može zaključiti sa koje strane dolazi najjače svetlo. Sa svake strane su po 2 LDR-a. Računajući prosečne vrednosti dobija se informacija na koji fotootpornik dolazi najviše svetlosti. Tada se šalje signal u servo motore koji treba da se okrenu za odgovarajući broj stepeni u stranu iz koje dolazi najveći intenzitet svetlosti. Vrednosti u serial monitoru su označene kao avt, avd, avl, avr. Računanje vrednosti je objašnjeno i prikazane su redom kako se štampaju u monitoru.

A. Opis komponenti korišćenih u radu

Solarne ćelije su poluprovodničke komponente koje pretvaraju Sunčevu energiju u električnu pomoću fotoelektričnog efekta. Mogu se izrađivati od različitih materijala, a najčešće solarne ćelije su na bazi silicijuma. Solarne ćelije se koriste kao izvori energije za različite primene, kao što su televizori, osvetljenje, mali kućni aparati. Više solarnih ćelija mogu se povezati i tada se nazivaju solarni paneli ili fotonaponske ploče.



Slika 1. Solarna ćelija.

Servo motor je rotacioni ili linearni pogon (aktuator) koji omogućava preciznu kontrolu ugaone ili linearne pozicije, brzine i ubrzanja. Sastoji se od odgovarajućeg motora povezanog sa senzorom za povratnu informaciju o položaju. Servo motor zahteva relativno softisticirani kontroler, često određeni modul dizajniran posebno za korišćenje sa servo motorima. Servo motor nije specifična klasa motora, ali pojam servo motor se često odnosi na motor pogodan za upotrebu u sistemu kontrole zatvorene petlje.



Slika 2. Servo motori.

Senzori za merenje osvetljenosti se nalaze iznad dela predviđenog za solarni panel. Na tom mestu se nalaze fotootpornici koji očitavaju intenzitet svetlosti koja pada na njih. Razdvojeni su uz pomoć malih pregrada kako bi očitavanje smera bilo lakše i tačnije. LDR ili fotootpornik je komponenta čija se otpornost menja pod dejstvom svetlosti. Izrađuju se od poluprovodničkih materijala. Najčešće rade na principu fotoprovodnosti. Rezultat ovoga

Đ. Pavlović, P. Tanasković, N. Plazinić, M. Šekularac su učenici Mašinsko - saobraćajne škole iz Čačka, dr Dragiše Mišovića 146, 32000 Čačak, Srbija, E-mail: milenkovic87@yahoo.com.

je da sve više i više struje krene prolaziti kroz uređaj kada je kolo zatvoreno i time se smanjuje otpornost komponente.

Ultrazvučni senzor generiše ultrazvučne talase koji se odbijaju od svih predmeta u prostoriji. Ako se u njoj ništa ne pomera, ultrazvučni talasi se vraćaju neizmenjeni. Ukoliko se neko ili nešto kreće po prostoriji, ultrazvučni talasi se izobličuju, i to izobličenje pobuđuje alarm. Ultrazvučni senzori su pogodni za merenje rastojanja od 20 mm do oko 10 m. Ovi senzori imaju visoku tačnost, ali je potrebno dobro poznavati kojom se brzinom prostire zvuk u datoj sredini. Ultrazvučni senzori rastojanja emituju periodične kratkotrajne visokofrekventne (učestanost veća od 20 kHz) zvučne impulse.



Slika 3. Ultrazvučni senzor.

HC-06 Bluetooth modul je *slave only* modul dizajniran za bežičnu serijsku komunikaciju. Bežična serijska komunikacija vrši se između Arduina i Bluetooth uređaja (uređaj koji može da šalje serijske podatke putem vazduha: pametni telefoni, računar) pomoću bežične Bluetooth veze. Kada Bluetooth modul prima podatke, oni se šalju na izlaz preko serijskog interfejsa. Arduinu nije potreban izvorni kod specifičan za Bluetooth modul. Aplikacija na telefonu se koristi za slanje ulaza modulu koji prima, a zatim ih prenosi na Arduino.



Slika 4. Bluetooth modul.

Kodovi za Arduino su pisani u prilagođenom C++ jeziku. Njegova glavna uloga je povezivanje računarskog i fizičkog sveta. Uz pomoć raznih senzora, motora i ostalih elektronskih komponenti Arduino može primati signale iz okoline, te i sam uticati na okolinu. Na samoj pločici se nalazi nekoliko ulaznih i izlaznih pinova, a uz upotrebu proto pločice nije potrebno lemiti komponente, jer će se koristiti *through hole* komponente, odnosno komponente sa izvodima. Arduino ploču čine 8-bitni Atmel AVR mikrokontroler sa pripadajućim komponentama koje omogućavaju programiranje i povezivanje sa drugom elektronikom.



Slika 5. Arduino Uno platforma.

B. Opis fizičkog rada sistema

Ceo projekat je potrebno prvo napraviti na proto ploči kako bi se na jednostavan način ispitao i po potrebi prilagodio kod sa kojim programiramo mikrokontrolersku jedinicu. Nakon toga, maketa od šperploče je izrađena prema nacrtu. Sledi spajanje delova makete, te ugradnja servo motora, fotootpornika sa štampanom pločom i ugradnja Arduino Uno sa Arduino *shield* na postolje (bazu) makete. Kada je sve ugrađeno i na svome mestu, sledi testiranje cele makete. U elektronskoj šemi sistema se nalaze 2 servo motora, 4 LDR otpornika, 4 otpornika od 10 kΩ i Arduino Uno. Vertikalni servo je spojen na pin 11, a horizontalni na pin 10 dok su LDR otpornici spojeni na analogne pinove na sledeći način: levi gornji A0, desni gornji A1, levi donji A2, desni donji A.



Slika 6. Finalni izgled projekta.

C. Programska realizacija

```
#include<Servo.h>
#include<SR04.h>
#define TRIG_PIN 12
#define ECHO_PIN
Servo myservo;
int pos=0
void setup() {
myservo.attach(9);
}

void loop() {
Serial.egin(9600);
  delay(1000);
}
void loop(){
  a=sr04. Distance();
Serial. print(a);
  Serial. println("Cm");
  delay(1000);
}
```

```
for(pos=0; pos<=180; pos+=1){
myservo.write(pos);
delay(15);
}

}
```

KOD ZA Bluetooth MODUL:

```
void setup(){
  pinMode(7,OUTPUT);
}
void loop(){
  digitalWrite(7,HIGH);
  delay(100);
  digitalWrite(7,LOW);
  delay(100);
```

III. ZAKLJUČAK

Upotreba obnovljivih izvora energije je sve više zastupljeno u praksi, domaćinstvima. Njihovim korišćenjem smanjuje se potrošnja struje, upotreba fosilnih goriva. Kombinacijom obnovljivih izvora energije sa IT tehnologijom, može se napraviti Pametna kuća (Smart Home) u kome bi vlasnici upravljali pomoću aplikacija, glasovnih komandi, automatizacijom ili pomoću veštačke inteligencije.

ZAHVALNICA

Zahvaljujemo se mentoru Stefanu Milenkoviću na pomoći i pružanju uslova za rad, kao i za svaki savet.

LITERATURA

[1] AVR mikrokontroleri i Atmel Studio, Autor Warwick, A. Smith.
[2] The Intel Microprocessors – Architecture Programming And Interfacing, Bary B. Bay.
[3] Microprocessor Architecture, Programming, and Applications with the 8085, Ramesh S. Gaonkar.
[4] Programming for Microprocessors, Endru Colin
[5] ARM Microprocessor Systems: CortexM Architecture, Programming, Muhammad Tahir
[6] Microprocessors: Principles, Programming, and Interfacing, Kenet Mukou, Bil R. Dim.

Projektovanje LED osvetljenja korišćenjem TLC5925 drajvera

Anastasija Avramović, Aleksandra Ilić, Milan Stojanović, Nemanja Stojanović

Sadržaj − U ovom radu prikazan je sistem za praćenje jačine svetlosti koju daje skup 16 LED dioda, koji se sastoji od 4 grupe različitih boja. Diode se pobuđuju pomoću drajvera TLC5925 i kontrolišu pomoću 4 tastera od kojih je svaki zadužen za uključivanje određene boje. Na OLED displeju se prikazuju broj uključenih LED dioda i jačina svetlosti koju one daju.

I. UVOD U OPTOELEKTRONSKE KOMPONENTE

Optoelektronika je naučna disciplina elektronike čiji je fokus rad sa komponentama koje mogu da emituju ili detektuju svetlost. Optoelektronske komponente su komponente koje proizvode svetlost ili reaguju na nju. Neke od optokomponenti su: LE diode [1], [2], foto-tranzistori, foto-diode, foto-otpornici, razni svetlosni indikatori, optokapleri [3]. Takođe, u optoelektronske poluprovodničke kompo-nente spadaju i laserske diode.

II. KORIŠĆENE KOMPONENTE ZA IZRADU PROJEKTA

A. Nucleo STM32F103RB

Nucleo [4] razvojno okruženje, korišćeno za realizaciju ovog 1istema, je okruženje koje omogućava korisnicima izradu prototipova I 1istema. Centralni deo okruženja predstavlja STM mikrokontroler STM32F103RB zasnovan na ARM Cortex-M3 RISC arhitekturi [5]. Ovaj mikrokontroler radi na frekvenciji od 72 MHz, poseduje ugrađenu memoriju velike brzine, širok spektar ulaza/izlaza I periferija, dva analogno-digitalna konvertora, 4 tajmera, kao I standardne komunikacione interfejse: dva I2C I SPI interfejsa, 3 USART-a, CAN I USB, dužina reči iznosi 32 bita, a ima I dovoljno memorije za pokretanje operativnog 1istema za rad u realnom vremenu (RTOS). *Nucleo* okruženje prikazano je na slici 1.

Okruženje na sebi poseduje I ST-LINK programator, tako da je prilikom programiranja mikrokontrolera dovoljno povezati okruženje sa računarom putem USB-a, a zatim učitati program na mikrokontroler.

A. Avramović, A. Ilić i N. Stojanović su studenti na modulu Elektronske komponente i mikrosistemi, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: anastasija.avram@elfak.rs, anchiii@elfak.rs, nemanja.stojanovic@elfak.rs.

M. Stojanović je student osnovnih studija na modulu Elektronika, Elektronski fakultet, Univerzitet u Nišu. Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: milancestojanovic@elfak.rs.



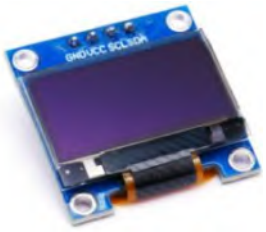
Slika 1. Nucleo okruženje.

B. TLC5925 drajver

Integrisano kolo TLC5925 [6] služi za pobuđivanje LED. Ovaj drajver istovremeno može da pobudi I do 16 dioda. Koristi SPI interfejs za povezivanje I komunikaciju sa mikrokontrolerom. U zavisnosti od poruke koju mu mikrokontroler pošalje, drajver će pobuditi određeni broj LED-ova. Drajver TLC5925 spada u grupu drajvera sa konstantnim izvorom struje (*constant current*), što omogućava da do svake LE diode dođe ista vrednost struje.

C. OLED SSD1306

OLED SSD1306 (slika 2) modul predstavlja moderni tip displeja za primenu u *embedded* sistemima, pre svega zbog svojih malih gabarita [7]. Sa mikrokontrolerom se povezuje preko samo 4 pina, korišćenjem I2C interfejsa, što rasterećuje MCU. Koristi SSD1306 drajver koji omogućava veliki broj podešavanja. Konkretno u ovom sistemu, OLED displej služi za uvid u to koliko je dioda uključeno i koliku jačinu svetlosti one daju što može biti bitno korisnicima.



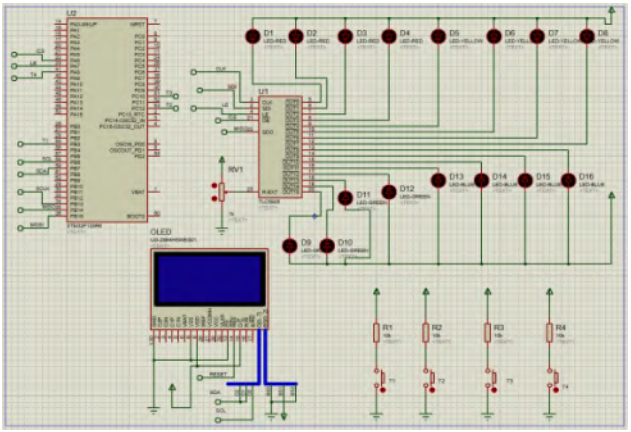
Slika 2. OLED displej.

III. Opis šeme sistema

Na slici 3 može se videti šema sistema. Centralni deo sistema je razvojno okruženje *Nucleo* sa STM32F103RB mikrokontrolerom [4,5].

Na ovo razvojno okruženje se povezuje drajver, TLC5925 [6], putem SPI protokola. Na šemi su prikazane oznake linija ove veze (MOSI, MISO, SCLK, /CS, LE), kao i odgovarajućih pinova mikrokontrolera i drajvera. Na sam drajver je povezano 16 dioda i eksterno jedan potenciometar (R_{EXT}), kojim se kontroliše kolika će struja proticati kroz LED diode. Određena grupa dioda se uključuje pomoću odgovarajućeg tastera. Tasteri su povezani u *pull-up* konfiguraciji i svaki taster uključuje određenu boju.

Na razvojno okruženje je povezan i OLED [7] displej, putem I2C protokola (linije SDA i SCL).



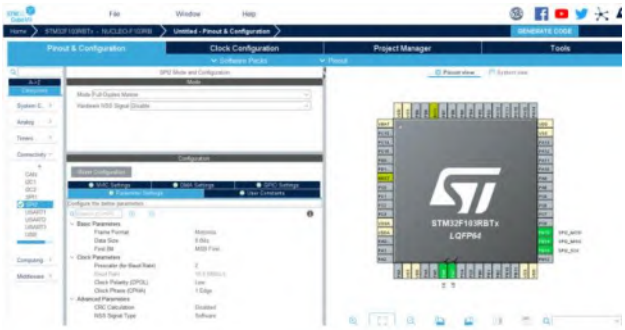
Slika 3. Šema sistema.

IV. Podešavanja u CubeMX

Odabir *Nucleo* okruženja i podešavanja periferijskih jedinica se obavlja u programu *CubeMX* [8]. *CubeMX* je grafički alat koji omogućava laku konfiguraciju STM32 mikrokontrolera i generisanje odgovarajućeg inicijalizacionog C koda. U ovom programu se bira odgovarajuće *Nucleo* okruženje i mikrokontroler.

Na slici 4 prikazana su podešavanja SPI protokola i odabir *Full-Duplex Master* moda rada što omogućava komunikaciju u oba smera između *Nucleo* [4] okruženja (master) i drajvera TLC5925 [6] (slejv). Time su postavljeni pinovi PB15, PB14 i PB13 kao MOSI, MISO i SCK, respektivno. Putem MOSI linije i pinova master šalje podatke slejvu, pomoću MISO master prima podatke koje šalje slejv i pomoću SCK master generiše takt rada slejva.

Pinovi PA6 i PA7 su podešeni kao izlazni i označeni kao CS i LE, respektivno. Pomoću /CS pina selektuje se slejv uređaj sa kojim master želi da komunicira, dok pin LE služi za dozvolu slanja podataka.



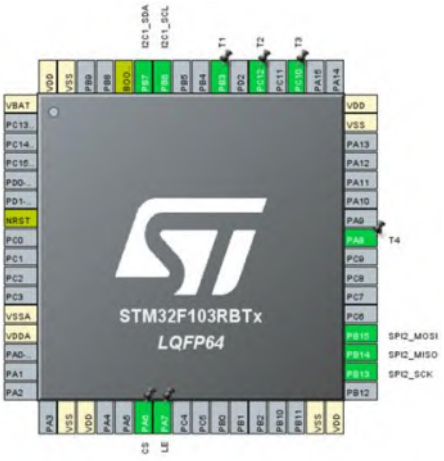
Slika 4. Podešavanje SPI protokola.

Na slici 5 se može videti podešavanje I2C protokola koji je neophodan za komunikaciju sa OLED [7] displejem. Pinovi PB7 i PB6 služe kao SDA i SCL. Preko SDA linije mikrokontroler šalje podatke displeju, dok SCL služi za generisanje signala takta za OLED. Koristi se I2C *fast mode* za brzo slanje podataka.



Slika 5. Podešavanje I2C protokola.

Na slici 6 su prikazani svi potrebni pinovi zajedno sa pinovima na kojima su povezani tasteri (T1, T2, T3 i T4). Pinovi za tastere su podešeni kao ulazni jer se na njima detektuje da li je taster pritisnut.



Slika 6. Svi potrebni pinovi obojeni zelenom bojom.

V. PROGRAMSKI KOD

Programski kod je pisan u IDE *Keil uVision* [9]. Ovo integrisano razvojno okruženje služi za kreiranje *firmware* i za "spuštanje" koda na čip.

Za rad sa OLED displejem bilo je potrebno uključiti dodatne biblioteke *fonts* i *ssd1306*. Biblioteke se uključuju naredbom *#include "ime_biblioteke.h"*.

Na slici 7 prikazan je programski kod funkcije *pisidrajver()*. Ova funkcija za argumente uzima dva 8-bitna podatka za uključivanje određenih grupa dioda. U promenljivoj P8D smeštaju se podaci za uključivanje prvih osam dioda, četiri plave i četiri žute (0x0F - plave, a 0xF0 - žute). U promenljivoj D8D smeštaju se podaci za uključivanje drugih osam dioda, četiri crvene i četiri zelene (0xF0 - zelene, a 0x0F - crvene). Podaci u okvru jednog nibla određuju koje diode će iz te grupe biti uključne (1 = ON).

Funkcija *HAL_GPIO_WritePin()* služi za dodeljivanje vrednosti određenom pinu. Na početku se pinu CS dodeljuje vrednost '0' čime se selektuje slejv, a zatim se i pinu LE dodeljuje vrednost '0' (njegova početna vrednost je stanje visoke impedanse (HIGH Z)).

Nakon toga se u prvi element niza pod (pod[0]) smeštaju podaci P8D kojima se uključuje prvih osam dioda, a u drugi element niza (pod[1]) podaci D8D kojima se uključuje drugih osam dioda.

Pozivom funkcije *HAL_SPI_Transmit()* master (mikrokontroler) šalje podatke slejvu (TLC5925 drajveru). Ova funkcija ima sledeće argumente: pinter na hendler za SPI (&hspi2), podatak koji se šalje, veličinu tog podatka i vreme čekanja u ms. U *while* petlji se ispituje uslov zauzetosti SPI-a, odnosno čeka se sve dok je on zauzet. Dodeljivanjem vrednosti '1' pinu LE dozvoljava se transfer podataka. Zatim se čeka dve sekunde da se transfer obavi, a onda se vrednost pina LE vraća na '0'. Na kraju se CS pin postavlja na '1' nakon obavljenog transfera podataka.

```
void pisidrajver(uint8_t P8D, uint8_t D8D) {
    HAL_GPIO_WritePin(CS_GPIO_Port, CS_Pin, 0);
    HAL_GPIO_WritePin(LE_GPIO_Port, LE_Pin, 0);
    pod[0] = P8D;
    pod[1] =D8D;
    //slanje podataka
    HAL_SPI_Transmit(&hspi2, pod, 2, 100);
    while (hspi2.State == HAL_SPI_STATE_BUSY) {};
    HAL_GPIO_WritePin(LE_GPIO_Port, LE_Pin, 1);
    HAL_Delay(2000);
    HAL_GPIO_WritePin(LE_GPIO_Port, LE_Pin, 0);
    HAL_GPIO_WritePin(CS_GPIO_Port, CS_Pin, 1);}
```

Slika 7. Programski kod funkcije *pisidrajver()*.

Funkcija *pisiOLED()* čiji je kod prikazan na slici 8 služi za ispivanje podataka na displej. Kao argumente uzima vrednosti broja dioda (*bd*) i vrednost jačine svetlosti u milikandelama (*bmcd*).

Funckija *SSD1306_Fill()* služi za uključivanje ili isključivanje svih dioda displeja u zavisnosti od argumenta koji joj je prosleđen ('0' isključuje, '1' uključuje sve piksele).

Funkcijom *SSD1306_GotoXY()* vrši se pozicioniranje na displeju. Argumenti ove funckije su X i Y koordinata tačke na displeju u kojoj se pozicioniramo.

SSD1306_Puts() je funkcija kojom se ispisuje tekst na displeju. Kao argumente uzima sam tekst napisan pod navodnicima, font kojim će on biti ispisan, a '1' na kraju služi za uključivanje piksela koji će prikazivati tekst.

Pošto se na OLED displeju samo mogu prikazivati promenljive tipa *string*, neophodno je vrednosti broja dioda i jačine svetlosti koje su tipa *unsigned integer* smestiti u novu promenljivu koja je tipa *string*. U tu svrhu se koristi funkcija *sprintf()* koja kao argumente uzima promenljivu tipa *string* u koju se smeštaju vrednsoti, zatim "%d" što označava da je brojna vrednost koja se smešta u novu promenljivu dekadno prikazana i poslednji argument je sama dekadna brojna vrednost. Broj dioda se smešta u promenljivu *Br_dioda*, a jačina svetlosti u promenljivu *Jacina_sv*.

Funkcija *SSD1306_UpdateScreen()* služi za osvežavanje displeja odnosno za primenu svih prethodno navedenih funkcija za pozicioniranje i ispis teksta na displej.

```
void pisiOLED(uint8_t bd, uint16_t bmcd) {
    SSD1306_Fill(0);
    SSD1306_GotoXY(10, 10);
    SSD1306_Puts("Br. dioda=", &Font_7x10, 1);
    sprintf(Br_dioda, "%d", bd);
    SSD1306_GotoXY(80, 10);
    SSD1306_Puts(Br_dioda, &Font_7x10, 1);
    SSD1306_GotoXY(1, 30);
    SSD1306_Puts("Jacina_sv=     mcd", &Font_7x10, 1);
    sprintf(Jacina_sv, "%d", bmcd);
    SSD1306_GotoXY(71, 30);
    SSD1306_Puts(Jacina_sv, &Font_7x10, 1);
    SSD1306_UpdateScreen();}
```

Slika 8. Programski kod funkcije pisiOLED().

U okviru beskonačne petlje (*while(1)*) ispituje se koji je od tastera pritisnut, odnosno, kao posledica toga, koje diode treba uključiti. Svakom tasteru je dodeljen *flag* koji vrednošću '1' signalizira da je taster pritisnut, a vrednošću '0' da taster nije pritisnut.

Na slici 9 prikazan je deo koda u kome se ispituju vrednosti *flag*-ova, nakon čega se OLED displeju i TLC5925 šalju odredjeni podaci pozivom funkcija *pisidrajver()* i *pisiOLED()*. Ovde su konkretno prikazani slučajevi kada nije pritisnut nijedan taster i kada su sva četiri tastera pritisnuta.

```
//Nijedan taster nije pritisnut, diode ne svetle
if(fleg1==0 && fleg2==0 && fleg3==0 && fleg4==0){
pisiOLED(0,0);
pisidrajver(0x00, 0x00);}

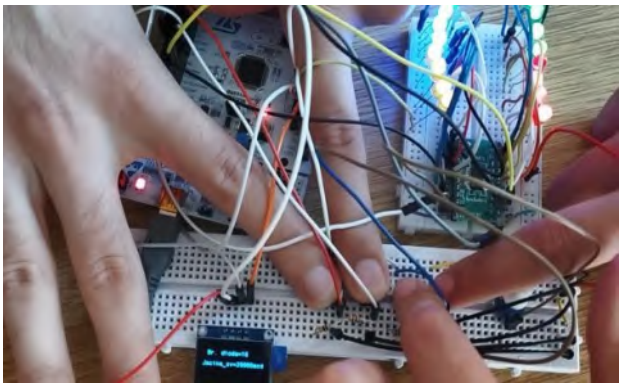
//Sve svetle diode
if(fleg1==1 && fleg2==1 && fleg3==1 && fleg4==1){
pisiOLED(16,(4*7200)+(4*29)+(4*3.7)+(4*19));
pisidrajver(0xFF, 0xFF);}
```

Slika 9. Programski kod funkcije pisiOLED().

VI. PRAKTIČNA REALIZACIJA

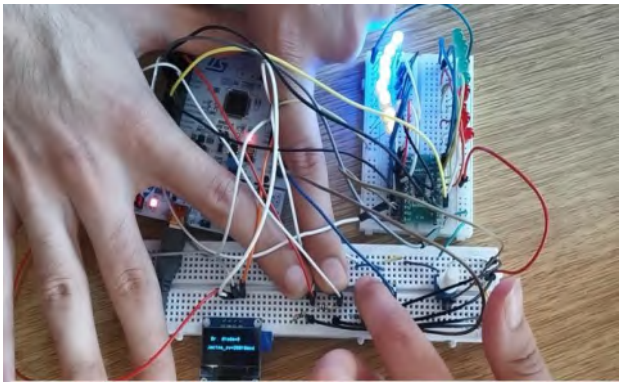
Praktična realizacija ovog sistema prikazana je na slikama 10 i 11.

Na slici 10 prikazan je slučaj kada su sva četiri tastera pristisnuta čime se uključuje svih 16 dioda. Na displeju se vidi da je broj uključenih dioda 16, kao i to da one daju jačinu svetlosti od 29006 mcd.



Slika 10. Prikaz praktične realizacije sistema kada su pritisnuta sva četiri tastera.

Na slici 11 prikazan je slučaj kada su dva tastera pritisnuta čime se uključuje 8 dioda (4 žute i 4 plave). Na displeju se vidi da je broj uključenih dioda 8, kao i to da one daju jačinu svetlosti od 28916 mcd.



Slika 11. Prikaz praktične realizacije sistema kada su dva tastera pritisnuta.

Jačina svetlosti koju daje svaka dioda može se videti u listi podataka o diodama [1], [2]. Ukupna jačina svetlosti svih uključenih dioda dobija se sabiranjem pojedinačnih vrednosti jačina svetlosti koje daju diode.

VII. ZAKLJUČAK

Projektovani sistem je pozdan jer korisnik u svakom trenutku, pomoću podataka prikazanih na OLED displeju, može imati uvid u to koliko je LE dioda uključeno, kao i koliku vrednost jačine svetlosti one daju, a takođe ima i mogućnost kontrole koliko će dioda biti uključeno.

Korak dalje u unapređenju rada sistema bilo bi omogućavanje da se jednim pritiskom na taster diode uključe i ostanu uključene sve vreme dok se taj taster ponovo ne pritisne, nakon čega bi se diode isključile. Estetski bi bilo lepše koristiti TFT displej koji ima mogućnost ispisvanja poruke na ekranu u različitim bojama. Na primer, kada su uključene crvene diode, poruka bi bila ispisvana crvenim slovima, a kada su uključene crvene i plave, poruka bi bila ispisana crvenim i plavim slovima naizmenično.

ZAHVALNICA

Rad je realizovan kao deo projektnog zadatka iz predmeta Optoelektronika, na modulu Elektronske komponente i mikrosistemi, na Elektronskom fakuletu u Nišu. Autori se zahvaljuju mentoru Nikoli Mitroviću na pruženom znanju, pomoći i podršci.

LITERATURA

[1] Tehnička dokumentacija za crvenu LE diodu. Dostupno na: https://www.tme.eu/en/details/ltl-307e/tht-leds-5mm/liteon/.
[2] Tehnička dokumentacija za diode. Dostupno na: https://www.tme.eu/html/EN/liteon-tht-leds/ramka_17807_EN_pelny.html.
[3] V. Paunovic, ”Optoelektronske komponente“. Dostupno na: http://mikro.elfak.ni.ac.rs/predmeti/optoelektronika/.
[4] Tehnička specifikacija za NucleoF103. Dostupno na: https://os.mbed.com/platforms/ST-Nucleo-F103RB/.
[5] Tehnička specifikacija za mikrokontroler STM32F103RB. Dostupno na: https://www.st.com/en/microcontrollers-microprocessors/stm32f103rb.html.
[6] Tehnička specifikacija za integrisano kolo TLC5925. Dostupno na: https://www.ti.com/lit/ds/symlink/tlc5925.pdf?ts=1617046705277&ref_url=https%253A%252F%252Fwww.google.com%252F.
[7] Tehnička specifikacija za OLED displej. Dostupno na: https://www.vishay.com/docs/37902/oled128o064dbpp3n00000.pdf.
[8] Grafički alat STM32CubeMX. Dostupno na: https://www.st.com/en/development-tools/stm32cubemx.html.
[9] Integrisano razvojno okruženje *Keil uVision*. Dostupno na: https://www2.keil.com/mdk5/uvision/.

Izvor jednosmernog napona sa digitalnom kontrolom

Nikola Stojiljković, Marko Zdravković

Sadržaj – U ovom radu biće prikazan princip projektovanja i realizacije izvora jednosmernog napona sa digitalnom kontrolom. Uređaj je realizovan korišćenjem nekoliko analognih i jednog digitalnog kola o kojima će biti reči u nastavku rada. Ovaj uređaj na svom izlazu imaće jednu vrednost negativnog napona od -12 V ili više različitih vrednosti pozitivnog napona u opsegu od 1.5 V do 12 V.

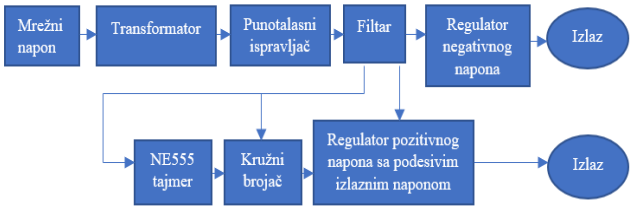
I. Uvod

Izvori napona napajanja predstavljaju električne uređaje koji transformišu naizmenični napon koji dolazi iz izvora napajanja, kao što je mreža, u vrednosti napona i struje neophodne za napajanje nekog opterećenja, kao što su motori ili razne vrste elektronskih uređaja.

Kvalitet izvora jednosmernog napona je od suštinskog značaja za pouzdan rad nekog uređaja. Savremeni izvori napajanja omogućavaju regulaciju izlaznog jednosmernog napona, smanjenje osetljivosti na promenu temperature, kao i dodatno filtriranje. Tradicionalni dizajn linearnog *AC/DC* izvora napona je evoluirao tokom godina, poboljšavajući se u smislu efikasnosti, opsega snage i gabarita samih uređaja. Veliko ograničenje u linearnom *AC/DC* izvoru napona je veličina transformatora. Zbog frekvencije ulaznog napona, neophodan transformator bi morao biti veoma veliki, mada je i za to pronađeno optimalno rešenje upotrebom feritnih transformatora.

II. Blok šema uređaja

Projektovani uređaj će se sastojati iz više delova, kao što je prikazano na slici 1.



Slika 1. Blok šema projektovanog izvora jednosmernog napona.

N. Stojiljković i M. Zdravković su studenti četvrte godine osnovnih akademskih studija Elektronskog fakulteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: stojiljkovicn@elfak.rs, zdravkovicmare@elfak.rs.

III. Princip rada i korišćene komponente

Za rad uređaja neophodno je obezbediti mrežni napon 230 V/50 Hz. Ovaj napon vodi se na mrežni transformator koji ima ulogu da smanji amplitudu ulaznog napona, u ovom slučaju na 15 V. Nakon toga ovaj naizmenični napon dovodimo na ulaz punotalasnog ispravljača pomoću koga se dobijaju dve vrednosti jednosmernog napona, 15 V i –15 V. Kako dobijeni napon nije idealan *DC* napon, neophodno je dodatno filtriranje istog uz pomoć dva elektrolitska kondenzatora. Negativna vrednost napona odvodi se na ulaz regulatora negativnog napona *LM7912* [1] sa fiksnim izlaznim naponom, na čijem izlazu će biti vrednost od -12 V. Napon od 15 V predstavlja napajanje za *NE555* tajmer [2], kružni brojač *CD4017* [3] i regulator pozitivnog napona sa podesivim izlaznim naponom *LM317* [4].

Kontrola pozitivne vrednosti napona na izlazu pozitivnog naponskog regulatora vrši se pomoću *NE555* tajmera i kružnog brojača. Praktično, pritiskanjem prekidača, *NE555* tajmer dobiće napon napajanja, pri čemu je povezan u monostabilnom režimu rada, tako da će na svom izlazu generisati kratkotrajni impuls. Ovo predstavlja *clock* signal kružnog brojača. Na ovaj način menjaće se aktivni izlazi brojača od Q0 do Q9. Aktiviranjem odgovarajućeg izlaza odgovarajući potenciometar formiraće naponski razdelnik sa otpornikom u kolu pozitivnog naponskog regulatora. Podešavanjem vrednosti otpornosti potenciometra na izlazu će biti vrednost napona od 1.5, 3.0, 4.5, 5.0, 6.0, 7.5, 9.0, 10.5, ili 12.0 V.

A. Transformator

Transformator predstavlja elektronsku komponentu. Sastoji se od dva kalema spregnuta na jednom magnetnom jezgru (*core*). Kalem na koji se dovodi ulazni signal naziva se primarni namotaj (primar), a kalem na kome se pojavljuje izlazni signal naziva se sekundarni namotaj (sekundar) [5].

Tokom rada transformatora pojavljuju se gubici u prenosu snage usled: „curenja“ magnetnog fluksa van jezgra, osobina materijala jezgra zbog kojih se ono zagreva, parazitnih otpornosti primara i sekundara, parazitnih kapacitivnosti primara i sekundara. Zato definišemo parametar efikasnost (koeficijent korisnog dejstva transformatora).

B. NE555 tajmer

NE555 tajmer predstavlja najčešće korišćeno integrisano kolo za precizno generisanje vremena (impulsa), uzrokovanje kašnjenja u kolu, širinsko-impulsnu modulaciju, frekvencijsku modulaciju, za kreiranje linearnog ramp signala i slično.

Ovo kolo može raditi u tri različita režima:

- monostabilni režim
- astabilni režim
- bistabilni režim,

pri čemu je za potrebe ovog projekta korišćen monostabilni režim rada.

C. Naponski regulator

Naponski regulator predstavlja kolo koje na izlazu daje stabilisani napon. Uzimajući u obzir činjenicu da napon na izlazu punotalasnog ispravljača i dalje sadrži određenu talasnost, a veliki broj kola zahteva jako stabilan i nepromenljiv napon da bi mogla da funkcionišu, naponski regulatori su neophodna kola.

Naponski regulatori se mogu podeliti u dve grupe:

- Regulatori sa fiksnim izlaznim naponom,
- Regulatori sa podesivim izlaznim naponom.

Posebnu pažnju treba obratiti na minimalni pad napona na regulatoru za koji se garantuje stabilna vrednost izlaznog napona [6].

D. Kružni brojač

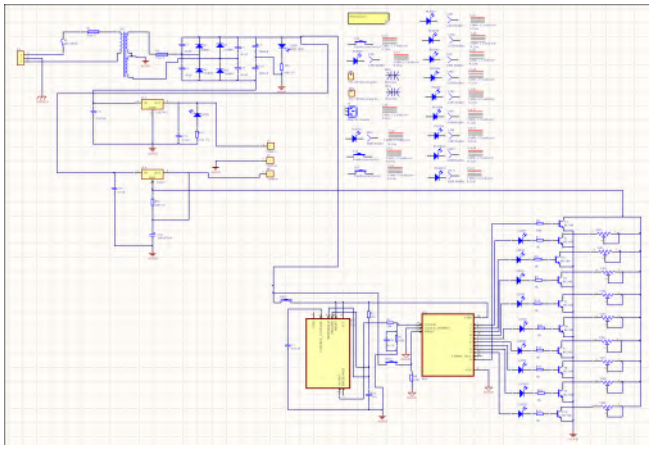
Kod kružnih brojača, korišćenih prilikom projektovanja, princip rada bazira se na upotrebi asinhronih ulaza karakterističnih za *flip-flop*-ove od kojih se sam registar realizuje. Koristi se ulaz */PRE* čija je osnovna uloga setovanje izlaza *flip-flop*-a nezavisno od *CLK* signala. Stanje logičke jedinice na izlazu prethodnog *flip-flop*-a ujedno se vodi na ulaz sledećeg *flip-flop*-a u strukturi pri čemu se sa nailaskom odgovarajuće ivice *CLK* signala prosleđuje njegov izlaz. Time je omogućeno, preko povratne sprege, kruženje logičke jedinice na izlazima registra, što je prikazano u tabeli I.

TABELA I
SEKVENCA STANJA 10- BITNOG PRSTENASTOG BROJAČA [7]

CLK	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9
0	1	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0
6	0	0	0	0	0	0	1	0	0	0
7	0	0	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	0	1

IV. PROJEKTOVANJE ŠTAMPANE PLOČE

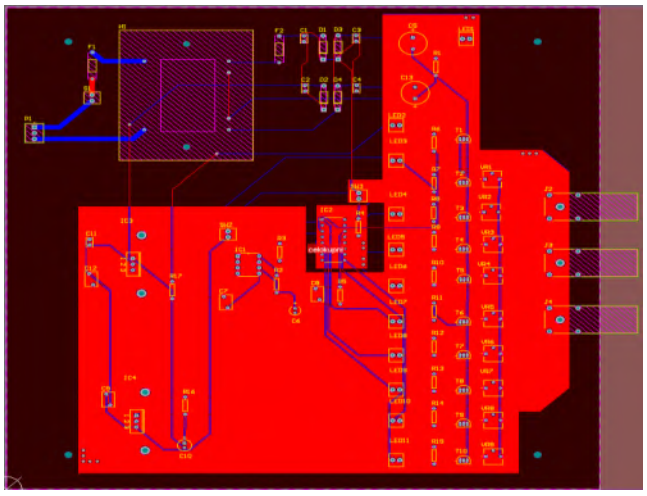
Sledeći korak pri projektovanju uređaja je crtanje odgovarajuće električne šeme koja je prikazana na slici 2. To je odrađeno u okviru softverskog paketa *Altium Designer* [8], u kome je projektovana celokupna štampana ploča uređaja.



Slika 2. Električna šema u softveru *Altium Designer*.

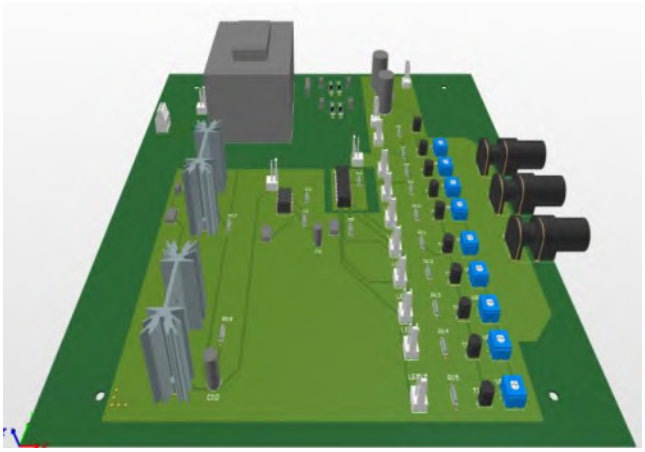
Svaka od komponenata koja se koristi za realizaciju projekta poseduje svoj odgovarajući simbol, footprint (otisak) i 3D model kako bi se mogla iskoristiti na štampanoj ploči. Za sve komponente korišćene u okviru ovog projekta, iskorišćeni su otisci koji se nalaze u okviru biblioteka koje su dodate u softver *Altium Designer*, osim za transformator čiji je otisak i 3D model realizovan na osnovu tehničke dokumentacije i osigurača čiji je 3D model takođe nacrtan na osnovu tehničke dokumentacije.

Na slici 3. prikazan je *layout* štampane ploče nakon raspoređivanja komponenata, izvršenog rutiranja, razlivanja mase i podešavanja širina linija za predviđenu jačinu struje na štampanoj ploči.



Slika 3. *Layout* štampane ploče u softveru *Altium Designer*.

Nakon izvršenih svih operacija na pločici i generisanja odgovarajućih fajlova potrebnih za njenu izradu ona se može videti u 3D prikazu (slika 4).



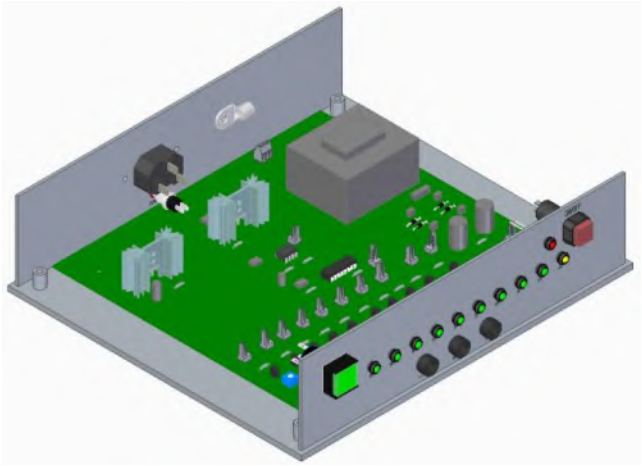
Slika 4. 3D prikaz projektovane ploče.

V. PROJEKTOVANJE KUTIJE UREĐAJA

Kutija za realizovani uređaj je projektovana u okviru softverskog paketa *Solid Edge* [9]. Na osnovu dimenzija pločice koja je realizovana, a koje iznose 210x210 mm, dimenzije kutije iznose 230x260 mm (slika 5). Kutija se sastoji iz četiri dela: gornjeg panela, donjeg panela i bočnih strana. Realizovana kutija ima mogućnost rasklapanja, nije fiksirana, kako bi se omogućio pristup štampanoj ploči u svakom trenutku ukoliko je neophodno servisiranje komponenata na njoj. Rastavljanje i sastavljanje kutije se vrši pomoću četiri zavrtnja. Materijal od kog je izrađena kutija je metal, pa to iziskuje i potrebu za uzemljenjem. Ako se pogleda kutija uređaja, na prednjem panelu su postavljene jedanaest LE dioda. Crvena, kao indikator rada (uključeno napajanje), žuta kao indikator aktivnog negativnog napajanja, i zelene diode koje omogućavaju korisniku da vidi koja je vrednost pozitivnog napona dostupna trenutno na izlazu. Na prednjem panelu su takođe postavljeni i odgovarajući tasteri, jedan koji služi za uključenje/isključenje uređaja i jedan koji služi za promenu trenutne vrednosti napona na izlazu. Mogu se uočiti i banana *plug* konektori, pomoću kojih će generisane napone moći da koriste odgovarajuća opterećenja i uređaji. Na zadnjem panelu kutije, nalazi se *AC male plug* pomoću koga se dovodi napajanje uređaju iz utičnice, kao i jedan taster koji služi za resetovanje brojača, a samim tim i vrednosti pozitivnog napona generisanog na izlazu. Na donjem panelu kutije pozicionirani su otvori pomoću kojih će se pričvrstiti ploča, kao i dodatni otvori za cirkulaciju vazduha i hlađenje. Na gornjem panelu mogu se uočiti prorezi širine 3mm koji služe za hlađenje unutrašnjosti kutije. Na krajevima gornjeg i donjeg panela (slika 6), postoje žlebovi za lakše sklapanje kutije koji ujedno i sprečavaju neželjeno pomeranje ovih delova kutije.



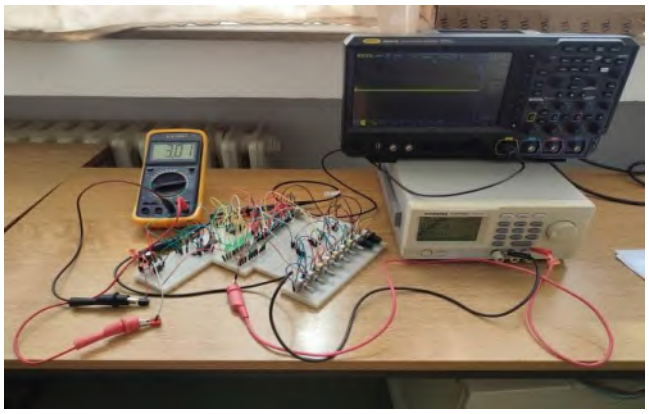
Slika 5. Prikaz celokupne kutije uređaja.



Slika 6. Kutija uređaja sa postavljenom štampanom pločom.

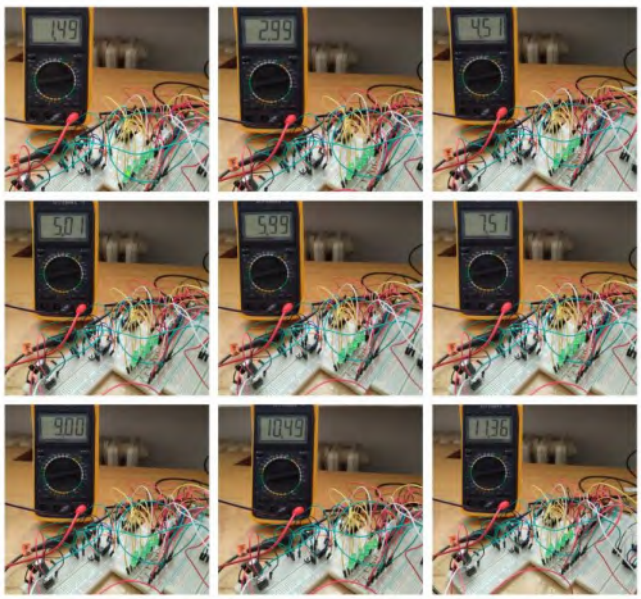
VI. PRAKTIČNA REALIZACIJA

Nakon projektovanja pločice u *Altium Designer* prešlo se na praktičnu realizaciju samog uređaja. Na slici 7. može se videti postavka merne opreme koja je potrebna kako bi se proverio rad projektovanog uređaja. Osciloskop se koristi u cilju provere promene vrednosti napona kojim se pobuđuje *NE555* tajmer.



Slika 7. Radno mesto u okviru Laboratorije za mikroelektroniku na Elektronskom fakultetu.

U nastavku (slika 8) prikazane su vrednosti napona koje su dobijene na izlazu uređaja nakon praktične realizacije i one se slažu sa vrednostima koje su očekivane pri čemu postoje eventualna mala odstupanja.



Slika 8. Prikaz vrednosti izmerenog napona na izlazu uređaja sa odgovarajućom svetlosnom signalizacijom u laboratoriji.

U tabeli II mogu se videti procentualna odstupanja dobijenog izlaznog napona u odnosu na očekivane vrednosti napona. Ova odstupanja javljaju se usled nemogućnosti finog podešavanja potenciometra u laboratorijskim uslovima.

TABELA II
PROCENTUALNA ODSTUPANJA IZLAZNOG NAPONA U ODNOSU NA OČEKIVANI

Očekivana vrednost napona [V]	Dobijena vrednost napona [V]	Odstupanje vrednosti napona u procentima [%]
1.5	1.49	0.6
3.0	2.99	0.3
4.5	4.51	0.2
5.0	5.01	0.2
6.0	5.99	0.16
7.5	7.51	0.13
9.0	9.0	0
10.5	10.49	0.09
12	11.36	5.3

VII. ZAKLJUČAK

Kroz ovaj rad prikazan je postupak projektovanja i praktične realizacije jednog izvora jednosmernog napona, u skladu sa trenutnim zahtevima i mogućnostima koji postoje u laboratorijskim uslovima.

Najpre je data blok šema uređaja nakon čega je prikazan princip rada i korišćene komponente tokom samog projekta. Jedan deo rada predstavlja i projektovanje štampane ploče i kutije za uređaj u okviru odgovarajućih softverskih alata. Na kraju rada prikazana je i praktična realizacija projekta u okviru Laboratorije za mikroelektroniku na Elektronskom fakultetu u Nišu.

Sam projekat podložan je raznim modifikacijama u smislu efikasnosti i funkcionalnosti.

ZAHVALINCA

Autori se zahvaljuju profesorima sa Katedre za mikroelektroniku na pruženoj pomoći i sugestijama u toku realizacije ovog projekta. Posebna zahvalnost prof. dr Danijelu Dankoviću i master inž. Aleksandri Stojković.

LITERATURA

[1] „*LM7912*“ Three-terminal negative 12 volts voltage regulator, Dostupno na: https://datasheetspdf.com/pdffile/662273/FairchildSemiconductor/LM7912/1

[2] Nikola Mitrović, Danijel Danković, Zoran Prijić, „LABORATORIJSKI PRAKTIKUM sa zadacima za samostalni rad iz predmeta ANALOGNA MIKROELEKTRONKA“, Edicija: Pomoćni udžbenici 2022. Elektronski fakultet Niš, ISBN 978-86-6125-245-7

[3] „*CD4017*“ 16 pin CMOS decade counter/ divider, Dostupno na: https://pdf1.alldatasheet.com/datasheet-pdf/view/50846/FAIRCHILD/CD4017.html

[4] „*LM317*“ Adjustable three terminal positive voltage regulator, Dostupno na: https://www.ti.com/lit/ds/symlink/lm317.pdf

[5] Z. Prijić, D. Danković, Elektronske komponente, Elektronski fakultet Niš, 2021. Dostupno na: http://mikro.elfak.ni.ac.rs/wpcontent/uploads/EK_Predavanja_01.pdf

[6] Z. Prijić, Analogna mikroelektronika, Elektronski fakultet Niš, 2018. Dostupno na: http://mikro.elfak.ni.ac.rs/wpcontent/uploads/2017/10/Literatura/Analogna/Analogna_lin_pred.pdf

[7] D. Danković, Z. Prijić, Digitalna Mikroelektronika, Elektronski fakultet Niš, 2007. Dostupno na: http://mikro.elfak.ni.ac.rs/wp-content/uploads/2017/10/Digitalna_Shift_registers.pdf

[8] „*Altium Designer*“, Dostupno na: https://www.altium.com/altium-designer

[9] „*Solid Edge*“, Dostupno na: https://www.plm.automation.siemens.com/store/en-us/trial/solid-edge.html

Projektovanje Power Bank uređaja

Nikola Milenković, Lazar Milenković, Aleksandra Stanković

Sadržaj – U ovom radu biće prikazan postupak realizacije Power Bank uređaja pomoću Altium Designer-a. Za ovaj projekat korišćeno je integrisano kolo TP4056 za punjenje litijum-jonske baterije i boost konvertor (XL6009) kao podizač napona na izlazu kola.

I. UVOD

Power Bank je u osnovi prenosivi uređaj koji može da napaja elektronske uređaje preko USB porta. U projektu je dizajniran Power Bank za napajanje, koji može da obezbedi izlaznu snagu od 5 V/3 A. Sadrži litijum-jonsku bateriju i kolo za punjenje, koje poseduje TP4056 modul, specijalno dizajniran za punjenje litijum-jonske baterije od 3.7 V i boost konvertor XL6009 za podizanje izlaznog napona. Litijum-jonska baterija skladišti energiju, koja se kasnije koristi za napajanje uređaja.

Mobilnim telefonima i drugim elektronskim uređajima neophodan je izvor napajanja od 5 V, ali litijum-jonska baterija sa jednom ćelijom daje maksimalni napon od 4.2 V. Zbog toga je potrebno na izlazu iz baterije dodati kolo za povećanje izlaznog napona na 5 V.

II. PRINCIP RADA

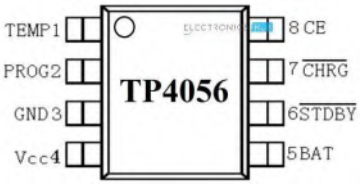
Kolo Power Bank-a se sastoji iz dva bloka:

- Kolo za punjenje baterije;
- Kolo za podizanje izlaznog napona;

A. Punjenje litijum-jonske baterije pomoću TP4056 kola

Litijum-jonska baterija od 3.7 V se koristi za skladištenje energije i potpuno je napunjena kada napon na njoj dostigne maksimalnih 4.2 V. Svaka baterija ima maksimalnu vrednost napona za koju je potpuno napunjena. Litijum-jonskom baterijom treba pažljivo rukovati, jer se baterija može zapaliti usled prekomernog punjenja. Zbog toga se za punjenje litijum-jonske baterije koriste posebna integrisana kola kao što je TP4056, koji automatski isključuju bateriju iz ulaznog napajanja kada je baterija potpuno napunjena [1].

Dodavanjem jednog programabilnog otpornika, kolo se može koristiti za punjenje litijum-jonske baterije od 3.7 V. Napon punjenja je fiksiran na 4.2 V i struja punjenja se može podesiti dodavanjem otpornika i kondenzatora u skladu sa baterijom koja se puni. Integrisano kolo takođe obezbeđuje unutrašnju termičku zaštitu i ograničenje struje. Nije potrebna dodatna dioda za blokiranje struje zbog unutrašnjeg P-MOSFET-a. TP4056 ima 8 pinova sa konfiguracijom [2] koja je prikazana na slici 1.



Slika 1. Raspored pinova kola TP4056.

TEMP (Pin 1): Ulazni pin temperaturnog senzora; Ako je napon na pinu ispod 45% ili iznad 80% napona napajanja Vin duže od 0.15 s, znači da je temperatura baterije ili previše niska ili previše visoka, pa se punjenje isključuje. Funkcionisanje ovog pina može da bude prekinuto njegovim kratkospajanjem na masu.

PROG (Pin 2): Pin za praćenje i podešavanje konstantne struje punjenja; Pin je setovan povezivanjem preko otpornika na masu. Napon na otporniku iznosi 0.2 V kada nema punjenja i 2 V u konstantnom režimu punjenja, dok se za merenje struje punjenja koristi srednja vrednost napona na otporniku (tipično 1 V). Odnos struje punjenja i otpornika za programiranje je data u sledećoj formuli (1):

$$I_{bat} = \frac{Vprog}{Rprog} \times 1200 \quad (V_{prog}=1\ V) \tag{1}$$

GND (Pin 3): Pin za masu;

VCC (Pin 4): Pin za napajanje; Kada ulazni napon padne na 30 mV od napona na pinu BAT, TP4056 prelazi u režim mirovanja i smanjuje struju BAT pina na manje od 2 µA.

BAT (Pin 5): Pin za povezivanje baterije; Mora biti povezan na pozitivan kraj baterije. BAT pin daje struju za napajanje baterije i reguliše napon na 4.2 V.

STDBY (Pin 6): Pin za status punjenja; Nakon završetka punjenja baterije, uključuje se pomoću unutrašnjeg prekidača (LOW), inače je u stanju visoke impedanse.

CHRG (Pin 7): Pin za status punjenja; Dok se baterija puni, pin se uključuje pomoću unutrašnjeg prekidača (LOW), inače je u stanju visoke impedanse.

CE (Pin 8): Pin dozvole rada; Logička jedinica na pinu stavlja uređaj u normalan režim rada, u suprotnom, integrisano kolo je isključeno.

Otpornik za programiranje (R3 ili Rprog) je podešen na 1.2 kΩ koji obezbeđuje brzinu punjenja baterije od

N. Milenković, L. Milenković i A. Stanković su studenti na modulu Elektronske komponente i mikrosistemi, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: a.aleksandra@elfak.rs, nikola.elfak@elfak.rs, milenkovic.lazar@elfak.rs.

1000 mAh. Ako kapacitet baterije nije 1000 mAh, onda se mora ukloniti otpornik (R3) i zameniti ga odgovarajućim, prateći podatke iz tabele:

TABELA I
VREDNOSTI PROGRAMABILNOG OTPORNIKA ZA ŽELJENU VREDNOST STRUJE PUNJENJA

R_{prog} (kΩ)	I_{bat} (mA)
10	130
5	250
4	300
3	400
2	580
1.66	690
1.5	780
1.33	900
1.2	1000

NTC otpornik, povezan na pin BAT, služi kako bi ograničio struju punjenja baterije, kada je baterija previše zagrejana, ili ako je previše hladna, dok kondenzatori C1 i C2 služe za uklanjanje šumova i regulaciju poskakivanja napona.

B. Boost konvertor XL6009

Kada se baterija u potpunosti napuni pomoću TP4056 kola, spremna je za korišćenje. Kako bi se dobili 5 V od litijum-jonske baterije, potrebno nam je kolo za povećanje napona. XL6009 regulator se koristi u kolu kojim se obezbeđuje regulisan i povećan napon za oko 1.6 puta veći od ulaznog napona iz baterije sa efikasnošću od 94%. XL6009 je DC-DC pretvarač koji je sposoban da generiše pozitivne ili negativne izlazne napone sa ulaznim naponom u opsegu od 3.6 V do 36 V. Konvertor ima ugrađeni N-kanalni MOSFET snage i oscilator fiksne frekvencije koji omogućava obezbeđivanje stabilnog izlaza u širokom opsegu ulaznih napona. XL6009 može da trpi napone napajanja na ulazu od -0.3 V do 42 V i da obezbedi maksimalni napon na izlazu od 42 V. Ima pet pinova sa sledećom konfiguracijom [3] (prikazana na slici 2):

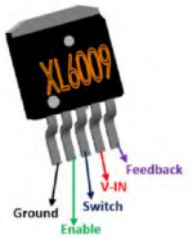
GND (Pin 1): Pin za masu;

EN (Pin 2): Pin za dozvolu rada; Postaviti EN pin u stanje logičke nule kako bi se kolo isključilo ili u stanje logičke jedinice kako bi se kolo uključilo. Podrazumevano je u stanju logičke jedinice.

SW (Pin 3): Izlazni pin prekidača za napajanje; Obezbeđuje pojačani napon.

VIN (Pin 4): Pin za napajanje; XL6009 radi sa jednosmernim naponom od 3.6 V do 36 V. Vin je povezano preko odgovarajućeg kondenzatora, koji eliminiše šumove sa ulaza, na GND.

FB (Pin 5): Povratni pin; Preko odgovarajućeg razdelnika napona, FB očitava i reguliše izlazni napon. Napon na izlazu pina je 1.25 V.



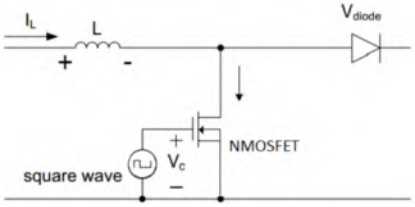
Slika 2. Raspored pinova kola XL6009.

Na ulazu i izlazu regulatora koriste se kondenzatori (C3 i C4) koji smanjuju neželjena poskakivanja i šum signala. C4 obezbeđuje regulisan, jednosmerni napon na izlazu. Kondenzator C6 vrednosti 1 µF je povezan paralelno sa kondenzatorom C4 da bi se smanjila ESR (ekvivalentna serijska otpornost) na izlazu (pošto kondenzatori visoke vrednosti imaju visoku ESR). Zener dioda (V_z = 5.6 V) na izlazu kola se koristi kao zaštita u slučaju od previsokog napona i mogućeg oštećenja uređaja prilikom punjenja.

Kalem povezan između pinova 3 i 4 igra važnu ulogu u pojačavaču. Glavna funkcija kalema je skladištenje energije. Što je veća vrednost induktivnosti, to će biti veća energija uskladištena u njemu, ali kalem visoke vrednosti takođe ima veću veličinu. Dakle, treba izabrati kalem koji može da obezbedi željenu struju na izlazu. U projektu su korišćeni kalem (L1) od 33 µH i Šotkijeva dioda D3 (SK34A). Dioda je izabrana jer ima manji napon vođenja i radi dobro na visokoj frekvenciji. Spisak pogodnih Šotkijevih dioda za integrisano kolo prema trenutnoj potražnji i ulaznom naponu može se naći u datasheet-u XL6009 konvertora.

C. Princip rada XL6009 konvertora

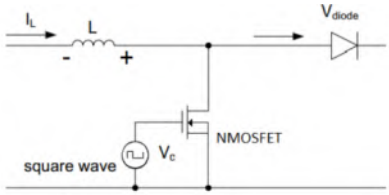
Interno XL6009 ima N-kanalni MOSFET snage sa fiksnom frekvencijom oscilatora. MOSFET deluje kao prekidački tranzistor, dok oscilator generiše kvadratni talas od oko 400 kHz (prema datasheet-u). Tokom pozitivne poluperiode kvadratnog talasa, kalem skladišti energiju i generiše magnetno polje tako da je na levom kraju kalema pozitivan, a na desnom negativan napon. Stoga je anoda diode na nižem potencijalu i deluje kao otvoreno kolo (slika 3).



Slika 3. Proces punjenja kalema tokom pozitivne poluperiode kvadratnog talasa.

Tokom negativne poluperiode, MOSFET se isključuje. Zbog toga, kalem ne može da se napuni. Sada

struja na kalemu generiše povratnu EMS (prema Lencovom zakonu) koja obrće polaritet kalema (kao što je prikazano na slici 4). Sada naelektrisanje uskladišteno u kalemu počinje da se prazni kroz diodu i na izlazu se dobija regulisani napon. U ovom slučaju, izlazni napon sada zavisi od uskladištenog naelektrisanja u kalemu, što je više uskladišteno naelektrisanje, to je veći izlazni napon. Prema tome, ako je vreme punjenja kalema duže, tada se povećava i uskladišteno naelektrisanje. Tako dobijamo dva izvora napajanja kao ulaz, jedan je kalem, a drugi je ulazno napajanje. Zbog toga uvek postoji izlazni napon veći od ulaznog.



Slika 4. Kalem kao izvor napajanja tokom negativne poluperiode.

D. Kolo sa razdelnikom napona

Budući da je granični napon unutrašnje povratne sprege XL6009 1,25 V, postoji konstantan napon na pinu 5 i konstantna struja će teći kroz potenciometar PR1 i otpornik R6. Prema tome, vrednost izlaznog napona Vout data je izrazom (2):

$$\text{Vout} = 1{,}25 \times (1+ (\frac{PR1}{R6})) \quad (2)$$

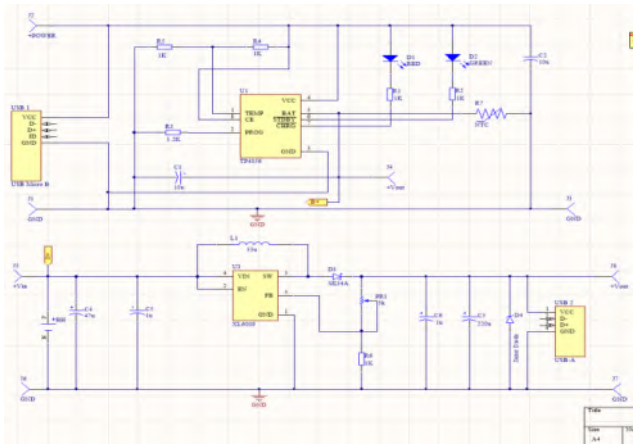
Izlazni napon nije tačno 5 V jer bilo koji uređaj koji zahteva 5 V ne radi tačno na 5 V. Potreban mu je veći napon od 5 V zbog nekih otpornih gubitaka u uređaju. Na primer, za 5.2 V na izlazu podešava se vrednost potenciometra na PR1 = 3.2 kΩ.

$$\text{Vout} = 1.25\ \text{V} \times (1+(\frac{3.2\ k\Omega}{1\ k\Omega})) \approx 5.2\ \text{V} \quad (3)$$

III. Realizacija u Altium Designer-u

Sistem zahteva stabilisano napajanje od 5 V koje se dobija iz adaptera ili računara. Jednosmerno napajanje se dovodi na ulaz USB1, koje se prosleđuje kolu za punjenje i baterija se puni. Dok se baterija puni, struja teče kroz crvenu LE diodu, ona svetli i pokazuje da je baterija u stanju punjenja. Kada napon na bateriji dostigne 4.2 V, odnosno, kada je baterija potpuno napunjena, TP4056 modul automatski isključuje pin CHRG i samim tim isključuje se crvena LE dioda i nema više struje kroz tu granu.

Sada struja teče kroz granu sa zelenom LE diodom, dioda vodi i pokazuje da je baterija napunjena. Vrednost napona na bateriji (4.2 V) je nedovoljna za punjenje uređaja, pa se prosleđuje na Boost konvertor, koji pojačava napon na željenu vrednost (u ovom slučaju na 5 V). Električna šema ovog sistema data je na slici 5 i realizovana je u softverskom paketu *Altium Designer* [5].



Slika 5. Električna šema u Altium-u.

A. Simboli, otisci i 3D modeli

Kako bi se realizovao projekat, bilo je neophodno ubaciti dodatne simbole, otiske i 3D modele, kojih nema u standardnim bibliotekama. U standardnim bibliotekama nema simbola za USB. Simboli, otisci i 3D modeli za USB1 (Micro B) i USB2 (USB A) su ubačeni i koriste se kao standardni ulazi za ovu vrstu uređaja. Svim elektrolitskim kondenzatorima promenjeni su otisak i 3D model, zbog njihove veličine. Previše veliki kondenzatori bi zahtevali veću zapreminu kućišta. NTC otporniku promenjen je otisak i 3D model, jer je korišćen SMD tip, a u standarnim bibliotekama je through hole. TP4056, držač za bateriju i XL6009 takođe se ne nalaze u biblioteci. Diode D3 i D4 u biblioteci su through hole, pa su promenjeni njihovi otisci i 3D modeli.

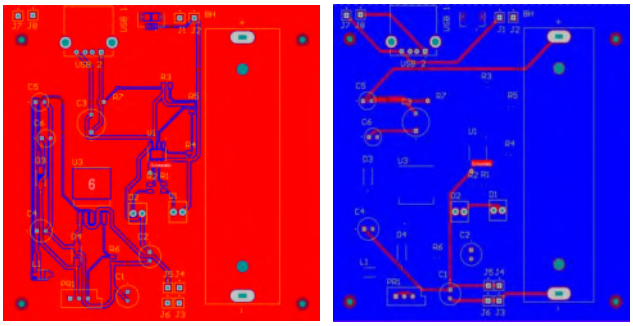
B. Layout

Sve komponente na štampanoj ploči povezane su linijama veze različitih debljina. U delovima kola, gde protiče jača struja, debljina linija je veća. Maksimalna debljina linija je 1.27 mm. Minimalna debljina linija, za delove kola gde protiče slabija struja je 0.254 mm. Šire linije su korišćene za GND, bateriju, USB1, USB2. Preko ploče, sa gornje i donje strane, razlivena je masa (prikazano na slikama 6 i 7).

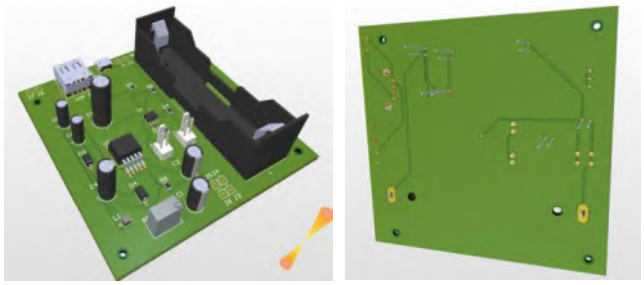
IV. Zaključak

Power Bank dizajniran u ovom projektu može se koristiti za punjenje bilo kog elektronskog uređaja kome je za rad potreban regulisan napon od 5 V i maksimalna struja od 3 A. Ima efikasnost oko 94% (efikasnost je odnos između energije koja se isporučuje na izlazu uređaja i energije uskladištene u njegovoj bateriji), a zbog upotrebe

odgovarajućih integrisanih kola ima unutrašnju prenaponsku i termičku zaštitu. Automatski se isključuje sa napajanja kada je potpuno napunjena baterija i ima LED indikatore za indikaciju tokom punjenja i kada se baterija potpuno napuni. Treba voditi računa da se baterija ne isprazni priključenjem na opterećenje, tokom procesa punjenja baterije. Istovremeno punjenje i pražnjenje može smanjiti trajanje baterije i može oštetiti TP4056. Najjednostavnije rešenje je da se uređaj nikada ne koristi dok se baterija puni. Čitav sistem mogao bi se optimizovati da ovo bude moguće, ali je potrebno uvesti neke dodatne izmene. Jedna od opcija koja bi mogla da funkcioniše zove se kolo „deljenja opterećenja" [4], koje omogućava ulaznom napajanju da direktno napaja opterećenje dok je uključeno, a takođe i nezavisno napaja TP4056 omogućavajući mu da napuni bateriju do kraja. Takvo kolo se sastoji od *Schottky* diode, P-kanalnog MOSFET-a i otpornika čiji princip rada i način povezivanja predstavlja zasebnu obimniju oblast.



Slika 6. 2D prikaz štampane ploče u Altium-u.



Slika 7. 3D prikaz štampane ploče u Altium-u.

ZAHVALNICA

Autori se zahvaljuju mentoru prof. dr Zoranu Prijiću i asistentkinji Aleksandri Stojković na pruženom znanju, pomoći i izdvojenom vremenu.

LITERATURA

[1] Designing a Power Bank. Dostupno na: https://www.engineersgarage.com/designing-a-power-bank-part-2-9/
[2] TP4056 Datasheet. Dostupno na: https://www.alldatasheet.com/view.jsp?Searchword=tp4056
[3] XL6009 Datasheet. Dostupno na: http://beriled.biz/data/files/XL6009.pdf
[4] Using the TP4056. Dostupno na: https://www.best-microcontroller-projects.com/tp4056.html
[5] Altium Designer. Dostupno na: https://www.altium.com/altium-designer

Život u Švajcarskoj i Master studije na Švajcarskom Federalnom Tehničkom Institutu u Cirihu (ETH Zürich)

Maša Nešić

Sadržaj – U ovom radu biće ukratko opisan studentski život u Cirihu, kao i Master studije na Švajcarskom Federalnom Tehničkom Institutu u Cirihu, poznatijem pod skraćenicom ETH (Eidgenössische Technische Hochschule). Cilj ovog rada je da studente upozna sa glavnim karakteristikama Švajcarske i grada Ciriha, kao i da pruži uvid u studentski život u ovoj oblasti i način realizacije Master studija na ETH.

I. UVOD

Švajcarska je površinski mala država koja se nalazi u srcu Evrope. Prepuna je prirodnih lepota i poznata je po visokom stepenu uređenosti i po opštoj umerenosti. Njen najurbaniji grad, često nazivan i njenim kulturološkim centrom, je Cirih. Iako najurbaniji grad Švajcarske, Cirih je po internacionalnim standardima zapravo jako miran grad. Ispunjen je prirodom i orijentisan oko velikog Ciriškog jezera i reka Limat i Sil. Pored toga, ovaj grad je poznat po tome što se u njemu nalazi najpoznatiji i najveći tehnički univerzitet u Švajcarskoj – ETH Cirih. ETH je univerzitet koji se vrlo često nalazi u prvih 10 na svetskim rang listama za oblasti nauke i inženjerstva. Poznat je po tome da je na njemu studiralo preko 20 laureata Nobelove nagrade, uključujući i Alberta Ajnštajna. Master studije na ovom univerzitetu odvijaju se u potpunosti na engleskom jeziku, što kao rezultat ima to da preko 50% studenata koji svake godine upišu Master studije na ETH dolazi iz inostranstva.

U nastavku ovog rada biće detaljnije opisani Švajcarska, Cirih i ETH iz ugla gledišta jedne studentikinje koja je nakon završetka Osnovnih studija na modulu Računarstvo i informatika na Elektronskom fakultetu, Univerziteta u Nišu upisala Master studije na departmanu za Računarstvo i informatiku na ETH.

II. ŠVAJCARSKA

Švajcarska je država koja se nalazi na spoju Zapadne, Centralne i Južne Evrope, pa se često kaže da se nalazi u samom srcu iste. Pokriva površinu od svega 41.285 km^2, pri čemu je više od 50% te, već prilično male, površine pokriveno Alpima. To znači da većina švajcarske populacije od 8,7 miliona stanovnika živi na jednoj trećini njene površine. Glavni grad Švajcarske je Bern, a njeni najveći gradovi su Cirih, Ženeva i Bazel. Ova 3 grada predstavljaju državne ekonomske centre, a svaki poseduje i internacionalni aerodrom.

Švajcarska, ili zvanično Švajcarska Konfederacija, je po uređenju federalna republika i sastoji se od 26 kantona. Svaki kanton predstavlja zasebnu saveznu državu, ima svoj Ustav, svoj parlament, vladu, policiju i svoje sudove. [1] Mapa švajcarskih kantona može se videti na Slici 1.



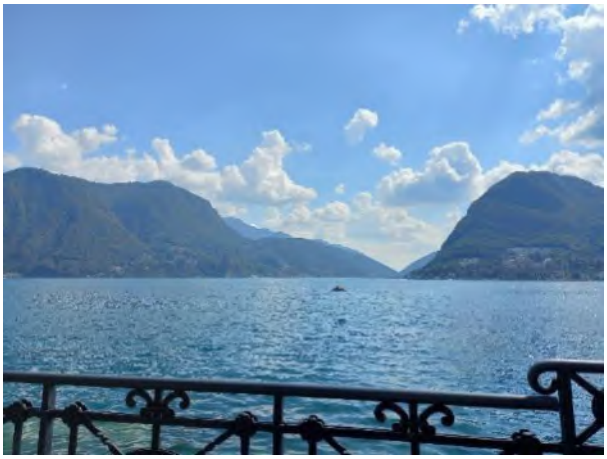
Slika 1. Mapa Švajcarskih kantona [2].

Interesantno je da na ovako maloj površini Švajcarska ima čak 4 različita lingvistička i kulturološka regiona – francuski, nemački, italijanski i retoromanski. Iako većina populacije govori nemački jezik i pripada nemačkom kulturološkom regionu (uključujući gradove Cirih, Bazel i Bern), sve ove kulture su duboko usađene u švajcarski identitet. Postojanje različitih lingvističkih regiona znači da ćete natpise u Švajcarskoj uglavnom viđati na 3 jezika – francuskom, nemačkom i italijanskom. To takođe znači da pri odlasku u lingvistički region različit od maternjeg, čak i Švajcarci pribegavaju korišćenju engleskog jezika radi uspostavljanja komunikacije. To predstavlja olakšavajuću okolnost za turiste i nas inostrane studente jer znači da se ovde možemo većinski snalaziti samo korišćenjem engleskog jezika. Još jedna interesantna činjenica je to da čak ni unutar jednog lingvističkog regiona govorni jezik nije homogen. Naime, ako pogledamo nemački region (koji je i najveći od pomenuta 4), unutar njega se može naći veliki broj različitih dijalekata. Pa je tako jezik koji se govori u Bernu drugačiji

M. Nešić, student Master studija na Departmanu za Računarstvo i informatiku (D-INFK) na Švajcarskom Federalnom Tehničkom Institutu u Cirihu (ETH Cirih), Rämistrasse 101, 8092 Cirih, Švajcarska, E-mail: mnesic@ethz.ch.

od onog koji se govori u Bazelu ili Cirihu, a svaki od njih je drugačiji od standardnog nemačkog koji se uči u školama (koji ovde zovu „visoki nemački“ ili hochdeutsch). Iako su ovi dijalekti ono što se koristi u svakodnevnom životu, gotovo svi Švajcarci iz nemačkog govornog regiona govore i hochdeutsch i upravo se na ovom „školskom“ nemačkom obavlja sva zvanična komunikacija. Suprotno od toga, švajcarske verzije italijanskog i francuskog vrlo malo se razlikuju od „školskih“ oblika ovih jezika.

Ova država bogata je jezerima i rekama – svaki veći ili grad srednje veličine leži na nekoj vodenoj površini. To doprinosi činjenici da su svi švajcarski gradovi u bliskom kontaktu sa prirodom. Iz urbanijih oblasti se vrlo lako, za manje od pola sata, pešaka ili biciklom može doći u predeo ispunjen prirodnim lepotama u kojem imate osećaj da se grad nalazi desetinama kilometara daleko od vas (Slika 2 i Slika 3).



Slika 2. Jezero Lugano, pogled iz centra istoimenog grada.



Slika 3. Reka Rajna, u blizini Šafhauzena i vodopada na Rajni.

Kako je površina ove države dobrim delom pokrivena Alpima, postoji puno opcija za planinarenje i zimske sportove tokom gotovo čitave godine (Slika 4).



Slika 4. Ski touring ka vrhu Rauflihorn, Bernski Alpi.

Još jedna karakteristika po kojoj je Švajcarska poznata je jako kvalitetna mreža javnog prevoza. U javni prevoz spadaju tramvaji, trolejbusi, vozovi i brodovi za rečni i jezerski saobraćaj. Metodama javnog prevoza se u Švajcarskoj može doći do bilo kog dela države, uključujući i brojne planinske vrhove. Prevoz je udoban, efikasan i vrlo strogo prati predviđeni raspored. Vrlo često i predstavlja najefikasniji način prevoza od tačke A do tačke B jer u urbanim područjima javni prevoz ima veći prioritet od privatnih automobila, a pri dužim putovanjima železnica prati najoptimalniju rutu, za razliku od autoputeva.

U švajcarskim planinskim predelima uzgoj krava je jako rasprostranjena delatnost. Mleko koje one daju koristi se za proizvodnju sira i čokolade u izobilju. Teško je reći koji od ta dva proizvoda Švajcarci vole više. Proizvode se razne vrste sira, uglavnom je u pitanju tvrd sir i važi pravilo da što je stariji (i jačeg mirisa), to bolje! Najpoznatija vrsta sira je Grujer, koji je dobio naziv po oblasti na zapadu države od čijih se krava prikuplja mleko za proizvodnju ove vrste sira. Najpoznatija jela u Švajcarskoj upravo sadrže sir, odnosno su gotovo u potpunosti sačinjena od njega, to su fondi i raklet.

III. Cirih

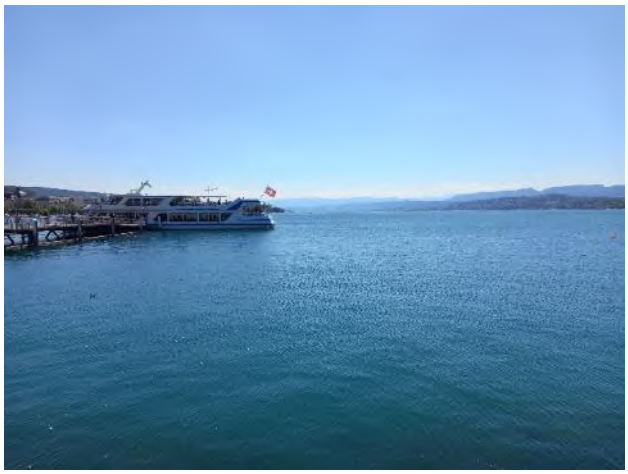
Cirih je najveći grad u Švajcarskoj i njen kulturološki centar. Površna same opštine je svega 87,88 km2 i na njoj živi nešto manje od pola miliona ljudi. Međutim, s obzirom na visoke cene stanovanja u samoj opštini, većina stanovništva gradskog područja živi na obroncima grada. Tako da kada se uzme u obzir čitavo gradsko područje, broj stanovnika postaje 1,83 miliona. Grad je prožet rustičnom germanskom arhitekturom, brojnim katedralama, parkovima i prirodom. Pogled na jezgro grada Ciriha iz parka Lindenhof može se videti na Slici 5.

Cirih pripada nemačkom lingvističkom i kulturološkom regionu i njegov zvanični jezik jeste nemački. Međutim, govorni jezik u ovom gradu je

takozvani ciriški nemački, lokalna varijanta švajcarskog nemačkog dijalekta. Ovaj dijalekt značajno se razlikuje od „školskog“ nemačkog jezika i čak je i govornicima kojima je nemački jezik maternji jako teško da razumeju razgovore na ovom dijalektu. Na sreću, sva zvanična dokumentacija je na „školskom“ nemačkom jeziku tako da je znanje istog ipak od velike koristi. Takođe, većina stanovnika pored ciriškog dijalekta govori i „školski“ nemački, te se mogu prilagoditi sagovorniku po potrebi. U većini slučajeva se u Cirihu može koristiti i engleski jezik. Međutim, poznavanje nemačkog jezika je od velike koristi za razumevanje natpisa, tumačenje pošte dobijene od gradskih, kantonskih ili državnih vlasti i sporazumevanje u prodavnicama, restoranima ili apotekama, gde radnici ne govore uvek engleski jezik.



Slika 5. Cirih, Lindenhof, pogled na reku Limat.



Slika 6. Polazna stanica za brodski saobraćaj na Ciriškom jezeru.

U samom srcu Ciriha nalazi se prilaz do Ciriškog jezera (Slika 6). Samo jezero je jako dugačko i prostire se i do obližnjih kantona Sen Galen i Švic. Oko jezera se sa obe strane nalaze parkovi i šetališta, pa se u večernjim satima i vikendom ovaj prostor ispuni ljudima koji uživaju u pikniku, rekreaciji ili odmoru. Pored jezera, Cirih ima i dve reke – Limat i Sil, koje protiču kroz njegov centar. Leti Limat predstavljaju omiljeno kupalište za Cirižane, a tokom čitave godine se njome odvija i javni prevoz brodićima. Na zapadnoj strani, tik pored grada, uzdiže se planina Itliberg čiji je najviši vrh na 870 m nadmorske visine [3]. Sa njenog vrha pruža se divan pogled na čitav grad koji se može videti na Slici 7. Itliberg je česta destinacija za planinarenje, a zimi je često prekriven snegom tako da postaje i destinacija za sankanje i druge zimske aktivnosti.



Slika 7. Pogled na Cirih sa planine Itliberg.

U gradu je dostupan javni prevoz tramvajima, trolejbusima, rečnim i jezerskim brodovima. Pored javnog prevoza, veliki broj ljudi kretanje po gradu obavlja i biciklama. Biciklističke staze su obeležene po kolovozu u čitavom gradu i biciklisti imaju prednost u saobraćaju u odnosu na automobile. Pored personalnih bicikala, u gradu postoje i programi za iznajmljivanje običnih i električnih bicikala po potrebi.

Kako je Cirih kulturološki centar Švajcarske, sadrži brojne muzeje i festivale. U samom centru grada, pored glavne železničke stanice, nalazi se Švajcarski Nacionalni Muzej. U blizini Ciriškog jezera, nalaze se zgrada Opere i Muzej umetnosti. Neki od festivala koji se održavaju na godišnjem nivou jesu Ciriška ulična parada i Open Air festival. Za vreme Ciriške ulične parade, čitav grad biva preplavljen stotinama hiljada ljudi koji čitavog dana imaju prilike da uživaju u muzici koja dolazi sa desetak različitih bina širom grada (Slika 8). Sa druge strane, Open Air festival traje nekoliko dana, održava se na obroncima grada i na njemu svake godine gostuju brojna svetska imena pop i rok muzike.

Cirih kao grad ima puno vrlina – prelepu prirodu, jezero, reke, veliki broj znamenitosti... Ali ima i jednu veliku manu – jako je skup. Za 2020. godinu bio je proglašen najskupljim gradom za život na svetu, za 2021. godinu zauzeo je 4., a za 2022. za sada zauzima 7. mesto na toj rang listi [5]. Ostaje samo da se nadamo da će u narednim godinama nastaviti sa opadanjem na ovoj rang listi.



Slika 8. Grad za vreme Ciriške ulične parade [4].

IV. ETH CIRIH

ETH je zapravo skraćenica za „Eidgenössische Technische Hochschule“, što u prevodu znači „Švajcarski Federalni Tehnički Institut“, ali je sama skraćenica toliko poznata i rasprostranjena da se koristi i u drugim jezicima, bez prevođenja. Logo univerziteta se može videti na Slici 9.

Slika 9. Logo ETH Cirih [6].

ETH je najveći i najpoznatiji švajcarski tehnički univerzitet koji je osnovan 1854. godine sa ciljem da obrazuje naučnike i inženjere. Glavne oblasti od interesa su nauka, tehnologija, inženjerstvo i matematika. Tako se na ovom univerzitetu ne mogu studirati društvene nauke, već samo prirodne nauke i inženjerstvo. ETH je poznat po tome da se godinama unazad nalazi u samom vrhu liste najboljih univerziteta na svetu. Za oblasti inženjerstva i tehnologije zauzima 6. mesto na svetu za 2022. godinu i 4. mesto za 2021. U oba slučaja to predstavlja 1. mesto za kontinentalnu Evropu. [7] Prestižu ovog univerziteta doprinosi činjenica da se za njega vezuje 22 laureata Nobelove nagrade, 2 dobitnika Fildsove medalje, 3 dobitnika Pritzkerove nagrade za arhitekturu i 1 dobitnik Tjuringove nagrade, među kojima su i Albert Ajnštajn i Džon fon Nojman. [8]

ETH Cirih se sastoji od 2 kampusa – centralnog (Slika 10), koji se nalazi u samom centru grada, i kampusa Hongerberg (Slika 11), koji se nalazi na istoimenom brdu na obroncima grada Ciriha. Svaki departman ima svoju zgradu na jednom od ova dva kampusa, a studentima se nudi i besplatan prevoz između njih. Prevoz se obavlja trolejbusima koji se nazivaju ETH Link koji voze direktno od jednog do drugog kampusa, bez međustanica. Dodatno, vremena početaka predavanja na oba kampusa i vremena polazaka ETH Link trolejbusa su tako usklađena da se uvek može stići sa jednog na drugi kampus za 2 uzastopna predavanja.



Slika 10. Centralni kampus ETH, glavna zgrada [9].



Slika 11. Kampus Hongerberg [10].

A. Stipendije za Master studente

Sama školarina za studiranje na ETH nije previše visoka i iznosi 1.229 franaka na godišnjem nivou, što je po trenutnom kursu 151.646 dinara (neki fakulteti u Srbiji imaju veće cene školarine). Međutim, život u Cirihu je jako skup pa je često studentima, pogotovo onima koji dolaze iz država sa znatno nižim standardom od švajcarskog, potrebna finansijska pomoć.

Sam ETH nudi odlične programe stipendiranja za Master studente koji se baziraju na dosadašnjem uspehu na studijama i perspektivi za doprinos nauci u oblasti rada. Prijava za ove programe stipendiranja vrši se pri samoj prijavi na Master studije i zahteva od dodatne dokumentacije još jedno pismo preporuke i predlog istraživačkog rada, odnosno Master teze kojom student planira da se bavi u budućnosti. Kandidati koji prođu prvi krug selekcije bivaju individualno kontaktirani za razgovor sa grupom profesora sa univerziteta koji se bave oblašću u kojoj je student napisao predlog istraživačkog rada. Na ovaj način student se istovremeno prijavljuje za program

stipendiranja ESOP – Excellence Scholarship & Opportunity Programme i ETH-D. Oba ova programa stipendiranja obezbeđuju oslobođenje od obaveze plaćanja školarine, a razlikuju se po količini novca koju student dobija na godišnjem nivou i po obavezama koje ima prema univerzitetu.



Slika 12. ESOP stipendisti za 2021. godinu.

ESOP program predstavlja stipendije rektora ETH i ETH fondacije i u okviru njega student dobija 24.000 franaka godišnje i nema nikakvu radnu obavezu prema univerzitetu, niti stupa u radni odnos sa istim. ETH svake godine dodeli oko 60 stipendija ovog tipa na nivou čitavog univerziteta, što znači da se 2-3 stipendije dodele svakom departmanu. Konkurencija za ovu stipendiju je jako velika jer se za 60-ak stipendija svake godine prijavi više stotina kandidata. Ali to ne znači da ne treba pokušati, studenti koji su pokazali izuzetan uspeh tokom osnovih studija i koji su zaista motivisani za istraživački rad i pokazuju visok nivo interesovanja i upoznatosti sa aktuelnim istraživačkim temama i te kako imaju šanse da dobiju ovu stipendiju. ESOP stipendisti za 2021. godinu, među kojima se nalazi i autorka ovog rada, nalaze se na slici 12.

ETH-D je program stipendiranja koji finansiraju sami departmani univerziteta. U okviru njega student dobija između 18.000 i 20.000 franaka godišnje, pri čemu se deo te sume isplaćuje kao stipendija ili nagrada, a deo kao plata. To znači da student stupa u radni odnos sa univerzitetom, potpisuje ugovor o radu i plaća porez na novac koji dobija kao kompenzaciju na taj način. Dodatno, zavisno od departmana, student mora da odradi određeni broj sati kao asistent da bi zadržao ovu stipendiju. Tačan broj ETH-D stipendija koji se dodeli svake godine nije poznat i zavisi od departmana koji ih dodeljuje, ali je svakako znatno veći od broja ESOP stipendija.

B. Studentski smeštaj

ETH nudi određeni broj smeštajnih jedinica za prvih godinu dana za studente koji dolaze iz država koje nisu članice Evropske Unije, niti Evropske asocijacije za slobodnu trgovinu (EU/EFTA) i koji nikada do tada nisu bili u Švajcarskoj. Link za prijavu za ove smeštajne jedinici stiže na email adresu svih studenata koji ispunjavaju uslovne kriterijume još aprila meseca za semestar koji počinje septembra iste godine. Broj ovih smeštajnih jedinica je mali pa je konkurencija velika te dodatna prednost studentu može biti ako je stipendista ESOP ili ETH-D programa. Dodatno, za poređenje studenata koji imaju jednake prioritete važno je vreme kada je prijava poslata pa je jako bitno prijavu popuniti u što kraćem roku od pristizanja email-a sa linkom ka formi za prijavu.



Slika 13. ETH Student Village [11].

ETH takođe nudi dva moderna studentska doma na Hongerberg kampusu – Student village (Slika 13) i Living science (Slika 14). Stanovi u ovim domovima su koncipirani tako da svaki student ima svoju sobu koja je opremljena osnovnim nameštajem, a kupatilo i kuhinju deli sa određenim brojem drugih studenata koji može da varira od 1 do 10. Cena ovakvog smeštaja se kreće između 400 i 1.000 franaka mesečno. Iako prijava za ovaj smeštaj teoretski jeste moguća preko veb sajta doma ([11] i [12]), najčešće studenti iz istog stana oglase o slobodnim sobama postavljaju na društvene mreže kako bi mogli da sprovedu niz intervjua sa kandidatima (kako se to ovde naziva – kasting), kako bi odabrali osobu koja im najviše odgovara za zajednički život.



Slika 14. ETH Living science [12].

Pored ove dve opcije, odličnu opciju za studentski smeštaj nudi i WOKO. Ova organizacija pruža smeštajne jedinice za studente širom Ciriha po subvencionisanim cenema. Slično kao i za studentske domove u okviru ETH Hongerberg kampusa, i za ove smeštajne jedinice se teoretski možete prijaviti putem veb sajta ([13]), ali se u realnosti većina soba dodeli studentima preko oglasa podeljenim na društvenim mrežama.

U svakom slučaju, povoljan smeštaj u Cirihu nije lako naći pa je ključ u tome da potragu počnete na vreme – 4 do 5 meseci pre predviđenog datuma selidbe. Pored prethodno opisanih studentskih opcija, potragu za smeštajem možete vršiti i putem sajta WG Zimmer ([14]).

C. Studentska ishrana

Kako je Cirih prilično skup grad za život, cene hrane su takođe prilično visoke pa su opcije za studentsku ishranu i te kako dobrodošle. Međutim, iako su cene hrane u ciriškim menzama niže nego u bilo kom restoranu i niže od cena ulične hrane, i dalje je povoljnije hranu spremati kod kuće nego kupovati je u menzi.

ETH studenti imaju na raspolaganju sve ETH menze i sve menze Univerziteta u Cirihu (skraćeno UZH). Ove menze su lokalizovane oko centralnih kampusa ova dva univerziteta, koji se nalaze tik jedan uz drugi u samom centru grada, oko ETH Hongerberg kampusa, UZH Irhel kampusa i UZH botaničke bašte.

Sve menze nude 1 ili 2 obroka dnevno (ručak i večeru ili samo ručak) radnim danima tokom trajanja semestra i ispitnih rokova, dok se za obroke vikendom, za vreme praznika ili raspusta studenti moraju snalaziti samostalno. Svaka menza nudi izbor između nekoliko različitih menija (2-6), pri čemu je uvek veći broj opcija veganski ili vegetarijaski. Takođe, za menije koji sadrže meso studenti moraju da plate cenu koja je za oko 1,1 frank veća od cene vegetarijanskih ili veganskih opcija. Cena obroka u menzi kreće se od 5,8 do 15 franaka.

D. Sport

Značajan deo studentskog života na ETH čini sport. Organizacija koja se bavi pružanjem mogućnosti za bavljenje sportom u okviru ETH naziva se ASVZ. Ova organizacija nudi prostore za bavljenje sportom u blizini kampusa ETH i UZH i na još nekoliko dodatnih lokacija širom Ciriha. Za korišćenje ovih usluga studenti plaćaju simboličnu cenu od 30-ak franaka po semestru. Spisak sportova kojima se studenti mogu baviti preko ogranizacije ASVZ je gotovo beskrajan i uključuje individualne i grupne sportove, borilačke veštine, ples, zimske sportove, vodene sportove i mnoge druge. Spisak svih sportova se može naći na veb sajtu organizacije ([15]), gde se može naći i mapa sa lokacijama svih prostorija u kojima se sportovi mogu obavljati. Jedan od događaja koji ASVZ organizuje svake godine je štafetna trka SOLA (Slika 15) u okviru koje svake godine učestvuje preko 10.000 ljudi.



Slika 15. Štafetna trka SOLA u organizaciji ASVZ.

E. Vannastavne aktivnosti

Krovno studentsko udruženje na nivou čitavog ETH naziva se VSETH, i u okviru njega svaki departman ima sopstveno studentsko udruženje koje je zaduženo za organizovanje društvenih događaja na nivou departmana. Ovih događaja ima tokom čitave godine i raznolike su prirode. Organizuju se vikend putovanja radi planinarenja ili skijanja, jednodnevni izleti do zabavnih parkova i akvaparkova, odlasci na aktivnosti kao što su Laser tag (Slika 16) i skakanje na trambolinama. Organizuju se i žurke, zajedničke večere i piknici u prirodi. Studentsko udruženje departmana za Računarstvo i informatiku pod nazivom VIS osnovalo je i kafić u podrumskim prostorijama departmana koji u potpunosti vode studenti i koji je otvoren svake srede tokom trajanja semestra i u kojem su cene pića prilagođene studentskom budžetu.

F. Studentski projekti

Ukoliko se interesujete i za teme koje nisu usko vezane za trenutni akademski put, ETH pruža okvir i za njihovo istraživanje.

Klub preduzetnika omogućuje studentima koji imaju želju da otvore sopstvenu kompaniju da upoznaju jedni druge, da nauče tehnike preduzetništva, ekonomije i poslovnog sveta. Zahvaljujući ovakvom pristupu od 2018. godine do sada na ETH je nastalo 429 kompanija. [8]

Dodatno, studenti koji vole da se bave kreativnim projektima imaju na raspolaganju 2 velike zgrade pod nazivom Student Project House (po jednu na svakom kampusu), u kojima imaju sve neophodne alate za kreativan rad poput 3D štampača, mašina za obradu drveta, elektronskih komponenti itd.

G. Master studije na departmanu za Računarstvo i informatiku

Master programi na Departmanu za računarstvo i informatiku (skraćeno D-INFK) traju 4 semestra, odnosno nose vrednost od 120 ESPB i sprovode se u potpunosti na

engleskom jeziku. Ovaj departman nudi 3 različita Master programa:

- Computer Science
- Data Science
- Cyber security

Computer Science je najopštiji Master program od pomenuta 3 i okviru njega student bira glavnu temu fokusa (major) i sporednu temu fokusa studija (minor). Kombinacijom ove dve teme definiše se profil kojim će student moći da se identifikuje u daljim koracima, nakon završetka Master studija. Postoji 5 različitih oblasti koje se mogu odabrati kao major [16]:

- Sistemi za upravljanje podacima
- Mašinska inteligencija
- Bezbedni i pouzdani sistemi
- Vizuelno i interaktivno računarstvo
- Teorijska informatika.

I 9 različitih oblasti koje se mogu odabrati kao minor [16]:

- Računarska grafika
- Računarski vid
- Upravljanje podacima
- Bezbednost informacija
- Mašinsko učenje
- Računarske mreže
- Programski jezici i softversko inženjerstvo
- Sistemski softver
- Teorijska informatika

pri čemu nije svaka kombinacija dozvoljena. Dozvoljene i zabranjene kombinacije mogu se videti na Slici 17.



Slika 16. Laser tag u organizaciji studentskog udruženja VIS.

Na ovom Master programu ne postoje obavezni predmeti, student u potpunosti samostalno određuje kada će i koje predmete da sluša, ali na kraju Master studija mora da stekne odgovarajući broj ESPB za svaku kategoriju iz tabele koja se može videti na Slici 18.

Kategorije uključuju praktičan rad, master tezu, major, minor, seminar, inter-fokus predmete i nauku iz druge perspektive. Nauka iz druge perspektive je kategorija koja obezbeđuje raznovrsnost u okviru Master programa. U okviru nje student sluša predmete koji su društvenog karaktera i prate tematiku istorije, filozofije, psihologije, jezika itd. Inter-fokus predmeti služe da odrede osnovni skup znanja koji svaki student mora da ima nakon Master studija. U ponudi su 4 predmeta od kojih student bira 2 i neophodno je da iz 4 pokušaja položi ta 2 predmeta kako bi mogao da završi Master studije. 4 ponuđena predmeta su:

- Napredni sistemi
- Bezbednost informacija
- Algoritmi
- Računarska inteligencija.

Predavač za svaki predmet zasebno definiše način polaganja, odnosno koliko procenata finalne ocene se alocira predispitnim obavezama, a koliko ispitu, kada se taj ispit održava i kojeg je tipa (pisani, usmeni, na računaru). Ispit se može održati u sredini semestra (midterm exam), na samom kraju semestra (semester exam) ili u ispitnom roku (session exam). Raspon ocena je od 1 do 6 sa korakom od 0,25, pri čemu je ocena 6 najbolja, a prelaznim ocenama se smatraju sve ocene veće ili jednake oceni 4.

Majors: \ Minors:	Computer Graphics	Computer Vision	Data Management	Information Security	Machine Learning	Networking	Programming Languages & Software Engineering	Systems Software	Theoretical Computer Science
Data Management Systems	✓	✓	×	✓	✓	✓	✓	×	✓
Machine Intelligence	✓	×	✓	✓	×	✓	✓	✓	✓
Secure and Reliable Systems	✓	✓	✓	×	✓	✓	×	✓	✓
Visual and Interactive Computing	×	×	✓	✓	✓	✓	✓	✓	✓
Theoretical Computer Science	✓	✓	✓	✓	✓	✓	✓	✓	×

Slika 17. Mogućnosti kombinovanja tema kao major i minor [16].

Master ETH Zurich in Computer Science	120
Major	26
Core Courses Core Electives	16
Minor	18
Inter Focus Courses	16
Seminar	2
Practical Work	8
Free Elective Courses	
Science in Perspective	2
Master's Thesis	30

Slika 18. Kategorije i minimalni broj neophodnih ESPB. [17]

V. ZAKLJUČAK

Kao i svaka država i grad, Švajcarska i sam Cirih poseduju specifičnosti na koje se treba privići. Neke od tih specifičnosti su pozitivne, poput neverovatne tačnosti i gotovo besprekornog funkcionisanja javnih servisa; a neke su negativne, poput nedostatka društvenog života tokom radne nedelje i osećaja otuđenosti među ljudima. Finalni

doživljaj ove države i ovog grada naravno zavisi od ličnih prioriteta i naklonosti, i toga koliku težinu pridajete pozitivnim i negativnim specifičnostima. Ali nezavisno od tog „finalnog suda“, studentski život u Cirihu i studiranje na ETH su zaista nešto posebno i vredno doživljavanja.

Brojne prirodne lepote i široka kulturološka raznovrsnost su karakteristike Švajcarske koje je čine divnom zemljom za istraživanje i obilazak, dok Cirih predstavlja jedinstvenu kombinaciju urbanog života, kulture i kontakta sa prirodom.

Čak i ako izuzmemo faktor lokacije, ETH Cirih kao univerzitet je mesto vredno posećivanja. On pruža priliku za lični razvoj u svim pravcima. Internacionalno i multikulturalno okruženje omogućuje proširenje vidika i upoznavanje sa običajima, kulturama i kuhinjama širom sveta. Vannastavne aktivnost, sport i prilike za bavljenje projektima koji nisu usko vezani za oblast studiranja čine studentski život u okviru ETH jako bogatim. Po pitanju samih studija i akademskog aspekta, ETH nudi vrlo izazovno okruženje, predavače koji su među najboljima na svetu u svojim oblastima rada i kurikulum koji prati najnovije tehnološke trendove i naučna otkrića. U okviru ETH studenti imaju priliku da rade na problemima koji do sada nisu rešeni pod mentorstvom vrhunskih stručnjaka.

LITERATURA

[1] „Switzerland“, https://en.wikipedia.org/wiki/Switzerland, (Datum poslednjeg pristupa 24.09.2022.)

[2] KarzA - Own work, Public Domain, https://commons.wikimedia.org/w/index.php?curid=5412012

[3] „Uetliberg“, https://de.wikipedia.org/wiki/Uetliberg, (Datum poslednjeg pristupa 25.09.2022.)

[4] „Street parade“, https://www.streetparade.com/en/multimedia/pictures/

[5] „The world's most expensive cities for 2022“, https://edition.cnn.com/travel/article/world-most-expensive-cities-2022-intl-hnk/index.html, (Datum poslednjeg pristupa 25.09.2022.)

[6] „ETH Zürich“, https://ethz.ch/, (Datum poslednjeg pristupa 25.09.2022.)

[7] „QS World University Rankings by Subject 2022: Engineering & Technology“, https://www.topuniversities.com/university-rankings/university-subject-rankings/2022/engineering-technology, (Datum poslednjeg pristupa 25.09.2022.)

[8] „ETH Zurich“, https://en.wikipedia.org/wiki/ETH_Zurich, (Datum poslednjeg pristupa 25.09.2022.)

[9] „Luftbild ETH Zentrum“, ETH Zürich / Alessandro Della Bella

[10] „Luftbild Hönggerberg“, ETH Zürich / Alessandro Della Bella

[11] „Student Village“, https://studentvillage.ch/en/accommodation/, (Datum poslednjeg pristupa 25.09.2022.)

[12] „Living science“, https://www.livingscience.ch/wohnen-studieren-zuerich/, (Datum poslednjeg pristupa 25.09.2022.)

[13] „WOKO“, https://www.woko.ch/, (Datum poslednjeg pristupa 25.09.2022.)

[14] „WG Zimmer“, https://www.wgzimmer.ch/en/wgzimmer/search/mate.html, (Datum poslednjeg pristupa 25.09.2022.)

[15] „ASVZ“, https://asvz.ch/, (Datum poslednjeg pristupa 25.09.2022.)

[16] „Master in Computer Science Regulations 2020“, https://inf.ethz.ch/studies/master/master-cs-2020.html, (Datum poslednjeg pristupa 25.09.2022.)

[17] „Brochure – Master in Computer Science“, https://ethz.ch/content/dam/ethz/special-interest/infk/department/Images%20and%20Content/Studies/Master/CS/Brochure%20Master%20in%20Computer%20Science.pdf, (Datum poslednjeg pristupa 03.10.2022.)

Linearni izvor napajanja sa promenljivom vrednošću izlaznog napona korišćenjem ATMEGA328P mikrokontrolera i digitalnog potenciometra

Boris Jevtić

Sadržaj – U ovom radu detaljno su opisani projektovanje i realizacija uređaja koji ima za cilj da naizmenični napon iz gradske mreže efektivne vrednosti 230 V ispravi na diskretne vrednosti jednosmernog napona od 3.3 V, 5 V, 9 V ili 12 V radi daljeg korišćenja.

I. UVOD

Uređaj, čije su projektovanje i realizacija opisani u ovom radu, pre svega je namenjen početnicima na polju elektronike. Realizacija uređaja je moguća pod pretpostavkom da je u radu dostupna sledeća oprema: štapna lemilica, pinceta i držač sa lupom. Sve elektronske komponente koje se postavljaju na štampanu ploču odlikuje *through-hole* način montaže.

II. PRINCIP RADA UREĐAJA

Naizmenični mrežni napon efektivne vrednosti 230 V dovodi se preko osigurača do primarnog namotaja transformatora. Naizmenični napon na sekundaru dovodi se na Grecov spoj gde se vrši njegovo ispravljanje.

Ispravljeni napon dovodi se na dva naponska regulatora, LM317 i L7805CV.

Odabir željene vrednosti napona na izlazu uređaja vrši se okretanjem klizača potenciometra koji je povezan na jedan od ulaza analogno-digitalnog konvertora mikrokontrolera. U fazi odabira, jedna od četiri RGB LE diode emitovaće crvenu svetlost, u zavisnosti od položaja klizača potenciometra odnosno željene vrednosti napona na izlazu. Pritiskom na taster za odabir, a u zavisnosti od vrednosti odabranog napona, jedna od četiri RGB LE diode emitovaće zelenu svetlost obaveštavajući time korisnika koja je vrednost napona na izlazu odabrana. Da bi korisnik promenio vrednost napona na izlazu uređaja, neophodno je da uređaj resetuje pritiskom na taster za reset i ponovo izvrši odabir.

III. PROJEKTOVANJE ŠTAMPANE PLOČE U PROGRAMU ALTIUM DESIGNER

U cilju jednostavnijeg postupka projektovanja, uređaj je podeljen na tri zasebna bloka: blok napajanja, blok mikrokontrolera, izlazni blok.

B. Jevtić je student osnovnih akademskih studija na modulu Elektronske komponente i mikrosistemi, Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija. E-mail: borisjevtic@elfak.rs.

A. Blok napajanja

Naizmenični napon se sa sekundara transformatora preko pinova J_1 i J_2 dovodi na integrisani Grecov spoj BR_1 (slika 1). Na ulaz u deo kola za regulaciju postavljeni su elektrolitski kondenzator C_1, čija je uloga da kompenzuje kratkotrajne propade i eventualne prekide mrežnog napona i keramički kondenzator C_2 koji ima ulogu bypass kondenzatora [1]. U nastavku se kolo za regulaciju deli na deo kola sa naponskim regulatorom sa promenljivom vrednošću izlaznog napona LM317 i na deo kola sa linearnim naponskim regulatorom L7805CV. Uloga regulatora LM317 je da obezbedi od korisnika izabranu vrednost napona na izlazu uređaja. Vrednost izlaznog napona određuje se na osnovu izraza:

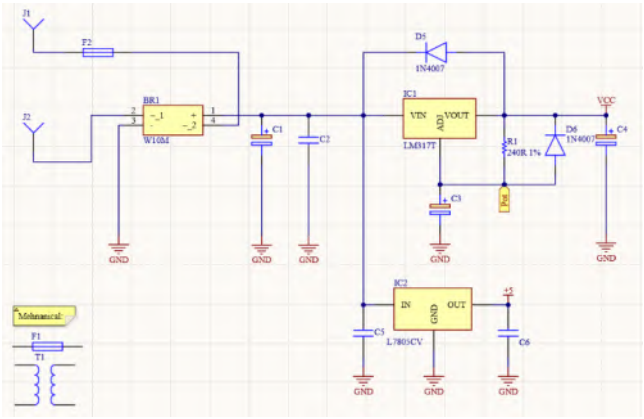
$$Vout = Vref \cdot (1 + \frac{Rpot}{R1}) \quad (1)$$

gde je V_{ref} temperaturno nezavisna naponska referenca i njena vrednost je 1.25 V, otpornost otpornika R_1 iznosi 240 Ω dok je R_{pot} vrednost otpornosti digitalnog potenciometra. Promenom vrednosti otpornosti digitalnog potenciometra menjaće se i vrednost napona na izlazu regulatora. Dioda D_5 sprečava pojavu inverznog napona na regulatoru. Ako na izlazu regulatora dođe do kratkog spoja, kondenzator C_4 će se umesto kroz regulator, isprazniti kroz diodu D_5 i time sačuvati regulator od uništenja. Kondenzator C_3 filtrira naponska poskakivanja i šumove dok dioda D_6 sprečava pražnjene kondenzatora C_3 preko regulatora [2]. Uloga linearnog naponskog regulatora L7805CV je da obezbedi stabilisanu vrednost napona od 5 V za napajanje mikrokontrolera i digitalnog potenciometra. Između ulaznog kontakta regulatora i mase vezan je bypass kondenzator C_5 dok je između izlaznog kontakta regulatora i mase vezan bypass kondenzator C_6 [3]. U svrhu smanjena dimenzija štampane ploče, mrežni transformator T_1 i osigurač primarnog namotaja F_1 montiraju se kao mehaničke komponente. Električna šema bloka napajanja prikazana je na slici 1.

B. Blok mikrokontrolera

Primarnu komponentu u ovom bloku predstavlja mikrokontroler ATMEGA328P (slika 2). Uloga mikrokontrolera je da izvrši analogno-digitalnu konverziju signala sa potenciometra R_{13} koji je na štampanu ploču

povezan preko *Male Headera* P_1, da na osnovu digitalnog signala dobijenog konverzijom podesi otpornost digitalnog potenciometra na željenu vrednost i upravlja RGB LE diodama obaveštavajući korisnika o izabranoj vrednosti izlaznog napona. Pritiskom na taster S_2, ostvariće se hardverski prekid i napon izabrane vrednosti biće prosleđen na izlaz uređaja. Pritiskom na taster S_1 mikrokontroler će se resetovati. Otpornici R_2 i R_3 imaju za cilj da pinove mikrokontrolera RESET i INT0 drže na visokom naponskom nivou. Kvarcni oscilator Y_1, njemu pridruženi otpornik R_{12} i kondenzatori C_7 i C_8 postavljaju se u skladu sa tehničkom dokumentacijom mikrokontrolera [4]. Kondenzator C_9 ima ulogu bypass kondenzatora i postavlja se što bliže moguće kontaktima za napajanje i masu mikrokontrolera. Digitalni potenciometar X9C103S biće obrađen u posebnom poglavlju. Električna šema bloka mikrokontrolera prikazana je na slici 2.



Slika 1. Električna šema bloka napajanja.

C. Izlazni blok

Izlazni blok sastoji se od četiri RGB LE diode sa zajedničkom katodom i dva konektora za dalje korišćenje dobijenog izlaznog napona (slika 3). Budući da se za indikaciju koriste samo crveni i zeleni segment RGB LE dioda, one su preko otpornika za ograničavanje vrednosti struje (od R_4 do R_{11}) povezane sa izlaznim pinovima mikrokontrolera. Konektor J_3 je *terminal block* konektor i služi za dovođenje izlaznog napona na licnaste ili provodnike punog poprečnog preseka dok je konektor P_2 *Male Header* konektor i njegova uloga je da izlazni napon dovede do drugog *Headera* (drugi uređaj, protoploča itd…). Električna šema izlaznog bloka prikazana je na slici 3. Konačni izgled štampane ploče prikazan je na slici 4.

IV. DIGITALNI POTENCIOMETAR X9C103S

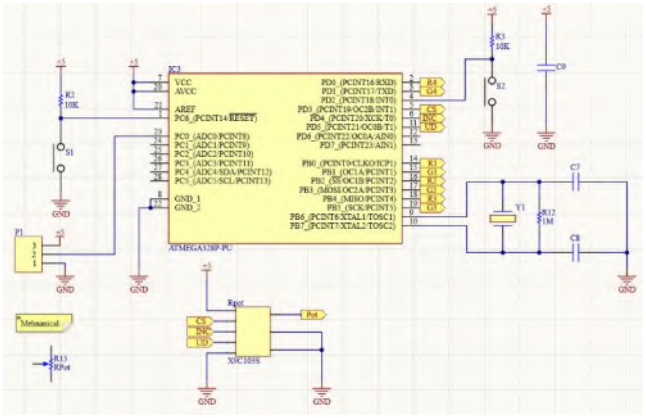
Korišćenjem izraza (1) za željene vrednosti izlaznih napona dobijaju se sledeće vrednosti otpornosti:

$$Vout = 3.3\,V \;\rightarrow Rpot = \;\; 393.6\,\Omega \qquad (2)$$

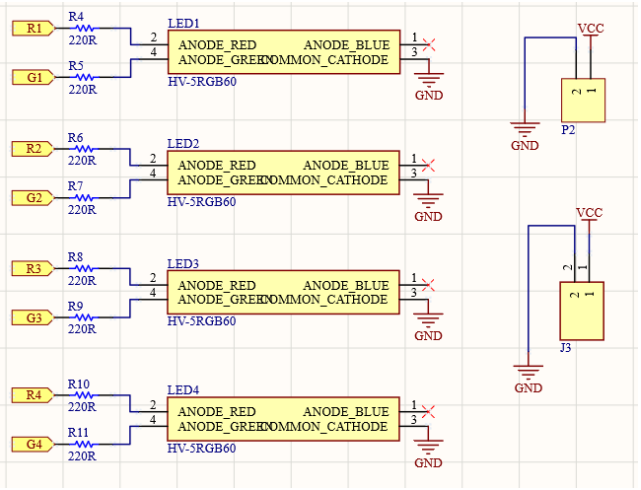
$$Vout = 5\,V \;\rightarrow Rpot = 720\,\Omega \qquad (3)$$

$$Vout = 9\,V \;\rightarrow Rpot = 1488\,\Omega \qquad (4)$$

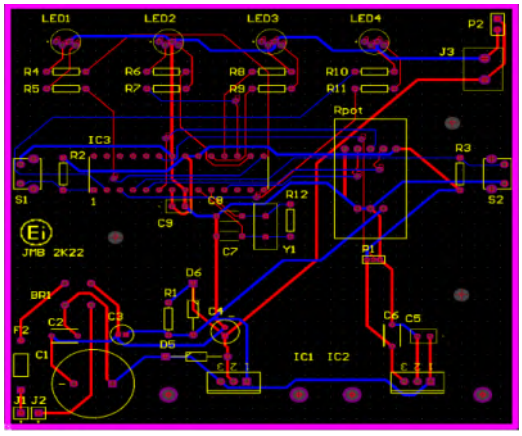
$$Vout = 12\,V \;\rightarrow Rpot = 2064\,\Omega \qquad (5)$$



Slika 2. Električna šema bloka mikrokontrolera.

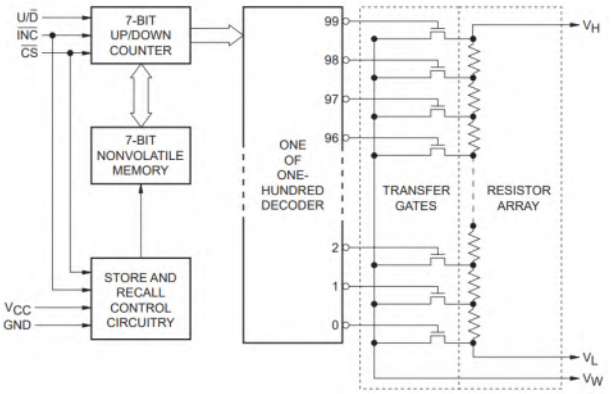


Slika 3. Električna šema izlaznog bloka.



Slika 4. Konačni izgled štampane ploče.

Digitalni potenciometar X9C103S predstavlja otporni niz od 99 otpornika. Na krajevima otpornog niza i između svakog otpornika nalaze se kontakti klizača potenciometra. Pozicija klizača kontroliše se preko pinova CS, U/D i INC. Postoji mogućnost da se pozicija klizača sačuva u memoriji digitalnog potenciometra ali se u konkretnom slučaju ova mogućnost ne koristi. Rezolucija digitalnog potenciometra računa se kao količnik maksimalne otpornosti (10 kΩ) i broja otpornika u otpornom nizu (99) [5]. Blok šema digitalnog potenciometra X9C103S prikazana je na slici 5.



Slika 5. Blok šema digitalnog potenciometra X9C103S.

Postavljanje pina CS na logičku nulu omogućiće rad digitalnog potenciometra. Postavljanjem pina U/D na logičku jedinicu vrednost otpornosti će rasti promenom stanja na pinu INC sa logičke jedinice na logičku nulu dok će se postavljanjem pina U/D na logičku nulu i promenom stanja na pinu INC sa logičke jedinice na logičku nulu vrednost otpornosti smanjivati. Potenciometar poseduje i pinove za napajanje i masu VCC i GND kao i pinove VH, VW i VL koji su analogni trima kontaktima klasičnog potenciometra. Za korišćenje digitalnog potenciometra preuzeta je odgovarajuća Arduino biblioteka. Zbog činjenice da se ovaj potenciometar na našem tržištu nalazi u SMD varijanti, na štampanu ploču uređaja postavljen je u vidu modula na kome se pored potenciometra nalaze i pull-up otpornik za pin INC, LE dioda kao indikator rada potenciometra i kondenzator za uklanjanje šumova. Tablica istinitosti digitalnog potenciometra prikazana je na slici 6.

$\overline{CS}$	$\overline{INC}$	U/$\overline{D}$	Promena
L	↓	H	Klizač gore
L	↓	L	Klizač dole
ƒ	H	X	Čuvanje položaja klizača
H	X	X	Potenciometar isključen
ƒ	L	X	Bez promene stanja

Slika 6. Tablica istinitosti digitalnog potenciometra.

V. PROGRAMSKI KOD U ARDUINO OKRUŽENJU

Mikrokontroler ATMEGA328P se najpre programira na Arduino UNO razvojnom okruženju a zatim se čip postavlja na svoj nosač na štampanoj ploči. Na početku programskog koda uvezena je biblioteka za digitalni potenciometar X9C.h i deklarisane su promenljive tipa integer za izlazne pinove mikrokontrolera kao i za pinove brojača i indikatora stanja. Promenljive koje se javljaju u funkcijama koje nisu podrazumevane funkcije deklarisane su kao volatile integer. U void setup delu programskog koda deklarisan je početak rada digitalnog potenciometra sa pinovima CS, INC i U/D, pinovi za uključivanje LE dioda podešeni su kao izlazni pinovi i definisan je hardverski interrupt na taster pinu (pin broj 2). Programski kod do void loop dela prikazan je na slici 7.

```
#include <X9C.h>
volatile int redOne=8, greenOne=9, redTwo=10, greenTwo=11;
volatile int redThree=12, greenThree=13, redFour=0, greenFour=1;
int cs=3, inc=4, ud=5;
int i, tasterPin=2, potVal;
volatile int selectedVoltage=0, potVoltage, y=1;
X9C pot;

void setup() {
  pot.begin(cs,inc,ud);
  for (i=8; i<14; i++){
    pinMode(i, OUTPUT);
  }
  pinMode(0, OUTPUT);
  pinMode(1, OUTPUT);
  attachInterrupt(digitalPinToInterrupt(tasterPin),tasterISR,FALLING);
}
```

Slika 7. Prvi deo programskog koda.

Void loop deo programskog koda započinje while petljom u okviru koje se naredbe izvršavaju samo dok korisnik vrši odabir vrednosti izlaznog napona (dok je vrednost promenljive y=1). Vrednost digitalnog signala sa analogno-digitalnog konvertora se očitava i smešta u promenljivu potVal. Zatim se na osnovu vrednosti promenljive potVal vrši izbor vrednost izlaznog napona pri čemu će jedna od četiri RGB LE diode koja predstavlja tu vrednost napona emitovati crvenu svetlost i to za vrednost potVal manju od 255 prva dioda, za vrednost potVal između 256 i 512 druga dioda, za vrednost potVal između 512 i 768 treća dioda i za vrednosti potVal veće od 768 četvrta LE dioda. Da bi se izbeglo prečesto ponavljanje komande digitalWrite, definisane su posebne funkcije koje će upravljati uključivanjem crvenih i zelenih LE dioda. Pritiskom na taster, mikrokontroler će ući u hardverski interrupt. U interrupt funkciji tasterISR trenutna vrednost promenljive potVoltage (vrednost 1 za 3,3 V, vrednost 2 za 5 V, vrednost 3 za 9 V i vrednost 4 za 12 V) biće upisana u promenljivu selectedVoltage i u promenljivu y upisaće se vrednost 2 (y=2) čime će se napustiti while petlja.

U nastavku void loop dela programskog koda se, u zavisnost od izabrane vrednosti napona na izlazu, vrši uključivanje zelenog segmenta jedne od četiri RGB LE diode i podešavanje vrednosti digitalnog potenciometra.

Zbog činjenice da su kontakt klizača digitalnog potenciometra VW i krajnji kontakt VH kratkospojeni i povezani na masu, za određivanje vrednosti otpornosti digitalnog potenciometra koristi se inverzna logika (da bi se dobila vrednost otpornosti ekvivalentna vrednostima otpornosti dva podeoka, softverski treba uključiti 97 podeoka otpornog niza). Funkcija setPot ima dva argumenta. Prvi argument predstavlja broj otpornih elemenata koje treba uključiti. Empirijskim putem se došlo do 96 elemenata za otpornost iz izraza (2), 94 elementa za otpornost iz izraza (3), 84 elementa za otpornost iz izraza (4) i 80 elemenata za otpornost iz izraza (5). Drugi argument je promenljiva tipa boolean i određuje da li će zadati položaj klizača biti sačuvan u memoriji. Segment koda za obaveštavanje korisnika o izabranoj vrednosti izlaznog napona i dodelu vrednosti digitalnom potenciometru prikazan je na slici broj 8.

```
if(selectedVoltage==1){
 redOn(LOW,LOW,LOW,LOW);
 greenOn(HIGH,LOW,LOW,LOW);
 pot.setPot(96,false);
}
else if(selectedVoltage==2){
 redOn(LOW,LOW,LOW,LOW);
 greenOn(LOW,HIGH,LOW,LOW);
 pot.setPot(94,false);
}
```

Slika 8. Deo koda za uključivanje zelenih segmenata LE dioda i dodelu vrednosti digitalnom potenciometru (za 3,3 V i 5 V).

Funkcija za kontrolu crvenih segmenata RGB LE dioda prikazana je na slici 9. Postoji identična funkcija za kontrolu zelenih segmenata RGB LE dioda.

```
void redOn(int first,int second, int third, int fourth){
    digitalWrite(redOne, first);
    digitalWrite(redTwo, second);
    digitalWrite(redThree, third);
    digitalWrite(redFour, fourth);
```

Slika 9. Funkcija za kontrolu LE dioda.

Programski kod tasterISR funkcije prikazan je na slici 10.

```
void tasterISR(){
  selectedVoltage=potVoltage;
  y=2;
}
```

Slika 10. Programski kod tasterISR funkcije.

VI. ZAKLJUČAK

Radu na ovom projektu pristupilo se sa idejom da se zaokruži proces projektovanja elektronskog uređaja i primeni znanje iz više predmeta stečenih na modulu Elektronske komponente i mikrosistemi, na Elektronskom fakultetu u Nišu. Dizajn štampane ploče odobrila je služba kontrole fabrike štapanih ploča JLCPCB (Shenzhen JIALICHUANG Electronic Technology Development Co.,Ltd.) i ploča je fizički realizovana u ovoj fabrici.

Poboljšanja uređaja su moguća, pre svega u načinu komunikacije sa korisnikom, korišćenjem LCD ili OLED displeja, korišćenjem SMD komponenata čime bi se znanto smanjile dimenzije štampane ploče. Ukoliko poseduje bazično znanje iz programiranja, korisnik može sam definisati vrednosti izlaznog napona koje su mu neophodne za realizaciju budućih projekata.

ZAHVALNICA

Zahvaljujem se asistentima sa Katedre za mikroelektroniku, Aleksandri Stojković, Milošu Marjanoviću i Nikoli Mitroviću, na izdvojenom vremenu i pomoći pruženoj tokom izrade projekta.

LITERATURA

[1] Zoran Prijić, Analogna mikroelektronika, Linearni izvori napajanja Dostupno na : http://mikro.elfak.ni.ac.rs/wp-content/uploads/2017/10/Literatura/Analogna/Analogna_lin_pred.pdf
[2] Tehnička dokumentacija naponskog regulatora LM317T Dostupno na : https://eu.mouser.com/datasheet/2/389/lm217-1849593.pdf
[3] Tehnička dokumentacija naponskog regulator L7805CV Dostupno na : https://eu.mouser.com/datasheet/2/389/l78-1849632.pdf
[4] Tehnička dokumentacija mikrokontrolera ATMEGA328P Dostpuno na : https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf
[5] Tehnička dokumentacija digitalnog potenciometra X9C103S Dostupno na : https://www.elecrow.com/download/X9C103S_Datasheet.pdf

Kontrola servo motora korišćenjem ARM mikrokontrolera

Mirjana Ćirić, Nemanja Stojanović, Lazar Milenković

Sadržaj – U ovom radu biće prikazan postupak realizacije sistema koji kontroliše servo motor korišćenjem ARM mikrokontrolera. Okretanje servo motora kontroliše se korišćenjem dva tastera – pritiskom tastera 1 / tastera 2 servo motor se pokreće u smeru kazaljke na satu / suprotno u odnosu na smer kazaljke na satu za 10 °. Okretanje je moguće vršiti u opsegu od -30 ° do +30 ° u odnosu na početnu poziciju. Obezbeđena je grafička interpretacija pozicije motora korišćenjem 7 LE dioda.

I. UVOD

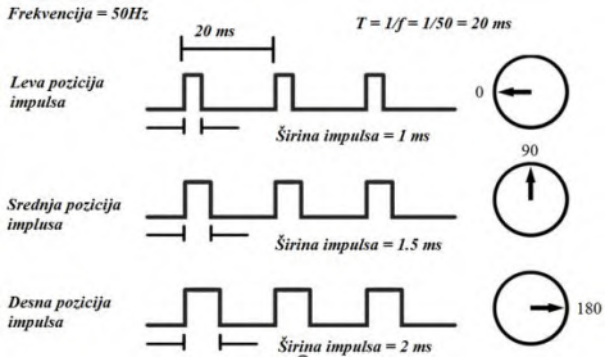
Servo motori se kontrolišu PWM (*Pulse Width Modulation*) signalima, tj. modulacijom širine kontrolnog impulsa. U ovom radu korišćen je hobi servo motor čija je frekvencija kontrolnog signala 50 Hz, što znači da je perioda tog signala 20 ms. Da bi servo motor zauzeo poziciju od 180° potrebno je da logička jedinica traje 2 ms, a logička nula 18 ms. Dakle, faktor ispune signala je 10%. Da bi servo motor zauzeo poziciju od 90°, potrebno je da u toku svake periode kontrolnog signala, signal bude 1.5 ms u stanju logičke jedinice. Slično, da bi servo motor zauzeo poziciju od 0°, kontrolni signal treba da bude u stanju logičke jedinice 1 ms, a 19 ms u stanju logičke nule. Na slici 1. prikazana je pozicija servo motora za određenu vrednost širine impulsa [1].

Servo motor napaja se sa 5 V jednosmernog napona. Za kontrolu servo motora odabrano je Nucleo razvojno okruženje sa ARM mikrokontrolerom. Prilikom razvoja ovog primera kontrole servo motora korišćeni su programi *Cube MX* za konfiguraciju mikrokontrolera i *Keil* za programiranje mikrokontrolera. Korišćen je programski jezik C.

II. KONFIGURACIJA TAJMERA MIKROKONTROLERA

Za upravljanje servo motora potrebno je generisati odgovarajući PWM signal, a za to se u ovom slučaju koristi tajmer mikrokontrolera. Da bi bilo moguće podešavanje tajmera u *Cube MX*-u, neophodno je odrediti preskaler i vrednost ARR (*Auto-Reload Register*) registra [2]. Preskaler predstavlja delilac učestanosti. Registar ARR pokazuje koliko puta treba da se inkrementira sadržaj registra tajmera da bi se desio prekid (*interrupt*). Za određivanje njihovih vrednosti korišćene su sledeće relacije i unapred zadate vrednosti: T = 20ms zato što je frekvencija servo motora f=50Hz. Faktor ispune određujemo za ugao od 90°:

$$\frac{1{,}5\ ms}{20\ ms} \times 100\% = 7{,}5\%. \quad (1)$$



Slika 1. Pozicija servo motora za određenu vrednost širine impulsa.

Pomoću faktora ispune nalazimo vrednost ARR registra:

$$7{,}5 = \frac{Pulse}{ARR} \times 100 \rightarrow ARR = 1000, \quad (2)$$

da bi vrednost promenljive Pulse bila ceo broj odnosno 75. Frekvenciju preskalera računamo iz formule:

$$f_{prescaler} = \frac{ARR}{T} = \frac{1000}{20 \times 10^{-3} s} = 50\ kHz. \quad (3)$$

Vrednost preskalera dobijamo iz količnika frekvencija preskalera i tajmera koja je podešena na 1 MHz:

$$Preskaler = \frac{ftim}{fprescaler} = \frac{1\ MHz}{50\ kHz} = 20. \quad (4)$$

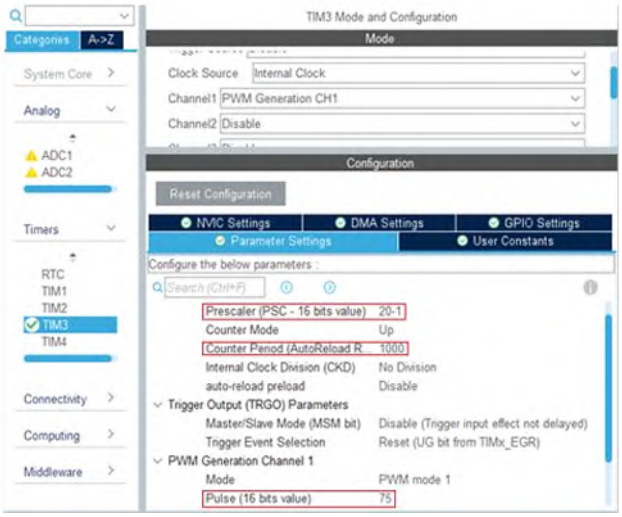
Na slici 2 prikazana su podešavanja za tajmer u programu *Cube MX*. Na osnovu ovih podešavanja prilikom generisanja koda predefinisane su vrednosti pojedinih registara mikrokontrolera.

U daljoj realizaciji ovog primera primene servo motora izvršeno je označavanje i podešavanje potrebih pinova. Na slici 3 prikazani su korišćeni pinovi. Pinovi koji su korišćeni u ovom projektu su:

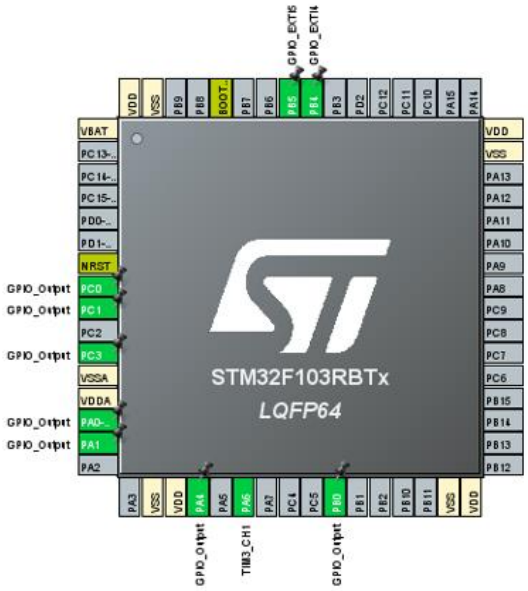
- PA6 za servo motor
- PB4 i PB5 za tastere (pull-up konfiguracija)

M. Ćirić, N. Stojanović i L. Milenković su studenti na modulu Elektronske komponente i mikrosistemi, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija. E-mail: mirjana.ciric@elfak.rs, nemanja.stojanovic@elfak.rs, milenkovic.lazar@elfak.rs.

• PA0, PA1, PA4, PB0, PC0, PC1 i PC3 za LE diode.



Slika 2. Prikaz kartice za podešavanje parametara tajmera.

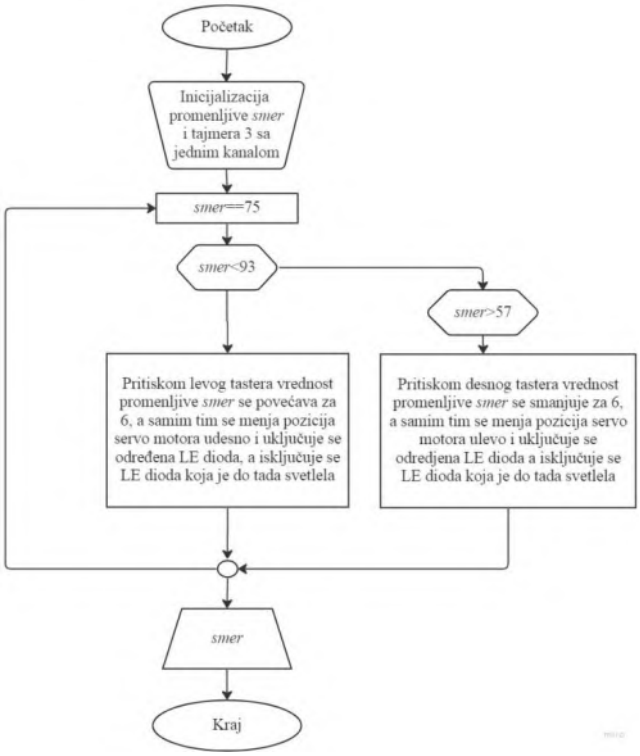


Slika 3. Prikaz izabranih pinova mikrokontrolera.

III. PROGRAMIRANJE MIKROKONTROLERA

Princip rada programa zasnovan je na pritisku tastera povezanih na pinove PB4 i PB5 na osnovu kojih se menja smer kretanja servo motora i smer uključivanja LE dioda. Pre početka pisanja glavnog dela programa uvedena je promenljiva *uint32_t smer=75*. Promenljivoj *smer* je dodeljena vrednost 75 (jer je tolika i vrednost Pulse-a) da bi se servo motor sigurno našao u poziciji od 90°. Smanjivanjem ili povećavanjem vrednosti promenljive *smer* servo motor se kreće ulevo ili udesno u zadatom opsegu (od -30° do +30°).

Sama realizacija projekta najpre je predstavljena u vidu algoritma (slika 4), a nakon toga je algoritam preveden u programski kod.



Slika 4. Algoritam za izvršenje programskog koda.

U delu *USER CODE BEGIN 2* pozivamo tajmer TIM3 i generišemo PWM signal preko kanala CHANNEL_1. Uključujemo LE diodu na pinu B0 koja odgovara poziciji od 90°.

U glavnom delu koda korišćena je funkcija *Set_Compare* koja služi za podešavanje vrednosti *duty cycle* na određenu vrednost i u zavisnosti od toga da li je povećavamo ili smanjujemo, menja se smer okretanja servo motora [3]. Od argumenata imamo redom: pointer na hendler za tajmer (&htim3), kanal na kojem je aktiviran PWM signal i vrednost promenljive *smer*.

Naredbom i*f(HAL_GPIO_ReadPin(GPIOB,GPIO_PIN_4)==GPIO_PIN_RESET && smer<93)* ispitujemo da li je pritisnut levi taster i da li je vrednost promenljive *smer* manja od 93 jer ta vrednost odgovara pomeraju od 30° udesno. Ako je taster pritisnut vrednost promenljive *smer* povećavamo za 6 zato što ta vrednost odgovara pomeraju od 10°. Tu vrednost smo dobili zato što znamo da krajnja leva i desna pozicija servo motora (gde je razlika u uglovima 180°) odgovaraju vrednostima promenljive *smer* od 25 i 125. Korišćenjem proporcije se došlo do vrednosti 6 koja mora da se inkrementira ili dekrementira u zavisnosti od toga koji od dva tastera je pritisnut. U zavisnosti od vrednosti promenljive *smer* uključuje se određena LE dioda i u isto vreme se isključuje LE dioda koja je do tada svetlela.

Naredbom *else_if(HAL_GPIO_ReadPin(GPIOB,GPIO_PIN_5)==GPIO_PIN_RESET && smer>57)* ispitujemo da li je prirtisnut desni taster i proces je potpuno analogan pritisku levog tastera osim toga što ovde ispitujemo da li promenljiva *smer* ima veću vrednost od 57 koja odgovara pomeraju od 30° ulevo. Naravno, ovde se svakim pritiskom tastera vrednost promenljive *smer* smanjuje za 6. U nastavku biće ispisan deo programa neophodan za rad ovog projekta.

```
/* USER CODE BEGIN PV */
uint32_t smer=75;
/* USER CODE END PV */

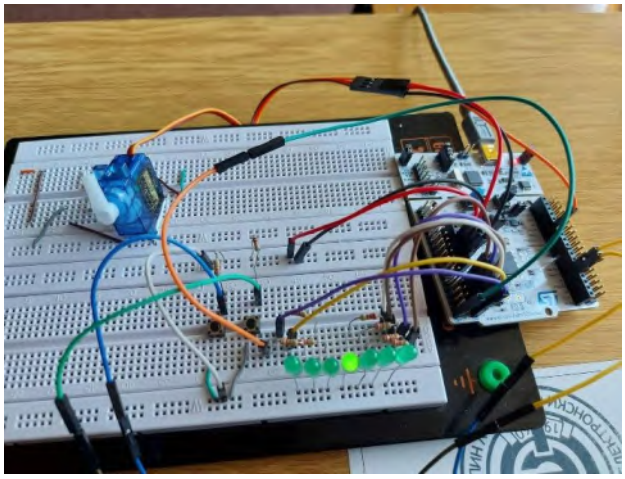
/* USER CODE BEGIN 2 */
HAL_TIM_Base_Start(&htim3);
HAL_TIM_PWM_Start(&htim3,TIM_CHANNEL_1);
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, 1);
 /* USER CODE END 2 */

/* USER CODE BEGIN WHILE */
  while (1)
  {
if(HAL_GPIO_ReadPin(GPIOB,GPIO_PIN_4)==GPIO_PIN_R
ESET && smer<93)
{
//servo
smer=smer+6;
__HAL_TIM_SetCompare(&htim3,TIM_CHANNEL_1, smer);
HAL_Delay(200);
if(smer==57){
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, 1);
}
if(smer==63){
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, 0);


HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, 1);
}
if(smer==69){
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, 0);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, 1);
}
if(smer==75){
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, 0);
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, 1);
}
if(smer==81){
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, 0);
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, 1);
}
if(smer==87){
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, 0);
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0, 1);
}
 if(smer==93){
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0, 0);
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_3, 1);
}
}
else_if(HAL_GPIO_ReadPin(GPIOB,GPIO_PIN_5)==GPIO_
PIN_RESET && smer>57)
{
smer=smer-6;
__HAL_TIM_SetCompare(&htim3,TIM_CHANNEL_1, smer);
HAL_Delay(200);
if(smer==93){
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_3, 1);
}
if(smer==87){
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_3, 0);
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0, 1);
}
if(smer==81){
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0, 0);
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, 1);
}
if(smer==75){
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, 0);
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, 1);
}
if(smer==69){
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, 0);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, 1);
}
if(smer==63){
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, 0);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, 1);
}
if(smer==57){
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, 0);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, 1);
}
}
/* USER CODE END WHILE */
```

IV. ZAKLJUČAK

Opisan primer je praktično testiran korišćenjem odgovarajućih komponenata na protoploči, uz upotrebu Nucleo razvojnog okruženja. Na slici 5 prikazana je ova realizacija. Opisani primer može biti iskorišćen kao sastavni deo složenih sistema koji koriste servo motore.



Slika 5. Praktična realizacija projekta.

ZAHVALNICA

Autori ovog rada se srdačno zahvaljuju profesorima i saradnicima sa Katedre za mikroelektroniku na Elektronskom fakultetu u Nišu.

Posebna zahvalnost asistentima Milošu Marjanoviću i Milanu Stojanoviću koji su nam obezbedili potreban materijal za izradu ovog projekta kao i sugestije koje su nam uputili tokom izrade istog.

LITERATURA

[1] Controlling Servo Motor with STM. Dostupno na: https://www.engineersgarage.com/
[2] STM32 Timers. Dostupno na: https://www.st.com
[3] STM32 PWM with variable Frequency. Dostupno na: https://community.st.com

Određivanje statičkog probojnog napona gasnog odvodnika prenapona *EPCOS*

Sanja Đokić, Marija Živković

Sadržaj – U radu su prikazani rezultati određivanja statičkog probojnog napona na osnovu eksperimentalnih podataka dobijenih korišćenjem gasnog odvodnika prenapona *EPCOS*. Date su definicije probojnog napona i metode za procenu statičkog probojnog napona. Eksperimentalni rezultati za prve tri brzine porasta napona su prikazani grafički i fitovani Vejbulovom raspodelom.

I. UVOD

Odvodnici prenapona punjeni gasom (u daljem tekstu: gasni odvodnici prenapona, engl. *Gas Filled Surge Arresters,* GFSA) predstavljaju klasične nelinearne komponente za zaštitu osetljivih elektronskih kola od prenapona. Opseg koji obuhvataju je od 70 do 1200 V, a prednost im je mogućnost provođenja visokih struja (> 5 kA) i mala kapacitivnost (< 1 pF).

Odvodnici se sastoje od najmanje dve elektrode smeštene u kućištu od neprovodnog materijala. U oblasti prostora između ravnih elektroda se nalazi gas, koji ne provodi struju pri normalnim vrednostima napona. Tek onda kada se u gasu ostvari određeni stepen jonizacije i nastupi proboj dolazi do provođenja struje. Elektrode su ravne i paralelne radi obezbeđivanja homogenog električnog polja između njih (slika 1). Kućište može biti napravljeno od keramike, stakla ili plastike. Može biti cilindričnog ili bilo kakvog drugog oblika u kome može da se održi gasna sredina bez curenja u okolni prostor.

Kod komercijalnih gasnih elektronskih komponenata statički probojni napon se najčešće određuje pri brzini porasta napona od nekoliko desetina volti u sekundi. U radu je kod EPCOS gasnog odvodnika prenapona statički probojni napon određen za brzinu porasta napona u opsegu od 1 do 10 V/s.



Slika 1. Gasni odvodnik prenapona EPCOS.

S. Đokić je student doktorskih studija modula Primenjena fizika na Katedri za mikroelektroniku, Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: sanjadjokic@elfak.rs.

M. Živković je student doktorskih studija modula Primenjena fizika na Katedri za mikroelektroniku, Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: marija.zivkovich@elfak.rs.

II. TEORIJSKI OSVRT

Električni proboj gasa može biti statički i dinamički što zavisi od oblika primenjenog napona. Proboj se naziva dinamičkim ako je brzina promene napona jednaka ili veća od brzine promene elementarnih procesa u gasu. Do statičkog proboja dolazi kada je brzina promene napona mnogo manja od brzine promene elementarnih procesa koji dovode do proboja. Statički proboj se odvija Taunsendovim ili strimernim mehanizmom, u zavisnosti od toga da li su sekundarni procesi na katodi dominantni u odnosu na sekundarne procese u gasu (Taunsendov mehanizam proboja) ili su sekundarni procesi u gasu dominantni u odnosu na sekundarne procese na katodi (strimerni mehanizam proboja).

Sa povećanjem električnog polja između elektroda, elektroni stiču dovoljno energije za jonizaciju gasa elektronskim udarom. Prilikom takve vrste sudara nastaju jedan novi elektron i jon, a daljim umnožavanjem nastaje lavina elektrona. Da bi opisao ovaj proces, Taunsend je uveo koeficijent jonizacije elektronskim udarom α, koji predstavlja broj jonskih parova koje stvara elektron po jedinici dužine puta driftujući ka anodi. Ovaj koeficijent se naziva prvim Taunsendovim koeficijentom, a proces jonizacije elektronskim udarom α-procesom [1].

Porast broja elektrona dn na putu ka anodi kada pređe rastojanje dx iznosi

$$dn = \alpha\, n\, dx. \tag{1}$$

Integracijom od katode do nekog rastojanja x, dobija se broj elektrona nastalih u lavini

$$n = n_0 e^{\alpha x}, \tag{2}$$

gde je n_0 broj primarnih elektrona koji napuštaju katodu u sekundi.

Broj elektrona koji dospeva na anodu je $n = n_0 e^{\alpha d}$, a struja na anodi je $i = i_0 e^{\alpha d}$.

Kada se jačina polja još više poveća, joni stvoreni u gasu driftuju ka katodi i izazivaju sekundarnu elektronsku emisiju. Sekundarna elektronska emisija je pojava izbijanja elektrona sa površine čvrstih ili tečnih tela pri bombardovanju elektronima ili jonima. Ako u prvoj generaciji (lavini) jedan elektron krene sa katode, broj elektrona u lavini koji stignu na anodu iznosi $e^{\alpha d}$, pri čemu u gasu nastaje $(e^{\alpha d} - 1)$ elektron-jonskih parova. Druga generacija daje $\gamma(e^{\alpha d} - 1)$ sekundarnih elektrona koji

krenu sa katode, itd. To je preglednije predstavljeno u tabeli 1.

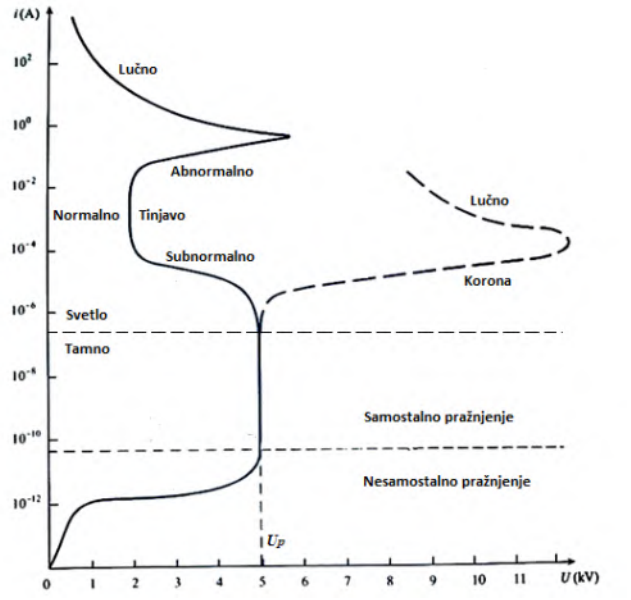
TABELA I
PROCES SEKUNDARNE EMISIJE NA KATODI

I	1	$e^{\alpha d}$	$e^{\alpha d}-1$
II	$\gamma(e^{\alpha d}-1)$	$\gamma(e^{\alpha d}-1)e^{\alpha d}$	$\gamma(e^{\alpha d}-1)^2$
III	$\gamma^2(e^{\alpha d}-1)^2$	$\gamma^2(e^{\alpha d}-1)^2e^{\alpha d}$	$\gamma^2(e^{\alpha d}-1)^3$
IV	----------	----------	----------

Ako je poznato da je γ (treći Taunsendov koeficijent koji daje broj elektrona koji krenu sa katode) mala veličina, a broj sukcesivnih lavina beskonačno veliki, sumiranjem srednje kolone dobija se ukupan broj elektrona na anodi po jednom elektronu koji krene sa katode

$$e^{\alpha d}(1+q+q^2+\cdots)=\frac{e^{\alpha d}}{1-q}, \qquad (3)$$

gde je $q=\gamma(e^{\alpha d}-1)$ koeficijent lavinske multiplikacije. Množenjem relacije (3) brojem primarnih elektrona n_0 dobija se ukupan broj elektrona na anodi n. Bez obzira na to što struja značajno raste zbog učešća γ-procesa (slika 2) pražnjenje je još uvek nesamostalno. Ukupna struja je proporcionalna struji primarnih elektrona i ako se ukloni spoljašnji izvor jonizacije pražnjenje se prekida [2].



Slika 2. Strujno-naponska karakteristika električnih pražnjenja u gasu [2].

Električni proboj gasa nastaje kada dođe do prelaza iz nesamostalnog u samostalno pražnjenje, a struja nastavlja da teče i posle uklanjanja spoljašnjeg izvora jonizacije. Uslov proboja može da se izvede ako se uzme da u trenutku proboja gustina struje naglo poraste. Tako se dolazi do uslova proboja

$$1-\gamma(e^{\alpha d}-1)=0, \qquad (4)$$

odnosno

$$\gamma(e^{\alpha d}-1)=1. \qquad (5)$$

Faktor $e^{\alpha d}-1$ predstavlja srednji broj jona nastalih od jednog elektrona u multiplikativnim procesima koji pređe rastojanje d ka anodi. Ako se ovaj faktor pomnoži koeficijentom γ dobije se broj elektrona naredne generacije emitovanjih sa katode. Prema jednačini (5) taj broj u graničnom slučaju treba da bude jednak jedan, što znači da svaki elektron koji pođe sa katode obezbedi stvaranje jednog novog elektrona naredne generacije na katodi. Koeficijent multiplikacije $q=\gamma(e^{\alpha d}-1)$ mora da bude veći ili, u graničnom slučaju, jednak jedinici $q\geq 1$. Samo tada se obezbeđuje stvaranje elektron-jonskih parova multiplikacijom u gasu koje ne zavisi od postojanja spoljašnjeg izvora jonizacije. Ovo znači da se proboj odigrava sa određenom verovatnoćom P. Za slučaj Taunsendovog mehanizma proboja ova verovatnoća je različita za svaki inicijalni elektron koji može dovesti do proboja. Verovatnoća proboja P je [3] [4]

$$P=0,\ q\leq 1; \quad P=1-\frac{1}{q},\ q>1. \qquad (6)$$

Određivanje probojnog napona U_b se vrši tako što se prvo na gasnu komponentu dovodi fiksni napon U_k koji je znatno manji od očekivanog probojnog napona U_b. Onda se napon povećava za unapred zadati naponski korak U_n pri čemu se unapred postavlja vreme trajanja naponskog stepa t_n. Brzina porasta napona je data odnosom $k=U_n/t_n$. Kada dođe do proboja naponom U_c se obezbeđuje zadata vrednost struje pražnjenja i_g za vreme pražnjenja t_g da bi se u cevi uspostavila stacionarna koncentracija naelektrisanih i pobuđenih čestica. Nakon toga se napon na elektrodama gasne komponente isključuje za vreme relaksacije τ koje se zadaje unapred. Nakon isteka zadatog vremena relaksacije proces se ponavlja.

U seriji ponovljenih merenja pri istim uslovima dobiće se različite vrednosti probojnog napona. To znači da se proboj dešava sa određenom verovatnoćom zbog čega je probojni napon U_b veličina statističkog karaktera sa određenom raspodelom.

III. STATIČKI PROBOJNI NAPON

Prilikom analize proboja u gasu potrebno je izvršiti veliki broj merenja pod istim uslovima i koristiti srednju vrednost $\overline{U}_b$. Potrebno je znati najmanji napon koji se može priključiti na gasnu komponentu kada dolazi do proboja i ta vrednost napona naziva se statički probojni napon U_s koji je manji od vrednosti $\overline{U}_b$ za brzine porasta napona veće od $1\,\mathrm{V/s}$.

Statički probojni napon se može definisati kao najveći napon U_ω za koji je verovatnoća proboja jednaka nuli

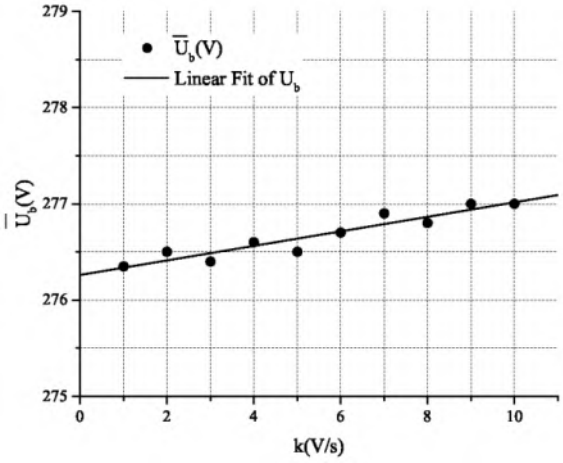
$$\lim_{P\to 0} U_\omega = U_s. \quad (7)$$

Najbolja procena statičkog probojnog napona može se izvršiti iz zavisnosti srednje vrednosti probojnog napona $\overline{U}_b$ od brzine porasta napona k. Na osnovu fitovanih eksperimentalnih rezultata $\overline{U}_b = f(k)$ polinomom određenog reda i preseka fitovane krive sa $\overline{U}_b$ osom, koja odgovara vrednosti $k = 0$, može da se proceni vrednost statičkog probojnog napona U_s [5]

$$\lim_{k\to 0} \overline{U}_b = U_s. \quad (8)$$

Naredni grafik predstavlja eksperimentalno dobijene rezultate merenja zavisnosti srednje vrednosti probojnog napona $\overline{U}_b$ od brzine porasta napona k u opsegu od 1 do 10 V/s.

Na slici 3 se može videti linearna zavisnost $\overline{U}_b$ od k, odnosno, vidi se prava linija dobijena nakon fitovanja eksperimentalnih rezultata. Vrednost statičkog probojnog napona U_s približno iznosi 276.3 V za gasni odvodnik *EPCOS*.

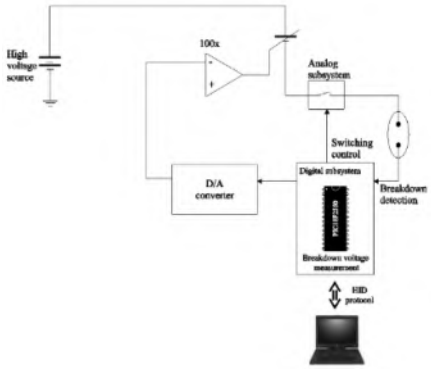


Slika 3. Srednja vrednost probojnog napona $\overline{U}_b$ u funkciji brzine porasta napona k od 1 do 10 V/s gasnog odvodnika prenapona proizvođača EPCOS.

IV. EKSPERIMENT

Merna oprema korišćena za realizaciju eksperimenta merenja probojnog napona prikazana je na slici 4. Preko računara se podešavaju parametri, aktivira softver i prikupljuju podaci. Merna oprema se sastoji od DC izvora napona, stepenastog generatora napona i digitalnog podsistema.

Sistem za merenje probojnog napona sastoji se od analognog i digitalnog podsistema. Analogni podsistem se sastoji od DC izvora napona i stepenastog generatora napona koji je projektovan kao naponski pojačavač sa x100 pojačanjem. Digitalni podsistem se sastoji od MICROCHIPs PIC18F2550 i D/A konvertora. Od napona na izlazu DC izvora se oduzima napon iz D/A konvertora pojačan 100 puta. Vrednost napona kada nastupi proboj se čuva u memoriju. Zatim je gasni odvodnik isključen za unapred zadatu vrednost vremena relaksacije τ. Nakon isteka ovog vremena počinje novo merenje i procedura se nastavlja 100 ciklusa sa unapred definisanom vrednošću brzine porasta napona.



Slika 4. Sistem za merenje probojnog napona [6].

V. VEJBULOVA RASPODELA

Vejbulova raspodela je jedna od najpoznatijih raspodela. Na adekvatan način opisuje greške koje se javljaju kod različitih tipova komponenata i fenomena. U poslednje tri decenije napisani su brojni radovi vezani za ovu raspodelu.

Vejbulova funkcija raspodele se pojavljuje u pet različitih oblika. Oblik raspodele koji se najčešće sreće je

$$F(t) = 1 - exp\left[-\lambda(t-\tau)^\beta\right], t \geq \tau. \quad (9)$$

Parametri raspodele su dati setom $\theta = [\alpha, \beta, \tau]$ gde su $\alpha > 0$, $\beta > 0$ i $\tau \geq 0$; α je parametar skale, β parametar oblika (u kom obliku se pojavljuje raspodela) i τ je parametar lokacije. Parametar λ predstavlja kombinaciju parametara α i β i dat je kao $\lambda = \alpha^{-\beta}$.

Funkcija gustine Vejbulove raspodele (9) je

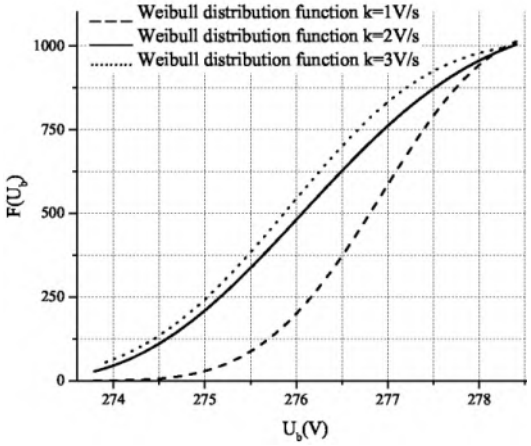
$$f(t) = \beta\lambda(t-\tau)^{\beta-1} exp\left[-\lambda(t-\tau)^\beta\right], t \geq \tau. \quad (10)$$

VI. REZULTATI MERENJA

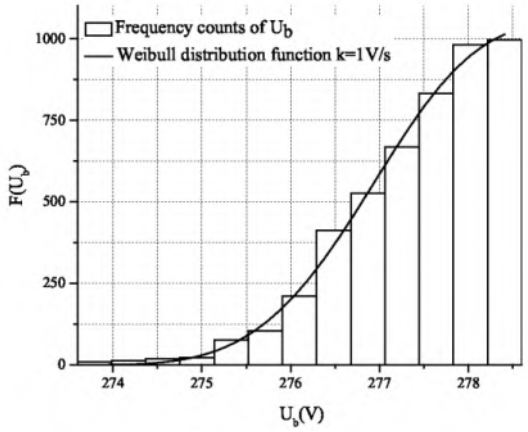
Naredni grafici prikazuju eksperimentalno dobijene rezultate (histogram) fitovane Vejbulovom raspodelom (puna linija) za svaku brzinu porasta napona kod komponente *EPCOS*. Ono što se meri jesu vrednosti probojnog napona za različite brzine porasta napona, gde se rezultati prikazuju pomoću histograma za svaku brzinu porasta napona za 1000 merenja.

Za brzine porasta napona veće od 3 V/s se ne dobijaju zadovoljavajući rezultati, odnosno ne postoji dobro slaganje između eksperimenta i teorije. Za brzine porasta napona do 3 V/s izraz za funkciju gustine raspodele dobro opisuje eksperimentalne podatke probojnog napona, što se vidi na narednim slikama.

Na slici 5 se mogu videti rezultati fitovani Vejbulovom raspodelom, dok se na slikama 6-8 može videti svaka raspodela posebno, rezultati merenja i Vejbulov fit.



Slika 5. Funkcija raspodele probojnih napona za k = 1 V/s, k = 2 V/s i k = 3 V/s.

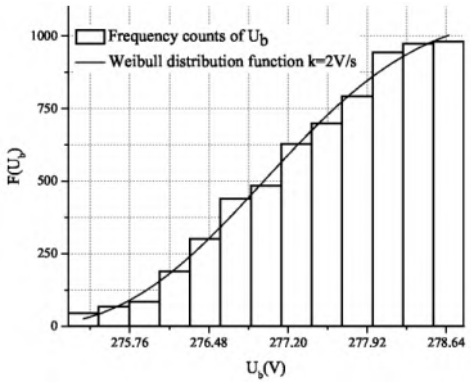


Slika 6. Funkcija raspodele probojnih napona za k = 1 V/s.

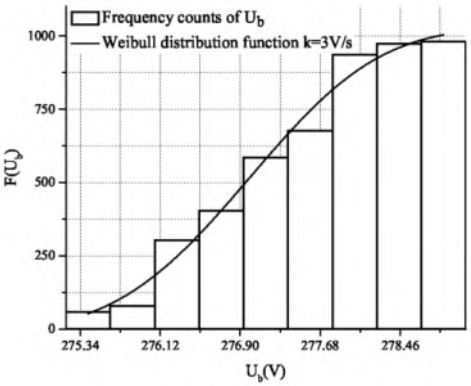
VII. ZAKLJUČAK

Na osnovu merenja prikazanih u ovom radu, zaključuje se sledeće. Gasni odvodnik prenapona *EPCOS* je komponenta koja se koristi za zaštitu osetljivih elektronskih kola od prenapona. Odrađena je statistička analiza električnog proboja, sa brzinama porasta napona od 1 V/s do 10 V/s, a samim tim određen i statički probojni napon. Eksperimentalni podaci su pokazali da se funkcija raspodele može fitovati Vejbulovom funkcijom raspodele što se može videti i na prikazanim graficima.

Buduća istraživanja će se ogledati u daljem upoređivanju različitih vrsta gasnih odvodnika prenapona (*Littelfuse, EPCOS, CITEL*), kao i gasnih cevi, sa različitim brojem merenja. Cilj će biti upoređivanje pouzdanosti novih gasnih komponenti usled određivanja najpouzdanije komponente.



Slika 7. Funkcija raspodele probojnih napona za k = 2 V/s.



Slika 8. Funkcija raspodele probojnih napona za k = 3 V/s.

ZAHVALNICA

Autori se zahvaljuju dr Emiliji Živanović, vanrednom profesoru, na saradnji i stručnoj pomoći i podršci koju nam je pružila tokom izrade ovog projekta.

LITERATURA

[1] M. M. Pejović, Električni proboj gasova, Niš: Elektronski fakultet u Nišu, 2009.

[2] V. Marković, Fizika jonizovanih gasova, Niš: Prirodno-matematički fakultet, 2004.

[3] J. Labat, Fizika jonizovanih gasova, Beograd: Fizički fakultet, 1991.

[4] B. S. Milić, Osnove fizike gasne plazme, Beograd: Građevinska knjiga, 1989.

[5] M. M. Pejović, Uvod u električna gasna pražnjenja, Niš: Elektronski fakultet u Nišu, 2008.

[6] E. Živanović, M. Živković and M. Pejović, "The evolution of breakdown voltage and delay time under high overvoltage for different types of surge arresters," vol. 34, pp. 307-322, 2021.

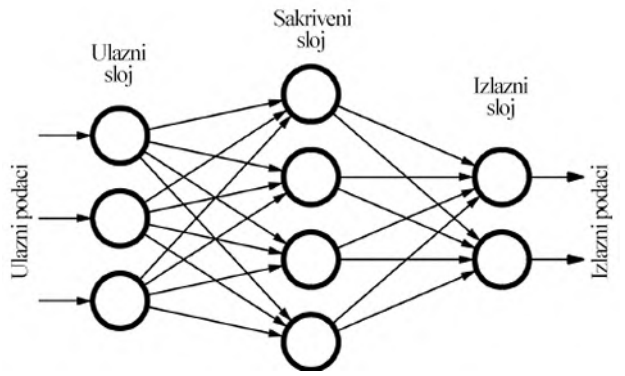
Platforma za treniranje neuronskih mreža genetskim algoritmom na jednostavnim video igrama

Lazar Mančić

Sadržaj – U ovom radu je izvršena analiza i prezentacija platforme namenjene za treniranje neuronskih mreža genetskim algoritmom na jednostavnim video igrama. Neuronske mreže su nepovratnog tipa sa fiksnom topologijom. Video igre na kojima je moguće treniranje su uglavnom arkadnog tipa. Platforma je izrađena u vidu veb aplikacije korišćenjem HTML-a, CSS-a i JavaScript-a.

I. UVOD

Neuronske mreže su deo mašinskog učenja, koje je podskup veštačke inteligencije. Inspirisane su biološkim nervnim ćelijama, ali sa njima nemaju naročitu vezu osim osnovne ideje postojanja neurona i veza među njima. Neuroni imaju svoju lokalnu memoriju u kojoj pamte podatke koji se obrađuju. Podaci se prenose između neurona uz pomoć veza od kojih svaka ima svoj težinski faktor. Težinski faktor određuje koliki uticaj će imati podatak koji putuje tom vezom. Pored memorije neuroni mogu imati *bias*, koji određuje uticaj tog neurona na obradu podataka. Neuroni su podeljeni na slojeve i obrada se vrši putem njih. Dva obavezna sloja su ulazni i izlazni sloj, dok mreža može, ali ne mora imati sakrivene slojeve.



Slika 1. Nepovratna (feedforward) neuronska mreža.

Postoje razne vrste neuronskih mreža podeljene prema različitim kriterijumima. Jedna od podela jeste podela prema smeru proticanja informacija i one se dele na nepovratne (*feedforward*) i povratne (*feedback*). Nepovratne su one u kojima protok ide samo u jednom smeru, sloj po sloj, od ulaznog sloja do izlaznog sloja, dok se kod povratnih podaci mogu vratiti nazad. Vizuelni prikaz neurona i veza između njih nepovratne neuronske mreže može se videti na slici 1.

Genetski algoritam je algoritam pretraživanja koji je inspirisan procesom prirodne selekcije. Primena ovog algoritma na neuronske mreže se svodi na posmatranje same neuronske mreže kao jedinke u sistemu. Veliku ulogu u razvoju ovde igra verovatnoća, jer je sama evolucija bazirana na nasumičnosti [3]. Kako bi rezultati bili što bolji, treba da postoje više jedinki koje se kasnije razmožavaju, pri čemu može doći i do mutacije, ali i do preživljavanja prethodnih jedinki. Cilj ovog procesa je da mreža evoluira i da se tako dođe do jedinke koja može rešiti zadati problem, odnosno u ovom projektu igranje video igre.

Projekat je osmišljen kao platforma za kreiranje i treniranje proizvoljnih modela koristeći neuronske mreže i genetski algoritam. Cilj istog jeste da se lakše prate i analiziraju rezultati treniranja proizvoljnih modela kako bi se utvrdilo koji model koristiti za određenu igru. Radovi na ovu i slične teme već postoje [1]-[4], ali su uglavnom potpuno naučno orijentisani ili praktično orijentisani za specijalni slučaj. Ovaj projekat se bazira na univerzalnom korišćenju postojećih metoda iz ove oblasti na različitim slučajevima, odnosno za sada različitim igrama.

II. PRINCIP RADA

Na platformi se može odabrati video igra na kojoj bi se izvršilo treniranje. Za svaku igru postoje unapred određeni ulazni i izlazni slojevi u neuronskoj mreži. Za potrebe rada u nastavku je objašnjena funkcionalnost jedne od igara, a to je poznata mobilna igra *Flappy Bird* [9]. Nakon igre objašnjena je i detaljna funkcionalnost korišćenih neuronskih mreža i metoda genetskog algoritma.

A. Flappy Bird igra

Flappy Bird je mobilna beskonačna igra u kojoj je jedini cilj ostvariti što veći rezultat. U pitanju je ptičica koja skače i kreće se napred, ali kako bi nastavila svoj put mora da prolazi između cevi koje je ometaju. Ukoliko ptičica udari u cev, ona umire i igrač gubi, za rezultat se uzima trenutni broj cevi koje je ptica prošla.

Kako su pravila igre veoma jednostavna, i ulazni i izlazni slojevi se mogu definisati kratko i jasno. Ulazni sloj poseduje dva neurona, jedan koji predstavlja horizontalno rastojanje ptice od sledeće cevi koja joj je na putu, a drugi neuron predstavlja vertikalno rastojanje ptice od sredine razmaka između cevi. Izlazni sloj ima samo jedan neuron, koji određuje binarno stanje da li će ptica skočiti ili ne.

L. Mančić je student prve godine na Elektronskom fakultetu, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, Email: lazarskolski@gmail.com.

Razlog za odabir malog broja neurona je da bi učenje bilo brže.

B. Neuronska mreža

Na platformi se može kreirati model sa nepovratnim neuronskim mrežama sa fiksnom topologijom. Njihova struktura se sastoji od slojeva, koji poseduju neurone, ali pored njih i aktivacionu funkciju (svi slojevi osim ulaznog). Pored neurona mreža poseduje veze, koje imaju težinski faktor. Vrednost neurona se računa sledećom formulom:

$$y_i = f(\sum_{j=1}^{|x|} x_j w_{ij} + b_i) \quad (1)$$

Vrednost *i*-tog neurona u sloju *y* je obeležena sa y_i, f predstavlja aktivacionu funkciju koja se primenjuje u tom sloju. Sloj koji prethodi *y* je sloj *x*, broj neurona u njemu je označen sa $|x|$, dok je vrednost *j*-tog neurona u njemu označena sa x_j. Težinski faktor veze između neurona y_i i neurona x_j je označen sa w_{ij}. Bias *i*-tog neurona u sloju *y* je označen sa b_i. Ova formula se primenjuje za svaki neuron koji nije ulazni, pri čemu je smer protoka informacija od ulaznog sloja ka izlaznom.

Aktivacione funkcije koje su korišćene na platformi su sledeće:

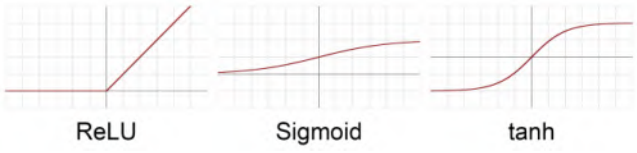
- Sigmoidna funkcija – formula (2)
- ReLU – formula (3)
- Hiperbolični tangens – formula (4)

$$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}} \quad (2)$$

$$f(x) = \begin{cases} 0 \; za \; x < 0 \\ x \; za \; x \geq 0 \end{cases} \quad (3)$$

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (4)$$

Grafici navedenih aktivacionih funkcija prikazani u Dekartovom koordinatnom sistemu mogu se videti na slici 2.



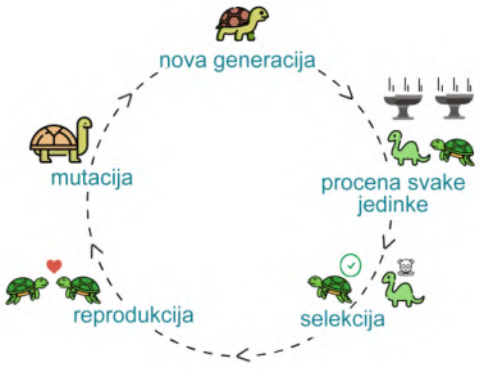
Slika 2. Grafici aktivacionih funkcija.

C. Genetski algoritam

Genetski algoritam se može podeliti na zasebne procese koji kada se izvrše odgovarajućim redom čine genetski algoritam. Ti procesi su redom:

1. Kreiranje nove generacije
2. Procena svake jedinke
3. Selekcija preživelih i selekcija roditelja
4. Reprodukcija jedinki
5. Mutacija jedinki

Vizuelno prikazan ciklus genetskog algoritma sa navedenim procesima može se videti na slici 3.



Slika 3. Ciklus genetskog algoritma.

Kreiranje nove generacije se razlikuje na početku i u toku rada algoritama. Na početku se generiše nova *populacija* jedinki koje će učestvovati u kasnijoj evoluciji. Platforma trenutno podržava samo nasumično generisanje *hromozoma* jedinki u populaciji. Pod *hromozomom* se podrazumeva neuronska mreža, dok pod *genom* se podrazumevaju težinski faktori i biasi u neuronskoj mreži, a *aleli* su vrednosti *gena*, odnosno težinskih faktora i biasa. U toku rada algoritama u novoj generaciji populaciju čine preživele jedinke i deca dobijena reprodukcijom.

Procena svake jedinke se vrši fitnes funkcijom, koja određuje koliko je uspešna svaka jedinka. Fitnes može biti i sam rezultat, ali se preferira precizniji način procene, ukoliko je rezultat neprecizan (poput celobrojnog rezultata).

Selekcija preživelih može da se vrši na osnovu starosti jedinke ili fitnes vrednost jedinke. Kod selekcije roditelja je malo drugačije. Postoje razne metode selekcije roditelja, od kojih se nekoliko njih mogu koristiti na platformi i one su:

- Rulet selekcija – Na osnovu fitnesa jedinki kreira se virtuelni točak, na osnovu koga se nasumičnim odabirom biraju roditelji. Bolje jedinke imaju veću šansu da budu odabrane, ali ni one loše nisu isključene iz selekcije, kako bi se diverzitet održao.
- Stohastičko univezalno uzrokovanje [6] – Veoma slično rulet selekciji, samo što se jednim „okretanjem“ točka pronalaze oba roditelja tako što su tačke odabira fiksirane.
- Turnir selekcija – Izdvajanje određenog broja jedinki od kojih se bira najbolja koja će biti jedan roditelj, isti proces se ponavalja i za odabir drugog roditelja.
- Nasumična selekcija – Oba roditelja se biraju nasumično bez ikakvog uticaja ostalih faktora.

Reprodukcija je proces kreiranja nove jedinke na osnovu dve roditeljske jedinke, pri čemu se vrši rekombinacija gena između hromozoma oba roditelja.

Metode rekombinacije koje se mogu koristiti na platformi su sledeće:

- Rekombinacija jednom tačkom – hromozom ili pravilni deo hromozoma se deli jednom nasumičnom tačkom na dva dela pri čemu dete uzima levu stranu jednog roditelja, a desnu stranu drugog roditelja.
- Rekombinacija sa više tačaka – Proces je isti kao kod rekombinacije jednom tačkom samo što postoji više nasumičnih tačaka, pri čemu dete naizmenično uzima delova.
- Nasumična rekombinacija gena – Svaki gen se gleda zasebno i dete nasumično uzima gen od roditelja na određenom mestu u hromozomu, pri čemu su šanse da gen roditelja bude odabran jednake.
- Aritmetička rekombinacija – Takođe se vrši na svakom genu posebno, na određenom mestu u hromozomu, pri čemu se primenjuje jednačina (5) za izračunavanje alela. Alel deteta je označen sa *c*, alel jednog roditelja sa *a*, alel drugog sa *b*, dok je α nasumičan realan broj između 0 i 1.

$$c = a * \alpha + b * (1 - \alpha) \quad (5)$$

Mutacija je proces koji se može izvršiti nad detetom tokom reprodukcije. Ovaj proces predstavlja promenu gena, odnosno alela, pri čemu je šansa da se to dogodi veoma mala. Metode mutacije korišćene na platformi su sledeće:

- Dodeljivanje nove nasumične vrednosti genu.
- Dodavanje, odnosno oduzimanje nasumične vrednosti genu.
- Negacija vrednosti gena – Ukoliko je vrednost pozitivna ona postaje negativna, ukoliko je negativna postaje pozitivna.

III. IMPLEMENTACIJA

Platforma je implementirana u vidu veb aplikacije [7], koja je pisana u HTML, stilizovana u CSS, i sva funkcionalnost joj je urađena u JavaScript programskom jeziku [8].

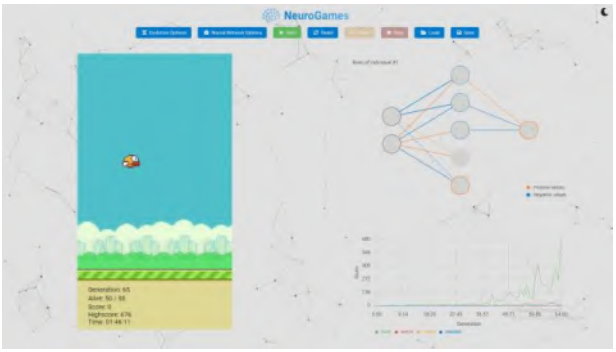
Na početnoj strani, prikazanoj na slici 4, se može odabrati igra koju korisnik može da igra kako bi je isprobao ili može izabrati opciju treniranja modela za odabranu igru pri čemu korisnik sam podešava parametre neuronske mreže i genetskog algoritma.

Na stranici za treniranje, prikazanoj na slici 5, postoji više opcija, one su konkretno:

- Podešavanja neuronske mreže – Korisniku pruža mogućnost da dodaje ili briše unutrašnje slojeve neuronske mreže i samim tim menja njenu topologiju, ovo je moguće samo pre početka evolucije. Takođe postoji opcija za uključivanje ili isključivanje vizualizacije neuronske mreže.
- Podešavanja evolucije – Korisniku pruža mogućnost odabira metoda selekcije preživelih i roditelja, metoda rekombinacije i mutacije, broja jedinki u populaciji i procenat preživelih. Pored toga takođe postoji mogućnost vizulizacije grafika zavisnosti rezultata ili fitnesa od generacije.
- Opcije za pokretanje (Start), pauziranje (Pause) i zaustavljanje (Stop) evolucije.
- Opcije za čuvanje (Save) i učitavanje sačuvane istrenirane veštačke inteligencije (Load).



Slika 4. Izgled interfejsa početne stranice.



Slika 5. Izgled stranice za treniranje sa uključenim vizualizacijama.

A. Implementacija neuroevolucije

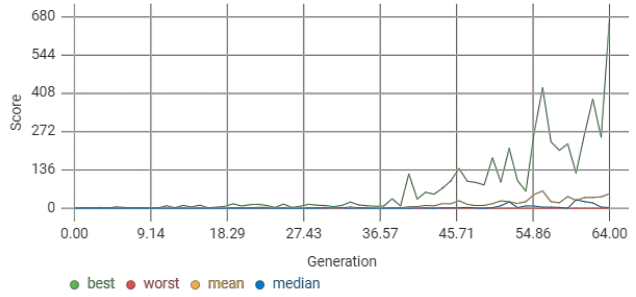
Svaka jedinka populacije na početku ima dodeljenu nasumično generisanu neuronsku mrežu, koja se izvršava svakog ciklusa igre kako bi se utvrdilo šta će jedinka da uradi. Neuronska mreža je implementirana odvojeno od same igre, tako da se može koristiti za više igara jedna ista implementacija. Implementacija je izvršena kroz nivoe, gde nivo predstavlja skup neurona dva susedna sloja i veza između između njih.

Ciklus genetskog algoritma se izvršava kada sve jedinke završe svoj život u igri. Procena jedinki se vrši tokom same igre kroz rezultat. Nakon toga se vrši selekcija preživelih i reprodukcija, tokom koje se odvijaju i rekombinacija i mutacija. Sve metode genetskog algoritma osim metode reprodukcije su univerzalno implementirane i mogu se koristiti i u drugim igrama. Metoda reprodukcije je vezana za jedinku, ali ona u sebi koristi univerzalne metode rekombinacije i mutacije.

IV. Rezultati

Rezultati platforme su delimično ispunili očekivanja. Trening je pokazao solidne rezultate, ali efikasnost i pouzdanost istog nisu najbolji. Grafik evolucije nakon treninga u trajanju od 100 minuta je prikazan na slici 6. Rezultati, zajedno sa generacijom i vremenom kad je dostignut taj tezultat su prikazani u tabeli I.

Model koji je treniran ima 2 ulazna neurona, 1 izlazni neuron i 5 neruona u skrivenom sloju. Broj jedinki u populaciji ima podrazumevanu vrednost 50.



Slika 6. Grafik evolucije u trajanju od jednog sata.

Tabela I
Rezultati treninga

Rezultat	Generacija	Vreme (min)
10	12	≈ 6
122	39	≈ 30
213	51	≈ 39
430	55	≈ 48
679	64	≈ 100

Grafik je nestabilan, iako je rast vidljiv, razlog tome je i priroda igre, odnosno nasumičnost datog nivoa. Cevi u igri se svaki put generišu nasumično, što znači teža, odnosno lakša kombinacija cevi se može generisati bilo gde u toku igre. Ovo može da dovede do devolucije jedinki, što može otežati ceo proces učenja. Moguća rešenja za ovaj problem još uvek nisu implementirana, ali postoje kao idejna i ona su:

- Promena funkcionalnosti igre tako da se sa napredovanjem u igri težina povećava.
- Treniranje jedinki iste generacije na više različitih nivoa, kako bi se nasumičnost iskoristila u korist učenja.

V. Moguće nadogradnje

Platforma je još na početku razvijanja, samim tim postoji i dosta mesta za nadogradnje. Osnovne nadogradnje bi bile dodavanje novih igara, novih metoda za procese genetskog algoritma, novih aktivacionih funkcija kod neuronskih mreža i drugo.

Moguće je i poboljšanje neuroskih mreža i neuroevolucije korišćenjem NEAT (Neuroevolution of augmenting topologies) algoritma [5]. Glavna razlika između NEAT algoritma i trenutne neuroevolucije na platformi jeste ta što je topologija neuronske mreže na platformi fiksa, samo se težinski faktori i biasi menjaju, dok se kod NEAT algoritma, mogu dodavati novi neuroni i nove veze između njih tokom mutacija čime se dobija neuroska mreža sa novom topologijom.

VI. Zaključak

Kombinacija genetskog algoritma i neuronskih mreža predstavlja dobar način za treniranje veštačke inteligencije da urade neke jednostavne zadatke, poput video igara u slučaju ove platforme. Naravno nije nemoguće koristiti ih za treniranje na kompleksnijim problemima. Jedna od mogućih primena je za treniranje modela veštačke inteligencije u autonomnim automobilima [1], iako to nije najefikasnije rešenje ono može ostvariti željene rezultate.

Literatura

[1] J. Rezgui, L. O. O'Leary, C. Bisaillon and L. C. Fourati, "Training Genetic Neural Networks Algorithms for Autonomous Cars with the LAOP Platform," 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), 2019, pp. 859-864.

[2] Bialas, Piotr. "Implementation of artificial intelligence in Snake game using genetic algorithm and neural networks." (2019).

[3] P. Krömer, V. Snáel and I. Zelinka, "Randomness and Chaos in Genetic Algorithms and Differential Evolution," *2013 5th International Conference on Intelligent Networking and Collaborative Systems*, 2013, pp. 196-201.

[4] M. G. Cordeiro, P. B. S. Serafim, Y. L. B. Nogueira, C. A. Vidal and J. B. Cavalcante Neto, "A Minimal Training Strategy to Play Flappy Bird Indefinitely with NEAT," *2019 18th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*, 2019, pp. 21-28.

[5] Neuroevolution of augmenting topologies https://en.wikipedia.org/wiki/Neuroevolution_of_augmenting_topologies (posećen 15.10.2022.)

[6] Stochastic universal sampling https://en.wikipedia.org/wiki/Stochastic_universal_sampling (posećen 15.10.2022.)

[7] Github Repository / Project source code https://github.com/lakimancic/neurogames (posećen 15.10.2022.)

[8] Live Demo https://lakimancic.github.io/neurogames/ (posećen 15.10.2022.)

[9] Flappy Bird Game https://en.wikipedia.org/wiki/Flappy_Bird (posećen 15.10.2022.)

Primena SCADA sistema u procesu proizvodnje sladoleda

Dejan Milenković

Sadržaj – U ovom radu biće opisan primer primene SCADA sistema u procesu proizvodnje sladoleda.

I. UVOD

Termin SCADA se odnosi na Supervisory Control And Data Acquisition. Bukvalan prevod bi glasio: Supervizorska kontrola i prikupljanje podataka. SCADA sistem je običan sistem procesne automatizacije koji se koristi da prikuplja informacije pomoću senzora i instrumentacije koji su locirani na udaljenim lokacijama kako bi se ti podaci preneli i prikazali na centralnoj lokaciji koja služi ili za upravljanje ili za svrhe monitoringa odnosno nadgledanja [1]. Primarna svrha SCADA sistema je da unapred daju upozorenje na problem koji može nastati.

SCADA sistem se obično sastoji od [2]:

- HMI – Human Machine Interface
- Nadzornog računarskog sistema
- RTU – Remote Terminal Units
- PLC – Programmable Logic Controller
- Komunikacione infrastrukture

HMI predstavlja vezu između operatera u nekom procesu i samog procesa. To je način na koji operater može nadgledati i kontrolisati process. Nadzorni računarski sistem služi za prikupljanje (akviziciju) podataka iz procesa i slanje komandi procesu. RTU su uređaji koji pretvaraju signale sa senzora u digitalne podatke koji se šalju nadzornom sistemu. PLC su industrijski računari tj. procesorske jedinice koje zadovoljavaju stroge zahteve sistema za rad u realnom vremenu i namenjene su za rad u industrijskim uslovima. Komunikaciona infrastruktura povezuje nadzorni računarski sistem sa RTU uređajima.

II. PRIMENA I CILJEVI SCADA SISTEMA

Glavni ciljevi SCADA sistema su da [3]:

- vrši nadzor
- uspostavi upravljanje nad sistemom i da osigura ponašanje sistema onako kako se od njega očekuje
- redukuje radnu snagu primenom automatizacije
- skladišti podatke o ponašanju sistema
- obezbedi informacije o radu sistema i uspostavi efektivno upravljanje resursima sistema
- uspostavi efikasno funkcionisanje sistema tako što će maksimalno smanjiti potrebu za rutinskim posetama udaljenim lokacijama i potencijalno smanjiti potrošnju električne energije
- obezbedi sistem upravljanja koji će omogućiti da se svi operativni ciljevi definišu i ostvare
- obezbedi sistem uzbunjivanja koji će omogućiti da se nepravilno funkcionisanje sistema evidentira sa centralne lokacije.

Primena SCADA sistema je dosta široka i najčešće je to u [3]:

- Proizvodnji i distribuciji električne energije
- Proizvodnji i distribuciji toplotne energije
- Petro-hemijskoj industriji
- Industriji vodosnabdevanja i prečišćavanje voda
- Prehrambenoj industriji
- Građevinskoj industriji
- Metalskoj industriji
- Industriji plastičnih masa.

III. PROJEKAT

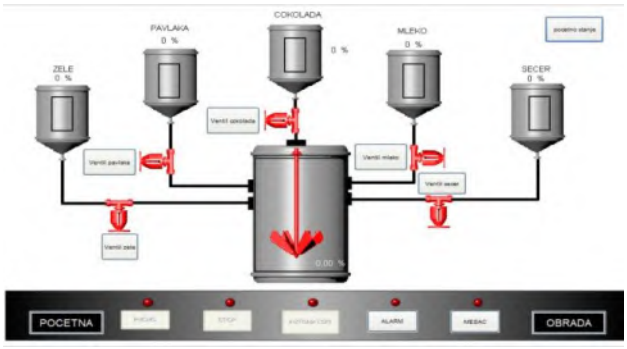
Projekat je zamišljen tako da omogući nadgledanje i upravljanje klasičnog automatizovanog procesa u industriji. Odlučeno je da taj proces bude proces proizvodnje sladoleda i realizovan je pomoću programa pod nazivom “CitectScada” verzije 7.1. Proces se sastoji od pravljenja smeše za sladoled, punjenja kalupa za sladoled, zamrzavanja istog i na kraju pakovanja finalnog proizvoda. Glavni cilj ovog rada je da se uspešno izvrši automatizacija celokupnog procesa proizvodnje sladoleda od početka do kraja procesa, odnosno pakovanja finalnog proizvoda.

A. Priprema smeše

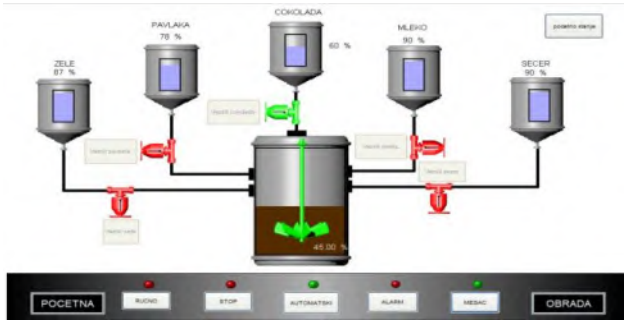
Cilj je da se pripremi smeša za dalju obradu (slika 1). Smeša se sastoji od 3% želea, 12% pavlake, 40% čokolade, 30% mleka i 15% šećera [4]. Možemo birati između automatizovanog i ručnog upravljanja. Pritiskom na dugme “pocetno stanje” dobijamo početno stanje koje je unapred definisano i to je po 90% u svakom od rezervoara sastojaka i 0% u rezervoaru za smešu. Tada smo u mogućnosti da biramo kako želimo da pravimo smešu, automatski ili ručno. Pritiskom na dugme “automatski” dobija se prikaz sa slike 2. Redom se otvaraju ventili i po unapred zadanom receptu se pravi smeša (3% želea, 12% pavlake, 40% čokolade, 30% mleka i 15% šećera). Vidi se da zelena boja

D. Milenković sa Katedre za automatiku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: dejanmilenkovic@elfak.rs.

na ventilu prikazuje da je ventil otvoren, dok je crvena boja za zatvoren ventil. Takođe, mešač je aktivan jer je zelene boje , a to i ukazuje indikator iznad dugmeta “MESAC”. U automatskom režimu su onemogućena sva dugmad za ventile.



Slika 1. Početna SCADA.



Slika 2. Prikaz za automatski režim rada.

Ako se odlučimo za ručno pravljenje smeše onda smo u mogućnosti da otvorimo ventile redosledom kako mi želimo ili pak da otvorimo više njih odjednom. Pritiskom na dugme “STOP” zaustavlja se proces i svi ventili se zatvaraju. Takođe ako se pritisne dugme “ALARM” svi ventili se zatvaraju i aktivira se alarm.

Funkcija koja omogućava sipanje sastojaka iz rezervoara sastojaka u rezervoar smeše realizovana je u odeljku za Cicode files na sledeći način:

```
FUNCTION automatizovano()

automatski=1;

              IF stop=0 AND rucno=0 THEN
                     ventil_zele=1;
                     zelee()
                     ventil_zele=0;
              END

              IF stop=0  AND rucno=0 THEN
                     ventil_pavlaka=1;
                     pavlakaa()
                     ventil_pavlaka=0;
              END

              IF stop=0  AND rucno=0 THEN
                     ventil_cokolada=1;
                     cokoladaa()
                     ventil_cokolada=0;
              END

              IF stop=0  AND rucno=0 THEN
                     ventil_mleko=1;
                     mlekoo()
                     ventil_mleko=0;
              END

              IF stop=0  AND rucno=0 THEN
                     ventil_secer=1;
                     secerr()
                     ventil_secer=0;
              END
END
```

Funkcije pod nazivom “secerr”, “mlekoo”, “cokoladaa”, “zelee” i “pavlakaa” su napravljene po modelu:

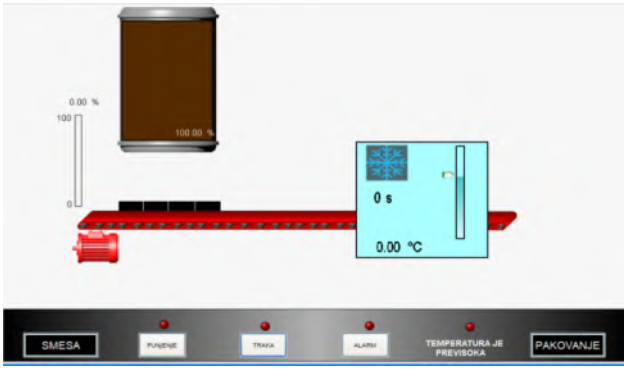
```
FUNCTION zelee()
       INT brojac;
              FOR brojac =0 TO 2 DO
                     IF stop=0 THEN
                            zele = zele - 1 ;
                            smesa=smesa+1;
                            Sleep(0.5);
                     END

              IF stop=1 THEN
                     brojac=brojac-1
              END
       END
END
```

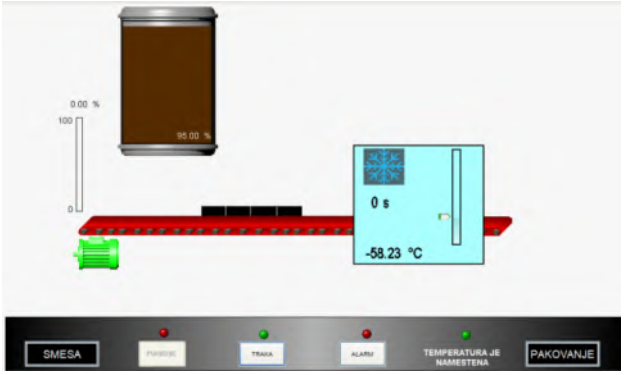
B. Punjenje kalupa

Cilj je da se napuni kalup već pripremljenom smešom i zaledi u zamrzivaču kako bi se pripremili za pakovanje. Najpre treba postaviti adekvatnu temperaturu (od - 50 stepeni celzijusa do - 60 stepeni celzijusa) u zamrzivaču preko mehanizma. Može se primetiti da se pojavio natpis da je temperatura previsoka (slika 3), a nakon postavljanja na adekvatnu da je nameštena (slika 4). Pritiskom na dugme “PUNjENjE” pokreće se punjenje kalupa. Kada dostigne 100% napunjenosti aktiviraće se pokretna traka i kalup ide ka daljoj obradi dok se punjenje stopira. Vidi se da nakon napunjenog kalupa uključila se pokretna traka i kalup je nastavio da se kreće ka zamrzivaču (slika 4). Indikator na motoru i LED dioda su postali zelene boje.

Kada kalup stigne u zamrzivač potrebno je da tu provede 120 sekundi da bi se smeša dovoljno zaledila. U svakom trenutku može se stopirati proces pritiskom na alarm, takođe, može se stopirati samo punjenje ili pokretna traka pritiskom na odgovarajuće tastere “PUNjENjE” ili “TRAKA” ukoliko postoji potrebe za tim.



Slika 3. Punjenje kalupa – temperatura previsoka.

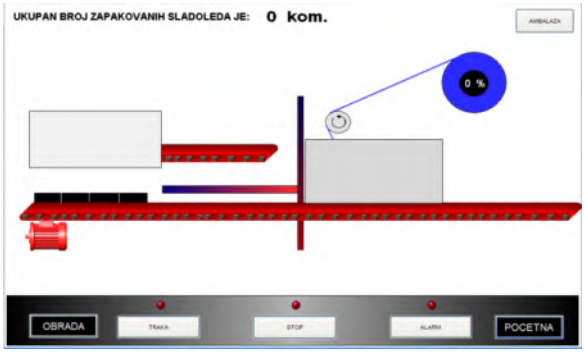


Slika 4. Punjenje kalupa – temperatura nameštena, pokretna traka uključena.

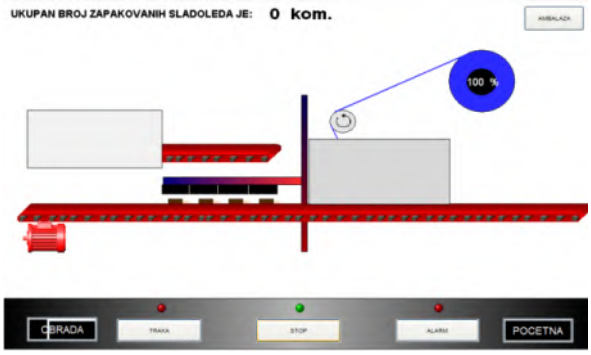
C. Pakovanje

Cilj je da se izvade sladoledi iz kalupa i da se zapakuju kao finalni proizvod. Može se primetiti da je rolna sa materijalom za pakovanje na 0% (slika 5). Pritiskom na dugme "AMBALAZA" postavlja se na 100%, što predstavlja početno stanje (slika 6). Pritiskom na dugme "TRAKA" pokrećemo proces pakovanja, takođe možemo zaustaviti traku ponovnim pritiskom na dugme "TRAKA" ili ukoliko želimo da zaustavimo ceo proces možemo pritisnuti dugme "STOP". Postoji mogućnost i da se uključi alarm koji zaustavlja celokupan proces proizvodnje sladoleda. Nakon pritiska dugmeta "AMBALAZA" i "TRAKA" može se videti kako je došlo do pomeranja kalupa i da je ambalaža na 100%. Ovo je ključni trenutak na ovoj strani gde se kalup odvaja od sladoleda, sladoledi nastavljaju da se kreću po pokretnoj traci do dela gde se pakuju, dok kalup ide na drugu pokretnu traku koja vodi u deo za ostavu iskorišćenih kalupa gde se pere i vraća na pokretnu traku kako bi se ponovo iskoristio.

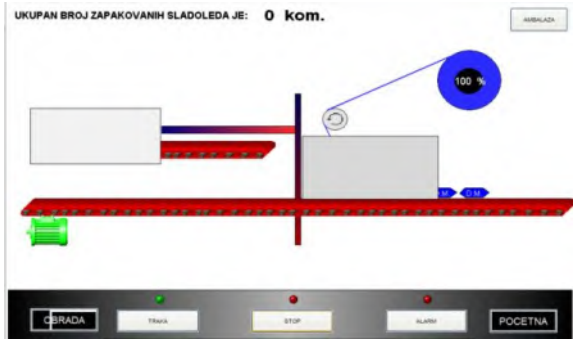
Trenutak kada su sladoledi zapakovani, a kalup se nalazi u objektu za ostavu kalupa spremnih za pranje prikazan je na slici 7. Trenutak kada je pakovanje završeno (slika 8), vidi se da je ambalaža na 99%, dok je broj zapakovanih sladoleda 4, jer se u jednom kalupu nalaze 4 sladoleda.



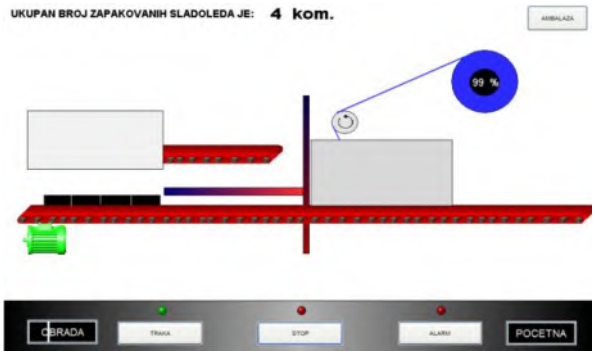
Slika 5. Pakovanje – rolna sa materijalom za pakovanje na 0%.



Slika 6. Pakovanje – rolna sa materijalom za pakovanje na 100%.



Slika 7. Pakovanje - sladoledi zapakovani, a kalup se nalazi u objektu za ostavu kalupa spremnih za pranje.



Slika 8. Pakovanje završeno.

IV. ZAKLJUČAK

U ovom radu je prikazan način razvoja i implementacije sistema za nadzor i upravljanje u postrojenju za proizvodnju sladoleda. Predstavljeni su tehnološki postupci kojima se upravlja sa korisničkog interfejsa SCADA-e. Takođe, predstavljeni su tokovi energije i energenata u samom procesu. Celokupan sistem je koncipiran i razvijen na osnovu standardne SCADA arhitekture uz dopunu funkcijama specifičnim za nadzor i upravljanje.

Savremena SCADA daje rešenja za bilo koji vid pogona proizvodnje, kao i uštedama u troškovima. Ovakvi sistemi omogućavaju lagodniji sistem rukovanja koji vodi ka boljem izvršenju operatera, smanjenom trošku energije, smanjenim gubicima, smanjenih troškova rada i većeg zadovoljstva korisnika. Uvođenjem centralnog sistema nadzora i upravljanja, u kombinaciji sa senzorima nove generacije, omogućava se visok stepen automatizacije pojedinih delova i postrojenja u celini, kao i nadzor kompletnog procesa i opreme. Iz priloženog možemo videti da se SCADA sistemi mogu primenjivati gotovo u svakom procesu i da je automatizacija jako bitna stavka za unapređenje industrije. Na konkretnom primeru je pokazano da je proces proizvodnje sladoleda uspešno automatizovan i da je omogućeno upravljanje i nadgledanje procesa u potpunosti od početka proizvodnje do zapakovanog proizvoda.

LITERATURA

[1] O SCADA sistemima na srpskom jeziku dostupno na: http://www.esi.ftn.uns.ac.rs/images/predmeti/SwKO/predavanja/Softver%20sa%20kriticnim%20odzivom%20-Predavanja%202.pdf

[2] Vežbe SCADA. dostupno na: http://automatika.elfak.ni.ac.rs/oglasna-tabla/nastavni-materijal-menu#scada-sistemi

[3] SCADA – Supervisory Control and Data Acquisition dostupno na: http://automatika.elfak.ni.ac.rs/files/Nastavni_materijal/SCADA/Predavanja/SCADA_2019_2020.PDF

[4] O receptu na engleskom jeziku dostupno na: https://dairyprocesinghandbook.tetrapak.com/chapter/ice-cream

Robot sa četiri ekstremiteta

Nikola Branković, Aleksandar Pavlovski

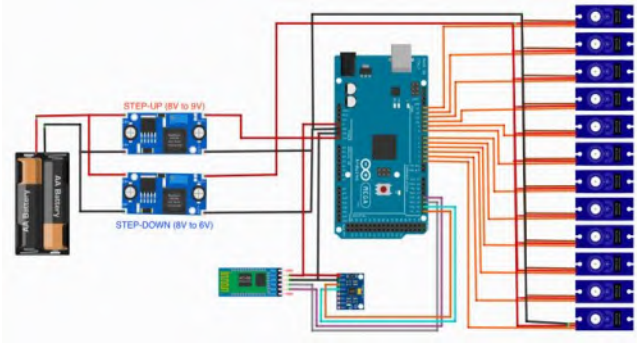
Sadržaj – U ovom radu prikazan je osnovni koncept dizajna, realizacija i način upravljanja robota sa četiri ekstremiteta sa ciljem da se demonstriraju široke mogućnosti ove vrste robota i da se ukaže na njihovu važnu ulogu koju će imati u bliskoj budućnosti u raznim aspektima života. Zbog njihove mogućnosti da vešto savladavaju kompleksne terene, njihova primena će biti značajna.

I. UVOD

Mobilini roboti predstavljaju odličnu alternativu u obavljanju poslova koji su za čoveka rizični ili teški za obavljanje. Oni se mogu podeliti na robote sa točkovima i na robote sa ekstremitetima. Jednostavniji dizajn i upravljanje imaju roboti sa točkovima, međutim, njihova primena je ograničena terenom po kojim se oni mogu kretati. Iz tog razloga raste potreba za robotima sa ekstremitetima koji sa lakoćom mogu da savladaju teren koji robot sa točkovima ne bi mogao. Roboti sa ekstremitetima su dugo bili veliki izazov za inženjere zbog njihove kompleksnosti, ali danas oni postaju sve popularniji. Najveći izazov kod razvoja ove vrste robota predstavlja kompleksna inverzna kinematika, algoritam za održavanje ravnoteže, odabir materijala za izradu robota, a posebno krajeva ekstremiteta koji dolaze u kontakt sa terenom po kojim se robot kreće i naravno dizajn samog robota. Postoji mnogo različitih konfiguracija ovakve vrste robota, od kojih je najpopularnija konfiguracija gde se robot oslanja na četiri ekstremiteta. Ovakav dizajn se pokazao kao jako stabilan, robustan i efikasan i iz tog razloga postaje sve popularniji u svetu. Trenutno se ovakvi roboti primenjuju u vojsci, raznim industrijama, svemirskim programima, a čak se prodaju i kao igračke na daljinski upravljač. Međutim, ovaj tip robota je još uvek na samom početku razvoja i potrebno je njihovo usavršavanje. U ovom radu biće prikazana praktična rešenja dizajna robota sa četiri ekstremiteta, inverzne kinematike potrebne za upravljanje i električna šema.

II. HARDVER ROBOTA

Aktuatori koji pokreću robot su 12 servo motora koji se upravljaju uz pomoć PWM signala. Za upravljanje je iskorišćena Arduino mega pločica sa Atmega2560 mikrokontrolerom iz razloga što ovaj mikrokontroler poseduje dovoljan broj pinova na kojima se može generisati PWM signal za upravljanje servo motorima [2-5]. Za komunikaciju sa korisnikom koji upravlja robotom korisiti se bluetooth modul, a za balans i ravnotežu koristi se modul sa žiroskopom i akcelerometrom. Na slici 1. se može videti kako su ovi moduli povezani sa arduinom, a na slikama 2. i 3. kako je ta električna šema praktčno realizovana na pertianks ploči. Radi uštede prostora i jednostavnosti koristi se jedno napajanje za servo motore i za svu ostalu elektroniku. Upotrebljena su dva modula koja regulišu napon i šumove, jedan step-up za mikrokontroler i jedan step-down modul za servo motore [1]. Ovakva podela je urađena iz razloga što servo motori proizvode šumove, a mikrokontroleru je potreban stabilan napon za rad. U većini slučejva se iz ovog razloga koriste dva zasebna izvora napajanja za mikrokontroler i servo motore, međutim, na pomenuti način je moguće postići isti rezultat i sa jednim izvorom napajanja. Kako motori zahtevaju veliku struju, za napajanje su odabrane litijumske baterije 18650 koje obezbeđuju autonomiju rada od sat vremena u našem slučaju.



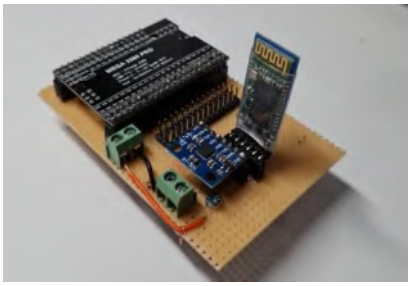
Slika 1. Električna šema robota sa četiri ekstremiteta.

TABELA I
ISKORIŠĆENE KOMPONENTE

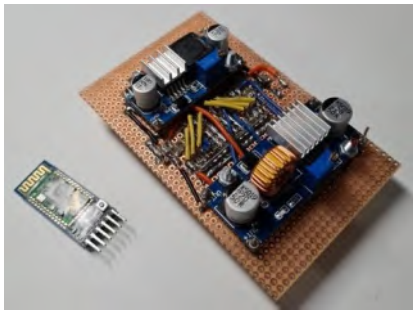
Komponenta	Model	Količina
Pločica sa mikrokontrolerom	Arduino Mega 2560	1
Bluetooth model	HC-05	1
Žiroskop i akcelerometar	MPU 6050	1
Litijumska baterija 18650 3.7V	N18650CK	2
Servo motor	MG90s	12

N. Branković je student na modulu Elektronske komponente i mikrosistemi, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: nikolabrankovic@elfak.rs.

A. Pavlovski je student na modulu Upravljanje sistemima, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: aleksandar.pavlovski@elfak.rs.



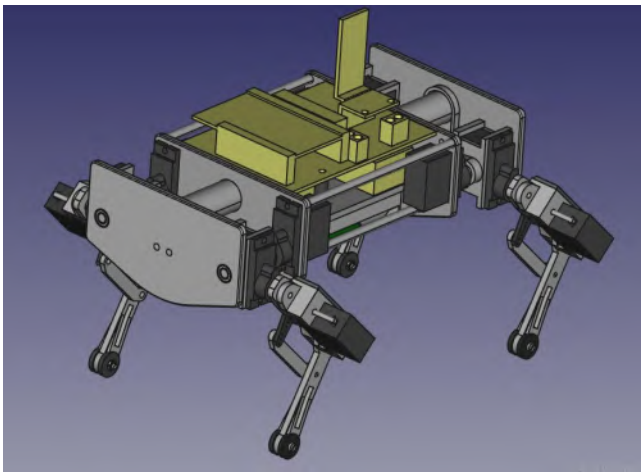
Slika 2. Praktična realizacija hardvera (gornja strana).



Slika 3. Praktična realizacija hardvera (donja strana).

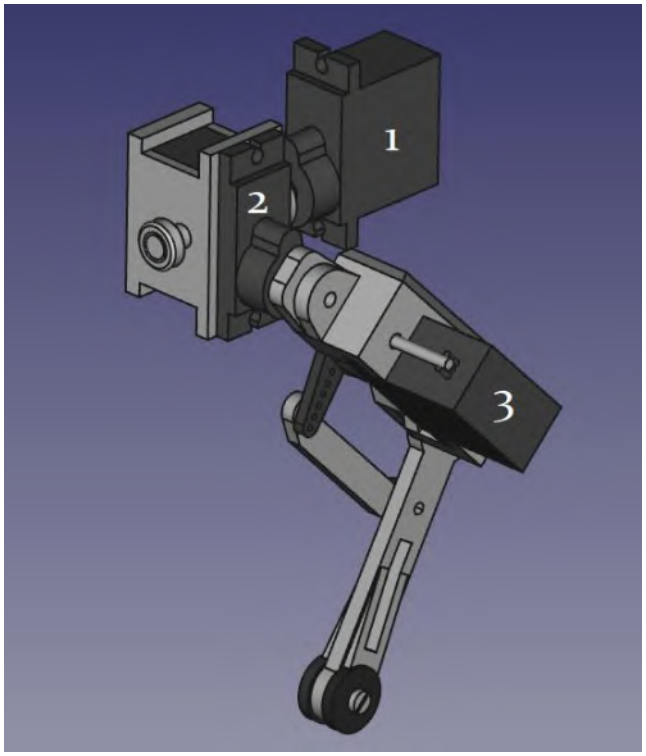
III. DIZAJN ROBOTA SA ČETIRI EKSTREMITETA

Dizajn se bazira na anatomiji životinja koje se kreću na četiri noge. Svaka noga se pokreće uz pomoć tri servo motora što omogućava robotu da se stabilno kreće u trodimenzionalnom prostoru. Na slici 4. se može videti u kom položaju su postavljeni pomenuti servo motori da bi se postigla ista funkcionalnost ekstremiteta kao kod životinja.



Slika 4. 3D model robota izrađen u FreeCAD programu. Robot se sastoji od 23 delova štampanih uz pomoć 3D štampača.

Na mestima na kojima je to potrebno, dodati su kuglični ležajevi da bi se smanjilo trenje između čvrstih delova. Da bi se povećalo trenje između robota i terena po kojem se on kreće, na krajevima ekstremiteta dodata je guma. Inspiracija za ovakvo rešenje pronađena je kod vozila (automobil, bicikl, motor...). Najteži deo predstavljaju baterije i njihov položaj je jako bitan za ravnotežu i stabilnost tako da su iz tog razloga smeštene u donjem delu na sredini šasije. Ovakvo rešenje se već primenjuje u automobilskoj industriji kod električnih automobila iz istog razloga.



Slika 5. Položaj motora.

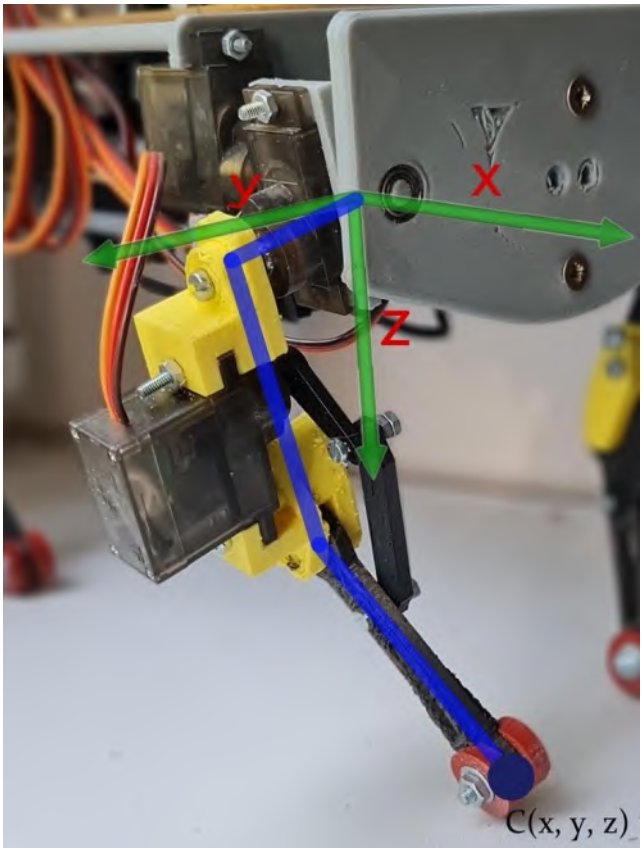
Na slici 5. se jasno vidi položaj motora. Motori 2 i 3 omogućavaju kretanje u dvodimenzionalnom prostoru i preslikavaju izgled ekstremiteta kod životinja. Kako dvodimenzionalno kretanje ekstremiteta nije dovoljno za upravljanje ovakvom vrstom robota, dodat je i motor 1 koji upotpunjava funkcionalnost ekstremiteta i omogućava njegovo kretanje u 3D prostoru. Takođe, motor 3 je podignut što je više moguće da bi se poboljšala stabilnost.

Ovako dizajniran robot sa gore pomenutim komponentama i 3D delovima ima masu od 497 grama, a dimenizije su: 174x134x110 mm.

IV. INVERZNA KINEMATIKA

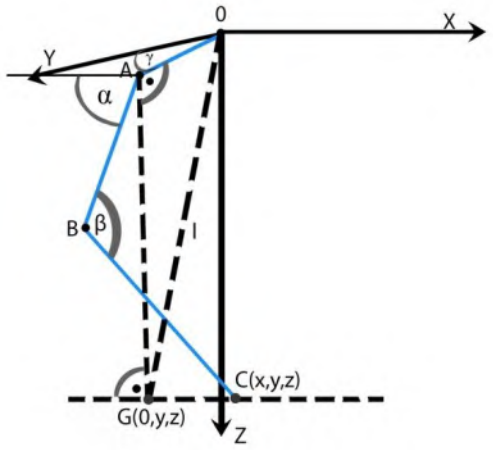
Glavni zadatak upravljanja robotom je postavljanje krajeva svakog od ekstremiteta na željenu poziciju. Svaki od ekstremiteta nalazi se u svom koordinatnom sistemu prikazanom na slici 6. na sledećoj strani, a svaki od vrhova ekstremiteta ima svoj položaj u ovom prostoru opisan koordinatama x, y i z tačke C. Da bi se određeni vrh ekstremiteta, to jest tačka C, dovela u željeni položaj u koordinatnom sistemu, potrebno je zadati određeni ugao servo motrima 1, 2 i 3, a ti uglovi se izračunavaju uz pomoć inverzne kinematike. Ovo podrazumeva određena

trigonometrijska izračunavanja uz pomoć kojih se dobijaju traženi uglovi.



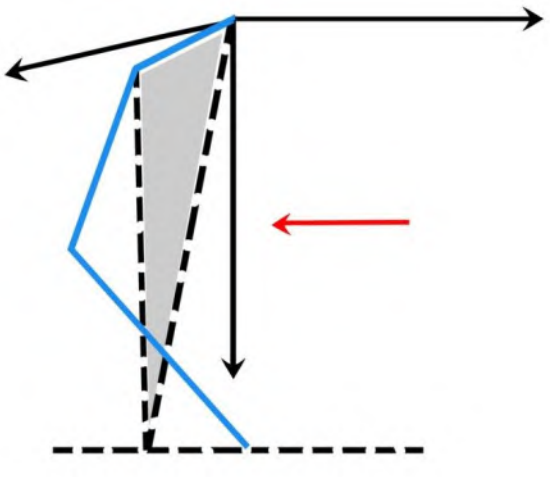
Slika 6. Položaj koordinatnog sistema i vrh ekstremiteta (tačka C).

Početak koordinatnog sistema nalazi se u preseku dveju pravih osovina motora 1 i 2. U nastavku će biti prikazan postupak dobijanja uglova α, β i γ koji predstavljaju položaj servo motora. Ovi uglovi su obeleženi na slici 7.

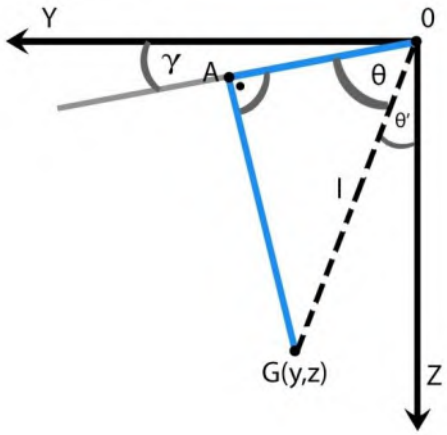


Slika 7. Geometrijski prikaz ekstremiteta. Označeni su uglovi α, β i γ koje treba izračunati, a poznate su koordinate x, y i z tačke C.

Uglove je najlakše izračunati uz pomoć trigonometrijskih funkcija iz trouglova u kojima se oni nalaze. Posmatramo prvo trougao OAG (slika 8), a detaljno prikazan i označen na slici 9.



Slika 8. Trougao OAG.



Slika 9. Trougao OAG i traženi ugao γ.

Na osnovu slike 9 izveden je sledeći postupak za dobijanje ugla γ [3]:

$$l = \sqrt{y^2 + z^2} \tag{1}$$

$$\theta = \arccos\left(\frac{AO}{l}\right) \tag{2}$$

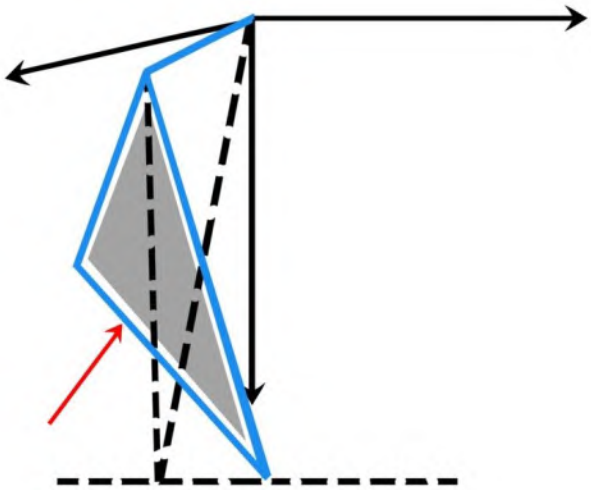
$$\theta' = \arctan\left(\frac{y}{z}\right) \tag{3}$$

$$\gamma = 90^{\circ} - (\theta + \theta') \tag{4}$$

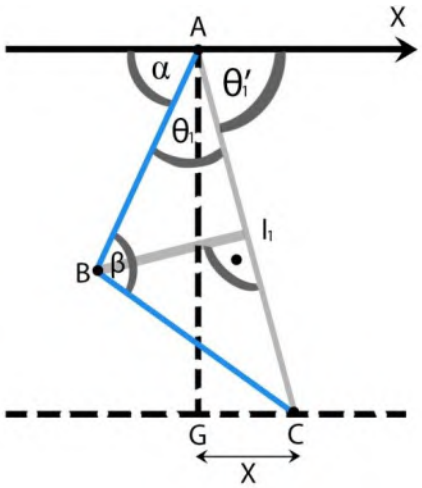
$$AG = \sqrt{l^2 - AO^2} \tag{5}$$

Napomena: dužina AG je potreba za izračunavanje ostalih uglova.

Sada posmatramo trougao ABC (slika 10), a detaljno obeležen na slici 11.



Slika 10. Trougao ABC.



Slika 11. Trougao ABC i traženi uglovi α i β.

Na osnovu slike 11 izveden je sledeći postupak za dobijanje uglova α i β [3]:

$$AC = \sqrt{x^2 + AG^2} \quad (6)$$

$$\theta_1 = \arccos\left(\frac{l_1}{2 * AB}\right) \quad (7)$$

$$\theta_1' = \arccos\left(\frac{x}{l_1}\right) \quad (8)$$

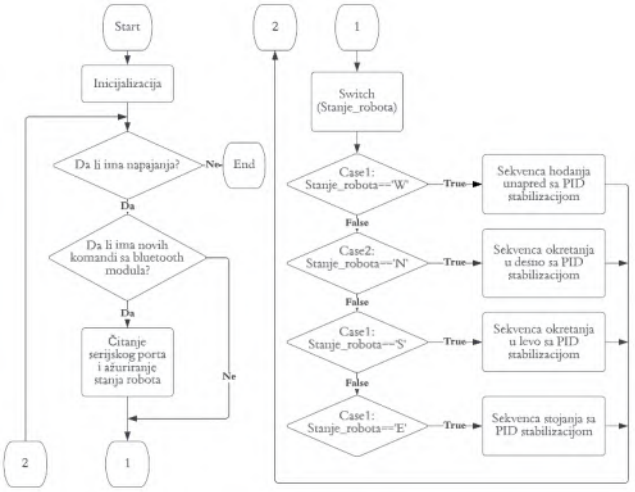
$$\alpha = 180 - (\theta_1 + \theta_1') \quad (9)$$

$$\beta = 180 - 2 * \theta_1 \quad (10)$$

U kodu je implementirana funkcija koja za parametre dobija tražene koordinate x, y i z, a kao rezultat daje tražene uglove servo motora α, β i γ.

V. ALGORITAM APLIKACIJE U ARDUINU

Na slici 12. je prikazan uprošćeni dijagram toka koji služi za upravljanje robotom. Kao što se može videti na slici, robot se može naći u jednom od četiri stanja, a nakon izvršenja jedne sekvence nekog od stanja se proverava da li je sa bluetooth modula stigla nova komanda za promenu stanja. Algoritam je prilagodjen mogućnostima realizovanog robota. Robot sa jačim motorima bi mogao da se kreće unapred i da skreće u istom trenutku

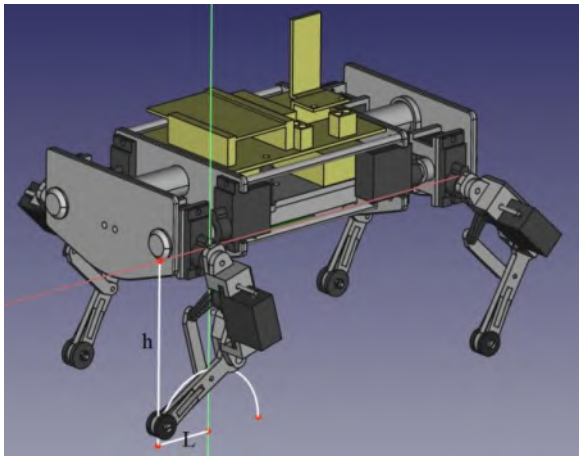


Slika 12. Dijagram toka.

VI. PARAMETRI KORAKA

U svakom trenutku je moguće promeniti dužinu koraka noge 2*L i visinu h na kojoj se određena noga postavlja. Na slici 13. su prikazani ovi parametri. Funkcija koja opisuje kretanje noge je:

$$z = h - \sqrt{L^2 - x^2} \quad (11)$$

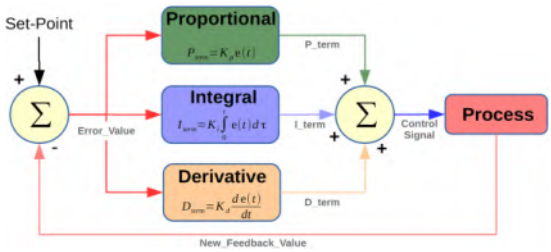


Slika 13. Parametri koraka i grafik funkcije kretanja noge.

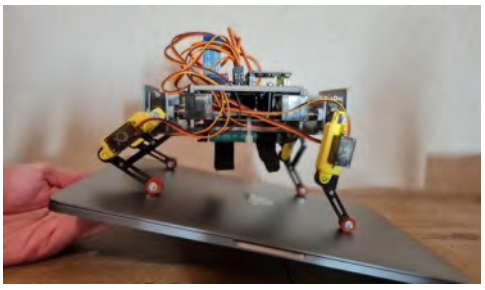
One noge koje se kreću u napred se pomeraju po gore pomenutoj funkciji i to tako da koordinata x raste od -L do +L, a one noge koje se kreću unazad se kreću po funkciji z = h i to tako da koordinata x opada od +L do -L. Koordinata y pri pravom kretanju ostaje konstantna i ima vrednost dužine OA da bi noge bile u normalnom položaju u odnosu na podlogu. Pri skretanju, potrebno je menjati i y koordinatu noge.

VII. STABILIZACIJA ROBOTA POMOĆU MODULA MPU 6050

Modul MPU 6050 komunicira sa arduino pločicom uz pomoć I2C komunikacije i iz njega se očitavaju dva ugla, a to su nagibi u odnosu na x i y osu [4]. Korišćenjem PID regulatora računa se novi položaj svakog ekstremiteta u zavisnosti od greške koja predstavlja razliku željenih uglova i uglova očitanih sa MPU 6050 modula. Uz pomoć PID regulatora se računaju vrednosti upravljanja koje se dodaju ili oduzimaju visini h sa slike 13 u zavisnosti od strane naginjanja robota. Vrednost upravljanja se dodaje parametru h nogama koje su na suprotnoj strani od strane naginjanja robota, a oduzima onim nogama koje se nalaze na strani na kojoj se robot naginje. Ovakav jednostavan algoritam omogućava robotu da se održava na podlozi koja predstavlja ravan, što je prikazano na slici 15., ali ne i na neravnom terenu koji se ne može predstaviti uz pomoć ravni. Takav algoritam je dosta kompleksniji. U našem slučaju parametri PID regulatora su eksperimentalno određeni. Na slici 14. prikazana je strukturna blok šema PID regulatora robota gde Set-Point predstavlja željeni ugao robota, New_Feedback_Value ugao očitan sa žiroskopa, a Control signal vrednost koja se dodaje parametru h. Proces označen na slici zapravo predstavlja robota.



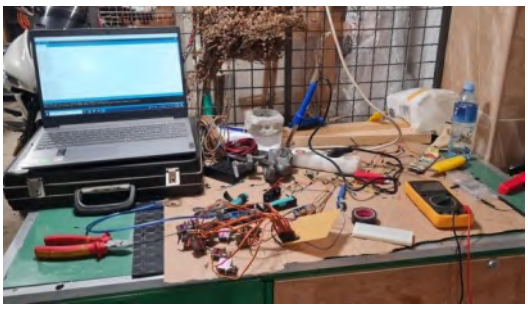
Slika 14. Strukturna blok šema PID regulatora robota.



Slika 15. Održavanje horizontalnog položaja robota.

VIII. PRAKTIČNA REALIZACIJA I RAZVOJNI PROCES

Ideje predstavljene na prethodnim strama realizovane su i praktično. Bilo je potrebno par meseci za prakičnu izradu projekta, to jest, od ideje do realizacije. Za to vreme bilo je neophodno savladati rad u nekom od CAD softvera, u ovom slučaju je to bio FreeCAD program, za uspešan 3D dizajn, savladati veštinu lemljenja, čitanje tehničkih dokumentacija radi odabira potrebnih komponenti, inverznu kinematiku ali takođe i savladavanje do sada nepoznatih problema koji su se često javljali u procesu izrade projekta. Na slici 16. prikazano je radno okruženje za vreme razvoja projekta.



Slika 16. Radno okruženje.

Razvoj robota je dug proces, a u našem slučaju se odvijao u sledećim fazama:

A. Ideja

Ideja za izradu ovakve vrste robota dolazi pre svega od činjenice da će ovakvi roboti uskoro biti prisutni u svakodnevnom životu iz razloga što su se pokazali kao jako dobro rešenje za obavljanje raznih zadataka, a pri tom mogu da se kreću po raznim terenima. Iz ovog razloga treba posvetiti što više pažnje njihovom razvoju.

B. Planiranje resursa

Pri izradi projekta uzete su obzir komponente koje su lako dostupne i koje zadovoljavaju potrebe projekta. Svaka komponenta se lako može zameniti u slučaju otkazivanja iz razloga što je robot dizajniran modularno.

C. Dizajn električne šeme i izrada

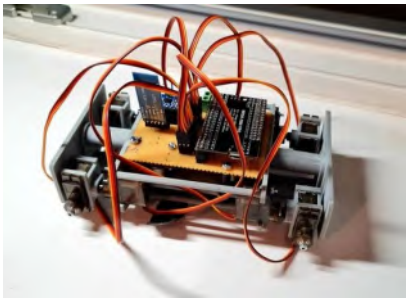
Na osnovu do sada stečenog znanja i kompoenti dizajnirana je električna šema koja je praktično realizovana na pertinaks pločici dimenzija 62.5x89.3mm

D. Dizajn 3D modela

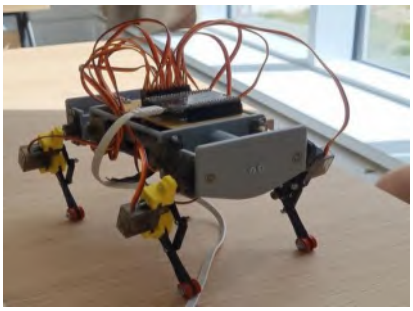
Ovaj deo zahteva poznavanje CAD softvera. Najpre se dizajniraju komponente, a onda i 3D delovi koji će se štampati. Nakon toga se svi delovi i komponente sastave u jednu celinu i koriguje se dizajn ukoliko je to neophodno.

E. Programiranje robota

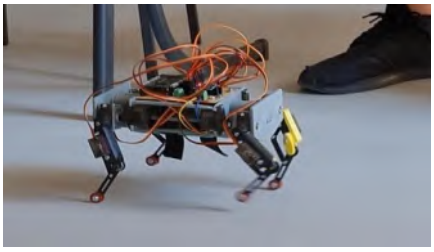
AtMega kontroleri na Arduino pločici se programiraju u programskom jeziku C/C++. Najbitnija stavka jeste implementacija inverzne kinematike.



Slika 17. Telo robota.



Slika 18. Programiranje robota.



Slika 19. Gotov projekat, robot u pokretu.

IX. ZAKLJUČAK

Razvoj ovakve vrste robota zahteva poznavanje raznih oblasti, ali i intuitivno razmišljanje iz razloga što se dizajn ovog robota oslanja na živa bića koja hodaju na četiri noge. Ovakav koncept može da posluži kao dobra osnova za dalji razvoj i usavršavanje. U zavisnosti od potreba, osnovni koncept prikazan na slikama 17, 18. i 19. se može prilagoditi kako bi se što efikasnije rešili konkretni problemi. Konkretno ovaj projekat će dobiti sledeću fazu i na njemu će se raditi i dalje.

ZAHVALNICA

Zahvaljujemo se asistentu Nikoli Mitroviću na sugestijama pri izradi projekta i 3D štampanju našeg modela.

LITERATURA

[1] Z. Prijić, slajdovi sa prezentacije „Prekidački izvori napajanja“, slajdovi 27-33. Dostupno na: http://mikro.elfak.ni.ac.rs/wp-content/uploads/2017/10/Literatura/Analogna/Analogna_smps_pred.pdf
[2] V. Sibinović „Literatura Mehatronika“
[3] Andreas Aristidou Inverse Kinematics: a review of existing techniques and introduction of a new fast iterative solver, strane 11-25. Dostupno na: https://www.researchgate.net/publication/273166356_Inverse_Kinematics_a_review_of_existing_techniques_and_introduction_of_a_new_fast_iterative_solver
[4] Tehnička dokumentacija MPU 6050 senzora. Dostupno na: https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf
[5] Michael Margolis „Arduino Cookbook“, strane 264 i 265. Dostupno na: https://juniorfall.files.wordpress.com/2011/11/arduino-cookbook.pdf

Primena dekadnog brojača u realizaciji zidnog sata

Lazar Arsić

Sadržaj – U ovom radu opisana je realizacija uz pomoć dekadnog brojača i logičkih kola. Na početku je dat kratak uvod u kom su date prednosti i nedostaci ovog uređaja. Zatim je dat kompletan postupak projektovanja samog uređaja.

I. UVOD

Danas živimo u svetu u kojem gotovo svaki uređaj koristi neku vrstu mikrokontrolera za svoje funkcionisanje. Mikrokontroleri su danas postali nezaobilazna komponenta u realizaciji uređaja. Međutim za upotrebu samih mikrokontrolera je potrebno i znanje iz oblasti programiranja. Postoje ljudi koji ne vole da pišu kodove. Upravo iz tog razloga je i nastao ovaj rad.

Za realizaciju samog sata korišćen je dekadni brojač i pomoćna logička kola za sinhronizaciju. Prikaz je uz pomoć LED dioda. Dekadni brojač je vrsta brojača koji prolazi kroz deset stanja prilikom brojanja. Ovde je korišćeno kolo CD4017 [1] dekadni džonsonov brojač koji ima 10 izlaznih pinova od kojih je jedan aktivan u zavisnosti od stanja brojača. Kao izvor taktnog signala korišćen je vrlo popularni tajmer NE555 [2] u astabilnom rezimu rada.

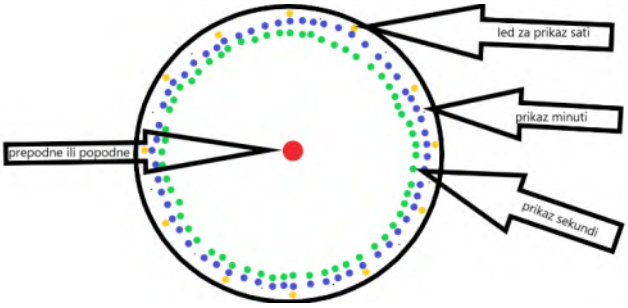
Prednost ovakve realizacije zidnog sata je to što nema programiranja i nema nikakvog kodiranja signala. Nedostatak je to što sat mora da se uključi u 00:00 jer samo tada će on raditi kako treba i prikazivati tačno vreme. Ovo mora jer ne postoji opcija podešavanja vremena.

II. RAZVOJ UREĐAJA

Ovaj sat prikazuje sate, minute i sekunde. Tako da je potrebno imati dva brojača (brojači za minute i sekunde su isti) jedan koji broji do 60, dok drugi broji do 12. Kako ovi brojači ne postoje potrebno ih je realizovati uz pomoć brojača koji već postoje.

Na slici 1 data je skica izgleda samog sata, crtana u *Paint*-u, onako kako bi on fizički trebao da izgleda. Žute tačkice predstavljaju LED diode kojima se prikažuju sati, plavim diode za prikaz minuta i zelene za prikaz sekundi. U centru se nalazi crvena tačka kojom je predstavljena dioda kojom se signalizira da li je pre podne ili popodne.

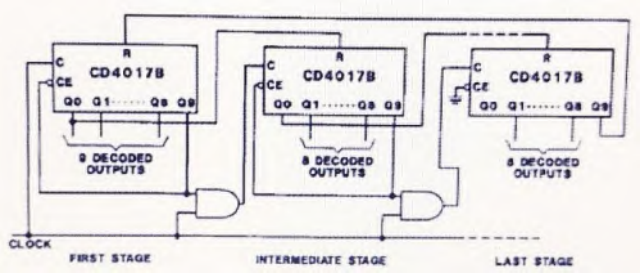
L. Arsić je student četvrte godine modula Elektronske komponente i mikrosistemi, Elektronski fakultet, Niš, Srbija, E-mail: lazar.arsic@elfak.rs.



Slika 1. Skica izgleda samog sata.

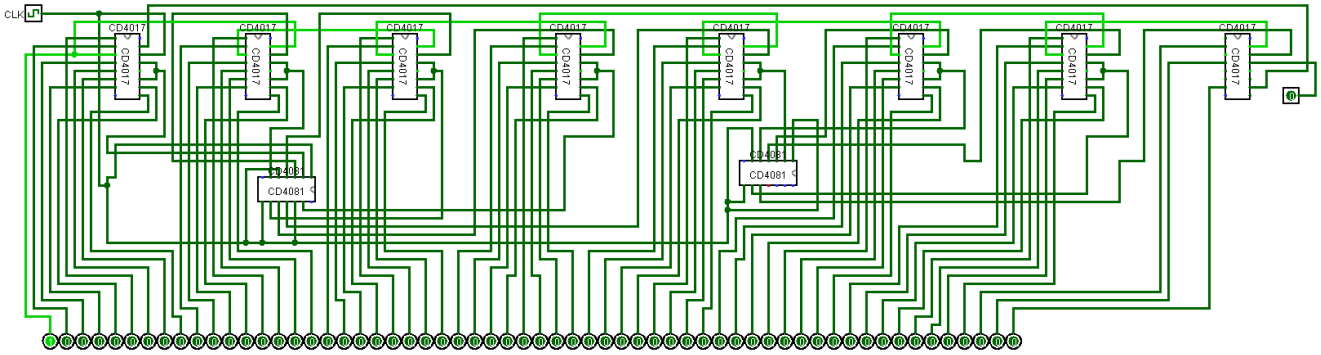
A. Realizacija brojača od 60 izlaza

Za realizaciju brojača koji ima 60 izlaza izabrano je kolo CD4017 koje ima 10 izlaza. Više ovakvih kola je kaskadno povezano na određeni način uz koričćenje dodatnih logičkih kola. Šema kaskadnog vezivanja brojača data je na slici 2.



Slika 2. Šema kaskadnog vezivanja brojača.

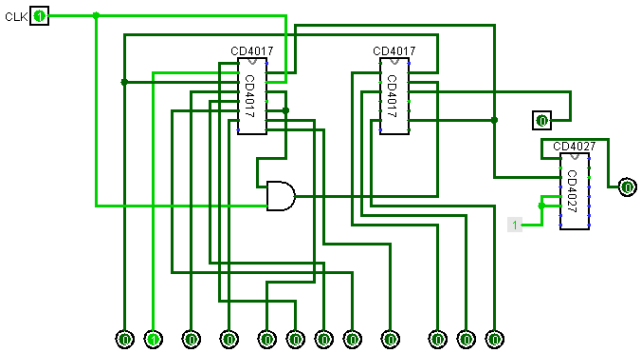
Kako ovo radi: *clock* signal je zajednički za sve brojače ali se samo na prvi u nizu vezuje direktno dok na sve ostale preko jednog AND logičkog kola. Kada prvi brojač izbroji do kraja pin Q9 je na logičkoj jedinici i na taj način omogućava da sledeći u nizu dobije *clock* signal. Pin Q9 je takođe vezan i za pin CE koji je dozvola za brojanje samog brojača koji je *active low* tako da kad na njega dođe jedinica on prestaje da broji. Pin Q1 svih brojača je iskorišćen za resetovanje sledećeg u nizu brojača. Pin Q1 prvog u nizu brojača je moguće koristiti kao izlaz konačnog brojača dok se kod ostalih on ne može koristiti. Pin CE poslednjeg brojača je vezan na masu odnosno logičku nulu. Realizovan brojač od 60 izlaza je prikazan na slici 3.



Slika 3. Realizacija brojača od 60 izlaza u *Logisim*-u.

B. Realizacija brojača od 12 izlaza

Realizacija ovog brojača je potpuno ista kao i brojača od 60 izlaza sti što se koriste samo dva kola CD4017. brojač je prikazan na slici 4.

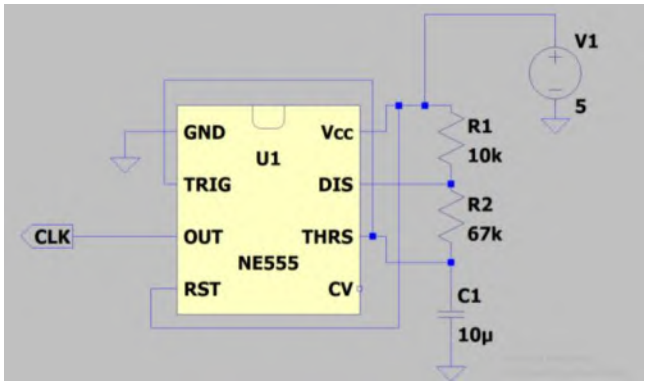


Slika 4. Realizacija brojača sa 12 izlaza.

Da bi se moglo odrediti da li je 10 sati pre podne ili popodne ovde je dodat jedan sklop. Sklop se sastoji od jednog T flip-flopa (realizvanog od jednog JK flip-flopa) i jedne LED diode. Kada brojač izbroji do kraja on okine flip-flop koji invertuje stanje na svom izlazu. LED dioda svetli ili ne svetli, svetli kada je prošlo podne.

C. Realizacija izvora clock signala

Izvor *clock* signala je realizovan uz pomoć kola NE555 i nekoliko otpornika i kondenzatora. Šema je prikazana na slici 5.



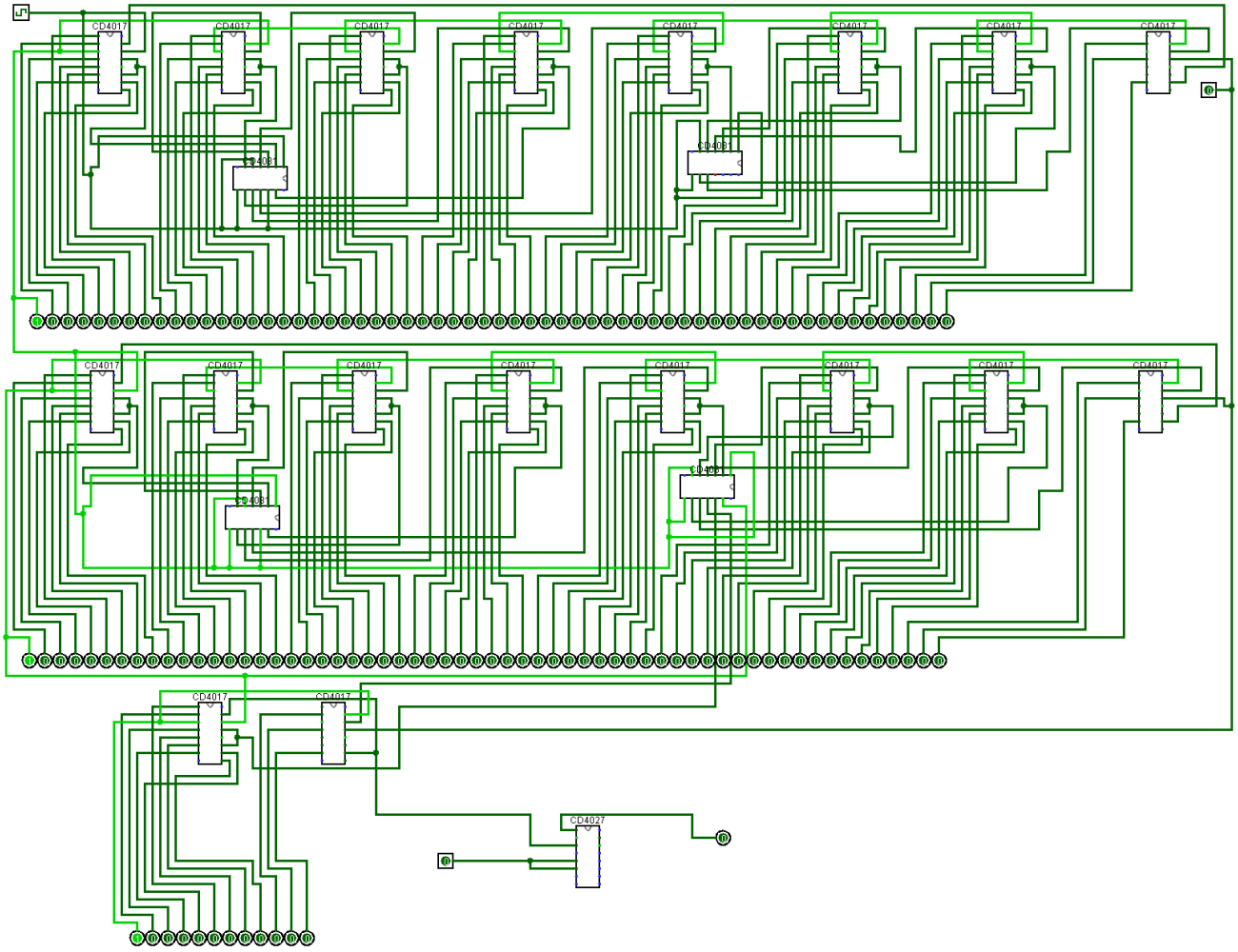
Slika 5. Šema generatora *clock* signala *LTspice*-u.

Kolo NE555 ovde radi u astabilnom režimu rada i generiše povorku impulsa frekvencije 1Hz. Vrednosti komponenti su Ra=10 kΩ, Rb=67 kΩ, C=10 µF. Pošto nepostoji standardna vrednost otpornika od 67 kΩ umesto fiksnog otpornika stavljen je trimer potenciometar od 100 kΩ i vezan kao promenljivi otpornik. Vrednost *duty cycle* je oko 50%.

D. Povezivanje u celinu

Brojači koji prikazuju sekunde, minute i sate su povezani na sledeći način: poslednji izlaz brojača sekundi je povezan na ulaz *clock* signala brojača minuta i na taj način on predstavlja izvor *clock* signala za brojač minuta, dok je poslednji izlaz brojača minuta vezan za ulaz *clock* signala brojača sati i predstavlja njegov izvor *clock* signala.

Šema realizovanog sata u programu *Logisim* data je na slici 6.



Slika 6. Šema sata u *Logisim*-u.

E. Komponente koje su korišćene u realizaciji

- CD4017
- CD4081 [3]
- CD4027 [4]
- NE555
- 10kΩ
- 100kΩ trimer
- 10uF

U tabeli I su dati nazivi pinova korišćenih integrisanih kola.

TABELA I.
PINOVI

Pin/ic	CD4017	CD4081	CD4027	NE555
1	Q5	A1	Q2	GND
2	Q1	B1	/Q2	TRIG
3	Q0	Q1	CLK2	OUT
4	Q2	Q2	RST2	RST
5	Q6	A2	K2	CON
6	Q7	B2	J2	THRES
7	Q3	GND	SET2	DIS
8	GND	A3	GND	VCC
9	Q8	B3	SET1	
10	Q4	Q3	K1	
11	Q9	Q4	J1	
12	C.OUT	A4	RST1	
13	CLK.EN	B4	CLK1	
14	CLK	VCC	/Q1	
15	MR		Q1	
16	VCC		VCC	

III. ZAKLJUČAK

Jedna od bitnijih stvari koje sam naučio prilikom realizacije samog ovog projekta jeste koliko je važno znati čitati tehničku dokumentaciju komponenti. Iako je ovo na početku delovalo lako za projektovanje kada se krene u samu realizaciju se ustvari vide poteškoće sa kojima se suočavaju svi projektanti na svetu pri projektovanju uređaja. Što se tiče tačnosti merenja vremena ovim uređajem ona pre svega zavisi od tačnosti *clock* signala koji opet zavisi od tačnosti odabranih komponenti prilikom izrade izvora *clock* signala.

ZAHVALNICA

Autor se zahvaljuje prof. dr Danijelu Dankoviću za pomoć i korisne savete prilikom izrade samog projekta.

LITERATURA

[1] Dekadni brojač CD4017. Dostupno na – https://pdf1.alldatasheet.com/datasheet-pdf/view/26855/TI/CD4017.html
[2] Tajmer NE555. Dostupno na – https://pdf1.alldatasheet.com/datasheet-pdf/view/355583/TI/NE555.html
[3] Četvorostruko dvoulazno I kolo CD4081. Dostupno na – https://pdf1.alldatasheet.com/datasheet-pdf/view/50864/FAIRCHILD/CD4081.html
[4] Dvostruki JK flip-flopCD4027. Dostupno na – https://pdf1.alldatasheet.com/datasheet-pdf/view/26863/TI/CD4027.html
[5] Prof. Dr Danijel Danković -Digitalna mikroelektronika pomerački registri i brojači-Predavanja

Aplikacija za upoređivanje temperatura zdrave i obolele strane lica u *Matlab* okruženju

Tamara Đorđević

Sadržaj – U radu je opisana *Matlab* aplikacija kojom se vrše transformacije termograma zbog što boljeg međusobnog poklapanja, u cilju određivanja temperaturne razlike između njih. Termogrami predstavljaju levu i desnu stranu regije glave i vrata osobe. Kod ispitivanja mogućnosti termovizijske kamere kao pouzdanog i bezbednog dijagnostičkog alata za bolesti maksilofacijalne regije, veoma je važno tačno i precizno utvrđivanje temperaturnih karakteristika regiona od interesa. U prikazanoj *Matlab* aplikaciji primenjuju se automatski i manuelni pristup definisanju regija od interesa i određivanju njihovih temperaturnih razlika.

I. Uvod

Temperatura kože je produkt toplote, koji se prenosi na površinu kože iz dubokih tkiva, prvenstveno iz unutrašnjih organa (jetre, srca, mozga i mišića) [1]. Temperatura kože oslikava stanje mikrocirkulacije, koje je pod kontrolom autonomnog nervnog sistema, a time i mogućnost termovizije da proceni disfunkciju autonomnog nervnog sistema i vaskularnog sistema. U stvari termovizija, detektujući promene vrednosti temperature kože može pomoći u objektivnoj proceni bolesti koje direktno ili indirektno utiču na vaskularni tonus mikrocirkulacije [2].

Svi objekti sa temperaturom iznad apsolutne nule (T>0 K) emituju infracrveno zračenje [3]. Infracrveno zračenje je deo elektromagnetnog spektra, koji se nalazi između vidljive svetlosti i radio talasa. Talasna dužina ovog zračenja je u opsegu od 0.7 μm do 1000 μm [4].

Koža je crno telo sa emisionim faktorom 0.98, što pokazuje da je savršen emiter infracrvenog zračenja na sobnoj temperaturi. Termovizijska kamera detektuje emitovano infracrveno zračenje sa površine kože i konvertuje intenzitet emitovanog zračenja u temperaturu [4]. Termogram predstavlja grafičku mapu regionalne distribucije temperaturnih vrednosti. Kod zdravih osoba prisutna je simetrična distribucija vrednosti temperature kože u odnosu na osu postavljenu u srednju ravan tela. Ukoliko se na termogramu javi regionalna temperaturna asimetrija, ona ukazuje na patološki proces [5].

Za realizaciju ovog rada primenjen je softver *Matlab* kompanije *Mathworks* [6]. Rezultati koji su ovde predstavljeni samo su deo većeg naučnog istraživanja koje proučava mogućnosti termovizijske kamere u dijagnostici tumora pljuvačnih žlezda i intraoralnih tumora [7], [8].

T. Đorđević je student na modulu Upravljanje sistemima, Elektronskog fakulteta Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: tamara.djordjevic@elfak.rs.

II. Eksperimentalna postavka

Za eksperimentalna merenja korišćen je termovizijski sistem *Varioscan 3021ST* kompanije *Jenoptic* [9]. Rezolucija kamere se definiše sa 360 piksela po liniji i 240 linija po slici, termalna rezolucija je ±0.03°C, temperaturni opseg od –40 do 1200°C, a spektralni opseg 8–12 μm.

Zbog specifičnog formata slika napravljenih termovizijskom kamerom, nije moguće primeniti standardni grafički softver za vizuelizaciju temperatura već samo specijalizovani softver za korišćenu kameru *Irbis* [10]. S obzirom da primenom softvera *Irbis* nije moguće izvršiti odgovarajuće transformacije termograma, ovaj softver je iskorišćen samo da eksportuje termograme u odgovarajuće *ASCII* tabele.

Dalja obrada podataka podrazumeva objedinjavanje dobijenih tabela koje se odnose na levu i desnu stranu lica svake pojedinačne osobe u jedan fajl, primenom *Matlab* softvera.

Imajući u vidu da se u procesu snimaju karakteristike i leve i desne strane lica posebno, neophodno je posvetiti pažnju i faktorima koji utiču na usaglašavanje strana lica, poput: razlika u uglovima pod kojima je odgovarajuća strana lica snimljena, razlika u rastojanjima kamere od lica, kao i od prirodnih odstupanja u simetriji leve i desne strane lica. Iz ugla programa, ove razlike se ispoljavaju u vidu podataka koje svaki piksel prikazuje, što znači da nije moguće primeniti jedinstvenu funkciju kojom bi se njihovo dejstvo uklonilo. Kreiranjem algoritma u *Matlab* okruženju moguće je smanjiti uticaj subjektivnih faktora na preciznost usaglašavanja termograma. S obzirom da termogrami sadrže merene vrednosti temperatura, bitno je napomenuti da se postupkom usaglašavanja termograma ne sme vršiti nikakva promena merenih podataka.

III. Realizacija aplikacije u *Matlab* okruženju

Za realizaciju programa primenjuje se *App Designer* koji predstavlja sastavi deo *Matlab* softvera. Proces projektovanja aplikacije započinje izborom grafičkih predstavnika kontrola, koje se postavljaju na strani *Design View*. Primenjene kontrole su: šest dugmeta, jedna padajuća lista, tri numerička spinera, jedno polje i pet labela za prikaz teksta. Dizajn realizovane aplikacije prikazan je na slici 1.

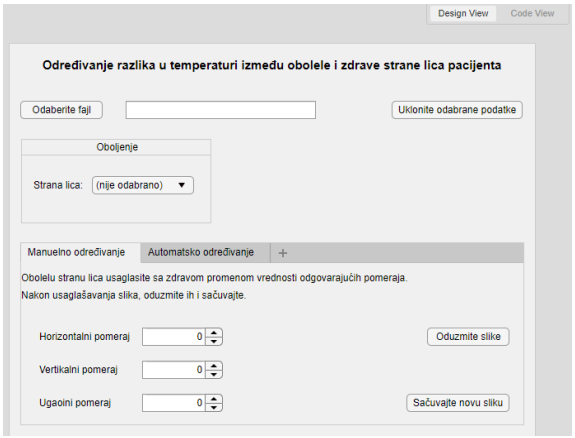
Funkcionalni deo kontrole definiše se selektovanjem kontrole i definisanjem funkcije koja se pokreće kada kontrola promeni vrednost ili se selektuje. Drugi naziv za

ovu funkciju je *callback*, odnosno funkcija povratnog poziva. Navedeno se definiše na strani *Code View*.

Termogrami čiji pikseli čuvaju vrednost merene temperature *Matlab* klasifikuje kao slike odgovarajućeg intenziteta, odnosno *grayscale* slike. Takođe, slike se numerički prikazuju u vidu matrica čije dimenzije odgovaraju dimenzijama slike termovizijske kamere. Drugim rečima, fajl sa ekstenzijom *mat* sadrži sliku leve i sliku desne strane lica i sačinjen je od dve *grayscale* slike dimenzija 240x360. Svaki element, odnosno piksel slike, sadrži vrednost izmerene temperature lica sa originalnih termograma.

Moguće je zaključiti da se operacije usaglašavanja termograma svode na operacije nad matricom i nad elementima u matrici.

Prilikom pokretanja programa prvi korak podrazumeva odabir željenog fajla pritiskom na dugme "Odaberite fajl" iz foldera koji korisnik selektuje, a kod koji se izvršava prikazan je na slici 2. Iz odabranog *mat* fajla učitavaju se matrice leve i desne strane lica u program, a zatim se čuvaju u promenljivama koje imaju prefiks *app*. Definicija promenljivih prikazana je na slici 3. Prefiks omogućava dostupnost podataka svim funkcijama u programu, ali poseduje ograničenja u vidu funkcija i operacija koje je moguće obaviti nad ovakvim tipom promenljive. Ograničenje se prevazilazi uvođenjem lokalnih promenljivih.



Slika 1. Grafički prikaz korisničkog prozora.

```
function OdaberitefajlButtonPushed(app, event)
    [ImeF, LokacijaF] = uigetfile('*.mat','Odaberite mat fajl');
    UlazFajl = fullfile(LokacijaF,ImeF);
    MatUlazFajl = load(UlazFajl);
    app.MatricaR = [MatUlazFajl.R];
    app.MatricaL = [MatUlazFajl.L];
    app.MatricaAO = MatUlazFajl;
    app.NazivfajlaTextArea.Value = ImeF;
end
```

Slika 2. Funkcija povratnog poziva dugmeta "Odaberite fajl".

```
properties (Access = private)
    MatricaR
    MatricaL
    Matrica
    Figura
    MatricaAO
    FiguraAO
end
```

Slika 3. Definicija promenljivih koje sadrže prefiks *app*.

```
function StranalicaDropDownValueChanged(app, event)
    Strana = app.StranalicaDropDown.Value;
    MatL = [app.MatricaL];
    MatR = [app.MatricaR];
    switch Strana
        case 'Desna'
                MatL = fliplr(MatL);
                app.MatricaL = MatL;
                prikazi(app,MatR,MatL);
         case 'Leva'
                MatR = fliplr(MatR);
                app.MatricaR = MatR;
                prikazi(app,MatL,MatR);
    end
end
```

Slika 4. Funkcija povratnog poziva padajuće liste.

```
function Mat = normalizacija(~, matrica)
    Tmax = max(matrica(:));
    Tmin = min(matrica(:));
    Mat = (matrica - Tmin)/(Tmax-Tmin);
end
```

Slika 5. Funkcija za normalizaciju matrice.

```
function  prikazi(app,matA,matB)
    matA = normalizacija(app,matA);
    matB = normalizacija(app,matB);
    imshowpair(matA,matB,'falsecolor');
end
```

Slika 6. Funkcija za istovremeno prikazivanje dve slike bez prikaza vrednosti piksela.

```
function UkloniteodabranepodatkeButtonPushed(app, event)
    app.NazivfajlaTextArea.Value = '';
    app.StranalicaDropDown.Value = '(nije odabrano)';
    app.HorizontalnipomerajSpinner.Value = 0;
    app.VertikalnipomerajSpinner.Value = 0;
    app.UgaoinipomerajSpinner.Value = 0;
    app.Matrica = [];
    app.MatricaL = [];
    app.MatricaR = [];
    app.Figura = [];
    app.FiguraAO = [];
    app.MatricaAO = [];
    close all
end
```

Slika 7. Funkcija povratnog poziva za uklanjanje svih podataka.

Sledeći korak predstavlja odabir strane lica na kojoj se nalazi oboljenje. Iz padajuće liste, korisnik može odabrati levu ili desnu stranu lica. Funkcija koja se poziva prilikom odabira strane, prikazana je na slici 4.

S obzirom da se slike predstavljaju preko intenziteta, neophodno ih je prikazati primenom odgovarajuće palete boja. Intenzitet jednog piksela u odnosu na intenzitet ostalih u matrici određuje se primenom funkcije za normalizaciju (slika 5). Funkcija za prikaz predstavljena je na slici 6. Nakon preklapanja slika, korisnik se može opredeliti za primenu manuelnog, odnosno ručnog, ili automatskog principa preklapanja i upoređivanja slika. U proces selekcije učestvuju samo fajlovi koji imaju ekstenziju *mat*, a naziv fajla biće prikazan u polju za tekst koji se drugačije naziva *Edit field (Text)*.

Na samom kraju, moguće je ukloniti podatke pritiskom na dugme "Uklonite odabrane podatke", čime svi parametri aplikacije uzimaju početne vrednosti. Odgovarajuća funkcija povratnog poziva prikazana je na slici 7.

A. Manuelno preklapanje termograma

U procesu manuelnog preklapanja termograma, korisnik sam podešava vrednosti horizontalnog, vertikalnog i ugaonog pomeraja slike s ciljem da uskladi levu i desnu stranu lica. Grafički prikaz kontrola kojim se vrše promene parametara prikazan je u panelu pod nazivom "Manuelno

određivanje" na slici 1. Prilikom promene jednog ili istovremene promene više parametara poziva se samo jedna funkcija povratnog poziva (slika 8). Iz ove funkcije, poziva se funkcija koja obavlja transformaciju i prikazuje rezultat transformacije za svaku pojedinačnu promenu vrednosti svih navedenih parametara (slika 9).

Kod funkcije za transformaciju prikazan je na slici 10 i analizira 4 različite situacije na osnovu kojih se obavlja translacija i/ili rotacija slike. Nakon transformacije, poziva se ponovo funkcija za prikaz. Ukoliko je korisnik zadovoljan transformacijom, pritiskom na dugme "Oduzmite slike" poziva funkciju koja oduzima vrednosti temperatura usaglašenih slika (slika 11), tako da se od termograma bolesne oduzima termogram zdrave strane lica. Korisnik može sačuvati sliku koja predstavlja rezultat oduzimanja strane lica, pritiskom na dugme "Sačuvajte novu sliku", koja poziva funkciju za čuvanje (slika 12).

```
function HorizontalnipomerajSpinnerValueChanged(app, event)
    NoviGrafik(app);
end
```

Slika 8. Zajednička funkcija povratnog poziva za horizontalni, vertikalni i ugaoni spiner.

```
function NoviGrafik(app)
    Strana = app.StranalicaDropDown.Value;
    h = app.HorizontalnipomerajSpinner.Value;
    v = app.VertikalnipomerajSpinner.Value;
    u = app.UgaoinipomerajSpinner.Value;
    MatR = app.MatricaR;
    MatL = app.MatricaL;
    switch Strana
            case 'Desna'
                    Slika = transforomacija(app,MatR,h, v, u);
                    prikazi(app,Slika,MatL);
            case 'Leva'
                    Slika = transforomacija(app,MatL,h, v, u);
                    prikazi(app,Slika,MatR);
    end
    app.Matrica = Slika;
end
```

Slika 9. Funkcija kojom se prikazuje nova slika za svaku promenu vrednosti parametra.

```
function TrenutnaSlika = transforomacija(~,mat, h, v, u)
    TrenutnaSlika = reshape(mat, 240, 360);
    if h~=0 && v~=0 && u~=0
          TrenutnaSlika = imtranslate(mat,[h, v],'FillValues',1,'OutputView','same');
          TrenutnaSlika = imrotate(TrenutnaSlika,u,'crop');
    elseif h~=0 || v~=0 && u~=0
          TrenutnaSlika = imtranslate(mat,[h, v],'FillValues',1,'OutputView','same');
          TrenutnaSlika = imrotate(TrenutnaSlika,u,'crop');
    elseif h~=0 || v~=0 && u==0
          TrenutnaSlika = imtranslate(mat,[h, v],'FillValues',1,'OutputView','same');
    elseif h==0 && v==0 && u~=0
          TrenutnaSlika = imrotate(mat,u,'crop');
    elseif (h==0) && (v==0) && (u==0)
    end
end
```

Slika 10. Definisanje funkcije za ugaonu ili translatornu transformaciju termograma.

```
function OduzmiteslikeButtonPushed(app, event)
    Strana = app.StranalicaDropDown.Value;
    switch Strana
        case 'Desna'
         matR = app.Matrica;
         matL = app.MatricaL;
         FinalnaSlika = imsubtract(matR,matL);
        case 'Leva'
         matR = app.MatricaR;
         matL = app.Matrica;
         FinalnaSlika = imsubtract(matL,matR);
    end
         app.Matrica = FinalnaSlika;
         RazlikaSlika = prikazifunkcija(FinalnaSlika);
         app.Figura = RazlikaSlika;
end
```

Slika 11. Funkcija povratnog poziva dugmeta za oduzimanje slika.

Funkcija za čuvanje poziva funkciju, koja je realizovana u *mlx* fajlu i koja omogućava primenu standardnih funkcija za čuvanje fajlova u željenom obliku (slika 13), što nije moguće ostvariti u okviru same aplikacije.

```
function SauvajtenovuslikuButtonPushed(app, event)
 [NazivSlike,LokacijaSlike] = uiputfile('*.mat','Odaberite lokaciju i naziv fajla');
 Slika = app.Matrica;
 FiguraSlike = app.Figura;
 NazivFigure = erase(NazivSlike,'.mat');
 sacuvatipodatke(LokacijaSlike,NazivFigure,NazivSlike,Slika,FiguraSlike);
end
```

Slika 12. Funkcija povratnog poziva za dugme za čuvanje rezultata manuelnog upoređivanja.

```
function sacuvatipodatke(LokacijaSl, NazivFigure,NazivMatF,Matrica,Figura)

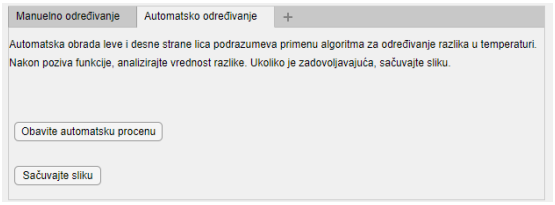
Fajl = fullfile(LokacijaSl,NazivMatF);
saveas(Figura,fullfile(LokacijaSl,NazivFigure),'fig');
save(Fajl,'Matrica');

end
```

Slika 13. Definicija funkcije za čuvanje rezultata obrade u *mlx* fajlu.

B. Automatsko preklapanje termograma

Automatski način preklapanja termograma primenjuje algoritam koji poziva standardne *Matlab* funkcije (slika 14). Automatsko upoređivanje se započinje pritiskom na dugme "Obavite automatsku procenu", čime se poziva funkcija definisana na slici 15. Iz funkcije povratnog poziva, poziva se funkcija definisana u *mlx* fajlu, a njen kod je prikazan na slici 16.



Slika 14. Panel za automatsko određivanje temperaturnih razlika.

```
function ObaviteautomatskuprocenuButtonPushed(app, event)
    Strana = app.StranalicaDropDown.Value;
    Mat = app.MatricaAO;
    matR = [Mat.R];
    matL = [Mat.L];
    [FiguraslikeAO,SlikaAO] = automatskaobrada(matL,matR,Strana);
    app.MatricaAO = SlikaAO;
    app.FiguraAO = FiguraslikeAO;
end
```

Slika 15. Funkcija povratnog poziva za automatsku procenu.

```
function [FiguraSlikeA, SlikaRazlikeA] = automatskaobrada(matricaL,matricaR,StranaLica)

[x,y] = imregconfig('multimodal');
x.MaximumIterations = 500;

switch StranaLica

    case 'Desna'
        matricaL = fliplr(matricaL);
        zajednickaOblast = imregister(matricaR, matricaL, 'affine', x, y);
        SlikaRazlikeA = imsubtract(zajednickaOblast,matricaL);
    case 'Leva'
        matricaR = fliplr(matricaR);
        zajednickaOblast = imregister(matricaL, matricaR, 'affine', x, y);
        SlikaRazlikeA = imsubtract(zajednickaOblast,matricaR);
end

figure('Name','Slika razlike u temperaturama dobijena primenom algoritma')
FiguraSlikeA = imshow(SlikaRazlikeA, []);

end
```

Slika 16. Algoritam automatske procene napisan u *mlx* fajlu.

Postoji mogućnost čuvanja slike pritiskom na dugme

"Sačuvajte sliku". Kod funkcije za čuvanje slike izvršava se nezavisno od koda za čuvanje slike prilikom manuelnog određivanja i prikazan je na slici 17. Kao i kod manuelnog upoređivanja, na kraju se poziva funkcija za čuvanje slike, prikazana na slici 13.

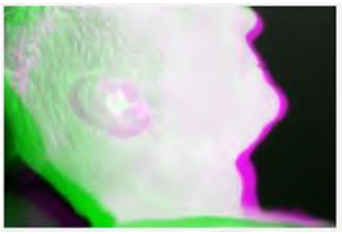
```
function SauvajteslikuButtonPushed(app, event)
[NazivSlikeAO,LokacijaSlikeAO] = uiputfile('*.mat','Odaberite lokaciju i naziv fajla');
SlikaAO = app.MatricaAO;
FiguraSlikeAO = app.FiguraAO;
NazivFigureAO = erase(NazivSlikeAO,'.mat');
sacuvatipodatke(LokacijaSlikeAO,NazivFigureAO,NazivSlikeAO,SlikaAO,FiguraSlikeAO);
end
```

Slika 17. Funkcija povratnog poziva za čuvanje rezultata automatske obrade.

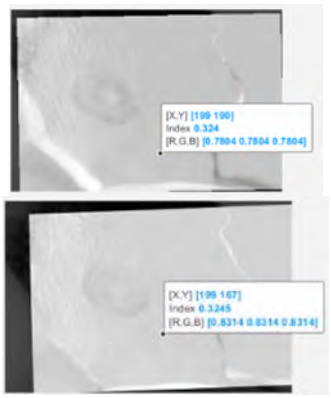
IV. Primer upotrebe aplikacije

Pokretanje programa se započinje opcijom *Run*. Za odabrani *mat* fajl ustanovljeno je da pacijent poseduje oboljenje na desnoj strani lica u predelu ispod uha. Očekuje se da za dati tip oboljenja temperaturne razlike budu pozitivne u tom predelu lica. Prikaz neusaglašenih strana lica, dat je na slici 18.

Postavljanjem vrednosti horizontalnog pomeraja na 10, vertikalnog na 8 i ugaonog na 2, u ovom slučaju je bilo moguće obaviti manuelno usaglašavanje slika. Nakon usaglašavanja, rezultat oduzimanja prikazan je na slici 19 (gore). Slično rešenje dobija se i primenom algoritma automatske obrade. Pritiskom na dugme "Obavite automatsku procenu" ostvaruje se rezultat prikazan na slici 19 (dole). Posmatranjem očitanih vrednosti može se zaključiti da obe tehnike ispunjavaju uslov da temperaturne razlike u oblasti oboljenja budu pozitivne sa malom međusobnom razlikom.



Slika 18. Neusaglašene strane lica.



Slika 19. Rezultati manuelne (gore) i automatske procene (dole).

V. Zaključak

Primenom *Matlab* programa moguće je realizovati aplikaciju kojom se vrši transformacija termograma neophodna za analizu razlika temperatura zdrave i obolele strane lica.

Ograničenja aplikacije prevazilaze se primenom *mlx* fajlova, kao i primenom lokalnih promenljivih. Nedostatak zajedničkih tačaka se negativno odražava na algoritam, tako da u toku procene postoji mogućnost dodeljivanja sistemske vrednosti, obično nule, elementima za koje nije moguće definisati pripadnost nekoj oblasti druge slike, što utiče na rezultat oduzimanja slika. Sa druge strane, primenom manuelne tehnike problem nultih vrednosti ne postoji.

S obzirom da automatska procedura primenjuje algoritam sa unapred definisanim brojem iteracija, postoji mogućnost da u pojedinim situacijama broj iteracija nije dovoljan, što dovodi do toga da procena bude gruba, a rezultati nakon oduzimanja netačni. Sa druge strane, manuelna tehnika upoređivanja zavisi isključivo od vrednosti pomeraja koje unosi korisnik.

Zahvalnica

Izrada ovog projekta ne bi bila moguća bez pomoći profesora Igora Jovanovića, koji je uloženim vremenom, sugestijama i pružanim informacijama, podržao pisanje ovog rada.

Literatura

[1] M. Vollmer, K. P. Möllmann, *Infrared Thermal Imaging: Fundamentals, Research and Applications*, Wiley – VCH, 2018.
[2] C. L. Herry, M. Frize, "Quantitative assessment of pain-related thermal dysfunction through clinical digital infrared thermal imaging", *Biomed Eng Online*, vol. 3, no. 1, pp. 1-19, 2004.
[3] M. Corosena, Infrared Thermography Recent Advances and Future Trends, Bentham Science Publishers, 2012.
[4] Ž. Barbarić, *Termovizija – formiranje i primena termovizijske slike*, Akademska Misao, 2014.
[5] I. Dimitrijević, *Primena infracrvene termovizijske kamere u dijagnostici i proceni terapijskog efekta fizikalnih agenasa kod lumbosakralne radikulopatije*, Doktorska disertacija, Univerzitet u Nišu, Medicinski fakultet, 2016.
[6] MathWorks, https://www.mathworks.com/
[7] A. Ćosić, I. Jovanović, M. A. Stošović, D. Krasić, D. Mančić, "Temperaturne Karakteristike Tumora Parotidne Regije Zabeležene Termalnom Kamerom", *20. Kongres stomatologa Srbije*, Beograd, Srbija, Septembar 2021., pp. 71-72.
[8] A. Ćosić, I. Jovanović, I. Kostić, M. A. Stošović, D. Krasić, D. Mančić, "Temperature Characteristics of the Face Region of Healthy People Obtained using Thermal Imaging Camera", in *Proc. 15th International Online Conference on Applied Electromagnetics - IIEC 2021*, August 30 – September 01, 2021, Niš, Serbia, pp. 125-128.
[9] VARIOSCAN high resolution Manual https://www.infratec.eu/
[10] IRBIS professional https://www.infratec.eu/ /thermography/thermographic-software/irbis3-active

Bez prepreka - jer život ne može da čeka. Projekat naplatne rampe za vozila hitne službe

Lazar Živković, David Petrović, Stefan Stratijev

Sadržaj – Ovaj rad se bavi problematikom bržeg i bezbednijeg prolaska vozila hitnih službi kroz naplatnu rampu. Korišćenjem ovakvog sistema naplate putarine izbegava se bespotrebno čekanje vozila hitnih službi i njihovo prvostepeno reagovanje.

I. UVOD

U vremenu kada je tempo života postao brži, sve više se nameće potreba za brzim načinom reagovanja u kriznim situacijama. Najveći rizik od opasnosti je, nesumnjivo na putevima. Po statističkom izveštaju Agencije za bezbednost saobraćaja, broj poginulih lica u prethodnoj godini je bio 492, a broj povređenih lica 17250 [1]. Jedan od mnogobrojnih problema je i zakasnelo reagovanje vozila hitnih službi, kao i njihovo zadržavanje u saobraćaju. Problem je i ulazak u traku za elektronsku naplatu putarine. Prvenstveno za vozila hitne pomoći, kao i za vozila vatrogasne službe minuti čekanja na naplatnoj rampi mogu biti presudni u reagovanju. Upravo iz tog razloga, tim naše škole je prepoznao problem i rešili smo da napravimo projekat naplatne rampe za vozila hitnih službi.

Naš projekat podrazumeva postojanje novog tipa naplate putarine. Ova traka za naplatu putarine je odvojena posebnom rampom i može da se koristi u hitnim slučajevima, ili kao traka za regularnu naplatu putarine. Pri detektovanju bilo kakvog vida ugrožavanja saobraćaja, radnik za naplatu putarine dobija obaveštenje o potrebi otvaranja trake samo za vozila hitnih službi. Otvaranje i zatvaranje rampe se reguliše preko potenciometra u prostoriji za radnike. To je ujedno i vremenski regulator otvorenosti tj. zatvorenosti rampe. Naime, svako vozilo hitne službe posedovaće integrisanu karticu koja će se očitavati putem senzora.

II. PRINCIP RADA I PRAKTIČNA REALIZACIJA

Sistem se sastoji od tri rampe koje funkcionišu na različite načine, kao što je to prikazano na slici 1. Prva rampa, postojeći vid naplate – simulira fizicku naplatu putarine putem kartice, u našem slučaju pritiskom na dugme. Druga rampa, postojeći vid naplate – simulira ENP.

L. Živković, D. Petrović i S. Stratijev su učenici četvrte godine Elektrotehničke škole “Mija Stanimirović“, Bulevar Svetog cara Konstantina 82-84, 18000 Niš, Srbija, E-mail: zivkovicl2004@gmail.com, davidpetrovic2810@gmail.com, stefanstratijev.etsms@gmail.com.

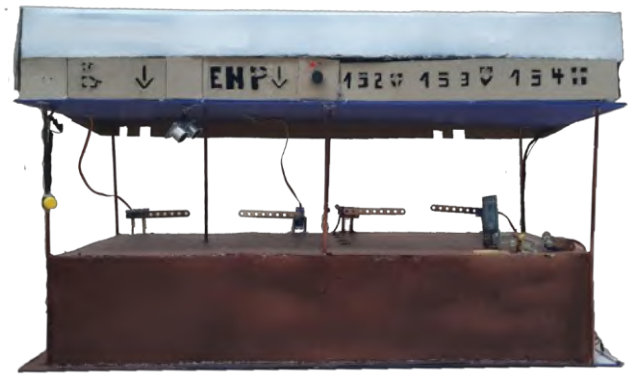
Proces naplate se odvija pomoću ENP uređaja smeštenog na vozilu, u našem slučaju ultrazvučni senzor. Treća rampa, idejno rešenje – prototip naplatne rampe za vozila hitne službe, radi po principu skeniranja uređaja, koji je integrisan na vozilo hitne službe. Prolazak skeniranog vozila se detektuje putem senzora koji se nalazi na određenoj udaljenosti od naplatne rampe. Po proračunima, bezbedna brzina za prolazak kroz naplatnu rampu je 30 km/h [2]. Uzevši u obzir vreme podizanja rampe i prosečnu bezbednu brzinu prolaska kroz rampu, možemo izračunati da senzor treba biti postavljen na minimalnoj udaljenosti od 800 m. Vrednost iz gore navedenih podataka smo dobili na osnovu formule:

$$s = v_{sr} \cdot t_{pr} \tag{1}$$

gde je v_{sr} - srednja bezbedna brzina vozila za prolazak kroz rampu, t_{pr} - vreme podizanja rampe. Nakon prijavljivanja opasnosti, radnik naplatne rampe okreće potenciometar u određeni položaj u kom će se kolovozna traka odvojiti pokretnim stubovima koji će izlaziti iz puta po vertikali. Senzor za detektovanje vozila pozicioniraće se na određenu udaljenost, potrebnu da vozilo bude skenirano (približiće se kolovoznoj traci kako bi vozilo moglo da bude skenirano). Na vrhu naplatne rampe upaliće se rotacija i sirena kao upozorenje učesnicima u saobraćaju u vidu zvučnog signala. Ceo projekat je realizovan u programskom okruženju Arduino IO. Kao prototip takve ideje napravili smo rešenje sa 3 različa tipa, kao što je dato na slici 2.



Slika 1. Perspektivni prikaz modela rampe.



Slika 2. Maketa sistema.

III. KORIŠĆENE KOMPONENTE

Arduino Uno(2) - mikrokontroler na štampanoj ploči. Radni napon je 5 V, potrebno je napajanje između 7 V i 12 V, ima 14 digitalnih I/O pinova, 6 analognih ulaza, programska memorija je 32 KB.



Slika 3. Arduino Uno.

RFID Senzor – MF522-MINI modul koristi originalni RC522 čip za dizajniranje kola čitača kartica. Modul radi na frekvenciji od 13.56 MHz.



Slika 4. RFID Senzor.

Ultrazvučni senzor - ovo je modul za merenje udaljenosti ultrazvučnog senzora koji je beskontaktni modul za merenje udaljenosti. Ima visoku tačnost merenja i njegova slepa oblast je blizu (do 2 cm).



Slika 5. Ultrazvučni Senzor.

LED – svetleća dioda, vrednost struje kroz diodu ne bi trebalo da prelazi 20 mA, dok je idealna vrednost struje između 16 i 18 mA. Jačinu struje kontrolišemo pomoću otpornika, koji vezujemo redno. Za naš sistem koristimo LED crvene, žute i zelene boje.



Slika 6. LED.

Servo motor- je komponenta koja može rotirati svoju ruku(obično izmedju 0 i 180 stepeni). Koristi se da kontroliše ugaonu poziciju objekta.



Slika 7. Servo.

Buzzer - stvara zujanje ili zvučni signal. Složeniji zvučnici uključuju sklop oscilatora i piezoelektrični element ili zvučnik u jednom paketu, tako da sve što trebate učiniti je da promenite napon i dobićete zvuk zujanja.

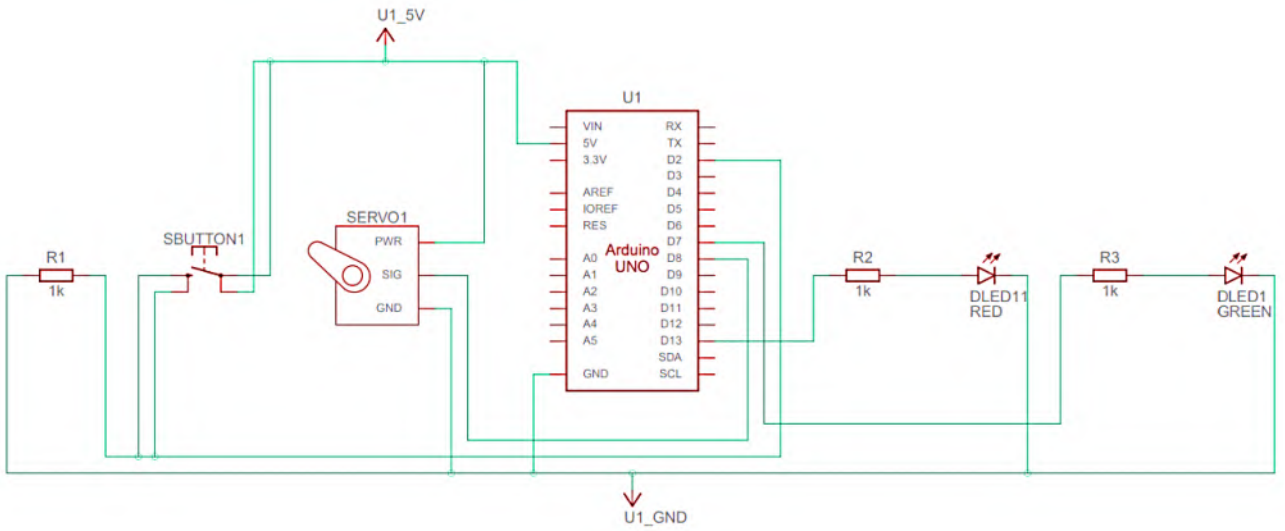


Slika 8. Buzzer.

DC Motor (*Direct Current Motor*) - je najrasprostranjeniji tip motora. DC Motor obično ima samo dva izlaza, jedan pozitivan i jedan negativan. Ako povežete ova dva izlaza direktno na bateriju, motor će se pokrenuti. Ako okrenete izlaze motor će se okrenuti na suprotnu stranu.



Slika 9. DC motor.



Slika 10. Električna šema sistema.

Programski kod:

```
#include <Servo.h>
Servo servo;
int led1=7;
int led11=13;
int button1=2;
int buttonState = 0;
void setup() {
Serial.begin(9600);
pinMode(led1, OUTPUT);
pinMode(led11, OUTPUT);
pinMode(button1, INPUT);
servo.attach(8); }
void loop() {
float motor = analogRead(A4);
buttonState = digitalRead(button1);
digitalWrite(led11, HIGH);
if (buttonState == HIGH) {
digitalWrite(led11, LOW);
digitalWrite(led1, HIGH);
servo.write(100);
delay(10000); }
else {
digitalWrite(led1, LOW);
servo.write(2); }
```

IV. REZULTAT

Vrednosti u tabelama I i II su rezultat proračuna kada je rampa postavljena na udaljenosti od 1 km. Pri tome je uzeto u obzir smanjenje brzine vozila na udaljenosti od 200 m od naplatne rampe. Smanjenje brzine se odvija postepeno: najpre na 150 m od rampe, a onda dodatno smanjenje brzine na 50 m od rampe. U slučaju ENP rampe uzeta je prosečna brzina svih vozila hitnih službi od 100 km/h na auto putu. Smanjenje brzine vozila na razdaljini od 150 m od naplatne rampe, kao i smanjenje brzine na udaljenosti 50 m od naplatne rampe su uzeti na osnovu praktičnih merenja brzina, koristeći vozila sličnih karakteristika. Računata je i prosečna brzina prolaska kroz naplatnu rampu u rasponu od 20 m, kao i prosečne brzine nakon prolaska rampe. Sledeće merenje brzine je nakon pređenih 200 m od naplatne rampe, a drugo je nakon 800 m od prolaska kroz naplatnu rampu. Vrednosti tih brzina su date u tabeli I. Za tabelarni prikaz brzina rampe za hitne slučajeve, uzete su obzir iste razdaljine puta pre, za vreme i nakon prolaska naplatne rampe. Vrednosti tih brzina date su u tabeli II.

Vrednosti u tabeli I su rezultat praktičnog merenja brzina u vozilima hitnih službi. Naime, autori ovog projekta su testirali brzine svakog od vozila hitnih službi na terenu. Uz pomoć prijatelja projekta, autori su bili u mogućnosti da merenja izvedu praktično i u realnim uslovima. Vozila koja su korišćena za testiranje su bila sličnih karakteristika kao i vozila hitnih službi. Vozilo policije zamenjeno je putničkim automobilom *Škoda Rapid* sa agregatom 1.6 tdi. Vozilo hitne pomoći zamenjeno je kombijem marke *Citroen Jumper* sa agregatom 3.0 hdi. Vozilo vatrogasne službe zamenio je kamion *Man TGM*. Sva merenja vršena su istog dana u istim uslovima.

Na osnovu datih brzina, računato je vreme prolaska vozila hitnih službi na svakom delu puta koji je obeležen u tabelama. Izračunavanja su vršena na osnovu obrasca za ravnomerno pravolinijsko kretanje na svakom segmentu puta.

$$t_{pr} = s / v_{sr} \tag{2}$$

Sumiranjem izračunatih vremena dobijeno je ukupno vreme prolaska vozila hitnih službi na deonici puta koja obuhvata kolimetar ispred i kilometar nakon naplatne rampe. Rezultati računanja prikazani su u tabeli III.

Poređenjem rezultata merenja ENP rampe i rampe za hitne slučajeve, kao što je prikazano u tabeli III, vidimo da je kod rampe za hitne slučajeve vreme prolaska od senzora do auto-puta smanjeno, što mnogo znači prilikom reagovanja u slučaju nesreće.

TABELA I
VREDNOSTI BRZINA ENP RAMPA

ENP rampa						
Tipovi hitnih službi	*Brzina vozila na udaljenosti 800 m od naplatne rampe(km/h)*	*Brzina vozila na udaljenosti 150 m od naplatne rampe(km/h)*	*Brzina vozila na udaljenosti 50 m od naplatne rampe(km/h)*	*Brzina vozila tokom prolaska kroz naplatnu rampu u rasponu od 20m(km/h)*	*Brzina vozila neposredno posle prolaska kroz naplatnu rampu u rasponu od 200m (km/h)*	*Brzina vozila pri izlaska na auto-put u razmaku od 800m(km/h)*
Hitna pomoć	100	70	50	30	55	95
Vatrogasci	100	65	45	30	50	90
Policija	100	75	55	30	65	100

TABELA II
VREDNOSTI BRZINA RAMPA ZA HITNE SUČAJEVE

Rampa za hitne slučajeve						
Tipovi hitnih službi	*Brzina vozila na udaljenosti 800 m od naplatne rampe(km/h)*	*Brzina vozila na udaljenosti 150 m od naplatne rampe(km/h)*	*Brzina vozila na udaljenosti 50 m od naplatne rampe(km/h)*	*Brzina vozila tokom prolaska kroz naplatnu rampu u rasponu od 20m(km/h)*	*Brzina vozila neposredno posle prolaska kroz naplatnu rampu u rasponu od 200m (km/h)*	*Brzina vozila pri izlaska na auto-put u razmaku od 800m(km/h)*
Hitna pomoć	100	80	75	70	85	95
Vatrogasci	100	75	70	65	75	90
Policija	100	90	85	80	90	100

V. ZAKLJUČAK

Ovim projektom želimo da unapredimo standardno funkcionisanje naplatnih rampi. Prednost ovog projekta je praktičnije i sveobuhvatnije rešenje. U zavisnosti od sledećih faktora: kolika je širina kolovozne trake, kolika je učestalost saobraćaja, kakav je geografski položaj mesta za naplatnu rampu, i sl.ovaj projekat je podložan promenama. U cilju unapeđivanja sistema postoji dosta mogućnosti. Jedna od njih je i automatizacija celokupnog sistema, tako što bi se eliminisao ljudski faktor u procesu regulisanja otvorenosti i zatvorenosti rampe. Tako bi se stvorio poseban softverski komunikacioni kanal između dežurnih službi i same naplatne rampe, gde bi se putem posebne signalizacije automatski vršilo otvaranje i zatvaranje rampe. Cilj nam je da bar u jednom delu bezbednosti na putu, damo i svoj doprinos.

TABELA III
VREDNOSTI BRZINA

ENP rampa			
Tipovi hitnih službi	*Brzina na udaljenosti od 1km od rampe(km/h)*	*Brzina prolaska kroz rampu(km/h)*	*Vreme potrebno da se pređe put od senzora do autoputa(s)*
Hitna pomoć	**100**	**30**	**85.93**
Vatrogasci	**100**	**30**	**89.63**
Policija	**100**	**30**	**81.6**
Rampa za hitne slučajeve			
Hitna pomoć	**100**	**70**	**77.75**
Vatrogasci	**100**	**60**	**81.3**
Policija	100	85	74.61

ZAHVALNICA

Autori rada zahvaljuju se Elektrotehničkoj školi „Mija Stanimirović“, koja nam je obezbedilila sredstva i uslove za rad, kao i mentoru Ivani Bogdanović, nastavniku u ETŠ „Mija Stanimirović“ na pruženoj podršci. Takođe se zahvaljujemo dipl. inž. arh. Goranu Radosavljeviću na izradi perspektivnog prikaza modela rampe, i drugovima iz odeljenja za tehničku podršku.

LITERATURA

[1] Preuzeto iz dokumenta Statistički izveštaj o stanju bezbednosti saobraćaja u Republici Srbiji u 2020. godini, postavljen 21.12.2021, na sajtu Agencije za bezbednost saobraćaja.
[2] Preuzeto sa sajta https://www.politika.rs/scc/clanak/20835/Tag-dize-rampe)))))
[3] https://www.arduino.cc/en/main/software link za preuzimanje ArduinoIDE softvera.
[4] Slike preuzete sa https://www.arduino.cc.

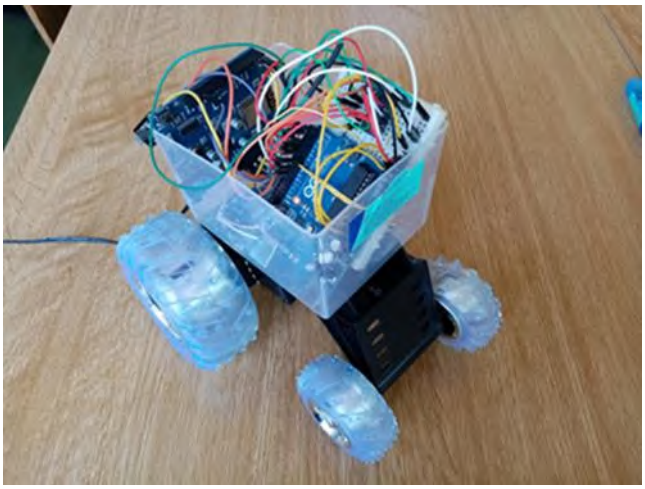
Pametan automobil i pametna garaža realizovani pomoću Arduino razvojnog sistema

Marko Manić, Dimitrije Nikolić, Milan Pavlović, Aleksa Ristić, Veljko Dunčić, Nikola Jevtović

Sadržaj – U ovom radu prikazan je pametan automobil i pametna garaža realizovani pomoću Arduino razvojnog sistema.

I. UVOD

Sve je veći broj slučajeva vožnje u alkoholisanom stanju. Takođe, zbog izduvnih gasova u zatvorenim prostorima i u automobilima, dolazi do smrtnih slučajeva. Cilj našeg projekta je bio realizacija pametne garaže i pametnog automobila, koji će uz pomoć standardnih bežičnih protokola i savremenih senzora međusobno interagovati kako bi se obezbedila sigurna vožnja i komfor prilikom upotrebe automobila. Bitno je napomenuti da su garaža i automobil dva zasebna sistema, pa je samim tim od velike koristi da dati sistemi budu automatizovani i da imaju mogućnost međusobne interakcije.



Slika 1. Automobil.

Strategija realizacije:

- Dekodiranje signala sa tastera koji su na daljinskom upravljaču i koji se sa daljinskog upravljača šalju IC prijemniku [1]. IC prijemnik se nalazi na automobilu.

M. Manić, D. Nikolić, M. Pavlović, A. Ristić, V. Dunčić, N. Jevtović, su studenti 4. godine modula Elektronika na Elektronskom fakultetu u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: marko.manic@elfak.rs.

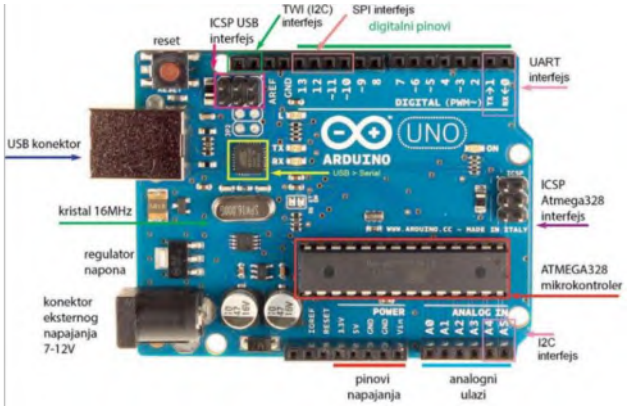
- IC prijemnik koji se nalazi na automobilu prima signal, signal se obrađuje i softverski procesira i kao rezultat automobil izvršava neku od odgovarajućih komandi kretanja (napred, nazad, skretanje levo, skretanje desno, stop ili start).
- Garaža poseduje mogućnost otvaranja na dva načina. Prvi način je pomoću daljinskog upravljača koji komunicira sa IC prijemnikom koji je u sklopu garaže: tako je vozaču olakšan pristup uparkiravanja i isparkiravanja automobila zato što ne mora da izlazi iz automobila. Drugi način otvaranja garaže je pomoću tastera koji se nalazi unutar objekta i i fizički je nedostupan za korisnika koji se nalazi izvan objekta, na taj način je objekat dodatno osiguran.
- Opremiti garažu tako da se vrata garaže, sijalica u garaži i zvučnik u garaži otvaraju (uključuju) i zatvaraju (isključuju) po želji i potrebi korisnika.
- Instalirati senzore za dim, alkohol i butan u garaži, koji će predstavljati podsistem koji je zadužen za osiguravanje same garaže i lica koje u tom trenutku boravi u garaži. Ukoliko dođe do prekoračenja limita količine dima, alkohola ili butana u vazduhu, aktivira se zujalica[3] kao način audio upozorenja da je bezbednost u datom trenutku narušena.
- Slanje obaveštenja na *e-mail* ako je senzor za dim detektovao dim (požar).
- Blokiranje garažnih vrata ako senzor za alkohol detektuje prisustvo alkohola u vazduhu, kako vozač ne bi vozio u alkoholisanom stanju.

II. PRINCIP RADA

Ovaj sistem je zasnovan na tri *Arduino* razvojne ploče (Dve *Arduino UNO* i jedna *Arduino WeMos D1R2* [2]).

Prva *Arduino* razvojna ploča (*Arduino WeMos D1R2*) je stacionirana u garaži. Takođe je zadužena za slanje obaveštenja putem *e-mail* poruke korisniku. Neke od sledećih funkcija se obavljaju uz pomoć sledećih perifernih komponenti. Kada kliknemo na taster na daljinskom upravljaču, pokreće se proces otvaranja garaže. Proces otvaranja garaže se realizuje mikro servo motorom *sg90*. Prilikom otvaranja garaže uključuje se i LED koja simulira osvetljenje u garaži. Takođe, dok se garaža otvara, aktivira se prva melodija zujalice [3]. Proces zatvaranja garaže se takođe odvija pritiskom tastera na daljinskom. Kada započne

proces zatvaranja, melodija se ponovo aktivira i LED se isključuje. Kao alternativa za daljinski upravljač, koristi se taster koji se nalazi unutar objekta. U garaži se nalaze i gasni senzori *mq2* i *mq3*. Senzor *mq2* se koristi za detekciju dima u garaži, a *mq3* senzor se koristi za detekciju alkohola i butana u garaži. Kada jedan od ova dva senzora detektuje nivo gasa koji prelazi graničnu količinu u vazduhu, aktivira se druga melodija na zujalici. Ta druga melodija je uvod u pesmu „*Electric funeral*“ grupe *Black Sabbath.*



Slika 2. Razvojni sistem Arduino UNO.



Slika 3. Razvojni sistem Arduino WeMos D1R2.

Druge dve *Arduino UNO* razvojne ploče se nalaze u automobilu. Prvi *Arduino UNO* je povezan sa drajverom L293D [4] koji kontroliše dva *DC* motora. Ovaj drajver ima dva H-mosta u sebi koji se koriste za pogon *DC* motora u oba smera. Ovi motori su povezani na točkove koji rade u istom smeru kao i sami *DC* motori. Važno je napomenuti da automobil ima pogon na zadnje točkove, što znači da se funkcije kretanja izvode na sledeći način:

1. NAPRED - oba *DC* motora rotiraju u pravcu napred;
2. NAZAD - oba *DC* motora rotiraju unazad;
3. LEVO - desni *DC* motor rotira napred;
4. DESNO - levi motor rotira napred;
5. STOP - oba *DC* motora su zaustavljena (točkovi se ne pomeraju);
6. OTKLJUČATI - motori su spremni da prime komande sa daljinskog upravljača.

Na drugom *Arduino UNO* sistemu nalazi se IC prijemnik. Ovaj prijemnik prima signale sa daljinskog upravljača, a potom signali odlaze do prvog *Arduino UNO* koji je povezan sa drajverom L293D. Ove dve ploče u automobilu komuniciraju jedna sa drugom koristeći linije za prenos i prijem, što predstavlja *UART* serijsku komunikaciju.

Taster 2: Napred (6375/0X18E7)
Taster 8: Nazad (19125/0X4AB5)
Taster 4: Levo (4335/0X10EF)
Taster 6: Desno (23205/0X5AA5)
Taster 5: Stop (14535/0X38C7)
Taster Start/Pauza: Start (49725/0XC23D)



Slika 4. Spoljašnost garaže.

Tok realizacije projekta:

Preko IC prijemnika dekodira se komanda sa daljinskog upravljača. Automobil takođe ima drajver L293D2 koji u sebi ima 2 H-mosta koji pobuđuje naše *DC* elektromotore kako bi se kretali u oba smera. Motori se nalaze na zadnjim točkovima, tako da naš automobil ima zadnju vuču. U odnosu na dekodiran signal, automobil izvršava neku od sledećih naredbi: kretanje napred, kretanje unazad, skretanje levo, skretanje desno, kočenje automobila, zaključati auto, otključati auto. Kada je auto zaključan ne mogu se izvršavati komande kretanja. Takođe, IC prijemnik dekodira i naredbe koje su namenjene garaži, šalje ploči koja se nalazi u garaži i garaža može izvršiti sledeće naredbe: otvaranje garaže i zatvaranje garaže. Garažna vrata se pokreću preko servo motora sg90, unutra se uključuje LED koja predstavlja sijalicu i zujalica koja nam predstavlja zvučnik i emituje prvu melodiju, a kada se vrata garaže zatvore, gasi se svetlo i emituje se druga melodija kraćeg trajanja. U sklopu garaže se nalaze senzori za alkohol i dim.

Senzor za dim ima ulogu monitoringa unutrašnjosti garaže i pri detektovanju dima iznad određenog nivoa, alarmira vlasnika *e-mail* porukom kako bi se hipotetički sprečio požar. Senzor za određivanje vrednosti alkohola ima ulogu monitoringa alkohola i pri detektovanju alkohola zaključava garažu i tako sprečava vozača da vozi u alkoholisanom stanju i time spašava njegov, ali i tuđe živote.



Slika 5. Unutrašnjost garaže.

III. REZULTATI

Omogućeno je uspešno kontrolisanje automobila (otključavanje/zaključavanje, kretanje i zaustavljanje) i garažnih vrata uz pomoć daljinskog upravljača.

Prilikom otvaranja ili zatvaranja garaže aktivira/deaktivira se LED (koji simulira osvetljenje u garaži) i aktiviraju se prva i druga melodija preko zujalice prilikom otvaranja/zatvaranja garaže.

Obezbeđen je sinhronizovan rad i pouzdana komunikacija između garaže i automobila.

Takođe je omogućen pouzdan rad senzora za detekciju kritičnih stanja unutar garaže (senzori za detekciju dima i alkohola u vazduhu).

IV. ZAKLJUČAK

Imajući u vidu veliku prisutnost tehnologije u svakodnevnom životu čoveka, realizacijom ovog projekta pokazali smo praktičnu primenu savremenih tehnologija u cilju očuvanja bezbednosti ljudskih života. Implementacija ovakvih i sličnih sistema bi takođe bila od pomoći ljudima sa invaliditetom. Pored toga, ovakav sistem mogao bi se personalizovati i služiti korisnicima različitih profila.

ZAHVALNICA

Ovaj rad je napisan kao deo grupnog projekta na osnovnim akademskim studijama na Elektronskom fakultetu u Nišu u sklopu predmeta „Bežične računarske mreže i uređaji“. Kompletna realizacija projekta izvedena je pod mentorstvom prof. dr Tatjane Nikolić. Posebnu zahvalnost dugujemo i Katedri za elektroniku, kao i prof. dr Branislavu Petroviću za finansiranje projekta.

LITERATURA

[1] https://github.com/Arduino-IRremote/Arduino-IRremote?fbclid=IwAR1u9gwTGcqAfGONYi4vZnRqg2NfUoo2MfWcgnMp0_um4REBxBssbOo5vwA
[2] https://www.instructables.com/Programming-the-WeMos-Using-Arduino-SoftwareIDE/
[3] https://gist.github.com/mikeputnam/2820675
[4] https://lastminuteengineers.com/l293d-DC-motor-arduino-tutorial/?fbclid=IwAR2gi9A1ZzRiv3NPUm_cItAuyPR1M0Cakc4UQ_nNML7seilNmvmE1CqYP-0

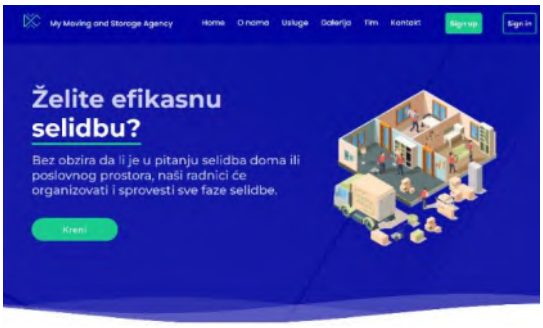
My MSA – Web aplikacija za pomoć pri selidbama

Kristina Petrović, Miljana Petković, Marija Vidović

Sadržaj – Selidba predstavlja prekretnicu u životu svakog čoveka koja podrazumeva promenu životne rutine, upoznavanje sa novim krajem, komšijama. Kako bi celokupna adaptacija, koja sledi neposredno nakon selidbe, lakše pala bitno je obezbediti da sam proces preseljenja bude dobro isplaniran, organizovan i realizovan. Oni koji se sele treba da budu upućeni u sve pojedinosti selidbe i najbitnije, njenoj ceni, kako bi se izbegli naizgled nepredviđeni troškovi i neprijatna iznenađenja na dan same selidbe. Ispitivanjem srpskog tržišta ustanovilo se nepostojanje adekvatne web aplikacije koja pruža sveobuhvatne usluge koje bi korisniku bile od značaja u toku preseljenja, kako domaćinstava, tako i poslovnih prostora. U ovom radu biće prikazana upravo takva jedna aplikacija realizovana na predmetu „Softversko inženjerstvo".

I. Uvod

Web aplikacija My MSA (*My Moving and Storage Agency*) (slika 1.) je namenjena agenciji, koja se bavi pružanjem usluga selidbe, njenim zaposlenima, kao i korisnicima kojima bi selidba bila potrebna. My MSA obezbeđuje pregledan prikaz dostupnih usluga, zakazivanje i poništavanje istih, uređivanje interne organizacije agencije, delegiranje poslova radnicima kao i praćenje uspešnosti poslovanja od strane samog vlasnika.



Slika 1. Početna strana aplikacije.

My MSA web aplikacija je proširiva, intuitivna i fleksibilna za korišćenje i u drugim agencijama, sa akcentom na izdvajanju grafičkog dizajna od funkcionalnosti da bi svaka instanca aplikacije mogla da bude jedinstvena. Korisnici pristupaju sistemu preko Web-a što zahteva minimum modemsku internet konekciju. Iz ovog razloga ne postoje posebna ograničenja u pogledu okruženja.

K. Petrović, M. Petković i M. Vidović, su studentkinje smera Računarstvo i informatika, Elektronski fakultet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: kristina.petrovic@elfak.rs.

II. Opis projekta

A. Profili prepoznatih korisnika

Administrator aplikacije − (skraćeno administrator ili admin): U ulozi administratora će se najčešće naći naći fakultetski obrazovana osoba sa visokim nivoom poznavanja rada na računaru i administriranja Web servera. Administrator će imati pristup svim funkcijama sistema. On je između ostalog zadužen i za postavljanje privilegija korisnicima.

Vlasnik agencije − Vlasniku agencije je omogućen pregled statistike o radu timova radnika i profitu agencije. Takođe, vlasnik ima mogućnost dodavanja administracije agencije.

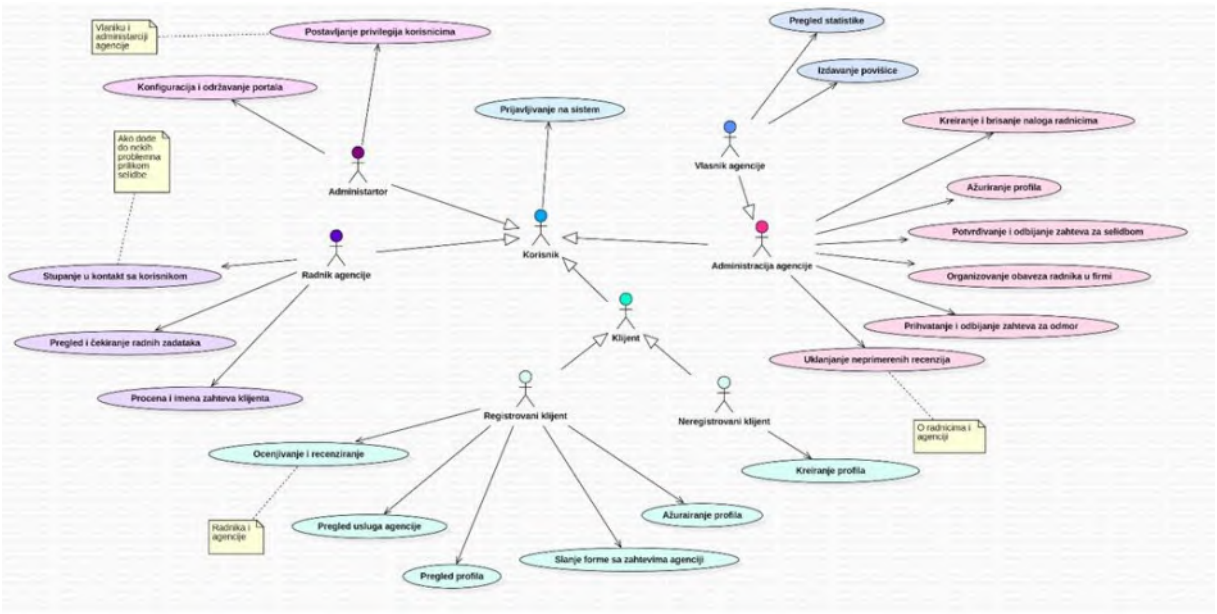
Administracija agencije − Administrator agencije je zadužen za kreiranje naloga radnika. On vrši potvrđivanje ili odbijanje zahteva za selidbom i organizuje obaveze radnika u firmi (selektuje radnike i dodeljuje dnevne zadatke). Administrator ima i mogućnost pregleda recenzija o samoj agenciji, izvršenim selidbama, ali i uklanjanja neprimerenih recenzija.

Radnik agencije − Radnik agencije ima mogućnost pregleda radnih zadataka za tekući dan i čekiranje izvršenih zadataka. Sa druge strane, radnik koji se bavi procenama selidbi takođe izvršava dnevne radne zadatke koji se odnose na popis stvari koje se sele, izdavanjem procene klijentu i u slučaju postizanja dogovora sa drugom stranom, specificira ostale pojedinosti vezane za selidbu, a koje se odnose na datum i krajnju tačku selidbe.

Korisnik agencijskih usluga (skraćeno korisnik) − Korisnik ima mogućnost kreiranja naloga i pregleda usluga koje agencija pruža. Moguće je slanje forme sa svojim zahtevima agenciji u cilju uspostavljanja saradnje. Najpre klijent treba da zakaže pregled selidbe, a nakon što se ona izvrši na uvid dobije specifikaciju iste. Ukoliko postigne dogovor sa agencijom specificira datum, vreme, krajnju tačku selidbe, dok dobija i određeni tim koji će ga tada uslužiti. Korisnik ima mogućnost ocenjivanja i recenziranja završene selidbe, ali i agencije, pri čemu bi one bile vidljive i ostalim korisnicima.

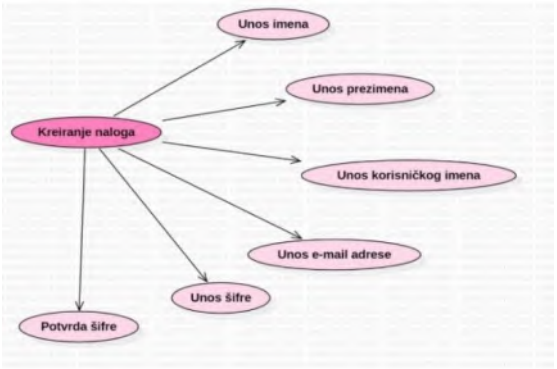
B. Opis slučajeva korišćenja

Osnovni UML dijagram koji prikazuje korisnike i slučajeve korišćenja My MSA Web aplikacije prikazan je na sledećoj slici (slika 2):

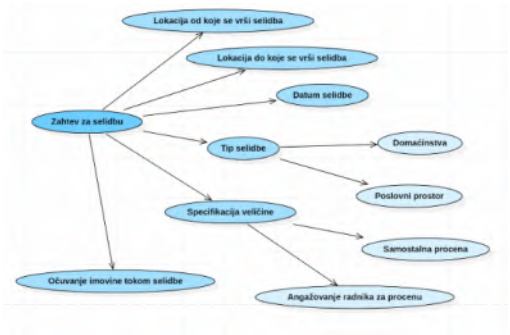


Slika 2. UML dijagram slučajeva korišćenja.

Slučajevi korišćenja *kreiranje naloga*, zahtev za selidbu i procena vrednosti selidbe su složenije radnje koje se mogu razložiti dalje na pojedinačne slučajeve korišćenja. Detaljni UML dijagram za slučaj korišćenja *kreiranje naloga* je prikazan na sledećoj slici (slika 3) :

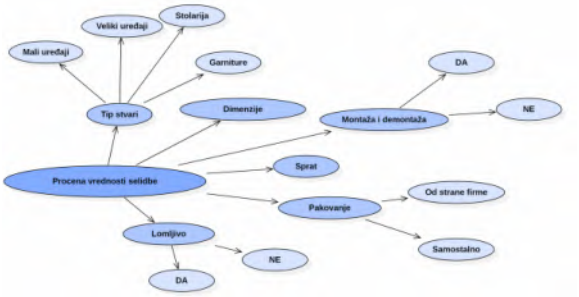


Slika 3. UML dijagram kreiranja naloga.



Slika 4. UML dijagram kreiranja zahteva za selidbom.

Detaljni UML dijagram za slučaj korišćenja *zahtev za selidbu* je prikazan na sledećoj slici (slika 4). Detaljni UML dijagram za slučaj korišćenja *procena vrednosti selidbe* je prikazan na slici 5.

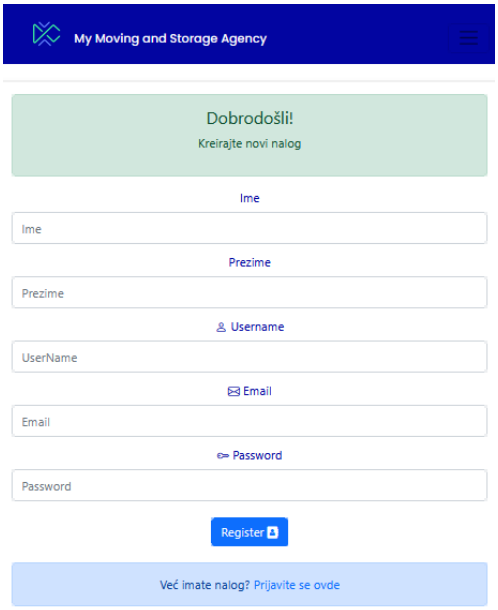


Slika 5. UML dijagram procene vrednosti selidbe.

C. Mogućnost kreiranja naloga

Kratak opis: Prikaz stranice sa formama za unos podataka o budućem korisniku (slika 6).
Akteri: Neregistrovani klijent.
Preduslovi: Klijent nije registrovan na sistem do sada.
Osnovni tok:
1. Prikazuje se stranica sa formama za unos podataka.
2. Korisnik unosi podatke o sebi, kao što su ime, prezime, korisničko ime, e-mail adresa i šifra [izuzetak: Korisnik je već registrovan sa tom e-mail adresom].
3. Korisnik dobija upozorenje u slučaju da korisničko ime nije jedinstveno.
4. Klikom na dugme *Register* se kreira nalog.
Izuzeci: [Korisnik je već registrovan sa tom e-mail adresom] U tom slučaju korisnik je registrovan na sistem i

neće biti u mogućnosti da kreira novi nalog sa istom e-mail adresom. Dobiće upozorenje na stranici.
Posledice: Dodat je novi korisnik na sistem. Registrovani korisnik može da koristi usluge sistema.



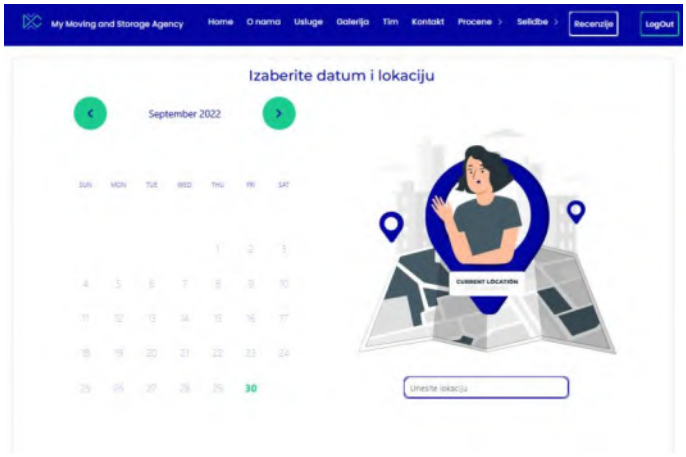
Slika 6. Stranica za registraciju korisnika.

D. Prijavljivanje na sistem

Kratak opis: Prijavljivanje korisnika na sistem u cilju pristupa specifičnim funkcijama koje zahtevaju autorizaciju. Akteri: Klijent, Agencija, Administrator.
Preduslovi: Registrovani korisnik nije trenutno prijavljen.
Osnovni tok:
1. Korisnik bira opciju prijavljivanje sa bilo koje stranice Web aplikacije.
2. Prikazuje se forma za prijavljivanje.
3. Korisnik unosi korisničko ime (ili e-mail) i lozinku, te nakon toga aktivira komandu Prijavi se.
4. Uneti podaci se proveravaju [izuzetak: pogrešno korisničko ime i/ili lozinka].
5. Generiše se jedinstveni identifikator sesije (SID).
6. Osvežava se polazna stranica sa SID-om kao parametrom.
Izuzeci: [Pogrešno korisničko ime i/ili lozinka] Prikazuje se intuitivno upozorenje na formi za prijavljivanje.
Posledice: Klijent je prijavljen na sistem i ima pristup specifičnim opcijama u skladu sa svojim privilegijama.

E. Slanje zahteva za zakazivanje selidbe

Kratak opis: Prikaz stranice sa formom za prikazivanje selidbe (slika 7).
Akteri: Registrovani klijent.
Preduslovi: Registrovani korisnik je prijavljen.
Osnovni tok:
1. Korisnik bira komandu zakaži.
2. Prikazuje se forma za zakazivanje.
3. Korisnik unosi podatke u formi i bira termin, radnju završava pritiskom na potvrdi.
Posledice: Registrovani klijent je poslao zahtev za selidbu.



Slika 7. Stranica za zakazivanje procene selidbe.

F. Prihvatanje zahteva

Kratak opis: U skladu sa svojim opterećenjem administracija šalje potvrde ili odbija zahteve korisnika.
Akteri: Administracija agencije
Preduslovi: Postojanje zahteva za selidbom.
Osnovni tok:
1. Administracija se prijavljuje na sistem.
2. Administracija na osnovu zahtevanog datuma i lokacije šalje formu o prihvatanju zahteva.
3. Ukoliko je zahtev prihvaćen administracija bira vođu tima radnika i dodeljuje mu privilegije.
4. Šalje formu radniku o terminima dolaska.
Posledice: Klijentov zahtev za selidbu je prihvaćen.

G. Odbijanje zahteva

Kratak opis: U skladu sa svojim opterećenjem administracija šalje potvrdu ili odbija zahteve korisnika.
Akteri: Administracija agencije
Preduslovi: Postojanje zahteva za selidbom.
Osnovni tok:
1. Administracija se prijavljuje na sistem.
2. Administracija na osnovu zahtevanog datuma i lokacije šalje formu o odbijanju zahteva.
Posledice: Klijentov zahtev za selidbu je odbijen.

H. Pregled recenzija agencije od strane klijenta

Kratak opis: Klijent ima mogućnost pregleda recenzija agencije.
Akteri: Registrovani neregistrovani klijent.
Preduslovi: Nema ih.
Osnovni tok:
1. Otvaranje glavne stranice.

Posledice: Klijent je upućen u osnovno poslovanje agencije.

I. Pregled recenzija agencije od strane administracije

Kratak opis: Administracija ima mogućnost pregleda statistike poslovanja agencije.
Akteri: Administracija agencije.
Preduslovi: Nema ih.
Osnovni tok:
1. Otvaranje stranice kojoj može da pristupi samo admistracija agencije.
Posledice: Vlasnik je upućen u osnovno poslovanje agencije.

J. Pregled radnih zadataka

Kratak opis: Radnik na svom profilu ima uvid u svoje radne zadatke.
Akteri: Radnik agencije.
Preduslovi: Radnik je prijavljen na sistem.
Osnovni tok:
1. Radnik se prijavljuje na sistem.
2. Radnik ima uvid u kalendar aktivnosti.
Posledice: Radnik je upućen u svoje aktivnosti.

K. Čekiranje radnih zadataka

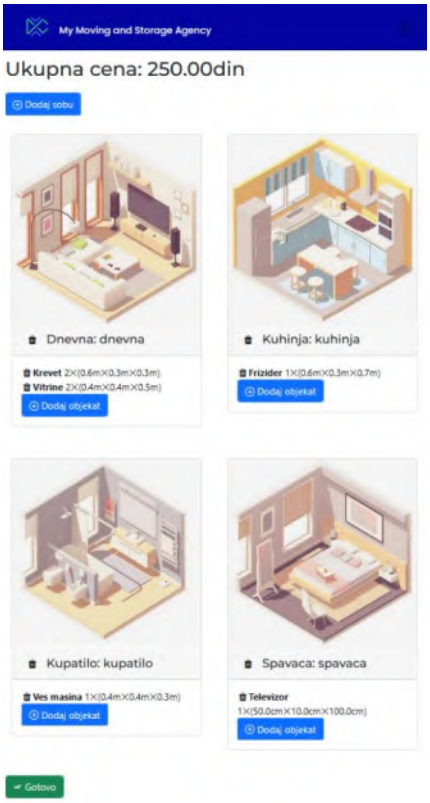
Kratak opis: Radnik na svom profilu ima uvid u svoje radne zadatke.
Akteri: Radnik agencije.
Preduslovi: Radnik je prijavljen na sistem i ima barem jedan radni zadatak.
Osnovni tok:
1. Radnik se prijavljuje na sistem.
2. Radnik ima uvid u kalendar aktivnosti. .
3. Klikom na zadatak on se označava kao završen.
Posledice: Označeni zadatak je završen.

L. Dodavanje radnih zadataka radnicima

Kratak opis: Radnicima se dodaju radni zadaci vezani za jednu selidbu.
Akteri: Administracija agencije.
Preduslovi: Radnik je prijavljen na sistem.
Osnovni tok:
1. Administracija agencije se prijavljuje na sistem.
2. Bira kriterijum na osnovu koga će selektovati radnike.
3. Radnicima se dodaju radni zadaci.
Posledice: Radnicima su dodati radni zadaci.

M. Procena cene selidbe

Kratak opis: Dodavanje liste stvari koje se sele radnik vrši procenu same cene (slika 8).
Akteri: Radnik agencije (vođa tima).
Preduslovi: Radnik je prijavljen na sistem.
Osnovni tok:
1. Radnik se prijavljuje na sistem.
2. Kreira listu stvari koje se sele i specificira određene pojedinosti za njih.
3. Automatski se generiše preliminarna cena selidbe.
Posledice: Određena je cena za selidbu i pojedinosti.



Slika 8. Stranica za popis stvari koje je potrebno preseliti.

N. Pregled statistike od strane vlasnika agencije

Kratak opis: Vlasnik ima uvid u statistiku poslovanja agencije.
Akteri: Vlasnik agencije.
Preduslovi: Vlasnik je prijavljen na sistem.
Osnovni tok:
1. Vlasnik se prijavljuje na sistem.
2. Odlazi na stranu statistike
3. Ima uvid u broj zaposlenih, ukupan godišnji i mesečni prihod.
Posledice: Vlasnik ima uvid u poslovanje firme i na osnovu toga može da donosi određene odluke.

O. Ostavljanje recenzije

Kratak opis: Klijent ima mogućnost recenziranja svoje konkretne selidbe, kao i same agencije.
Akteri: Registrovan klijent.

Preduslovi: U slučaju ostavljanja recenzije agencije potrebno je da je pregled selidbe zakazan. Ukoliko se radi o recenziranju selidbe, potrebno je da ona bude realizovana.
Osnovni tok:
1. Korisnik se prijavljuje na sistem.
2. Odlazi u poseban odeljak ostavljanja recenzija.
3. Ostavlja ocenu i komentar.
Posledice: Ostavljanje recenzije agenciji.

P. Brisanje neprimerenih komentara

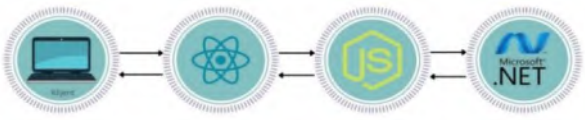
Kratak opis: Komentari koji nisu u skladu sa kodeksom definisanim od strane agencije će biti uklonjeni.
Akteri: Administrator.
Preduslovi: Komentar se odlikuje neprimerenim sadržajem.
Osnovni tok:
1. Uočavanje komentara gore pomenutog tipa od strane administratora sistema.
2. Uklanjanje sadržine koja nije u skladu sa propisanim ponašanjem korišćenjem ukloni komentar akcije.
3. Potraga za novim slučajevima neadekvatnog ponašanja u cilju održavanja standarda sistema.
Posledice: Izabrani komentar je obrisan.

III. TEHNOLOGIJE

A. Sistemski zahtevi

My MSA sistem je realizovan korišćenjem *HTML*-a, *CSS*-a, *JavaScript*-a i *React framework*-a uz primenu svih potrebnih biblioteka u cilju bogatijeg korisničkog interfejsa. Implementiran je i serverski deo uz primenu *ASP.NET*, pri čemu su podaci čuvani pomoću baze podataka. (Slika 9). Korisnički interfejs Web aplikacije mora da bude optimizovan za sledeće Web čitače:

- Google Chrome
- FireFox (Mozilla)
- Internet Explorer 6+
- Microsoft Edge
- Opera 8+
- Safari.



Slika 9. Dijagram pregleda korišćenih tehnologija.

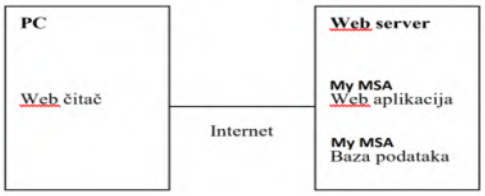
B. Perspektiva proizvoda

MY MSA portal je Web aplikacije koja omogućava digitalni prikaz za zakazivanje i planiranje selidbi. Dijagram koji pokazuje kontekst sistema dat je na slici 10.



Slika 10. Dijagram konteksta sistema.

My MSA sistem je zasnovan na klijent – server arhitekturi Web aplikacija (slika 11).



Slika 11. Dijagram arhitekture Web aplikacije.

Serverski deo se izvršava u kontekstu Web servera na personalnom računaru koji je za to namenjen. Pri izboru tehnologija vođeno je računa o tome da sistem funkcioniše ispravno na većem broju platformi (Windows, Linux, itd.). Serverske komponente komuniciraju sa *DBMS*-om na istoj mašini. Klijentska strana sistema se izvršava na personalnim i prenosivim računarima, u okviru Web čitača koji se sa Web serverom povezuje pomoću Interneta. Ne postoji potreba za posebnom instalacijom klijenta, potrebno je voditi računa o kompatibilnosti sistema sa različitim popularnim tipovima Web čitača.

C. React

React (takođe poznat kao *React.js* ili *ReactJS*) je besplatna *JavaScript open source* biblioteka za kreiranje korisničkih interfejsa [1]. Interfejsi su zasnovani na komponentama korisničkog interfejsa. *React* se može koristiti kao osnova u razvoju *single-page*, mobilnih ili serverskih aplikacija sa okvirima kao što je *Next.js*. Kreiranje *React* aplikacija obično zahteva upotrebu dodatnih biblioteka za rutiranje, kao i određene funkcionalnosti na strani klijenta.

D. ASP.NET

ASP.NET je proces koji izvršava funkcionalnost opsluživanja zahteva prispelih sa više Web čitača i parsira zahteve [2]. Ukoliko je zahtevana stranica JavaScript skripta, *ASP.NET* inicira izvršenje procesa koji obrađuje odgovarajuću skriptu i generiše sadržaj koji se vraća čitaču. *ASP.NET* može paralelno da inicira veći broj procesa. Proces obavlja posao obrade zadate *JavaScript* skripte i generiše odgovarajući tekstualni sadržaj koji *ASP.NET* šalje Web čitaču. Za izvršenje *JavaScri*pt skripti ovaj proces može da zahteva usluge *Azure SQL* servera.

Komunikacija između procesa i *Azure SQL* servera se obavlja tako što *ASP.NET* prosleđuje zahtev za podacima *Azure SQL* serveru, a on vraća rezultate.

IV. AUTORIZACIJA I AUTENTIFIKACIJA

Autorizacija i autentifikacija su bezbednosni procesi vezani za očuvanje sigurnost podataka korišćenih od strane administatora u svrhu zaštite sistema i informacija. Kombinovanjem ovih procesa postiže se sigunost sistema. Sistem se ne može smatrati bezbednim bez pravilno konfigurisane autentifikacije i autorizacije.

A. Autorizacija

Autorizacija određuje nivo pristupa korisnika [3]. Koristimo autorizaciju kako bi korisnicima ili uslugama obezbedili pristup podacima ili obezbedili mogućnost obavljanja radnji. Ovim procesom određujemo koje sve delove sistema korisnik sa određenim pravima pristupa može da koristi.

B. Autentifikacija

Autentifikacija potvrđuje da je korisnik ono što tvrdi da jeste [4]. Koristimo autentifikaciju u sistemima kako bi kotisnicima obzbedili mogućnost pristupa podacima. Sistem potvrđuje autentičnost korisnika koristeći princip nečega što samo taj korisnik zna. Autorizacija određuje šta korisnici mogu da urade kada im sistem odobri pristup, a autentifikacija ih verifikuje pre nego što im da pristup.

C. JSON Web Token (JWT)

JWT je otvoreni standard koji omogućava bezedan način da dve strane razmene podatke u format *JSON* objekata [5]. Informacija je digitalno potpisana i kao takva može biti verifikovana. U web aplikaciji My MSA za postizanje autorizacije korišćeni su *JSON Web* tokeni. Kada se korisnik prijavi na sistem u okviru svakog narednog zahteva tog korisika poslaće se i *JWT*. Pomoću ovog tokena koriniku će se omogućiti pristup rutama, uslugama i resursima dozvoljenim tim tokenom.

V. PREGLED SRODNIH REŠENJA

Poteškoće koje nastaju prilikom selidbe prepoznate su i u svetu, te se danas nude srodne aplikacije koje bi olakšale ovu životnu etapu [6]. Neke od njih su:

MoveAdvisor – aplikacija koja korisniku omogućava evidentiranje budućih obaveza, popis lične svojine i selekciju određene lokalne agencije za prevoz stvari. Ova aplikacija je besplatna i može se pribaviti za iPhone, Android i Amazon tablete.

Todolist – softver za upravljanje zadacima za kreatore lista. Pored pakovanja, premeštanja i raspakovanja postoje i mnogobrojne obaveze vezane za prenos komunalnih usluga, upravljanje periodičnim plaćanjem zakupnine ili hipoteke, promena adrese u SUP-u. Sve ovo se može pratiti preko Todolist aplikacije. Cena je 3-5 dolara mesečno, ali postoji i već gotov besplatan plan. Može se nabaviti za iPhone i Android.

TaskRabbit – aplikacija koja povezuje slobodne radnike i klijente kojima je selidba potrebna brzo. Može se unajmiti neko za košenje trave, sklapanje IKEA nameštaja ili jednostavno, prenos stvari sa starog na novo mesto. Ova aplikacija je prevashodno namenjena za one koji su u poslednjem trenutku odlučili da se ne sele sami ili one koji su zaboravili da unajme radnu snagu, jer je slobodne radnike tada teško naći pogotovo ukoliko se radi o velikom gradu. Aplikacija je besplatna i dotupna za sve platforme.

VI. ZAKLJUČAK

Postojanje ovakve aplikacije značajno olakšava jedan životni aspekt čoveka, ali i potpomaže u optimalnom poslovanju firme koja pruža ovakve usluge. Projekat i aplikacija My MSA je otvorena za dodavanje novih i inovaciju postojećih funkcionalnosti. Postoji ideja za unapređenjem postojeće aplikacije dodavanjem mogućnosti smeštanja i čuvanja robe u magacinima (uz dodatnu naplatu usluge) ukoliko selidbu nije moguće izvršiti po samom iseljenju iz polazne tačke preseljenja. Bilo bilo bi omogućeno i praćenje jedinstveno identifikovanih paketa u svakom trenutku od strane klijenta i zaposlenih u magacinu. Postoji i potreba za unapređenjem organizacije vozila i tereta koji bi mogli da prenose. Razvojni tim aplikacije će nastojati da u narednom periodu realizuje ove ali i mnogo drugih zamisli, obzirom da ova tema predstavlja nepresušni izvor novih ideja za njene članove.

Svim čitaocima ovog rada, preporučujemo da posete sledeći link i pogledaju video prezentaciju projekta: https://www.loom.com/share/0772f527301c47fd84e4a832d902a81a

ZAHVALNICA

Autori se posebno zahvaljuju prof. dr Emiliji Živanović na izdvojenom vremenu, sugestijama i pomoći pri pisanju rada.

LITERATURA

[1] https://reactjs.org/tutorial/tutorial.html#what-is-react
[2] https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet
[3] https://www.onelogin.com/learn/authentication-vs-authorization
[4] https://www.sailpoint.com/identity-library/difference-between-authentication-and-authorization/
[5] https://jwt.io/introduction
[6] https://junehomes.com/blog/2022/05/24/the-12-best-moving-apps-for-2022/

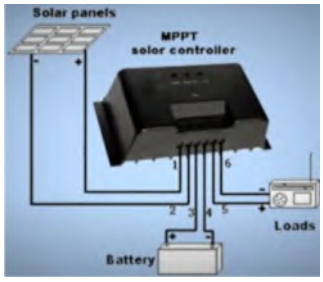
Projektovanje štampane ploče za MPPT kolo

Anastasija Avramović, Marija Simonović, Nemanja Stojanović, Vukašin Nikolić

Sadržaj − Rad opisuje kolo za praćenje tačke maksimalne snage za primenu sa solarnim panelima i daje predlog PCB dizajna ovog kola. Direktno povezivanje solarnog panela sa opterećenjem nije dobra ideja. Veliki deo energije koju solarni panel ima potencijala da generiše, uopšte se ne generiše. Svaki panel ima svoju unutrašnju otpornost koja varira izlaznu snagu kada panel generiše maksimalnu snagu, Tako da u stvarnosti izlaz solarnog panela ne zavisi samo od količine svetlosti koju prima i temperature, već zavisi i od vrednosti opterećenja. Ukoliko se ne uskladi otpornost panel se neće koristiti u potpunosti, jer je veoma teško imati opterećenje koje ima sličnu vrednost unutrašnje otpornosti ili impedanse kao solarni panel. Tako da je rešenje koristiti MPPT (*Maximum Power Point Tracking*) uređaj između panela i opterećenja.

I. MPPT

MPPT izoluje opterećenje od panela (slika 1). Ovaj uređaj nije ništa drugo nego DC DC pretvarač sa različitim unutrašnjim otpornostima. MPPT će se prebaciti na unutrašnji otporni nivo koji će uvećati snagu na panelu za datu količinu solarne insolacije ili opterećenja. Velika prednost MPPT je što štiti bateriju od prekomernog punjenja (to se dešava kada se baterija puni naponom većeg od navedenog) i nedovoljnog punjenja [1].



Slika 1. MPPT.

Solarni panel daje nelinearni izlaz, zbog toga postoji maksimalna radna tačka, na toj tački panel postiže najveću efikasnost. MPPT algoritam kontinuirano upoređuje i prilagođava različite tačke na izlazu da locira najveću efikasnost. Sam proces praćenja je automatski i ne zahteva učešće korisnika. Algoritam ima 3 stepena punjenja baterije za brzo, efikasno i bezbedno punjenje baterije, a to su: *bulk* punjenje, *absortion* i *float* punjenje. U *bulk* fazi napon na bateriji nije dostigao vrednost konstantnog napona *(Equalize/Boost*) i kontroler radi u režimu konstante struje, isporučujući svoju maksimalnu struju do baterije. Kada napon baterije dostigne zadatu vrednost konstantnog napona, kontroler će početi da radi u *absortion* režimu. Ovaj proces više nije MPPT punjenje; struja punjenja će postepeno opadati tokom ove faze. Tokom *float* faze korisnik može podesiti konstantno vreme i unapred podešenu vrednost napona pojačanja u podešavanjima. Pojačanje predstavlja odnos ulaznog i izlaznog napona. Ova faza se koristi za sprečavanje prekomernog zagrevanja baterije, u ovoj fazi napon i struja su konstantni.

A. Avramović, M. Simonović, N. Stojanović i V. Nikolić su studenti osnovnih studija na modulu Elektronske komponente i mikrosistemi, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: anastasija.avram@elfak.rs.

II. ELEKTRIČNA ŠEMA U ALTIUM DESIGNER-U

Altium Designer je softverski paket namenjen za dizajn štampanih ploča. Svaka komponenta u okviru Altium-a poseduje svoj odgovarajući simbol, *footprint* i 3D model. Na slici 2 prikazana je električna šema, može se primetiti da se MPPT sastoji se iz više blokova.

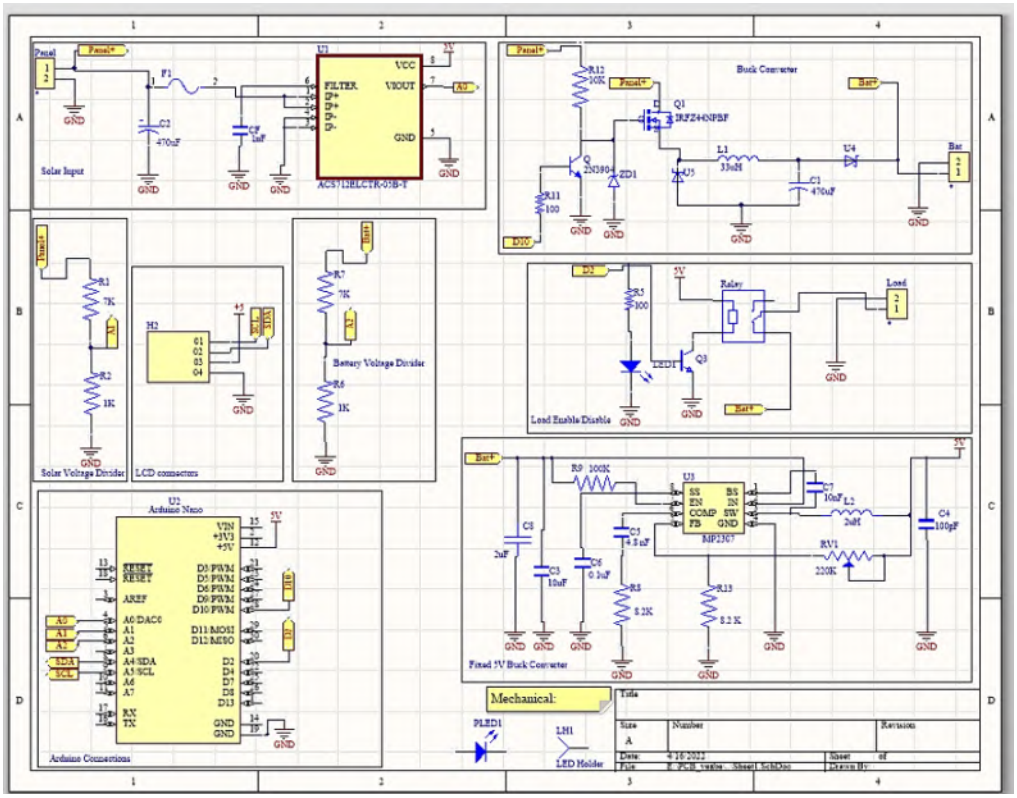
A. Solar input

Sastoji se od konektora sa dva pina na koji se dovodi solarni panel. Zatim, elektrolitskog kondenzatora od 470 µF koji predstavlja rezervoar za skladistenje energije, pošto napon na solarnim panelima varira u odnosu na količinu sunčeve energije. Osigurač od 3 A služi da zaštiti strujni senzor ACS712 od pregorevanja u slučaju nailaska veće struje. Strujni senzor ACS712 [2] je korišćen kako bi se očitavala i ograničavala vrednost struje na željenu vrednost.

ACS712 radi na principu Holovog efekta, on detektuje određenu struju i konvertuje u napon proporcionalno od 66 do 185 milivolti po amperu. Izlaz iz strujnog senzora se dovodi na analogni ulaz A0 Arduino mikrokontrolera koji manipuliše sa vrednostima struje i napona.

B. Naponski senzor

Naponski senzor realizovan je preko naponskog razdelnika korisćenjem otpornika od 1 kΩ i 7 kΩ zato što se očekuje da ulazni napon bude 36 V, a napon na Arduino treba da bude maksimalno 5 V, ovaj naponski razdelnik omogućava napon na Arduinu 4.5 V. Postoji jos jedan naponski senzor realizovan preko naponskog razdelnika koji služi da meri napon na bateriji, izlaz iz naponskog razdelnika povezuje se na analogni ulaz A2 mikrokotrolera. U odnosu na fazu punjenja baterije (*bulk, apsortion, float*) napon će imati određenu vrednost, to će biti kasnije objašnjeno u radu.



Slika 2. Električna šema.

C. LCD konektor

Sastoji se od 4 pina, od kojih je jedan za masu, drugi za napajanje (napaja se na 5 V), SDA (*Serial Data*) pin se povezuje na A4 pin mikrokontrolera, dok se SCL (*Serial Clock*) povezuje na analogni ulaz A5. Po *datasheet*-u pinovi A4 i A5 su predviđeni za I2C komunikaciju. Ovaj način komunikacije predstavlja I2C komunikaciju. *Master* je mikrokontroler, a *slave* LCD *display* na kome se prikazuju informacije.

D. Buck convertor

Tranzistor 2N3904 služi kao MOSFET (*metal–oxide–semiconductor field-effect transistor*) drajver, odnosno on će isključivati i uključivati MOSFET tranzistor (koji predstavlja prekidač za buck konvertor). Baza bipolarnog tranzistora je preko otpornika 100 Ω povezana na digitalni pin 10 Arduino mikrokontrolera. Takođe, kolektor bipolarnog tranzistora je povezan preko otpornika od 10 kΩ koji služi da ograniči struju kroz Zener diodu na pozitivan kraj panela. Zener dioda služi kao regulator napona, takođe ona se integriše s mosfetom da bi zaštitili gejt oksida od visokog ESD (*Electrostatic discharge*) napona. Kada je bipolarni tranzistor vodi na gejtu mosfeta se pojavi Zenerov napon koji automatski uključuje mosfet. Napon vođenja MOSFET tranzistora je od 2 do 4 V. U zavisnosti od moda punjenja baterije očitava se struja i napon preko strujnog i naponskog senzora, na osnovu toga šalje se PWM signal iz pina 10 mikrokontrolera. Zahvaljujući PWM signalu može da se poveća tj.smanji vrednost struje i napona koju šalje solarni panel.

U bulk modu, prvoj fazi punjenja baterije, struja se ograničava na konstantnu vrednost, dok se vrednost napona menja. Kao što je naglašeno struja se kontroliše pomoću PWM signala čija je frekvencija reda 30 kHz, što znači da je perioda signala mala. U bulk modu struja punjenja baterije je 2 A. Pošto se struja meri pomoću strujnog senzora, kada ima vrednost ispod 2 A, onda se poveća duty cycle, odnosno PWM i obrnuto. Kada se dostigne maksimalni napon bulk moda punjenja baterije, prelazi se u drugi mod, odnosno apsorbciju, u toj fazi napon je na konstantnoj vrednosti, dok vrednost struje opada, kada ona dostigne vrednost oko 0.1 A, tada započinje treća faza, float, punjenja baterije, napon je konstantan i iznosi oko 13 V, dok struja ima malu vrednost ispod 0.1 A.

Kada MOSFET vodi, energija magnetnog polja se skladišti u kalemu vrednosti 33 µH, a energija električnog polja se skladišti u kondenzatoru. Struja kroz kalem raste linerano kada mosfet vodi, odnosno kada se poveća *duty cycle*, u suprotnom ona opada linearno. Kada MOSFET ne vodi, odnosno kada je prekidač podignut, pošto kalem teži da zadrži stacionarno stanje, odnosni da zadrži isti smer struje, on okreće polaritet, zatim šotkijeva dioda (U5) počinje da vodi. Šotkijeva dioda (U4) služi da zaštiti bateriju od inverzne polarizacije. Dok kondenzator reda 470 µF služi kao napajanje u slučaju kada je prekidač podignut.

E. Opterećenje

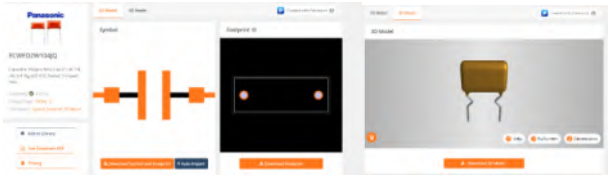
Ukoliko napon na bateriji dostigne vrednost manju od minimalnog napona na koji je potreban da bi se povezalo opterećenje, putem *relay*-a koji igra ulogu prekidača ujedno i zaštite isključujemo, odnosno uključujemo opterećenje. Tako što na digitalni pin D2 šaljemo logičku 0, odnosno logičku 1. Kao indikator toga da li je opterećenje priključeno na bateriju koristimo LE diodu i otpornik koji će ograničiti struju kroz diodu na 20 mA. Tranzistor Q3 igra ulogu prekidača, odnosno kada na bazu tranzistora dođe 5 V on ode u zasićenje, pošto je njegov kolektor priključen na GND pin *relay*-a i kada je tranzistor u zasićenju ostvaren je put ka masi, što će uključiti *relay*.

F. MP2307

MP2307- buck konvertor, u suštini ovo kolo služi da nam obezbedi napajanje za Arduino mikrokontroler, strujni senzor, LCD i *relay*. Ulaz u ovo kolo predstavlja baterija, a na izlazu se dobija fiksnih 5 V [3].

III. Layout PCB-a u Altium Designer-u

Svaka komponenta u okviru Altium-a poseduje svoj odgovarajući simbol, *footprint* i 3D model. Sve komponente su skinute sa sajta *SnapEDA* (slika 3), komponente su imale u okviru biblioteka integrisane simbole, *footprint*-ove i 3D modele [4].



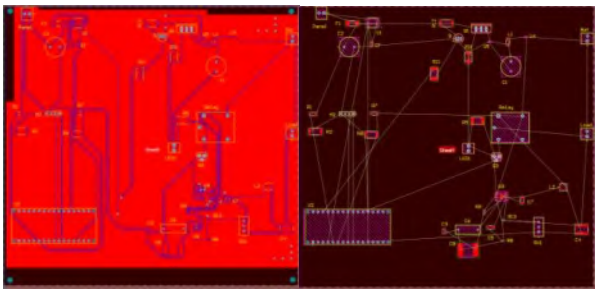
Slika 3. Primer 2D i 3D modela komponente.

Na slici 5 se može primetiti 2D prikaz pločice, linije su deblje tamo gde teče jača struja, u ostalim delovima smo podesili standardnu širinu linija. Širinu debljih linija smo odredili pomoću online kalkulatora (slika 4), znajući struju koja teče kroz deo kola gde se očekuju veće struje [5].



Slika 4. Online kalkulator.

Nakon izvršenog rutiranja vide se veze u top i bottom layer-u. Crvene su u top layeru, plave u bottom layer-u. One su na određenim mestima povezane vijama, koje imaju ulogu da spoje linije veze iz top u bottom layer. Masa je razlivena i sa donje i sa gornje strane pločice, kako bi obezbedili elektromagnetnu kompatibilnost i smanjenje šuma u kolu. Takođe, masa iz donjeg i gornjeg dela je povezana odgovarajućim vijama.



Slika 5. Prikaz pločice pre i nakon rutiranja i izlivanja mase.

Na slici 6 je prikazan 3D prikaz pločice, sa komponentama u 3D. Može se primetiti kako bi pločica izgledala u realnosti.

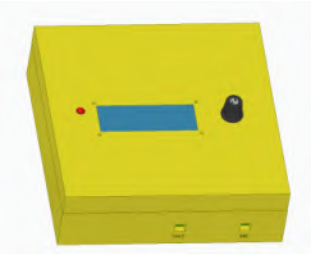


Slika 6. 3D prikaz pločice.

IV. Mehanički dizajn u Solid Edge-u

Kutija je napravljena u skladu s pločicom, dimenzija 150·140 mm. Sastoji se iz dva dela, gornjeg i donjeg. Oba dela su napravljena tako što je prvobitno nacrtan pravougaonik koji je zatim ekstrudovan. Nakon toga je nacrtan pravougaonik na kvadru i cut-ovan je, odnosno izdubljen iznutra. Nakon toga su nacrtani držači za kutiju, prorezi za hlađenje, izbušeni su otvori za konektore i na kutiji je obeležen konektor prema njegovoj nameni.

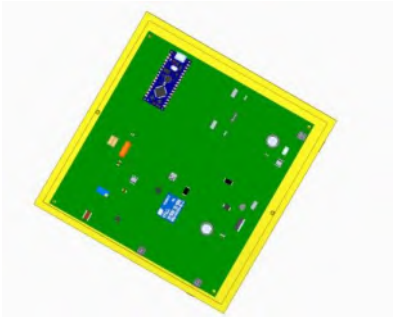
Na gornjem panelu je (slika 7) nalazi se LED dioda koja ima ulogu indikatora da li je opterećenje povezano, LCD i potenciometar koji služi za podešavanje osvetljenosti displeja. Sa bočne strane se nalaze otvori za konektore (slika 8). Takođe postoje prorezi koji omogućavaju hlađenje uređaja. Na slici 9 može se videti unutrašnjost kutije bez poklopca.



Slika 7. Prednja strana kutije.



Slika 8. Prikaz bočnih strana kutije.



Slika 9. Unutrašnjost kutije.

V. ZAKLJUČAK

Projekat MPPT predstavlja prototip koji se može koristiti za praćenje tačke maksimalne snage različitih solarnih panela i pomoći kupcima da odrede najefikasnije solarne panele dostupne na tržištu. Ko želi da poboljša performanse uređaja može dodati još jedan strujni senzor na izlazu kako bi se poredile snaga na ulazu i izlazu, takođe se može dodati čitač memorijske kartice kako bi se podaci skladištili u nju.

LITERATURA

[1] MPPT dostupno na: https://www.solar-electric.com/learning-center/mppt-solar-charge-controllers.html/

[2] Tehnička specifikacija za strujni senzor ACS712. Dostupno na: https://www.alldatasheet.com/view.jsp?Searchword=Acs712%20datasheet&gclid=Cj0KCQjwyt-ZBhCNARIsAKH1177AhXQ8kzfdW9NocaiONzhSpt1aNZc4NFiE8ZbhIKWqp7KYwQOVf8UaAj5dEALw_wcB.

[3] Tehnička specifikacija za MP2307. Dostupna na: https://cdn-shop.adafruit.com/datasheets/MP2307_r1.9.pdf.

[4] Sajt SnapEDA. Dostupno na: https://componentsearchengine.com/?gclid=Cj0KCQjw166aBhDEARIsAMEyZh7d4wPfoUIQCTIVUNARH3wG0jqG9X4kYrL76hw7TjZsVoTwlqIvNS0aAhHaEALw_wcB.

[5] Online kalkulator za širinu linija. Dostupno na: https://www.4pcb.com/trace-width-calculator.html.

Sistem za praćenje vlažnosti zemljišta i parametara zagađenosti vazduha korišćenjem razvojne platforme STM32F7508-DK

Anđelina Zdravković, Anđela Pešić

Sadržaj – U ovom radu biće predstavljena realizacija sistema za praćenje vlažnosti zemljišta i parametara zagađenosti vazduha korišćenjem STM32F7508-DK razvojnog sistema, *Hydro Probe Click* senzora i *Air Quality 7 Click* senzora.

I. UVOD

Porast stanovništva na zemlji suočiće urbana područja sa mnogim izazovima. Nedostatak vode i zagađenost vazduha su jedan od najizazovnijih problema širom sveta, kako zbog povećanja stanovništva, tako i zbog klimatskih promena. U današnjim uslovima ubrzanog razvoja, veoma malo se vodi računa o životnoj sredini i vrednosti elemenata životne sredine su veoma narušene, što svakim danom ima sve izraženije efekte na naš svakodnevni život. Jedno od rešenja za takve izazove je razvoj tehnologija koje koriste senzore, prikupljaju informacije i koriste te informacije za obavljanje nekih zadataka koji se odnose na stvaranje efikasnijeg okruženja.

Državne stanice za merenje kvaliteta vazduha ne pokrivaju dovoljan deo teritorije, pa se podstiče građansko merenje kvaliteta vazduha sa ciljem dobijanja preciznijih i tačnijih podataka kao i pokrića većeg dela teritorije. Na ovaj način dobijamo preciznije rezultate i više se podiže svest ljudi o zagađenosti vazduha.

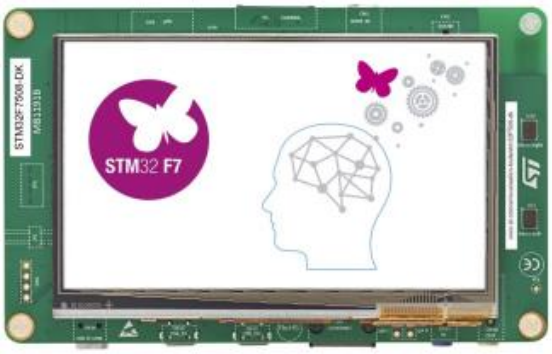
II. PRAĆENJE PARAMETARA VLAŽNOSTI ZEMLJIŠTA I ZAGAĐENOSTI VAZDUHA

U ovom radu koristi se razvojni sistem STM32F7508-DK, koji se putem I2C interfejsa povezuje sa senzorima, nakon čega se vrši očitavanje vrednosti.

A. STM32F7508-DK razvojni sistem

Razvojni sistem STM32F7508-DK [1], [2] je kompletna platforma za demonstraciju i razvoj koja nudi korisnicima sve što je potrebno da brzo počnu sa radom i lako razviju aplikacije kao što su npr. aplikacije za praćenje parametara vazduha ili vlažnosti zemljišta.

Ovaj razvojni sistem je zasnovan na mikrokontroleru STM32F750N8H6 MCU, a serija mikrokontrolera STM32F7 zasnovana je na procesoru Arm Cortex-M7. STM32F7508-DK omogućava korisnicima da razvijaju aplikacije i dele ih sa mikrokontrolerima serije STM32F7, koji su zasnovani na ARM Cortex M7 jezgru. STM32F7508-DK ploča je dizajnirana tako da se napaja sa 5 V. Na slikama 1 i 2 prikazane su gornja i donja strana ploče.



Slika 1. Pogled na gornju stranu razvojnog sistema STM32F7508-DK.



Slika 2. Pogled na donju stranu razvojnog sistema STM32F7508-DK.

Mikrokontroler STM32F750N8H6 sadrži sledeće: četiri I2C-a, šest SPI-a sa tri multipleksirana simpleks I2S-a, SDMMC, četiri USART-a, četiri UART-a, dva CAN-a, tri 12-bitna ADC-a, dva 12-bitna DAC-a, dva SAI-a, 8- do 14-bitne interfejs module za digitalnu kameru, interni 320+16+4-kilobajtni SRAM i 64-kilobajtna Flash

A. Zdravković i A. Pešić su studenti četvrte godine modula Elektronika, podmodula Embeded sistemi na Elektronskom fakultetu u Nišu, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: andjelinaa@elfak.rs.

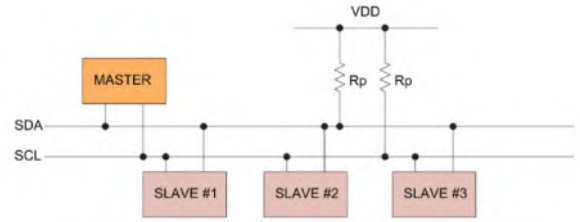
memorija, USB OTG HS i FS, Ethernet MAC, FMC interfejs, Quad-SPI interfejs, SWD podrška za debagiranje.

B. I2C komunikacija

Sama komunikacija između STM32F7508-DK razvojne ploče i senzora vrši se putem I2C interfejsa. I2C je interfejs za komunikaciju između integrisanih kola koji koristi samo dve linije: SCL (linija serijskog takta) i SDA (linija serijskih podataka).

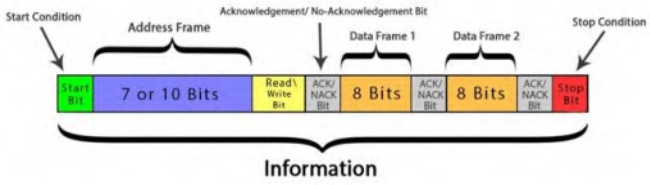
I2C komunikacioni interfejs prati hijerarhiju *master*/*slave*-a. Zbog toga je ključno izbegavati dupliranje adresa među *slave*-ovima. *Slave*-ovi nikada ne pokreću prenos podataka.

I2C magistrala koristi samo dve dvosmerne linije, liniju serijskih podataka (SDA) i liniju serijskog takta (SCL), kao što se može videti na slici 3. I2C kompatibilni uređaji se povezuju na magistralu sa otvorenim kolektorom ili „*open-drain*" pinovima koji se postavljaju na naponski nivo logičke nule. Kada nema prenosa podataka, I2C linije magistrale ne rade u *HIGH* stanju; linije se pasivno pull-uju visoko.



Slika 3. Generalizovani dijagram I2C veze.

Podaci se prenose u porukama. Poruke su razbijene u okvire podataka. Svaka poruka ima adresni okvir koji sadrži binarnu adresu *slave*-a i jedan ili više okvira podataka koji sadrže podatke koji se prenose. Poruka takođe uključuje START i STOP bit, bit za čitanje/upis i ACK/NACK bit između svakog okvira podataka. Format poruke I2C komunikacije prikazan je na slici 4.



Slika 4. Format poruke I2C komunikacije.

C. STM32CubeIDE

Razvojno okruženje koje se koristi za realizaciju projekta jeste STM32CubeIDE. STM32CubeIDE je napredna C/C++ razvojna platforma sa mogućnošću konfiguracije periferija, generisanjem koda, kompajliranjem koda i funkcijama za uklanjanje grešaka za STM32 mikrokontrolere i mikroprocesore. Zasnovan je na *Eclipse®/CDT™ framework*-u i GCC *toolchain*-u za razvoj i GDB-u za debagiranje.

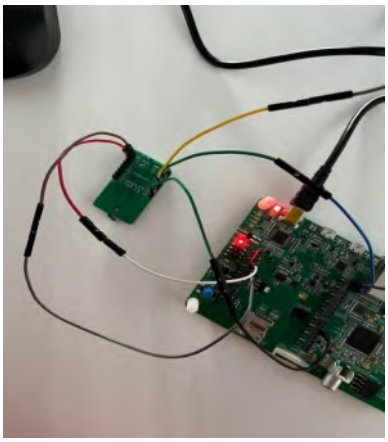
D. Air Quality 7 Click

Za očitavanje vrednosti parametara zagađenosti vazduha koristimo *Air Qualiy 7 Click* senzor (slika 5), [4] . *Air Quality 7 Click* je kompaktna dodatna ploča koja kombinuje najsavremeniju tehnologiju MOS senzora sa inteligentnim algoritmima za detekciju za praćenje lako isparljivih organskih jedinjenja - VOC i varijacija ugljen-dioksida, CO_2, u prostorijama. Ova ploča sadrži MiCS-VZ-89TE, integrisani senzorski modul za praćenje kvaliteta vazduha u zatvorenom prostoru iz kompanije *Amphenol*.



Slika 5. Air Quality 7 Click senzor.

Kao što se može videte na slici 6, prilikom povezivanja senzora sa razvojinm okruženjem moramo povezati pin 3.3 V sa senzora sa napajanjem mikrokontrolera, zatim GND pin sa masom, zatim SCL i SDA pinove sa I2C Clock i Data pinovima, ukoliko koristimo I2C interfejs, i OUT pin senzora sa PWM izlazom sa mikrokontrolera, ukoliko koristimo PWM za komunikaciju.

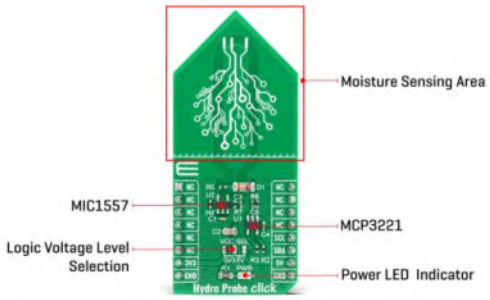


Slika 6. Povezivanje *Air Quality 7 Click* senzora i razvojnog sistema.

E. Hydro Probe Click senzor

Model sistema za merenje zapreminskog sadržaja vode u zemljištu zasnovan je na senzoru *Hydro Probe*

Click (slika 7), [3]. Ovaj senzor je odabran zbog svoje dostupnosti i zato što su testiranja u laboratorijskim uslovima pokazala da ima brz odgovor koji linearno zavisi od zapreminskog sadržaja vode u zemljištu.

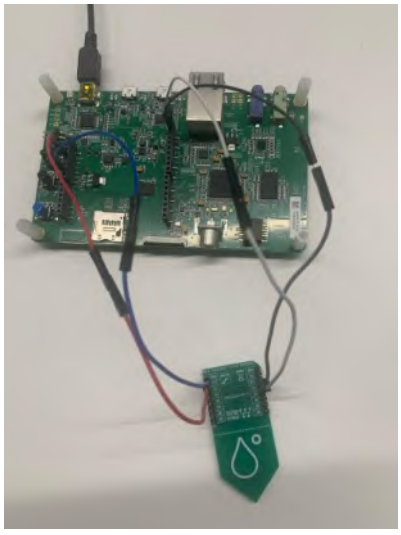


Slika 7. *Hydro Probe Click* senzor.

Hydro Probe Click je kapacitivni senzor vlage u zemljištu zasnovan na kapacitivnim promenama koje se koriste za detekciju zapreminskog sadržaja vode u zemljištu. Detekcija vode se postiže korišćenjem funkcije oscilatora MIC1557 i ADC konvertora MCP3221 kompanije *Microchip* (slika 7), koji omogućavaju konvertovanje prisustva vlage u digitalnu vrednost.

U poređenju sa otpornim senzorima vlage u zemljištu, njegov kapacitivni model smanjuje eroziju elektroda što ga čini otpornim na koroziju i boljim izborom za aplikacije kao što su detekcija vlage u zemljištu i automatsko zalivanje biljaka. Osim toga, *Hydro Probe Click* se može koristiti i u poljoprivredi, upotrebi u domaćinstvu za baštovane, u istraživanju i prognozi i još mnogo toga.

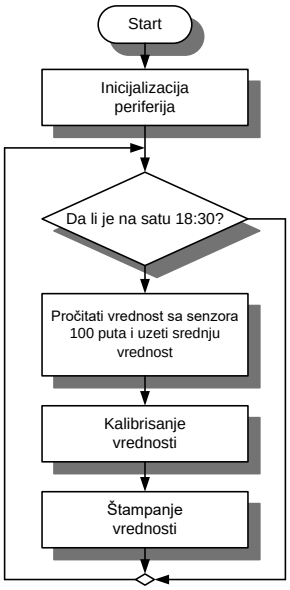
Kao što se može videte na slici 8, povezivanje senzora sa razvojnim sistemom vršimo na isti način kao što se vrši povezivanje *Air Quality 7 Click* senzora, opisano u prethodnom odeljku.



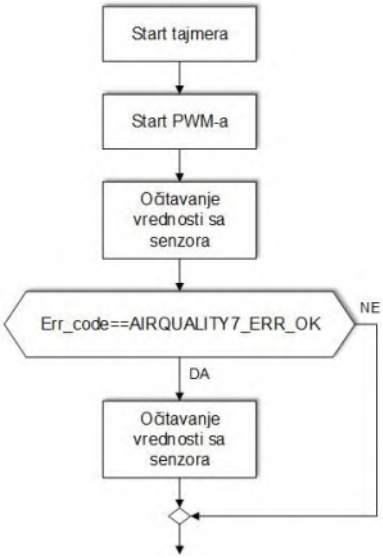
Slika 8. Povezivanje *Hydro Probe Click* senzora sa razvojnim sistemom.

III. REZULTATI

Nakon povezivanja svih hardverskih komponenti, kao što je prethodno opisano, sledeći korak je softverski deo. Za pisanje programa korišćen je C jezik. Softver za realizaciju sistema za praćenje vlažnosti zemljišta u definisanim vremenskim intervalima zasnovan je na algoritmu prikazanom na slici 9, dok je na slici 10 prikazan algoritam na kome je zasnovan softver za realizaciju sistema za praćenje kvaliteta vazduha. Kao što se može primetiti i na jednom i na drugom algoritmu nema bloka „Kraj", zato što je ovo sistem u realnom vremenu.



Slika 9. Algoritamski prikaz procesa očitavanja i prikazivanja vrednosti sa senzora vlage.

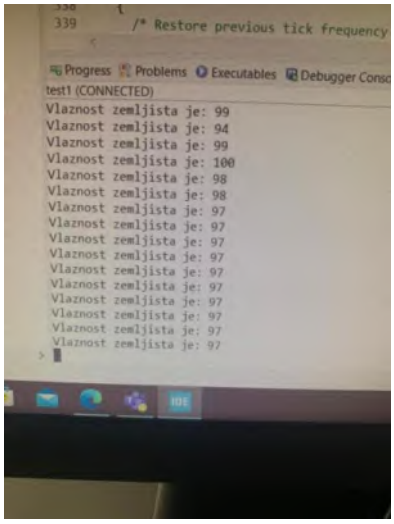


Slika 10. Algoritamski prikaz procesa očitavanja i prikazivanja vrednosti sa *Air Quality 7 Click* senzora.

Odabrano je da se očitavanje sa senzora vlage vrši jednom dnevno, u 18:30:00, u trenutku kada sat dostigne tu vrednost. U cilju povećanja tačnosti, merenja i očitavanja će se vršiti 100 puta i izračunavaće se srednja vrednost. Nakon dobijene srednje vrednosti, izvršava se skaliranje te vrednosti kako bi se dobio konačan rezultat u opsegu 0 do 100%.

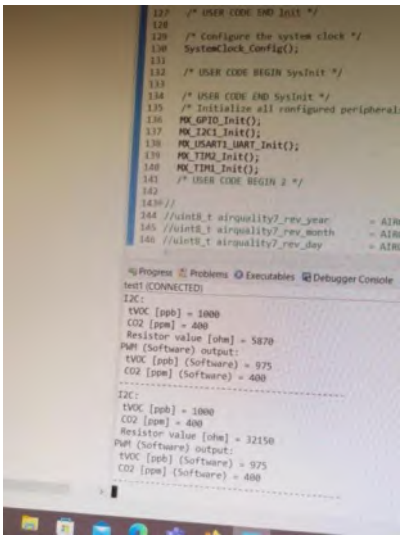
U slučaju očitavanja vrednosti sa senzora za merenje parametara vazduha važno je napomenuti sledeće. Nakon što povežemo senzor i pokrenemo kod trebalo bi da sačekamo 15 minuta da se senzor zagreje kako bismo dobili tačne rezultate. Nakon toga proveravamo status senzora i ukoliko dobijemo pozitivan odgovor prikazujemo očitane vrednosti. Ako za očitavanje vrednosti koristimo PWM onda koristimo prekidnu rutinu u okviru koje merimo period i faktor ispune signala. Na osnovu vrednosti faktora ispune korišćenjem jednačine prave kroz dve tačke računamo CO2 i VOC vrednosti.

Dobijeni rezultati očitavanja izmerenih vrednosti sa oba senzora se štampaju u konzoli na računaru kako bi se prikazalo kolika je vlažnost zemljišta i kakav je kvalitet vazduha u određenom trenutku.



Slika 11. Prikaz očitanih vrednosti sa senzora vlage.

Prskanjem alkohola u blizini *Air Quality 7 Click* senzora primećujemo značajno povećanje očitanih VOC vrednosti, što nam je jedan od načina za testiranje ovog senzora. Što se tiče *Hydro Probe Click* senzora, njega možemo testirati tako što ga stavimo u saksiji sa zemljom, ukoliko je suva zemlja, očitaće se mali procenat vlažnosti, dok s druge strane ukoliko bismo zalili zemlju, videli bi porast očitane vrednosti.



Slika 12. Prikaz očitanih vrednosti sa senzora za merenje parametara vazduha nakon prskanja alkohola.

IV. ZAKLJUČAK

U radu je prikazana praktična realizacija sistema za praćenje vlažnosti zemljišta i parametara zagađenosti vazduha. Predstavljen je model sistema za merenje vlažnosti zemljišta i kvaliteta vazduha, opisane su tehnike koje se mogu koristiti pri merenju, prikazan je detaljan opis povezivanja i komunikacije mikrokontrolera i senzora koji se koriste za izračunavanje.

ZAHVALNICA

Autori se zahvaljuju profesorima sa Katedre za elektroniku na izdvojenom vremenu, sugestijama i pomoći pri realizaciji i pripremi ovog rada.

LITERATURA

[1] Data brief „Discovery kit with STM32F750N8 MCU”
[2] UM2470 User manual *„Discovery kit for STM32F7 Series with STM32F750N8 MCU”*
[3] Hydro Probe Click, https://www.mikroe.com/hydro-probe-click
[4] Air Quality 7 Click https://www.mikroe.com/air-quality-7-clic

Kućna automatizacija bazirana na IoT tehnologijama

Rastko Dragić, Luka Đorđević, Stefan Đilas, Stefan Karać, Sergej Salonski, Filip Goldberger

Sadržaj – U ovom radu je opisan princip rada projekta "Kućna Automatizacija Bazirana na IoT tehnologijama (KABIT)", kao i komponente i alati korišćeni za njegovu izradu.

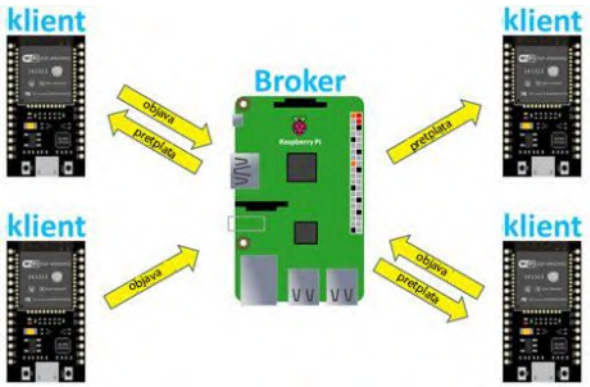
I. UVOD

KABIT je *IoT* (*Internet of Things*) projekat razvijen sa ciljem jednostavne akvizicije podataka umreženih resursa kao i njihovim daljinskim upravljanjem putem interneta.

Glavne komponente, korišćene za izradu ovog projekta su mikroračunar *Raspberry Pi* [1], centralni web server sa podrškom za *NodeJs* [2] i *Python* [3], *NodeMCU* [4] mikrokontroler, *MQTT* broker [5] i *Node-RED* [6].

II. PRINCIP RADA

Princip rada zasnovan je na konstantnoj komunikaciji između centralnog servera i *NodeMcu* mikrokontrolera preko *MQTT brokera* (Slika 1). Centralni server je u svakom trenutku u mogućnosti da upravlja kontrolerima. U isto vreme prikuplja, prikazuje i čuva podatke koje kontroleri šalju. Pregled prikupljenih podataka i live-feed kamera mogu se naći na adresi: [7].

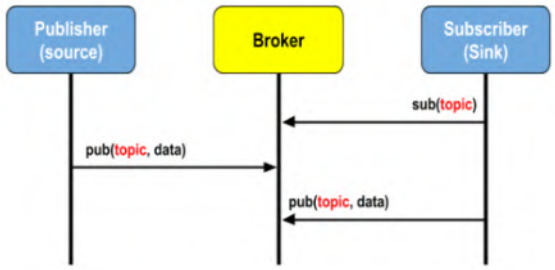


Slika 1. Šema komunikacije.

R. Dragić, L. Đorđević i S. Đilas su učenici IV razreda smer Elektrotehničar računara, S. Karać je učenik IV razreda smer Elektrotehničar informacionih tehnologija. S. Salonski je učenik III razreda smer Administrator računarskih mreža, elektrotehnička škola „Mihajlo Pupin" Novi Sad, Futoška 17, 21000 Novi Sad, Srbija, E-mail: rastko.dragic@gmail.com.

F. Goldberger je student prve godine računarstva i automatike, Fakultet tehničkih nauka u Novom Sadu, Trg Dositeja Obradovića 6, 21000 Novi Sad, Srbija, E-mail: filip.gold159@gmail.com.

NodeMCU mikrokontroleri koriste *MQTT protokol* za razmenu podataka sa centralnim serverom. *MQTT* je jednostavan protokol koji je baziran na sistemu pretplate i objave (Slika 2) koji je namenjen prenosu kratkih poruka. Očitavanja senzora se objavljuju na njima namenjenu temu (*topic*), kao na primer "dht-sprat1" (topic na koji se objavljuju temeperaturna merenja *DHT11* senzora na prvom spratu) , a uređaji koji žele da prime poruku moraju da budu pretplaćeni na tu istu temu. Poruka se prvo šalje do *broker*-a, a on je zatim prosleđuje pretplatnicima. Prednost ovog pristupa je to što je omogućena jednostavna distribucija iste poruke do svih zainteresovanih klijenata, a da pri tome nije potrebno da uređaj koji objavljuje poruku zna detalje o pretplatnicima (njihovu adresu, stanje veze itd). KABIT projekat koristi *mosquito broker* koji je instaliran na *Raspberry Pi* mikroračunaru.



Slika 2. Mehanizam pretplate i objave kod *MQTT* protokola.

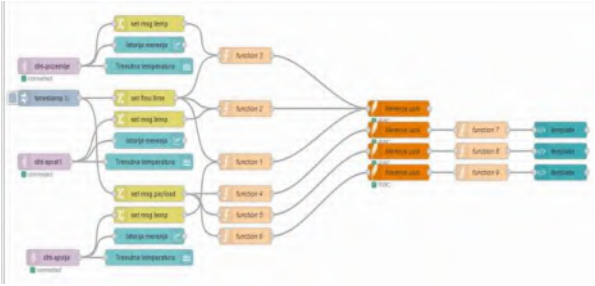
Dalji proces obrade podataka odvija se na serveru. *Node-Red* zadužen je za kontrolu resursima i prikaz podataka.

Node-RED je programski alat koji funkcioniše na principu *flow-based* [8] programiranja. Prvobitno je razvijen od strane *IBM*-ovog *ETS* tima (*Emerging Tehnology Services team*), a sada je deo *JS* fondacije.

Node-RED pokreće *NodeJs* okruženje, a *flow-editor*-u se pristupa preko porta na kojem je otvorem pristup *Node-RED okruženju*. U *editor*-u aplikacija se kreira prevlačenjem čvorova (*nodes*) iz menija na radni prostor i njihovim međusobnim povezivanjem. Paleta čvorova se može proširiti kreiranjem čvorova ili instaliranjem čvorova kreirani od strane *Node-RED* zajednice, a tokovi (*flows*) koji se kreiraju mogu biti lako podeljeni kao *JSON* datoteke.

Tok „Merenja" (Slika 3) sadrži tri *MQTT_in* čvora, svaki pretplaćen na zasebnu *MQTT* temu za merenja tri različita *DHT11* senzora povezana na dva *NodeMCU*

miktrokontrolera. Pri pristizanju podataka na odgovarajuću temu oni se prikazuju (Slika 4) na *dashboard-u (veb interfejs ugrađen u Node-RED)* u vidu trenutne temperature i na grafiku kao temperatura tokom određenog vremenskog perioda.

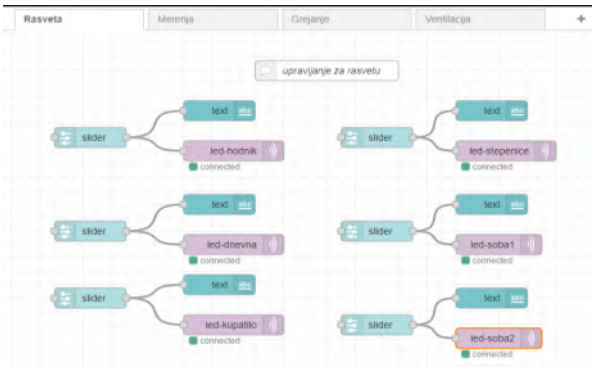


Slika 3. Tok „Merenja“.



Slika 4. Node-RED prikaz rezultata.

Tok „Rasveta“ (Slika 5) sastoji se iz klizača koji korisniku omogućavaju podešavanje intezitet svetala. Svako podešavanje se prosleđuje na zaseban *MQTT_out* čvor koji vrednosti šalje na odgovarajuće teme. Zadate vrednosti se prikazuju i čuvaju sve dok je *Node-RED* instanca aktivna.



Slika 5. Tok „Rasveta”.

III. IMPLEMENTACIJA

A. Hardverska implementacija

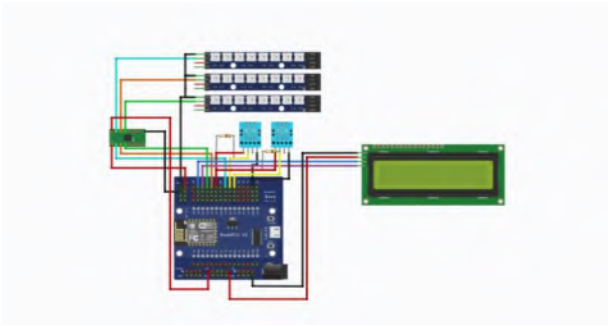
Glavna komponenta upravljačkog dela projekta KABIT je *NodeMCU* mikrokontroler.



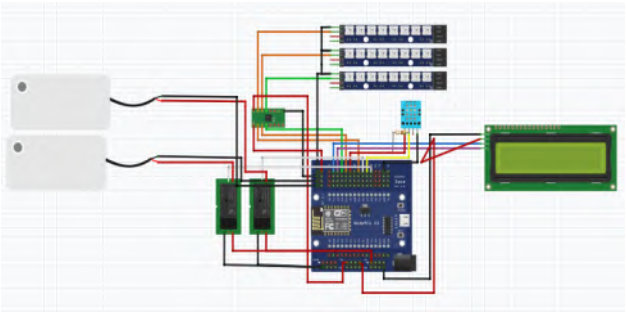
Slika 6. *NodeMCU*.

NodeMCU je jeftina *open-source* platforma zasnovana na *ESP8266* [9] mikrokontroleru sa ugrađenim *wireless* funkcionalnostima. *NodeMCU* takođe pruža neke od najbitnijih odlika mikrokontrolera kao što su *GPIO* (*general purpose input - output*) pinovi, *PWM* (*pulse-width modulation*), *A/D* konvertor, *I2C* komunikacija, itd.

NodeMCU miktrokontoleri imaju dualnu funkciju u projektu KABIT. Prva funkcija im je prikupljanje vrednosti merenja sa senzora (DHT11, ..) i slanje podataka na MQTT broker (*publish*), a druga funkcija je kontrolisanje i upravljanje različitim resursima u kući po naredbama koje dobijaju daljinski, MQTT protokolom (*subscribe*) ili automatski, u zavisnosti od ispunjenosti određenih uslova.



Slika 7. Šema povezivanja *NodeMCU (prvi sprat)*.



Slika 8. Šema povezivanja *NodeMCU (prizemlje)*.

DHT11 [10] *senzori* služe za merenje temperature i vlažnosti vazduha. Na maketi kuće ugrađena su 3 *DHT11 senzora*, unutar makete nalazi se po jedan senzor za prizemlje i prvi sprat, a spolja se nalazi senzor za spoljašnja merenja. Merenja ovih senzora svakih 10 sekundi očitavaju kontroleri (u kodu npr. dht.temperature()).



Slika 9. DHT11 senzor temperature i vlažnosti vazduha.

Osvetljenje u maketi je realizovano niskonaponskim *LED* trakama čiji se intenzitet kontroliše pomoću *PWM (pulse-width modulation).*

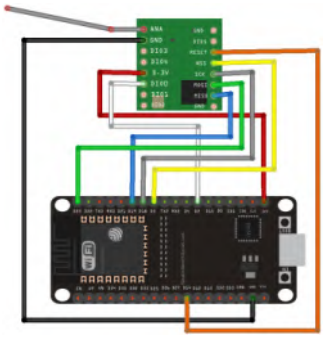
Na maketi postoje i dva elektronski kontrolisana prozora. Prozori su izrađeni od PLA plastike na „3D" štampaču „Creality Ender 3 MAX".

Mehanizam za kontrolu prozora pokreću koračni motori.

Sistem vodenog grejanja je pojednostavljen za potrebe makete i sastoji se iz spremnika vode, grejača, silikonskih creva i dve pumpe za vodu koje kontrolišu dotok tople vode za prizemlje i prvi sprat. Program koji se bavi kontrolom temperature upoređuje temperaturna merenja *DHT11 senzora* i temperaturu zadatu od strane korisnika. Ukoliko je temperatura na datom spratu niža od zadate temperature pumpa koja kontroliše dotok se uključuje preko releja i topla voda iz spremnika počinje da cirkuliše kroz sistem. Pumpa se gasi kada temperatura na senzoru pređe zadatu temperaturu.

LoRa [11] protokol omogućava slanje kratkih poruka na relativno velike distance uz pomoć LoRa modula.

Komunikacija preko *LoRa* protokola u projektu realizovana je direktno, bez *gateway-a*, između dva *NodeMCU* mikrokontrolera od kojih je jedan izvan *WiFi* mreže na udaljenoj lokaciji. Kontroler sa udaljene lokacije (u ovom slučaju to je maketa sušare za meso) kontroliše tok sušenja mesa i putem *LoRa* protokola šalje rezultate merenja sa senzora kontroleru na mreži. Nakon pirmanja informacija kontroler na mreži prosleđuje dobijene podatke *MQTT brokeru,*a on pretplaćenim korisnicima.



Slika 10. *LoRa* i *NodeMCU* šema povezivanja.

MQTT Broker, Node-RED i baza podataka se nalaze na *Raspberry pi*-u. Paralelno, *MQTT broker* podatke prosleđuje još jednom web serveru, koji takođe sem prikaza rezultata ima mogućnost upravljanja resursima.

Raspberry Pi 3 b+ model koristi četvorojezgarni *ARM Cortex-A53* procesor koji radi na frekvenciji od 1,4 GHz. Poseduje 1 GB *LPDDR2 RAM*-a. Za skladištenje koristi *MicroSDHC* slot kao i njegovi prethodnici. Na njega je instaliran operativni sistem *Raspberry Pi OS* koji je baziran na *Debian*-u.

NodeMCU kontroleri se napajaju preko eksternog napajanja od 5 V koje služi i za LED trake. Grejač i pumpe za sistem grejanja i ventilator za ventilaciju napajaju se preko eksternog napajanja od 12 V i kontrolišu se preko releja. Senzori se napajaju pomoću 3.3 V preko *NodeMCU* napajanja. Koračni motori se napajaju preko 12 V iz eksternog napajanja.

B. Softverska implementacija

Za softversko razvijanje korišćeno je razvojno okruženje *Arduino IDE* [12]. Program je napisan u programskom jeziku *C* [13].

Softverska implementacija KABIT je modularna.

Glavni delovi embeded koda za *NodeMCU* su kod za povezivanje na *WiFi* mrežu i podešavanje parametara preko web interfejsa realizovan pomoću „*WiFiManager"* biblioteke, kao i kod za povezivanje na *MQTT broker*, za koji je u ovom projektu upotrebljena biblioteka „*PubSubClient"*.

```
void loop() {
    if(millis() - tempTime > TEMPDELAY)
    {
      int t = dht.readTemperature();
      char tempString[64] = "";
      itoa(t , tempString, 10);
      client.publish("dht-spratl", temp);
    }
```

Slika 11. Primer objavljivanja podataka na njima namenjene teme.

```
if(digitalRead(13)) forceConfig = true;

WiFiManagerParameter mqttserver_text_box("key_text", "MQTT Broker Adresa", mqttServer, 50);

wifiManager.addParameter(&mqttserver_text_box);

if (forceConfig)
{
  if (!wifiManager.startConfigPortal("Esp-settup", "configure445"))
  {
    Serial.println("connection failed");
    delay(3000);
    ESP.restart();
    delay(5000);
  }
}
else
{
  if (!wifiManager.autoConnect("Esp-settup", "configure445"))
  {
    Serial.println("connection failed");
    delay(3000);
    ESP.restart();
    delay(5000);
  }
}
strcpy(mqttServer, mqttserver_text_box.getValue());
```

Slika 12. Funkcija za podešavanje *NodeMCU* preko web interfejsa.

Za svaku novu poruku na nekoj od *MQTT* tema na koje je klijent pretplaćen poziva se *callback()* fukncija (Slika 13). U zavisnosti od teme na koju je stigla i vrednosti poruke u *callback()* funkciji pozivaju se druge funkcije iz koda ili bibioteka.

```
if(strcmp(topic, "led-stepenice") == 0)
{
  int i_payload = atoi(s_payload);
  analogWrite(LEDPIN2, map(i_payload, 0, 100, 0, 255));
  Serial.print("Set lights to: ");
  Serial.println(i_payload);
}
else if(strcmp(topic, "led-soba2") == 0)
{
  int i_payload = atoi(s_payload);
  analogWrite(LEDPIN1, map(i_payload, 0, 100, 0, 255));
  Serial.print("Set lights to: ");
  Serial.println(i_payload);
}
```

Slika 13. Primer *Callback()* funkcije.

Softverska implementacija na serverskoj strani uključuje kreiranje tabele u bazi podataka u koju se podaci supisuju, kreiranje veb interfejsa i njegove funkcionalnosti.

U *Node-RED*-u se prevlače željeni *node*-ovi na tablu, konfigurišu se i povezuju. U slučaju potrebe za pisanjem funkcije koja nije predefinisana, koristi se *function node* i kod funkcije se piše u *JavaScript* jeziku.

IV. Zaključak

Zbog ograničenja koja su definisana, ovaj tekst ne može da prikaže sve mogućnosti projekta. Detaljniji opis projekta, kao i linkovi ka *NodeRED* kontrolnoj i javnoj tabli (praktična realizacija) nalaze se na veb portalu *Pametni šor*, na adresi [14].

Potrebni podaci za zaštićene delove (upravljanje resursima) su *user: admin; password: Pilab001*.

Izvorni kodovi se mogu videti na adresi: [15].

Hardversko-softverska rešenja (merenje, upravljanje…) su potpuno primenljiva na realnom sistemu. Opremu za sistem grejanja, osvetljenje, kao i kvalitetne senzore treba odabrati shodno potrebama realnog sistema.

Unapređenja predviđena za sledeći period: dodavanje funkcija koje omogućavaju upotrebu još jednog *MQTT brokera* (veći stepen sigurnosti transfera podataka), kao i zaštita informacija - šifrovaje *MQTT* poruka i *SSL*.

Zahvalnica

Zahvaljujemo se mentorima Ušan Branislavu, Bogićević Nebojši, Vukobrat Milanu i Pavlović Jovi, Razrednom starešini Hajder Radmili, kao i ostalim, sadašnjim i bivšim, članovima sekcije *PiLab* na usmeravanju, sugestijama i podršci pri izradi ovog projekta.

Literatura

[1] Raspberry Pi, Dostupno na: https://www.raspberrypi.org
[2] Node.js, Dostupno na: https://nodejs.dev/learn
[3] Python, Dostupno na: https://www.python.org/
[4] NodeMCU, Dostupno na: https://electropeak.com/learn/nodemcu-esp8266-on-arduino-ide/
[5] MQTT, Dostupno na: https://www.rt-rk.uns.ac.rs/sites/default/files/materijali/predavanja/Orm2_Beleske_Predavanje12.pdf
[6] Node-RED, Dostupno na: https://nodered.org/about/
[7] https://pametnisor.pilab021.com/2022/09/28/kucna-automatizacija/
[8] Flow-based programming, Dostupno na: https://en.wikipedia.org/wiki/Flow-based_programming
[9] ESP8266, Dostupno na: https://www.espressif.com/en/products/socs/esp8266
[10] DHT-11, Dostupno na: https://www.adafruit.com/product/386#technical-details
[11] LoRa protokol, Dostupno na: https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/
[12] Arduino IDE, Dostupno na: https://www.arduino.cc/en/software
[13] C, Dostupno na: https://www.tutorialspoint.com/cprogramming/index.htm
[14] https://pametnisor.pilab021.com/2022/09/28/kucna-automatizacija/
[15] https://github.com/RastkoDr/KABIT_emb

Kontrola brzine i smera DC motora korišćenjem kola L293D i mikrokontrolera sa ARM arhitekturom

Nikola Stojiljković, Hristina Todorović, Nikola Milenković

Sadržaj – U ovom radu detaljno je opisana realizacija sistema kojim se pomoću *ARM* mikrokontrolera i integrisanog kola *L293D* kontroliše brzina i smer *DC* motora. Iskorišćeno je *Nucleo* razvojno okruženje koje poseduje *ARM* mikrokontroler kao i integrisano kolo *L293D* koje sadrži četiri tranzistora snage. Kao kontrolor promene brzine i smera okretanja *DC* motora upotrebljen je potenciometar, a kao indikatori istog crvena i zelena *LE* dioda.

I. UVOD

U današnje vreme jedan od glavnih razloga zašto se *DC* motori koriste više od drugih tipova motora leži u njihovoj sposobnosti da odgovore preciznoj kontroli brzine i smera okretanja što je neophodno za industrijske primene. Oni menjaju smer kretanja što je suštinski faktor za kontrolu proizvodne opreme i imaju brz vremenski odziv.

Postoji sve veća potražnja za modele *DC* motora sa napajanjem od 12 V i 24 V. Tržište solarne, brodske i kamionske opreme u mnogome zavisi od tehnologije *DC* motora kao najisplativijeg rešenja. Iako je tehnologija motora na jednosmernu struju starija od tehnologije motora na naizmeničnu struju neprestano se razvijaju i formiraju nove metode za održavanje i kontrolu *DC* motora [1]. Mnogi tipovi ovih motora prilagodljivi su i podesivi da odgovaraju širokom spektru aplikacija.

II. KOMPONENTE KORIŠĆENE U REALIZACIJI PROJEKTA

U nastavku su navedene sve komponente neophodne za realizaciju ovog projekta sa najvažnijim teorijskim objašnjenjima, kako bi preciznije objasnili sam princip rada. Korišćene komponente su: *STM32 Nucleo-64 [2]* razvojno okruženje, *L293D* [3] drajver induktivnih opterećenja, *DC* motor, potenciometar, *LE* dioda, otpornici.

A. STM32 Nucleo-64 razvojno okruženje

Razvojno okruženje *STM32 Nucleo-64* (slika 1) sa *STM32F103RB* mikrokontrolerom pripada familiji *ARM* mikrokontrolera. *Arm Cortex-M* je koncept korišćenja 32-bitnih *RISC ARM (Advanced RISC Machine)* arhitektura za kreiranje mikroprocesora i mikrokontrolera.

N. Stojiljković, H. Todorović i N. Milenković su studenti četvrte godine na modulu Elektronske komponente i mikrosistemi, Elektronskog fakulteta Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: stojiljkovicn@elfak.rs.

STM32 Nucleo-64 razvojno okruženje poseduje mogućnost povezivanja sa *Arduino* razvojnim okruženjem i *STM Morpho* razvojnim okruženjem za primenu u kompleksnim aplikacijama. Ima fleksibilno napajanje od 3.3 V, 5 V, 7 V-12 V. Poseduje dve korisničke *LE* diode i dva tastera kao i mini *USB* konektor. Konfiguriše se korišćenjem softverskog alata *STM32CubeMX [4]* dok se odgovarajući *firmware* piše u odgovarajućem programu, pri čemu je ovde upotrebljen *Keil uVision [5]*.



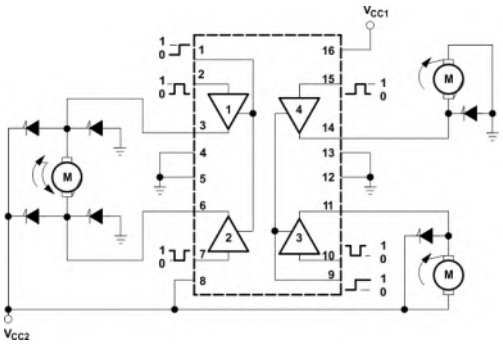
Slika 1. STM32 Nucleo-64 razvojno okruženje.

B. L293D drajver induktivnih opterećenja

Integrisano kolo *L293D* predstavlja drajver induktivnih opterećenja kao što su releji, solenoidi, *DC* motori, bipolarni *stepper* motori i slično. Svaki izlaz ovog integrisanog kola je povezan sa internim Darlingtonovim parom tranzistora (struja do 600 mA). Rad drajvera se omogućava u paru (kombinovanjem prvog i drugog ili trećeg i četvrtog kao što je prikazano na slici 2). *L293D* je jedan od najpopularnijih drajvera na tržištu. Postoji nekoliko razloga koji čine *L293D* važnim drajverom za korisnike. To su niska cena (u poređenju sa drugim drajverima), pravilan oblik i veličina, laka kontrola, nema potrebe za zaštitnim kolom i diodama, nema potrebe za hladnjakom i dobrom otpornošću na varijacije temperature i velike brzine. Ovo integrisano kolo (*IC*) može regulisati motore sa naponom od 5 V do 36 V i strujom do 600 mA. Međutim, može da izdrži struju do 1200 mA za maksimalno 100 µs. Frekvencija rada ovog *IC* je 5 kHz.

Kod ovog integrisanog kola postoje četiri pina na koje se može povezati *DC* motor. To su pinovi 3, 6, 11, 14. Takođe, poseduje i dva *ENABLE* pina za dozvolu rada, 1 i 9, kao i četiri *DRIVER INPUT* pina (2, 7, 10, 15) na koje se može dovoditi *PWM* signal čime se zapravo kontroliše brzina okretanja *DC* motora.

U zavisnosti od *PWM* signala koji se dovodi na *DRIVER INPUT* pinove, za potrebe ovog projekta koriste se pinovi 2 i 7, motor će se okretati odgovarajućom brzinom.



Slika 2. Funkcionalni blok dijagram integrisanog kola L293D.

C. DC motor

DC motori (slika 3) su bili prvi oblik motora koji se široko koristio, jer su se mogli napajati iz postojećih sistema za distribuciju energije za osvetljenje. Brzina *DC* motora se može kontrolisati u širokom opsegu, korišćenjem promenljivog napona napajanja ili promenom jačine struje u njegovim namotajima. Mali *DC* motori se koriste u alatima, igračkama i uređajima. Veći *DC* motori se trenutno koriste u pogonu električnih vozila, liftova i dizalica, kao i u pogonima za valjanje čelika.



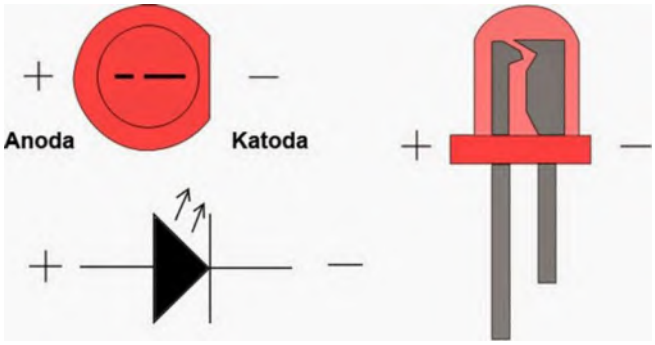
Slika 3. Primer *DC* motora korišćenog u okviru ovog projekta.

D. Potenciometar

Potenciometar predstavlja otpornik sa kontrolom promene otpornosti. Sastoji se od otpornog dela na čijim su krajevima vezani kontakti i klizača koji se kreće po otpornom delu potenciometra i takođe ima poseban priključak. Okretanjem osovine klizača, korisnik menja vrednost otpornosti.

E. LE dioda

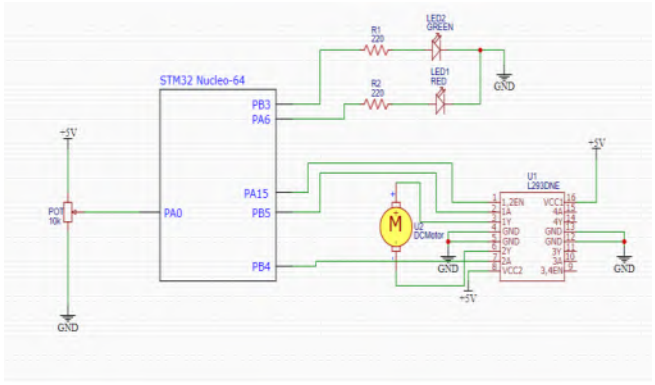
Direktno polarisane *LE* diode (slika 4) će provoditi struju jer se veliki broj nosilaca naelektrisanja kreće ka kontaktima anode i katode. Elektroni iz provodne zone (sa višom energijom) se rekombinuju sa šupljinama iz valentne zone (sa nižom energijom). Procesom rekombinacije se oslobađa energija u vidu toplote (fononi) i/ili svetlosnog zračenja (fotoni). Ukoliko dominiraju fotoni javlja se efekat fotoluminiscencije – zračenje u vidu vidljive, *IC* ili *UV* svetlosti. Za izradu *LE* dioda se koriste poluprovodna jedinjenja sa direktnim energetskim procepom [6].



Slika 4. Izvodi i električni simbol *LE* diode.

III. ELEKTRIČNA ŠEMA

U nastavku, na slici 5 je predstavljena električna šema po kojoj je realizovan ovaj projekat.



Slika 5. Električna šema u softverskom alatu *EasyEDA* [7].

IV. PRINCIP RADA

Razvojno okruženje *STM32 Nucleo-64* povezano je preko pina na interno ugrađen *A/D* konvertor sa potenciometrom vrednosti 10 kΩ pri čemu se na tom pinu očitava vrednost napona između mase i izvoda potenciometra koji je povezan na klizač potenciometra. Potenciometar služi da se njime kontroliše brzina i smer okretanja *DC* motora i to na način ako je otpornost potenciometra u opsegu od 0 % do 20 %, motor će se okretati u smeru suprotnom od smera kazaljke na satu i od 20 % do 100 % za smer kazaljke na satu, pri čemu se brzina menja srazmerno vrednosti otpornosti. Prilikom očitavanja vrednosti otpornosti okruženje će uz pomoć

odgovarajućeg programskog koda vršiti *A/D* konverziju i na pinovima koji su definisani kao izlazni generisati odgovarajući *PWM* signal za kontrolu motora.

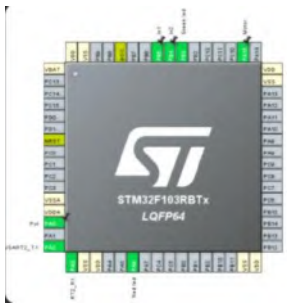


Slika 6. Podešavanje tajmera 2 za potrebe generisanja *PWM* signala na kanalu 1 i 2.



Slika 7. Podešavanje tajmera 3 za potrebe generisanja *PWM* signala na kanalu 1.

Kako bi se izvršila konfiguracija pinova razvojnog okruženja STM32 Nucleo-64 i podesili odgovarajući tajmeri i pinovi za vršenje A/D konverzije iskorišćen je softverski alat STM32CubeMX. U nastavku, na slici 6 i slici 7 data su podešavanja tajmera za potrebe generisanja impulsno širinske modulacije tj. PWM na kanalima 1 i 2, dok je na slici 8 prikazano podešavanje pinova. Pored ovih pinova imamo još dva pina koja služe da se povežu odgovarajuće *LE* diode zadužene za indikaciju smera okretanja motora i to crvena za smer suprotan od smera kazaljke na satu i zelena za smer kazaljke na satu. Takođe, diode će u skladu sa promenom brzine kojom se okreće motor, tj. vrednosti otpornosti menjati jačinu kojom trepere.



Slika 8. Konfiguracija pinova u *STM32CubeMX*.

V. PROGRAMSKI KOD

Programski kod za potrebe ovog projekta napisan je u softverskom alatu *Keil uVision* i prosleđen mikrokontroleru. Za potrebe provere smera okretanja *DC* motora koristi se virtuelni terminal. Najpre su definisane dve promenljive u delu koda predviđenom za korisnika.

```
/* USER CODE BEGIN 0 */
int adcVrednost;
int duty;
/* USER CODE END 0 */
```

Nakon toga potrebno je pokrenuti odgovarajuće tajmere za vršenje *A/D* konverzije.

```
/* USER CODE BEGIN 2 */
HAL_ADC_Start(&hadc1);
HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1);
HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_2);
HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_1);
/* USER CODE END 2 */
```

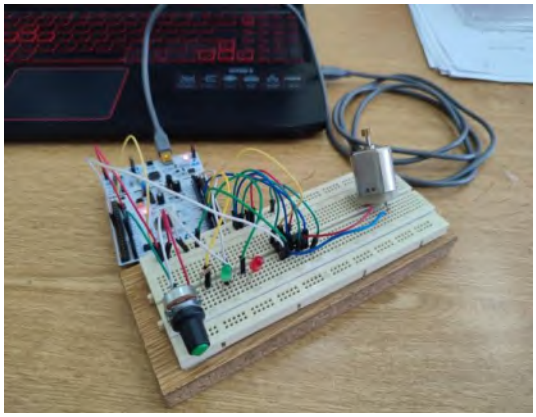
U okviru glavne programske petlje nalazi se programski kod u nastavku.

```
/* USER CODE BEGIN 3 */
adcVrednost=HAL_ADC_GetValue(&hadc1);
duty=adcVrednost*100/4095;
htim2.Instance->CCR1=duty;
if ( duty==20 || duty==0 || duty==100 )
        {
        htim2.Instance->CCR1=0;
        htim2.Instance->CCR2=0;
        htim3.Instance->CCR1=0;
        }
else { if ( duty<20 )
        {
        HAL_GPIO_WritePin(GPIOB,GPIO_PIN_5,GPI
        O_PIN_RESET);
        HAL_GPIO_WritePin(GPIOB,GPIO_PIN_4,GPI
        O_PIN_SET);
        HAL_UART_Transmit(&huart,"smer1\r\n", strlen
        ("smer1\r\n"),10);
        htim3.Instance->CCR1=50;
        htim2.Instance->CCR2=0;
        }
else
        {
        HAL_GPIO_WritePin(GPIOB,GPIO_PIN_5,GPI
        O_PIN_SET);
        HAL_GPIO_WritePin(GPIOB,GPIO_PIN_4,GPI
        O_PIN_RESET);
        HAL_UART_Transmit(&huart,"smer2\r\n", strlen
        ("smer2\r\n"),10);
        htim2.Instance->CCR2=duty;
        htim3.Instance->CCR1=0;
```

```
        }
    }
/* USER CODE END 3 */
```

VI. PRAKTIČNA REALIZACIJA

Nakon napisanog progamskog koda za mikrokontroler izvršeno je praktično povezivanje (slika 9) svih komponenata kako bi se proverila funkcionalnost samog projekta. Razvojno okruženje *STM32 Nucleo-64* je putem *USB* porta povezano na ručunar kako bi mu se obezbedilo odgovarajuće napajanje i prosledio programski kod.



Slika 9. Praktična realizacija projekta.

VII. ZAKLJUČAK

Kroz ovaj projekat prikazan je jedan od principa kontrole brzine i smera *DC* motora. Najpre je odrađena električna simulacija u okviru odgovarajućih softverskih paketa. Nakon toga izvršeno je praktično povezivanje, odrađena su odgovarajuća softverska podešavanja razvojnog okruženja *STM32 Nucleo-64* i napisan programski kod koji je prosleđen mikrokontroleru. Projekat je podložan određenim modifikacijama u smislu efikasnostii funkcionalnosti.

Alternativa za promenu smera *DC* motora bila bi primena *H* - mosta, koji sadrži četiri tranzistora snage *IRF510* povezana u takozvani *H* - most, pri čemu se u dijagonali nalazi motor. Kao kontrolor smera *DC* motora povezanog u *H* - most se koriste tasteri koji pokreću po dva, odgovarajuća tranzistora.

Za kontrolu brzine i smera okretanja *DC* motora moguće je koristiti i već gotove štampane ploče, kao što je to na primer „*Brushless DC motor drive board controller*“.

Međutim, sve ove modifikacije izlaze iz okvira trenutnih mogućnosti i zahteva ovog projekta.

ZAHVALNICA

Autori ovog rada se srdačno zahvaljuju profesorima i saradnicima sa Katedre za mikroelektroniku na Elektronskom fakultetu u Nišu.

Posebna zahvalnost asistentu Milošu Marjanoviću koji nam je obezbedio sav potreban materijal za izradu ovog projekta kao i sugestije koje nam je uputio tokom izrade istog.

LITERATURA

[1] “*DC* motor”, Dostupno na: https://www.automatika.rs/baza-znanja/mehatronika/dc-elektromotori.html

[2] “*STM32 Nucleo-64*”, Dostupno na: https://www.st.com/en/evaluation-tools/nucleo-f401re.html

[3] “Integrisano kolo L293D”, Dostupno na: https://www.ti.com/lit/ds/symlink/l293d.pdf?ts=1664721885581&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FL293D

[4] “STM32CubeMX”, Dostupno na: https://www.st.com/en/development-tools/stm32cubemx.html

[5] “*Keil uVision ‘’,* Dostupno na: https://www2.keil.com/mdk5/uvision/

[6] A. Prijić, Poluprovodničke komponente, Elektronski fakultet Niš, 2021. Dostupno na: http://mikro.elfak.ni.ac.rs/wpcontent/uploads/Poluprovodničke-komponente-diode-2021-1.pdf

[7] “EasyEDA”, Dostupno na: https://easyeda.com

Web aplikacija za sortiranje i rangiranje studenata po studijskim grupama

Nikola Simić

Sadržaj – U ovom radu je opisana ideja, princip rada i struktura veb aplikacije RADB (*Računarstvo i automatika database*) koja služi za sortiranje i rangiranje studenata po studijskim grupama. Aplikacija je napisana u programskom jeziku *Python* u *Django* random okviru i skladišti podatke u *PostgreSQL* bazi podataka.

I. Uvod

Na velikom broju fakulteta postoji dodatno deljenje unutar samih smerova gde se studenti opredeljuju u kojoj oblasti tačno žele da se usavršavaju – opredeljuju se za studijsku grupu na svom smeru. Ako je broj mesta na nekoj studijskoj grupi ograničen studenti se rangiraju na osnovu kriterijuma propisanih od strane samog fakulteta. Primer možemo videti na Fakultetu tehničkih nauka u Novom Sadu – konkretno na smeru Računarstvo i automatika, popularnom *E2*. Studenti po završetku druge godine biraju dve od tri studijske grupe koje žele da pohađaju u trećoj i četvtroj godini. Odluka mora da se donese do kraja meseca juna, a rezultati o upisanoj grupi izlaze tek sredinom septembra iz razloga što položeni ispiti u junskom, julskom i avgustovskom ispitnom roku utiču na rangiranje. Ovde se stvara ideja o aplikaciji na koju studenti mogu da se registruju, unesu svoje podatke o ispitima i ocenama kao i odabrane studijske grupe i budu sortirani i rangirani na tabelama sa ostalim prijavljenim studentima. Studenti tako mogu da vide svoju trenutnu poziciju na rang listama, kao i da vide koji ispiti će im koliko pomoći u njihovom cilju – upisivanju željene studijske grupe.

Backend aplikacije je izrađen u *Django* [1] radnom okviru. *Django* [1] je veb radni okvir visokog nivoa, otvorenog koda napisan u *python* [2] programskom jeziku. Prednosti *django* [1] radnog okvira su skalabilnost aplikacija, brzina razvijanja samih aplikacija kao i *django ORM - Object-Relational Mapper* [3] koji značajno olakšava komunikaciju sa sistemom za upravljanje bazama podataka. *Django* [1] takođe poseduje ugrađen sistem za verifikaciju korisnika koji pruža visoki stepen bezbednosti. *Frontend* je izrađen uz pomoć *Django template* jezika [4] koji na jednostavan način pomaže sa formatiranjem i prikazom sadržaja na stranici. Pored *Django template* jezika [4], korišćeni su *HTML, CSS i JavaScript* koji obezbeđuju željeni izgled aplikacije.

N. Simić je student treće godine na smeru Računarstvo i automatika, Fakultet tehničkih nauka Novi Sad, Univerzitet u Novom Sadu, Trg Dositeja Obradovića 6, 21102 Novi Sad, Srbija, E-mail: nikola7simic@gmail.com.

II. Struktura aplikacije i princip rada

Struktura same veb aplikacije se sastoji od dve celine, odnosno dve zasebne *django*[1] aplikacije – *data* i *user_auth* (Slika 1). Aplikacija *user_auth* služi za registraciju, verifikaciju i autentifikaciju korisnika. Koristeći *djangov Object-Relational Mapper* [3] model korisnika je namapiran na odgovarajuća polja tabele koje je *ORM* kreirao u bazi podataka na osnovu definisanog modela. Korisnik se registruje sa svojim univerzitetskim imejlom koji će služiti kao njegovo korisničko ime.



Slika 1. Struktura aplikacije.

Pored standardne zaštite prilikom registrovanja korisnika koja podrazumeva da korisnik mora da unese validan univerzitetski imejl i dovoljno jaku šifru (minimum šest karaktera) dva puta, nakon uspešne registracije korisnik dobija link za aktivaciju naloga na imejl koji je uneo prilikom registracije (Slika 2). Korisnik koji nema aktiviran nalog ne može da izvrši prijavu na aplikaciju. Ovako pružamo zaštitu od korisnika koji žele maliciozno da iskoriste tuđi univerzitetski imejl kako bi se registrovali na aplikaciju. *User_auth* celina takođe služi za kontrolu prilikom prijave kao i za podnošenje zahteva za novi aktivacioni link ili promenu lozinke.



Slika 2. Imejl sa linkom za aktivaciju naloga.

Aplikacija *data* služi za kontrolu i prikaz podataka o korisnikovim odabranim studijskim grupama, položenim ispitima i ocenama, kao i prikaz početne i „O projektu“

stranice. Pristup stranici sa korisničkim profilom i stranici sa rang listama je zabranjen korisnicima koji nisu prijavljeni. Konačni rezultat po kojim se studenti rangiraju se računa na osnovu formule propisane od strane fakulteta, a koja je implementirana u *norm.py* (Slika 3) pomoćnom modulu koji aplikacija *data* koristi.

```
def remap(value, min_value, max_value):
    if value < min_value:
        return 0
    return round(value/max_value, 4)


def return_score(espb, prosek, indeks):

    godina = int(indeks.split('/')[1])
    #ako je student rasx/2021 u drugoj godini racunamo da se prebacio nas smer a da se upisao 2020
    if godina == 2021:
        godina = 2020

    #za donje granice stavljamo donja_granica-1 zato sto nacemo da student za dovoljnim minimalnim scorom donje scora 0
    espb_score = remap(espb, 71, 120)
    prosek_score = remap(prosek, 6.00, 10.00)
    godina_score = remap(godina, 2020, 2016, 1, 0)

    return round((espb_score*prosek_score*godina_score), 4)
```

Slika 3. Računanje konačnog rezultata za rangiranje.

Pored *data* i *user_auth* sekcija kreirani su posebni moduli kao što je već spomenuti *norm.py* koji sadrže odvojene funkcionalnosti koje aplikacije implementiraju po njihovoj potrebi. *Modulfunctions.py* modul sadrži funkcije koje šalju upite bazi podataka koji treba da vrate studente rangirane po studijskoj grupi u odnosu na njihove konačne rezultate. *Decorators.py* modul implementira funkciju koja proverava da li je korisnik prijavljen i ne dozvoljava pristup stranicama za registraciju i prijavu ako jeste (korisnik ne može da se prijavljuje ili registruje ako je već prijavljen). Poslednji pomoćni modul, *views.py*, služi za obradu odgovora servera tipa 404 i 500. Odgovor servera sa kodom 404 označava da server nije pronašao resurs koji je klijent zatražio, dok odgovor sa kodom 500 označava neobrađenu grešku unutar servera koja je sprečila normalan odgovor servera.

Podaci se čuvaju u *PostgreSQL* [5] objektno-relacionoj bazi podataka. *PostgreSQL* [5] je sistem za upravljanje bazama podataka otvorenog koda, koji je poznat po svojoj stabilnosti i pouzdanosti. Hosting servis na kom se hostuje ova veb aplikacija – *Heroku* [6] koristi ovaj sistem za upravljlanje bazama podataka.

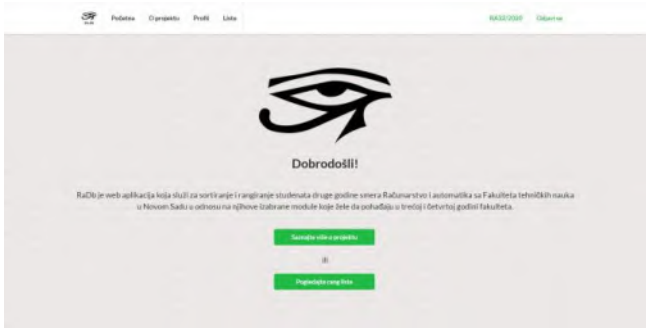
III. KORISNIČKI INTERFEJS

A. Korisnički prikaz

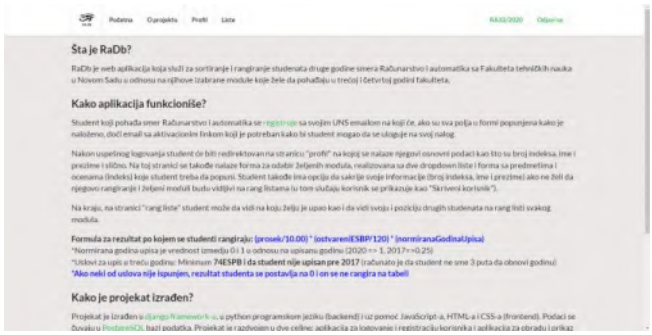
Korisniči interfejs je kreiran tako da bude jednostavan i intuitivan kako korisnik ne bi imao problema prilikom navigacije i korišćenja sajta.

Prilikom pristupa sajtu korisnik se nalazi na početnoj stranici (Slika 4) jednostavnog izgleda sa kratkim opisom koji objašnjava čemu aplikacija služi i sa dva dugmeta koji ga upućuju na stranicu za prijavu ili na stranicu na kojoj može da sazna više o projektu. U koliko je korisnik prijavljen, dugme za prijavu se menja u dugme koje vodi korisnka na rang liste studijskih grupa.

Stranica „O projektu“ (Slika 5) sadrži detaljan tekst o tome kako aplikacija funkcioniše i uvek je dostupna na traci za navigaciju koja je se nalazi na vrhu svake stranice. Pored teksta kako aplikacija funkcioniše, nalazi se i kratak tekst koji opisuje kako je projekat izrađen kao i kontakt na koji mogu da se obrate ako imaju neke nedoumice ili probleme sa aplikacijom.



Slika 4. Početna stranica.

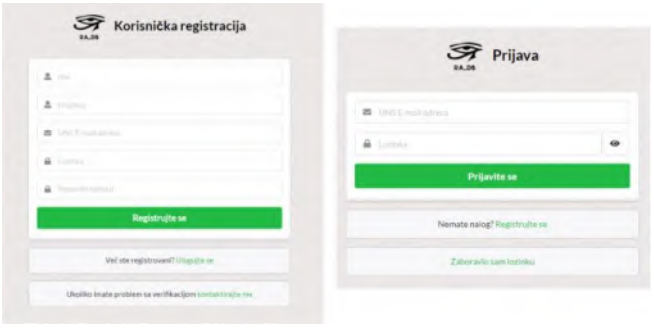


Slika 5. Stranica „O projektu“.

Ako korisnik nije ulogovan na traci za navigaciju će u desnom uglu biti prikazana dugmad koja vode do stranica za prijavu i registraciju korisnika.

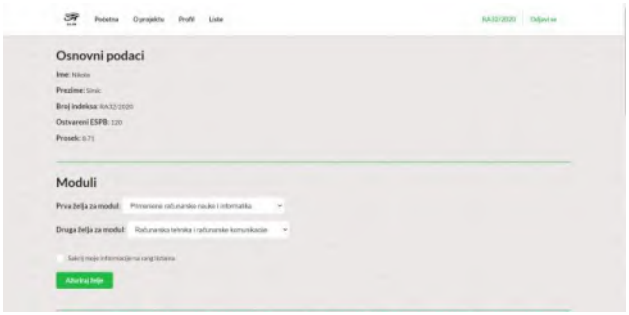
Stranica za registraciju korisnika u svojoj formi (Slika 6) pored polja za imejl i šifru takođe ima polja za ime i prezime korisnika. Potrebno je uneti podatke u svako polje u suprotnom korisnik neće moći da izvrši registraciju. U slučaju uspešne registracije korisniku se na vrhu forme prikazuje poruka koja ga informiše da je na njegov imejl pristigao aktivacioni link kojem mora da pristupi kako bi mogao da se prijavi na aplikaciju. Stranica za prijavu korisnika pored forme za prijavu (Slika 6) na dnu ima linkove do stranice za registraciju u slučaju da korisnik nema registrovani nalog i link koji vodi do stranice za slanje zahteva za promenu lozinke. Ako korisnik koji nije aktivirao svoj nalog pokuša da se prijavi poruka će ga obavestiti da njegov nalog mora da se aktivira, a na dnu forme će se pojaviti link za zahtev za novi imejl sa aktivacionim linkom.

Po uspešnoj prijavi korisnik je preusmeren na stranicu svog profila. Stranica ima tri sekcije poređane jedna ispod druge. Prva sekcija (Slika 7) sadrži osnovne podatke studenta – njegovo ime, prezime, indeks, broj ostvarenih esp bodova i njegov prosek. Broj ostvarenih espb i prosek studenta se menjaju svaki put kad student ažurira svoje položene ispite i ocene u sekciji „Indeks“.



Slika 6. Forme za registraciju i za prijavu korisnika.

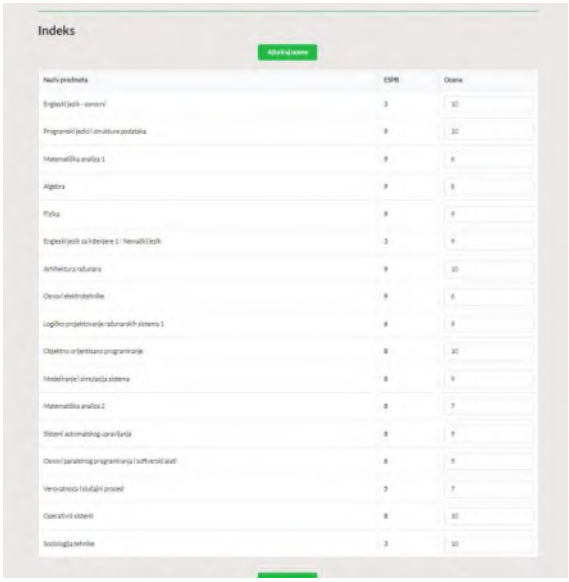
Druga sekcija (Slika 7) sadrži formu za izbor modula (studijskih grupa). Forma je realizovana sa dve spregnute *dropdown* liste – korisnik ne može da izabere istu grupu za svoju prvu i drugu želju. Na formi se takođe nalazi opcija koja na rang listama sakriva korisnikovo ime, prezime i indeks ako on to želi.



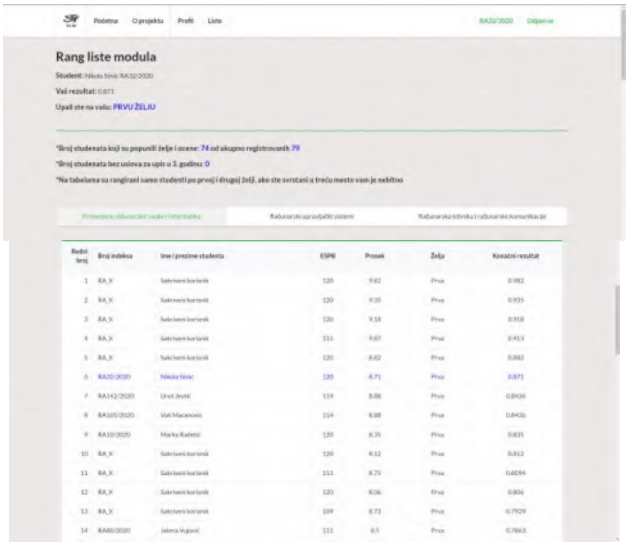
Slika 7. Sekcije sa osnovnim podacima i formom za izbor studijskih grupa na stranici korisničkog profila.

Poslednja sekcija – „Indeks“ (Slika 8) u tabeli sadrži sve predmete iz prve dve godine sa smera računarstvo i automatika, njihove espb i na kraju ocenu koju je student postigao na ispitu. Ako je polje za ocenu (koje može da se ažurira) prazno taj ispit se računa kao nepoložen.

Nakon što korisnik izabere i prvu i drugu želju za studijsku grupu koju želi da pohađa na stranici „Liste“ (Slika 9) moći će da vidi da li je upisao neku od njegove 2 želje i tabele rangiranih studenata na sve tri studijske grupe. Na vrhu stranice se nalaze podaci o korisnikovom konačnom rezultatu i informacija koju želju je upisao. Ispod toga se nalazi informacija o trenutnom broju registrovanih korisnika i broju korisnika koji je zapravo popunio svoje želje i indeks. Takođe je prikazan i broj registrovanih studenata koji nemaju uslov za upis u treću godinu. Na kraju informacija se nalazi meni u kojem korisnik bira za koju studijsku grupu želi da vidi rang liste. Ako je uspešno rangiran, korisnikov red na rang listi biće ispisan plavim slovima kako bi mogao lakše da se pronađe.



Slika 8. Sekcija „Indeks“ na stranici korisničkog profila.



Slika 9. Stranica sa rang listama studijskih grupa.

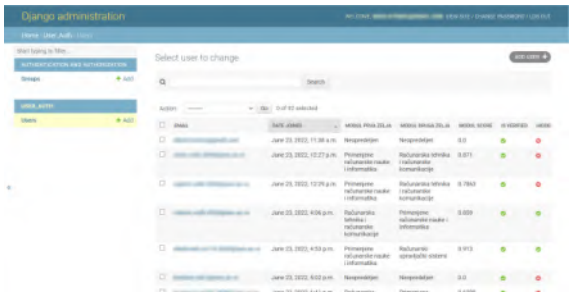
Obezbeđena je komunikacija sa korisnikom kroz povratne informacije u vidu poruka (Slika 10) koje se prikazuju na vrhu stranice kada korisnik šalje zahteve kao što su zahtevi za novu lozinku ili novi verifikacioni imejl ili kada vrši izmene svojih podataka kroz forme na stranici.



Slika 10. Komunikacija sa korisnikom kroz poruke.

B. Administratorski prikaz

Django [1] automatski kreira administratorski panel, ali taj panel na početku nema opciju dodavanja, brisanja ili izmena podataka dok se oni ručno ne implementiraju. Administratoru je omogućen rad sa unapred određenim podacima vezanim za samog korisnika, da ih po potrebi dodaje, briše, ili da pravi izmene. Administratorski panel (Slika 11) nema pristup menjaju korisničkih podataka bitnih za prijavu na sajt kao što su njegov imejl i šifra, ali ima pristup ručnoj aktivaciji naloga u slučaju da korisnik nema pristup imejlu ili nije dobio svoj aktivacioni imejl. Pristup administratorskom panelu ima samo poseban administratorski nalog, a putanja do panela je promenjena sa uobičajne */admin* putanje čime je povećana sigurnost.



Slika 11. Administratorski panel.

IV. Rezultati

U periodu od 24. juna 2022. godine, kada je aplikacija predstavljena studentima druge godine smera računarstvo i automatika na Fakultetu tehničkih nauka u Novom Sadu, do 20. septembra 2022. godine, kada je fakultet obavestio studente koju studijsku grupu su upisali, na sajt je registrovano 90 studenata od kojih je 79 aktiviralo svoj nalog i popunilo svoje želje i položene ispite. To je približno 33% od ukupnog broja studenata na smeru (oko 240). Studenti smera su samo jednom obavešteni o aplikaciji u tri velike grupe na aplikacijama koje koriste - *Facebook*, *Whatsapp* i *Discord*. Pretpostavka je da su studenti ponovo obavešteni, barem još jednom u toku julskog i avgustovskog ispitnog roka, približno bi 50% studenata smera bilo registrovano na sajt, što pokazuje da postoji potražnja za ovim servisom. Takođe je primećeno da je 32 od 79 studenata izabralo opciju koja sakriva njihovo ime, prezime i indeks sa rang listi.

V. Zaključak

Aplikacija uspešno izvršava svoju funkciju i pruža studentima značajne informacije koje mogu da iskoriste da povećaju svoju šansu za upis željene studijske grupe. Trenutno aplikacija radi samo za studente smera računarstvo i automatika na Fakultetu tehničkih nauka u Novom Sadu, ali ovaj koncept je primenjiv na bilo koji smer koji ima neki vid podele na studijske grupe.

Unapređenja predviđena za sledeći period: Kreirati novi korisnički interfejs koristeći *Angular* [7] radni okvir i obezbediti lepši prikaz sajta prilikom pristupa sa prenosivih uređaja kao što su mobilni telefoni i tableti.

Zahvalnica

Zahvaljujem se svojim prijateljima na konstruktivnim kritikama i pomoći oko testiranja aplikacije, kao i profesoru iz srednje škole, Ušan Branislavu, koji me je motivisao da napišem ovaj rad.

Dodatak

Prijava na razvijenu aplikaciju je omogućena kroz javnog korisnika. Podaci za prijavu javnog korisnika su: imejl: testuser.ra0.0@uns.ac.rs, šifra: radbTSTuser123. Aplikacija je dostupna na adresi: radb-dx.herokuapp.com.

Literatura

[1] Django, „Django dokumentacija“: https://docs.djangoproject.com/en/4.1/, datum pristupa sajtu: 06.10.2022. godine.
[2] Python: https://www.python.org/, datum pristupa sajtu: 06.10.2022. godine.
[3] Django ORM: https://opensource.com/article/17/11/django-orm, datum pristupa sajtu: 06.10.2022. godine.
[4] Django template language: https://docs.djangoproject.com/en/4.1/ref/templates/language/, datum pristupa sajtu: 06.10.2022. godine.
[5] PostgreSQL: https://www.postgresql.org/docs/, datum pristupa sajtu: 06.10.2022. godine.
[6] Heroku: https://devcenter.heroku.com/, datum pristupa sajtu: 06.10.2022. godine.
[7] Angular: https://angular.io/docs, datum pristupa sajtu: 06.10.2022. godine.

Karakterizacija gasnih komponenata proizvođača *Littelfuse*

Nemanja Tomić

Sadržaj − U ovom radu opisuje se deo eksperimentalnih rezultata dobijenih merenjem probojnog napona i vremena kašnjenja komercijalnih gasnih odvodnika proizvođača Littelfuse na radnom naponu 110 i 145 V. Eksperimenti su odrađeni u Laboratoriji za vakuumsku i gasnu tehniku na Elektronskom fakultetu. Glavni cilj ovog rada je da se izvrši karakterizacija gasnih odvodnika proizvođača Littelfuse na radnom naponu 110 i 145 V. Karakterizacija ovih gasnih komponenata podrazumeva u ovom slučaju određivanje statičkog probojnog napona, kao i vremena odziva ovih komponenata. Osnovu ovog rada čine rezultati više eksperimenata koji su omogućili procenu ovih fizičkih veličina. Detaljno je opisana procedura rada, kao i oba sistema za merenje koja su tom prilikom korišćena. Za procenu statičkog probojnog napona korišćena je dinamička metoda. Provera postojanja memorijske krive omogućila je određivanje vremena odziva komponenata za različite vrednosti primenjenog napona na komponente. Određivanje ovih veličina i upoređivanje sa karakteristikama komponenata iz tehničke dokumentacije omogućuje procenu pouzadnosti komponenata, kao i mogućnost njihove dalje primene.

I. UVOD

Razvoj elektronike i računarske tehnike, koji se naročito može videti u trenutnoj težnji da se sve komponente minijaturizuju zbog krajnjeg smanjenja uređaja, odnosno zbog sve većeg stepena integracije elektronskih komponenata, ukazao je na veoma veliku važnost njihove zaštite od prenapona. Problem koji je nastao usred minijaturizacije elektronskih komponenata je upravo smanjena otpornost na pojavu prenapona. Zato je veoma bitno posvetiti što veću pažnju problemu prenaponske zaštite komponenata. Efikasna zaštita od prenapona može zaštititi komponentu, kao i celo elektronsko kolo, od oštećenja koja ponekad mogu biti i trajna. Takođe, dolazi i do javljanja šumova koje je posle teško ukloniti. Gasni odvodnici prenapona (slika 1) su jedni od najboljih i najčešće primenjivanih nelinearnih elemenata za zaštitu od prenapona na niskonaponskom nivou. Njihov rad se zasniva na procesima koji dovode do električnog proboja i pražnjenja. Gasni odvodnici se koriste za zaštitu od prenapona u opsegu od 70 do 1200 V uz prednost u mogućnosti provođenja visokih struja (> 5 kA) i maloj kapacitivnosti (< 1 pF). Stabilnost njihovih karakteristika je posebno važna u specifičnim uslovima rada [1].

N. Tomić je student master studija na modulu Elektronika i mikrosistemi, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: n.tomic@elfak.rs.

Gasni odvodnici, danas, nalaze sve veću primenu u vojnoj industriji i svemirskoj tehnologiji.



Slika 1. Gasni odvodnik sa dve i tri elektrode [1].

Korišćeni uzorak u našim eksperimentima bio je gasni odvodnik proizvođača *Littelfuse*. Uopšteno gledano, gasni odvodnici se sastoje od dve ili tri elektrode koje su zatopljene u keramičko ili stakleno kućište. Elektrode, koje su ravne i paralelne kako bi se u međuelektrodnom prostoru obezbedilo približno homogeno električno polje, se nalaze u kućištu koje se sastoji od keramike i metala. Rastojanje između elektroda je reda milimetra ili dela milimetra. Kao izolacioni materijal koriste se ili plemeniti gasovi (argon, neon, kripton ili ksenon) ili njihove smeše na pritiscima od 0.1 kPa do 70 kPa.

Nekada su u upotrebi bile dve vrste gasnih odvodnika prenapona, i to oni bez ugrađenog radioaktivnog izvora i sa ugrađenim radioaktivnim izvorom. Radioaktivni izvor obezbeđuje povećani broj jonskih parova u međuelektrodnom prostoru što dovodi do bržeg reagovanja odvodnika na pojavu prenapona [2]. Danas se svi gasni odvodnici izrađuju bez radioaktivnog izvora. Gasni odvodnici prenapona sa tri elektrode predstavljaju dva gasna odvodnika prenapona u istom kućištu.

II. TEORIJSKI OSVRT

Prenapon je pojava koja definiše stanje gde je potencijal jedne tačke komponente, uređaja ili voda u odnosu na neku drugu tačku (npr. tačku nultog potencijala) veći od dozvoljenog. Takvi prenaponi iznad određene vrednosti mogu dovesti do ugrožavanja bezbednosti ljudi koji rukuju uređajima, kao i do oštećenja samih uređaja.

Ako se sporo povećava jednosmerni napon na ravne paralelne elektrode postavljene na rastojanju *d* u gasu čiji je pritisak *p*, koncentracija atoma ili molekula gasa *n*, pri nekoj kritičnoj vrednosti napona javiće se električni proboj. Proboj se karakteriše brzim prelaskom gasa od lošeg provodnika električne struje sa otpornošću reda 10^{14} Ωm u relativno dobar provodnik čija je otpornost veoma mala (za slučaj tinjavog pražnjenja ona je reda 10^{3} Ωm).

Napon na kome se javlja prelazno stanje naziva se probojni napon i njegova vrednost, pored ostalog, zavisi od vrste i pritiska gasa, materijala elektroda i stepena prethodne jonizacije [3].

Do pojave električnog proboja u gasu i samostalnog pražnjenja dolazi tek onda kada su vrednosti električnog polja *E*, pritiska gasa *p* i međuelektrodnog rastojanja *d* takve da svaki elektron koji se oslobodi sa katode izazove sekundarne procese pri kojima će njega zameniti novi elektron sa katode. Dakle, do proboja dolazi samo onda kada je porast broja elektrona formiranih jonizacijom veći od njihovog gubitka usled difuzije, pripajanja ili rekombinacije. Tokom proboja, tj. pri prelasku nesamostalnog u samostalno pražnjenje, dolazi do naglog (teorijski beskonačnog) porasta struje. Ova struja praktično raste do neke konačne vrednosti, ograničene impedansom spoljašnjeg kola i konačnom snagom izvora.

Električni proboj gasa može biti statički ili dinamički zavisno od oblika primenjenog napona. Ako je brzina promene napona jednaka ili veća od brzine promene elementarnih procesa u gasu, proboj se naziva dinamičkim. Statički (jednosmerni) proboj nastaje kada je brzina promene napona mnogo manja od brzine promene elementarnih procesa koji dovode do proboja. Statički proboj se može odvijati Townsend-ovim ili strimernim mehanizmom. Townsend-ov mehanizam proboja se pojavljuje kada su sekundarni procesi na katodi dominantni u odnosu na sekundarne procese u gasu. U protivnom se javlja strimerni mehanizam proboja.

Napon pri kojem se javlja električni proboj gasa naziva se probojni napon. U slučaju statičkog proboja ovaj napon se naziva statički probojni napon (eng. *DC spark-over voltage*) koji je deterministička veličina i njegova vrednost zavisi od proizvoda *pd*. U slučaju dinamičkog proboja, napon pri kojem se on javlja naziva se dinamički (eng. *impulse spark-over voltage*) i stohastička je veličina[4].

Ono što je karakteristično za gasove je upravo njihova veća inertnost pri proticanju električne struje u odnosu na provodnike. Mnoga eksperimentalna istraživanja u gasovima su pokazala da se proboj ne dešava u trenutku priključivanja napona, čak i onda kada je inicijalni elektron oslobođen sa katode, a priključeni napon veći od statičkog probojnog napona U_s. To znači da se proboj dešava posle određenog vremena koje se naziva vreme kašnjenja električnog proboja. Ova pojava kašnjenja električnog proboja se javlja kod svih gasova. Jedna od mnogih definicija vremena kašnjenja električnog proboja t_d je da je to vreme koje protekne od momenta kada napon priključen na elektrode cevi dostigne vrednost statičkog probojnog napona, pa do momenta kada zbog proboja njegova vrednost počne da se smanjuje. U slučaju ovog eksperimenta, ovo je vreme koje protekne od momenta priključenja radnog napona na elektrode komponente, pa do pojave električnog proboja. Podrazumeva se da je vreme porasta napona zanemarljivo malo u odnosu na merena vremena t_d. Osim toga potrebno je da unutrašnji otpor izvora napona bude zanemarljivo mali.

Vreme kašnjenja električnog proboja sastoji se iz statističkog vremena kašnjenja t_s i vremena formiranja pražnjenja t_f.

III. OPIS EKSPERIMENATA

Komponente koje su korišćene u eksperimentima kao uzorci pri merenju probojnog napona i vremena kašnjenja su gasni odvodnici proizvođača *Littelfuse* radnog napona 110 i 145 V.

Prilikom izbora savršene komponente za dalji dizajn određenog kola, potrebno je da ona zadovolji određene zahteve koje potrošač zadaje proizvođaču. Karakteristike koje su bitne za gasne odvodnike i koje se obavezno nalaze na tehničkoj dokumentaciji komponente su: statički probojni napon, dinamički probojni napon meren na različitim naponima, životni vek komponente, otpornost izolacionog dela komponente, kapacitivnost na određenoj frekvenciji (oko 1 MHz) za koju je bitno da bude što manja, kao i radna temperatura komponente.

Ono čime *Littelfuse* gasni odvodnici navode da se odvajaju od drugih su upravo to što imaju stabilan rad u toku radnog veka, imaju brz odziv, malu parazitnu kapacitivnost i visoku otpornost izolacionog materijala.

U zavisnosti od naših zahteva biramo komponente sa različitim nivoom probojnog napona, a tokom eksperimenata je odrađen veliki broj merenja nad gasnim odvodnikom proizvođača *Littelfuse.* Ova komponenta ima definisani nominalni probojni napon, čija se devijacija takođe određuje u ovom radu. Ovim eksperimentom urađena je analiza uz mogućnost upoređivanja dobijenih rezultata sa karakteristikama iz njihove tehničke dokumentacije.

Tehničke karakteristike korišćenih *Littelfuse* gasnih odvodnika prenapona su: DC probojni napon, tipična vrednost DC proboja, napon luka, izolatorska otpornost, kapacitivnost i nominalna impulsna struja pražnjenja dati su u tabeli I.

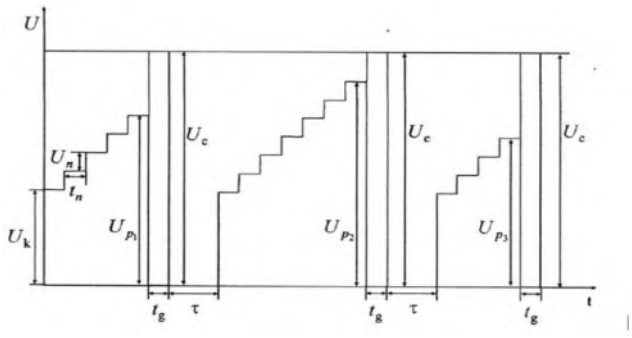
TABELA I
SPECIFIKACIJE KORIŠĆENIH GASNIH ODVODNIKA PRENAPONA IZ TEHNIČKIH DOKUMENTACIJA NA TEMPERATURI OD 25 ^{O}C [5]

Komponenta	DC probojni napon (V)	DC tipicni probojni napon (V)	Napon luka (V)	Izolatorksa otpornost na 50V (Ω)	Kapacitivnost (pF)	Nominalna impulsna struja praznjenja (x5@8/20μs) (kA)
LF CG145	116-174	145	15	1010	1.5	20
LF CG110	88-132	110	15	1010	1.5	20

Određivanje probojnog napona U_p najčešće se vrši u uslovima stepenastog porasta napona na elektrodama gasne komponente i uslovima linearnog porasta napona. Na komponentu se u prvom slučaju dovodi fiksni napon U_k koji je znatno manji od očekivanog probojnog napona U_p.

Nakon toga napon biva povećan za određeni naponski korak U_n koji ima određeno vreme trajanja t_n. Odnos $k = U_n/t_n$ predstavlja brzinu porasta napona. Kada nastane finalni proboj naponom U_c, kroz komponentu je obezbeđena struja pražnjenja za određeno vreme pražnjenja t_g kako bi se u cevi uspostavila stacionarna koncentracija naelektrisanih i pobuđenih čestica. Napon na elektrodama gasne komponente se nakon toga isključuje za vreme τ koje se naziva vreme relaksacije. Kako bi rezultati eksperimentalno izmerenog napona proboja gasne komponente bili što tačniji, bitno je da se korak povećanja napona U_n ne menja, kao ni vreme između dva naponska koraka. Struja pražnjenja i vreme relaksacije se takođe treba održavati konstantnim [6].

Kada istekne zadato vreme relaksacije proces počinje od početka kako je to već opisano. Ovaj proces prikazan je na slici 2.



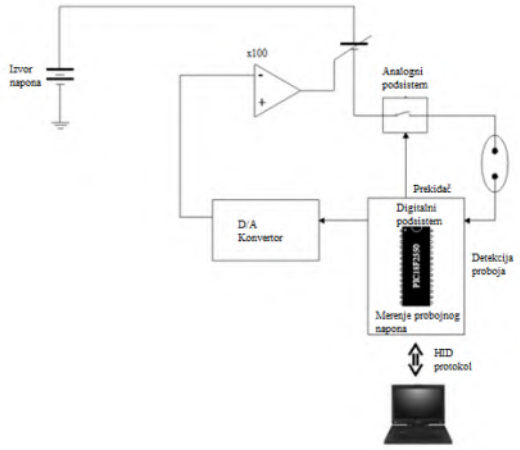
Slika 2. Stepenasti porast napona na gasnoj komponenti za seriju od tri uzastopna merenja probojnog napona [3].

Prilikom određivanja probojnog napona dinamičkom metodom, u seriji ponovljenih merenja pri istim uslovima, dobijene vrednosti probojnog napona biće različite [7]. To pokazuje da se proboj dešava sa određenom verovatnoćom zbog čega je probojni napon U_p veličina statističkog karaktera sa određenom raspodelom.

Za realizaciju eksperimenata merenja probojnog napona, korišćena je sledeća oprema (slika 3):

1. Računar, preko kojeg se podešavaju parametri, aktivira softver i prikupljaju podaci;
2. DC izvor napona, stepenasti generator napona, digitalni podsistem.

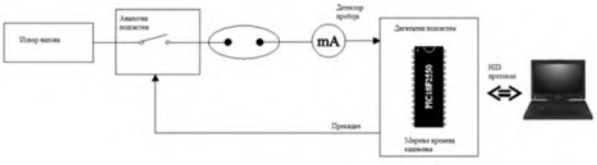
Sistem za merenje probojnog napona, prikazan na slici 3, sastoji se od analognog i digitalnog podsistema. Analogni podsistem sastoji se od DC izvora napona i stepenastog generatora napona. Stepenasti generator je projektovan kao naponski pojačavač sa x100 pojačanjem. Digitalni podsistem sadrži mikrokontroler MICROCHIPs PIC18F2550 i D/A konvertor. Napon iz D/A konvertora se pojačava 100 puta i oduzima od napona na izlazu DC izvora. Napon pri kojoj vrednosti dolazi do proboja se čuva u memoriji. Nakon toga, gasni odvodnik je isključen za unapred definisanu vrednost vremena relaksacije τ. Nakon ovog vremena, počinje novo merenje i ova procedura se nastavlja sto ciklusa sa unapred definisanom vrednošću brzine porasta napona.



Slika 3. Blok dijagram sistema za merenje probojnog napona.

Kao što je već pomenuto, do električnog proboja u gasovima ne dolazi odmah posle priključivanja napona čak i kad je on znatno veći od statičkog probojnog napona, već posle izvesnog vremena koje se naziva vreme kašnjenja električnog proboja t_d. S obzirom na statističku prirodu procesa u gasu, vreme kašnjenja električnog proboja, kao i probojni napon, je veličina statističkog karaktera. Prilikom dobijanja mnogo podataka vezanih za t_d pored održavanja konstantnog vremena relaksacije τ potrebno je održavati konstantnim i vreme trajanja pražnjenja t_g i radni napon. Upravo je ovo kašnjenje glavni problem prilikom korišćenja ovih gasnih komponenata.

Pojava memorijske krive predstavlja neželjeni efekat kod gasnih odvodnika, i taj efekat nije prisutan kod izložene komponente. Sama memorijska kriva predstavlja zavisnost srednje vrednosti vremena kašnjenja u funkciji vremena relaksacije. Merenja su izvršena preko softvera koji naizmenično prikazuje svako uneto vreme relaksacije i vreme žarenja. Blok dijagram sistema koji se koristi za merenje vremena kašnjenja električnog proboja kod gasnih komponenata prikazan je na slici 4.



Slika 4. Blok dijagram sistema za merenje vremena kašnjenja.

Struktura sistema se može podeliti na dva odvojena podsistema:

1. Analogni i
2. Digitalni.

Namena analognog podsistema je da obezbedi brzo i tačno uključivanje napona na gasnom odvodniku prenapona. Digitalni podsistem je, sa druge strane, odgovoran za prikupljanje i smeštanje izmerenih podataka, kao i kontrolu izvršenog merenja.

Procedura merenja je sledeća. Serija impulsa konstantnog napona se primenjuju na komponentu, i meri se vreme koje protekne između primenjenog naponskog impulsa i pojave električnog proboja, tj. trenutka kada struja tačno određenog intenziteta protekne kroz komponentu. Izmerene vrednosti vremena kašnjenja t_d se čuvaju u memoriji računara, a naponski impuls se održava na komponenti tokom tačno definisanog vremenskog intervala, vremena pražnjenja t_g. Da bi se održali isti eksperimentalni uslovi prilikom svakog merenja, gasna komponenta je isključena za jednake vremenske intervale u svakom ciklusu, odnosno tokom vremena relaksacije. Za svako zadato vreme relaksacije mereno je po sto vrednosti vremena kašnjenja električnog proboja. Struja pražnjenja je održavana na 0.5 mA, a vreme pražnjenja je tom prilikom iznosilo 1 s.

IV. Opis metoda

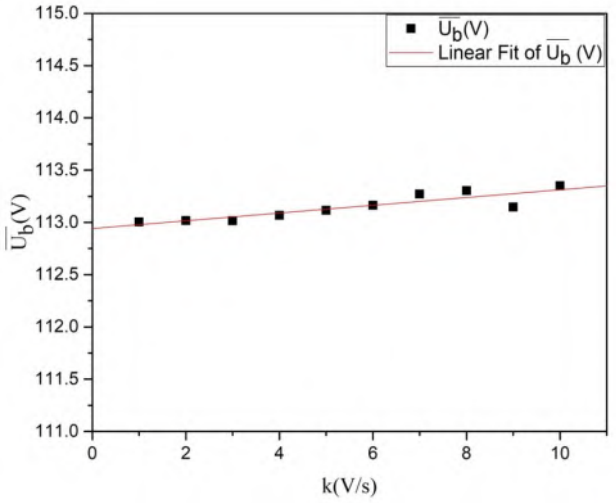
Sam statički probojni napon je upravo početna tačka za sva dalja proučavanja i izračunavanja vezana za probojni napon. On se ne može izjednačiti sa jednosmernim probojnim naponom koji se javlja u tehničkoj dokumentaciji komponente. Kao što je već pomenuto u radu, probojni napon je veličina statističkog karaktera. Zbog toga je prilikom analize proboja u gasu potrebno izvršiti veliki broj merenja (u ovom radu je izvršeno po 100 merenja za svaku brzinu porasta napona) pod istim uslovima i koristiti srednju vrednost U_b. Kod većine gasnih komponenata, kako onih koje rade u oblasti nesamostalnog pražnjenja, tako i kod onih koje rade u oblasti proboja, potrebno je poznavati najmanji napon koji se može dovesti na gasnu komponentu da bi se izazvao proboj.

Druga metoda koja je korišćena u ovom radu zasnovana je na metodi vremena kašnjenja. Osnovu ove metode čini snimanje zavisnosti srednje vrednosti vremena kašnjenja električnog proboja u funkciji vremena relaksacije τ za različite vrednosti prenapona kod komercijalne komponente. Ova grafička zavisnost predstavlja memorijsku krivu. I kod vremena kašnjenja električnog proboja je svaka srednja vrednost računata na osnovu 100 izmerenih vrednosti za svaku zadatu vrednost vremena relaksacije. Vreme odziva se očitava na t_d – osi za svaku vrednost primenjenog napona na komponentu.

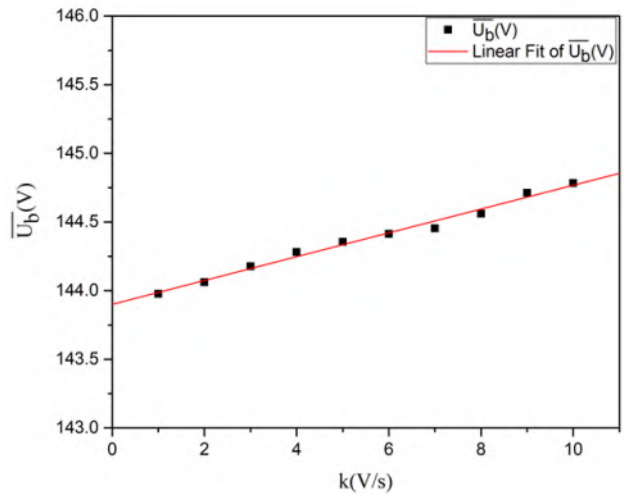
V. Eksperimentalni podaci

Na grafiku prikazanom na slici 5 se može uočiti zavisnost srednje vrednosti probojnog napona u funkciji brzine porasta napona za gasni odvodnik prenapona proizvođača *Littelfuse* na radnom naponu od 110 V.

Svaka srednja vrednost probojnog napona izračunata je na osnovu 100 izmerenih vrednosti. Brzina porasta napona menjana je od 1 do 10 V/s. Na slici se može uočiti blagi porast srednje vrednosti probojnog napona sa povećanjem brzine porasta napona što je u skladu sa teorijskim osnovama. Eksperimentalni podaci su fitovani linernom funkcijom i u preseku prave sa U_b - osom dobija se vrednost statičkog probojnog napona. Procenjena vrednost statičkog probojnog napona za komponentu LF-110 iznosi 112.9 V (slika 5).



Slika 5. Srednje vrednosti probojnog napona u funkciji brzine porasta napona za gasni odvodnik LF-110.



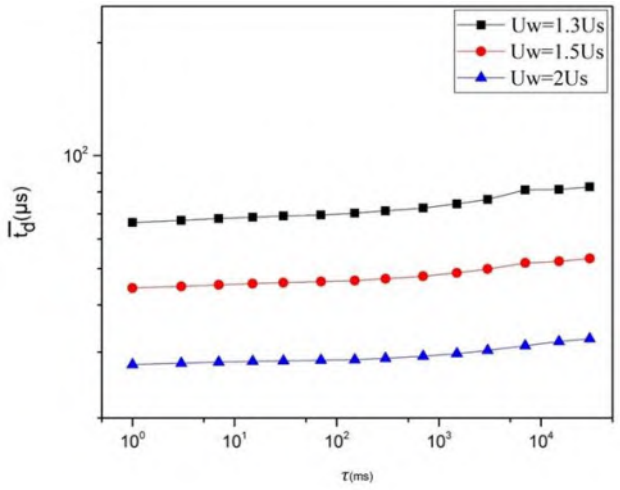
Slika 6. Srednja vrednost probojnog napona u funkciji brzine porasta napona za gasni odvnodnik LF-145.

Na slici 6 prikazani su rezultati procene statičkog probojnog napona za gasni odvodnik prenapona proizvođača *Littelfuse* pri radnom naponu 145 V. Procenjena vrednost iznosi 143.8 V.

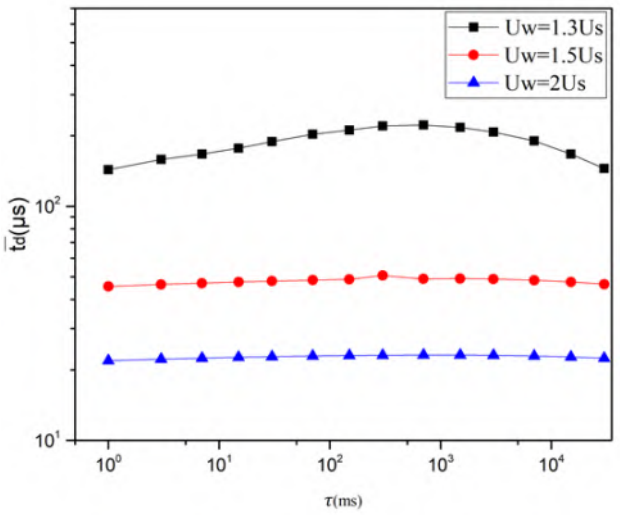
Zaključuje se da su oba tipa gasnih odvodnika vrlo pouzdana po pitanju stabilnosti napona na šta ukazuju

procenjene vrednosti statičkog probojnog napona u poređenju sa vrednostima iz tehničke dokumentacije.

Na slikama 7 i 8 prikazane su zavisnosti srednje vrednosti vremena kašnjenja u funkciji vremena relaksacije za gasne odvodnike proizvođača *Littelfuse* pri radnim naponima od 110 i 145 V, redom. Dobijene srednje vrednosti vremena kašnjenja električnog proboja dobijene su iz skupova od sto izmerenih vrednosti vremena kašnjenja. Vreme relaksacije se tokom eksperimenta menjalo od 1 ms do 30 s. Eksperimentalni rezultati koji su dobijeni ukazuju na stabilnost i visoku pouzdanost ovih gasnih komponenata zato što rezultati ukazuju konstantnost u odnosu na vreme kašnjenja.



Slika 7. Srednje vrednosti vremena kašnjenja u zavisnosti od vremena relaksacije za gasni odvodnik LF-110.



Slika 8. Srednje vrednosti vremena kašnjenja u zavisnosti od vremena relaksacije za gasni odvodnik LF-145

Pored toga, provera je izvršena za tri različite vrednosti prenapona. Prenapon u ovom slučaju podrazumeva koliko je vrednost primenjenog napona veća od procenjene vrednosti statičkog probojnog napona. Eksperimenti su odrađeni za vrednosti prenapona od 30, 50 i 100 %. Naime, ovom prilikom bez obzira na vrednost primenjenog napona na gasnu komponentu nije uočena memorijska kriva koja se inače javlja kod gasnih dioda čime je potvrđena pouzdanost ovih komponenata. Pored toga, potvrđeno je da sa porastom primenjenog napona na elektrode gasnih odvodnika dolazi ranije do proboja, odnosno dolazi do smanjenja vremena kašnjenja električnog proboja bez obzira na vrednost vremena relaksacije.

VI. Zaključak

Gasni odvodnici prenapona se danas sve više koriste kao elementi za zaštitu od prenapona. Rad gasnih odvodnika se zasniva na procesima koji dovode do električnog proboja i pražnjenja u gasu. Izvršen je veliki broj istraživanja koja ispituju različite osobine različitih gasnih odvodnika.

Istraživanja koja su realizovana u okviru ovog rada analiziraju raspodele vremena kašnjenja električnog proboja i probojnog napona gasnog odvodnika prenapona. Analiza je izvršena za komercijalni gasni odvodnik proizvođača *Littelfuse* pri radnim naponima od 110 i 145 V. Procena statičkog probojnog napona izvršena je dinamičkom metodom za deset različitih brzina porasta napona, od 1 do 10 V/s. Procenjene vrednosti statičkog probojnog napona su iz opsega vrednosti datih u tehničkoj dokumentaciji, dok je standardna devijacija jako mala. Time se zaključuje da je komponenta pouzdana za korišćenje. Zaključuje se da su oba tipa gasnih odvodnika vrlo pouzdana po pitanju stabilnosti napona na šta ukazuju procenjene vrednosti statičkog probojnog napona u poređenju sa vrednostima iz tehničke dokumentacije. Procenjena vrednost statičkog probojnog napona za gasni odvodnik *Littelfuse* -110 iznosi 112.9 V, dok je kod gasnog odvodnika *Littelfuse* -145 ta vrednost procenjena na 143.8 V.

Analiza vremena kašnjenja kod ovih komponenata izvršena je za vrednosti vremena relaksacije od 1 ms do 30 s. Kao što se može videti iz priloženih grafika, kod ove komponente se ne javlja memorijska kriva za merenja pri prenaponu većem od 30 %. Do tog zaključka se došlo snimanjem zavisnosti srednje vrednosti vremena kašnjenja u funkciji vremena relaksacije za vrednosti prenapona od 30, 50 i 100 % čime je potvrđena pouzdanost ovih komponenata.

Literatura

[1] Emilija Živanović, Sandra Veljković, Marija Živković, Milić Pejović, "Reliability of Various Type of Gas-filled Surge Arresters Under DC Discharge", Proc. 2019 IEEE 31st International Conference on Microelectronics, MIEL 2019, Niš, Serbia, September 2019, pp. 113-116.

[2] Momčilo M. Pejović, Gasne elektronske komponente – Uvod u električna gasna pražnjenja i proboj gasa, Elektronski fakultet, Univerzitet u Nišu, 2008.
[3] Momčilo M. Pejović, Milić M. Pejović, Električni proboj gasova- merni sistemi i eksperimenti i eksperimentalna istraživanja, Elektronski fakultet, Univerzitet u Nišu, 2009.
[4] Emilija N. Živanović and Čedomir A. Maluckov, "Investigation of statistical behaviour of electrical breakdown voltage distribution for nitrogen-filled diode at 13.3 mbar pressure", *Contributions to Plasma Physics*, Vol. 58, No. 4, pp. 293-301, 2018.
[5] Tehnička dokumentacija gasnog odvodnika: https://es.datasheetq.com/CG145-doc-Littelfuse-2013
[6] Emilija Živanović, Marija Živković, Sandra Veljković, "Study of Breakdown Voltage Stability of Gas-Filled Surge Arresters in the Presence of Gamma Radiation", *Electronics*, Vol. 11, p. 2447, 2022.
[7] Emilija Živanović, Marija Živković, Milić Pejović, "The evolution of breakdown voltage and delay time under high overvoltage for different types of surge arresters", *Facta Universitatis, Series: Electronics and Energetics*, Vol. 34, No. 2 pp. 307-322, 2021.

Praktična realizacija elektronskog kola tranzistorskog dvokvadrantnog čopera u regulaciji brzine motora jednosmerne struje

Ognjen Bunjac, Vid Vuković

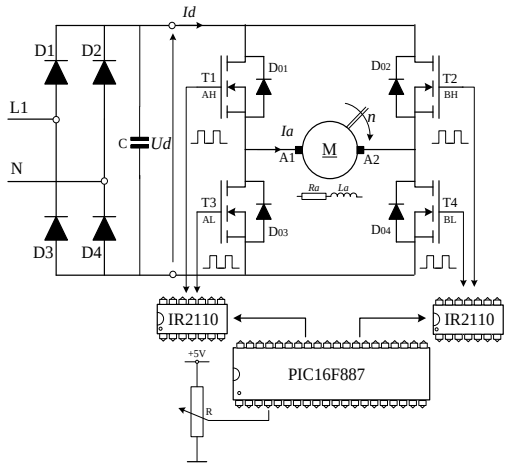
Sadržaj – U ovom radu je prikazan jedan način praktične realizacije tranzistorskog dvokvadrantnog čopera za regulaciju brzine jednosmernih motora sa permanentnim magnetima male snage. Upravljanje je realizovano pomoću mikrokontrolerske kartice i odgovarajućeg algoritma upravljanja. U radu su takođe prikazani i rezultati merenja na hardverskoj realizaciji pretvarača.

I. DVOKVADRANTNI TRANZISTORSKI ČOPER

Razvojem energetske elektronike i mikrokontrolerskih upravljačkih sistema stekli su se preduslovi za razvoj i primenu naprednih tehnika upravljanja električnim motorima [1].

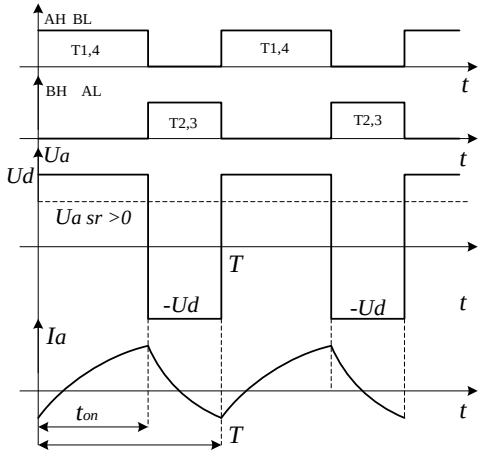
U ovom radu prikazana je hardverska realizacija tranzistorskog čopera u H mostu koji može da radi u dva kvadranta. Na taj način omogućava se regulacija brzine jednosmernih motora u širokom opsegu brzina.

Principski izgled pretvarača prikazan je na slici 1 [2].



Slika 1. Principska šema tranzistorskog čopera u H mostu.

Algoritam upravljanja podrazumeva dijagonalno uključenje tranzistora u H mostu. Pri tome se istovremeno uključuju dijagonalni tranzistori T_1 i T_4, odnosno T_2 i T_3. Parovi tranzistora koji se istovremeno uključuju imaju jednaka vremena uključenja. Vremenski dijagrami pobudnih impulsa obeleženi sa AH, AL, BH, BL, kao i vremenski dijagrami napona i struje motora prikazani su na slici 2.



Slika 2. Karakteristični vremenski dijagrami pobudnih impulasa i napona i struja na pretvaraču.

Srednja vrednost napona na motoru može se odrediti primenom principa volt-sekundnog balansa za dijagram napona $U_a=f(t)$ sa slike 2.

Princip volt-sekundnog balansa pokazuje da je površina koju ograničava napon U_d na periodi T jednaka površini ekvivalentnog pravougaonika čija je jedna stranica jednaka srednjoj vrednosti armaturnog napona motora a druga stranica jednaka je periodi T. Kod ove tehnike upravljanja faktor ispune ćemo definisati kao odnos vremena uključenja (t_{on}) prvog para tranzistora (T_1, T_4) i vremena periode T.

$$U_{a\,sr}T = U_d t_{on} + (-U_d)\cdot(T - t_{on}) \;/:T$$

$$U_{a\,sr} = U_d \frac{t_{on}}{T} - U_d\left(1 - \frac{t_{on}}{T}\right),$$

$$\frac{t_{on}}{T} = a, \qquad (1)$$

$$U_{a\,sr} = U_d a - U_d(1-a) = (2a-1)U_d\,,$$

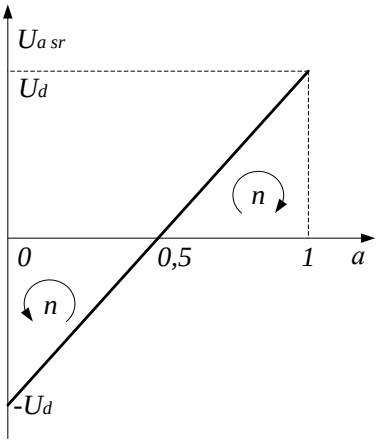
$$U_{a\,sr} = (2a-1)U_d$$

O. Bunjac i V. Vuković su učenici na smeru Elektrotehničar procesnog upravljanja, Elektrotehničke škole "Mihajlo Pupin", Novi Sad, Futoška 17, 21000 Novi Sad, Srbija, E-mail: bunjaco05@gmail.com.

Izvođenje prikazano jednačinama (1) pokazuje da se srednja vrednost napona na rotoru motora menja saglasno izrazu (2).

$$U_{a\,sr} = (2a-1)U_d \tag{2}$$

Izgled regulacione karakteristike promene srednje vrednosti napona u zavisnosti od faktora ispune prikazan je na slici 3.

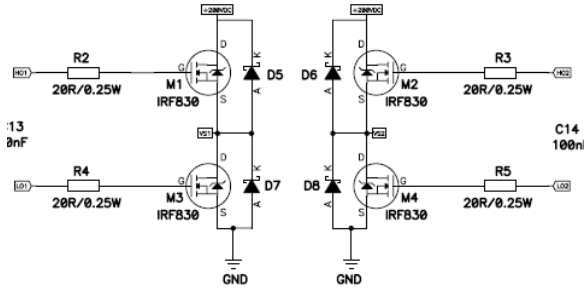


Slika 3. Regulaciona karakteristika pretvarača $U_{a\,sr}=f(a)$.

Važno je naglasiti da prikazana topologija tranzistorskog čopera omogućava i rad u sva četiri kvadranta. Međutim, cilj ovog istraživačkog rada je rad pretvarača u prvom i trećem kvadrantu i algoritmom upravljanja nije obuhvaćena mogućnost rada u drugom i četvrtom kvadrantu.

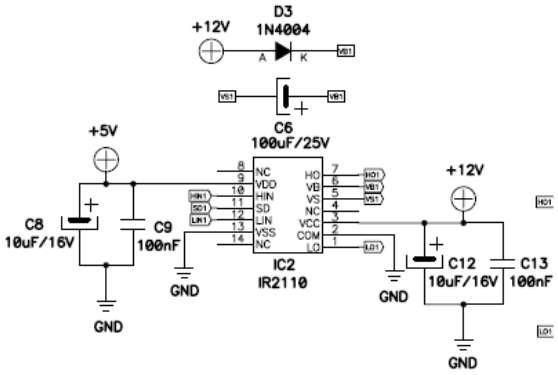
II. HARDVERSKA REALIZACIJA PRETVARAČA

Nakon teorijske analize pretvarača prelazi se na hardversku realizaciju pretvarača. Za hardversku realizaciju pretvarača na tržištu se može naći veliki izbor softvera koji omogućavaju razvoj hardvera. Kao prvi korak nameće se razrada električne šeme pretvarača čiji je izgled prikazan na slici 4.



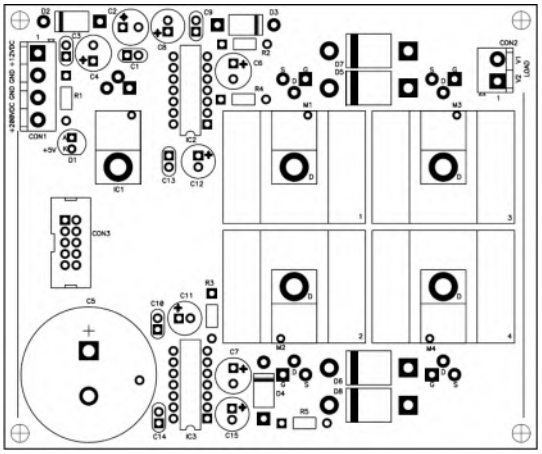
Slika 4. Električna šema hardverske realizacije pretvarača.

Na slici 5 prikazana je električna šema povezivanja drajverskog kola IR2110 za pobuđivanje tranzistora u granama pretvaračkog mosta. Hardvesrka kartica sadrži dva pobudna kola IR2110 za svaki par tranzistora po jedno. Kondenzator C6 predstavlja butstreb kondenzator tranzistora M1 prvoj grani H mosta, isti takav kondenzator postoji za tranzistor M2 u drugoj grani H mosta.



Slika 5. Principska šema povezivanja drajverskog kola IR2110.

Hardverska organizacija pločice pripremljene za štampu prikazan je na slici 6.



Slika 6. Hardverska organizacija pločice tranzistorskog pretvarača.

Elektronska pločica i izbor komponenti predviđen je za radni napon jednosmernog međukola od U_d=200 V. Zbog bezbednosti, hardversko kolo pretvarača je testirano za vrednost napona jednosmernog međukola U_d=15 V i za navedenu vrednost napona izabran je odgovarajući jednosmerni motor.

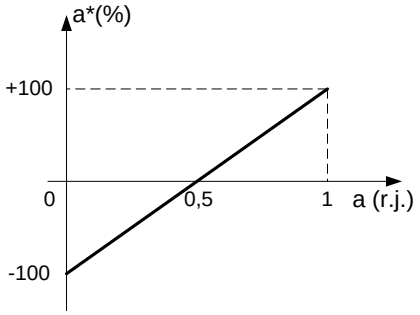
III. REZULTATI TESTIRANJA

Testiranje hardverske realizacije ogleda se u sledećem. Za promenu faktora ispune izmeren je opseg promene napona na armaturnom kolu i brzine obrtanja

jednosmernog motora. Digitalnim osciloskopom snimljeni su karakteristični vremenski dijagrami napona i struje na jednosmernom motoru [3]. Na slici 7 prikazan je izgled hardverskog kola i opreme u fazi testiranja.



Slika 7. Izgled ispitne stanice za testiranje rada hardverske realizacije pretvarača.



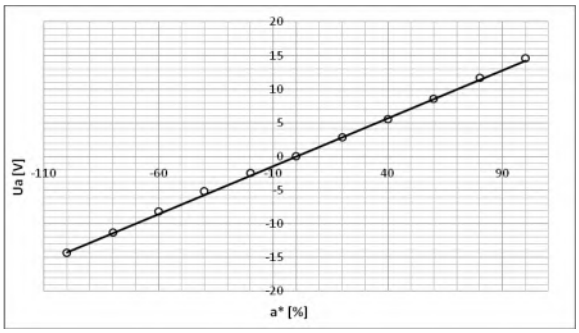
Slika 8. Zavisnost analitičke vrednosti faktora ispune *a* i zadate vrednosti faktora ispune na analognom pinu mikrokontrolera a^*.

U Tabeli I prikazane su vrednosti napona i brzine obrtanja na jednosmernom motoru za različite vrednosti faktora ispune. Merenje brzine izvršeno je laserskim tahometrom.

TABELA I
VREDNOSTI NAPONA I STRUJE MOTORA ZA RAZLIČITE VREDNOSTI FAKTORA ISPUNE ČOPERA

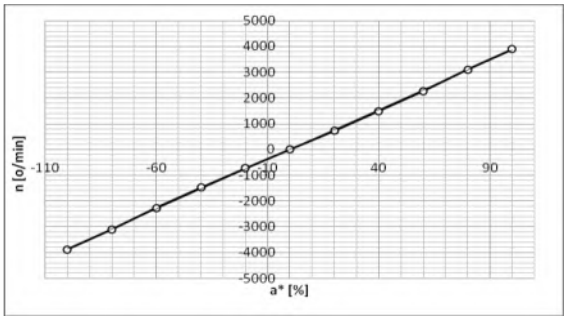
a^* [%]	U_a[V]	n [o/min]
100	14,6	3896
80	11,7	3106
60	8,6	2277
40	5,6	1486
20	2,8	727,6
0	0	0
-20	-2,4	-726,3
-40	-5,2	-1483
-60	-8,2	-2272
-80	-11,3	-3104
-100	-14,3	-3887

Na slici 9 prikazana je funkcionalna zavisnost napona na motoru u zavisnosti od faktora ispune pretvarača.



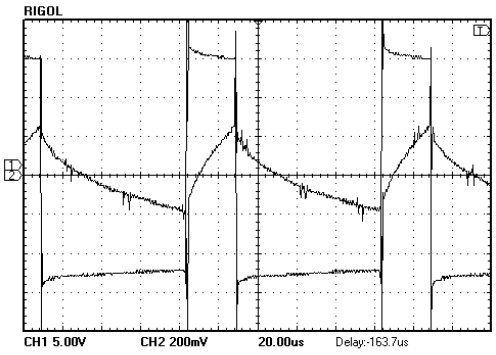
Slika 9. Dijagram promene napona na motoru u zavisnosti od faktora ispune, $U_a=f(a^*)$.

Na slici 10 prikazana je funkcionalna zavisnost brzine obrtanja jednosmernog motora u zavisnosti od faktora ispune.

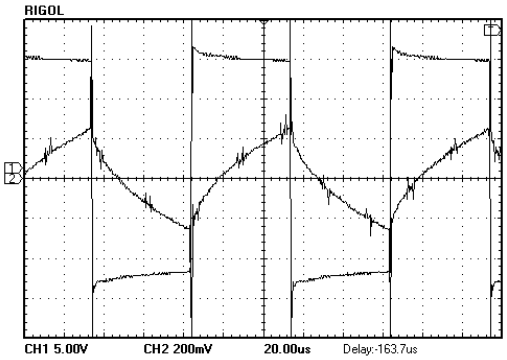


Slika 10. Dijagram zavisnosti brzine obrtanja u zavisnosti od faktora ispune pretvarača, $n=f(a^*)$.

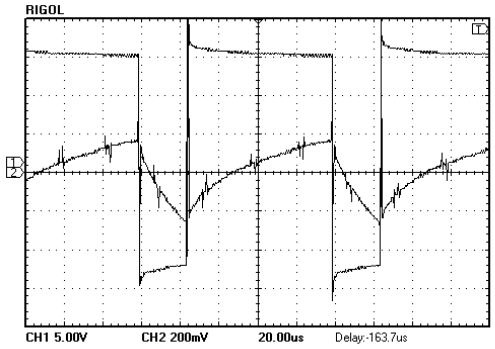
Na osnovu dijagrama sa slike 9 i slike 10 može se zaključiti da su regulacione karakteristike pretvarača linearne funkcije. Na slikama 11, 12 i 13 prikazani su vremenski dijagrami napona i struje jednosmernog motora za različite vrednosti faktora ispune tranzistorskog čopera a^*.



Slika 11. Vremenski dijagram napona i struje motora za faktor ispune a^*= -50 %.



Slika 12. Vremenski dijagrami napona i struje motora za faktor ispune a^*= 0 %.



Slika 13. Vremenski dijagrami napona i struje motora za faktor ispune a^*= +50 %.

IV. ZAKLJUČAK

U ovom radu obrađena je praktična realizacija tranzostorskog pretvarača-čopera u H mostu. Praktična realizacija predstavlja zahtevan postupak osmišljavanja hardverske pločice, postavljanje komponenti i njihovo lemljenje što zahteva izuzetnu preciznost. Posebnu pažnju nepophodno je posvetiti implementaciji odgovarajućeg upravljačkog algoritma za šta nam je poslužila odgovarajuća mikrokontrolerska kartica. Testiranje hardvera postavljeno je u funkciju provere osnovnih teorijskih principa. Ono što je izuzetno važno zapaziti je da su regulacione karakteristike pretvarača linearne što je izuzetno povoljno sa aspekta upravljanja i regulacije brzine obrtanja jednosmernog motora.

ZAHVALNICA

Ovom prilikom želimo da se zahvalimo našem profesoru mentoru, Skoko Saši, na pomoći pri realizaciji ovog rada. Takođe se zahvaljujemo Goranu Georgijevu na projektovanju elektronske pločice pretvarača.

LITERATURA

[1] D. Martinović, Z. Pendić, J. Menart, Ž. Janda, *Energetska elektronika*, Zavod za udžbenike, Beograd, 2007
[2] M. Ivanov, S. Skoko, *Elektroenergetika*, Beograd, Zavod za udžbenike, 2017.
[3] S. Skoko, P. Kolarski, *Energetska elektronika-praktikum laboratorijskih vežbi*, Novi Sad, Elektrotehnička škola "Mihajlo Pupin", 2018.

UGV – Robot SCORPION

Veljko Aleksić, Aleksandra Aleksić, Predrag Šubarević

Sadržaj −U ovom radu prikazano je zemaljsko vozilo bez posade (*Unmanned ground vehicle*), u daljem tekstu UGV robot [1]. Kao osnovni paket ovaj UGV robot na sebi nosi senzor za otrovne gasove. Robotska ruka, koja se nalazi na samom transporteru je jedan od modula UGV-a, i ima mogućnost manipulacije u polusferi, a namena joj je uzimanje uzoraka, dostava i uklanjanje raznih kontaminiranih predmeta, i sve to u zavisnosti od namene samog robota. Cilj ovog rada je stvoriti UGV robota koji ima posebnost u jednostavnosti korišćenja, bez ikakvih predhodnih znanja u programiranju. Jednostavnost rukovanja dodatno olakšava razvijeni aplikativni softver namenjen isključivo ovom vozilu. Pored robotske ruke ovaj UGV robot može biti dodatno opremljen raznim oruđima u skladu sa budućom namenom robota i željama kupaca. Osvrtom na istraživanja kompanije IDTechEx [2], naš UGV robot u budućnosti vidimo, pored ostalih primena, kao nezamenjivu poljoprivrednu mašinu. On će u mnogome olakšati posao na terenu kako bi lakše poljoprivrednici došli do zdravih i kvalitetnih proizvoda. Namenjen je prvenstveno da zameni ljude na opasnim radnim mestima. Sva istraživanja, radi realizacije ovog rada, obavljena su u laboratorijama ETŠ „Nikola Tesla" u Nišu.

I. UVOD

Paralelno sa društveno-ekonomskim razvojem odvija se i industrijski razvoj. Sve češće mašine i roboti zamenjuju čoveka. U mnogim industrijskim granama postoje radna mesta sa povećanim rizikom, i takva radna mesta su opasna po zdravlje i život zaposlenog. Navešćemo neka radna mesta sa povećanim rizikom na kojima bi ovaj robot imao primenu:

- Radna mesta u topionicama gde je radnik izložen stalnim povišenim temperaturama koje su izuzetno neprijatne za rad. Pored visokih temperatura česta su i isparenja otrovnih gasova, koje nikada ne možemo isključiti, već ih samo svodimo na neki prihvatljiviji nivo što umanjuje zdravstvenu ili životnu ugroženost radnika, ali je i ne isključuje u potpunosti.
- Radna mesta koja podrazumevaju rad sa opasnim materijama i hemikalijama gde greške u tehnološkom procesu ili pogreške samog zaposlenog dovode u pitanje i ostale koji se nalaze u blizini.
- Transport lakših paketa po nepristupačnim terenima
- Geološka istraživanja
- Manipulacija i rad sa opasnim materijama u poljoprivredi (zaprašivanje terena i kultura)
- Spasilačke ekipe u katastrofama izazvanih bilo prirodnim ili društvenim uticajem
- U vojnim zadacima: izviđačke delatnosti, dekontaminacija terena, provera miniranih područja i demineriranje terena...

Kod radnika se javlja zamor kao i pad koncentracije, a samim tim smanjuje se i njegova efikasnost i preciznost u radu. Premor radnika veoma često je i uzrok grešaka, a posledice mogu biti nesagledive.

UGV roboti su, dakle, mašine koje uspešno mogu zameniti čoveka u mnogim krajnje opasnim i teškim poslovima. Zbog toga se izrađuju i razne vrste UGV robota u skladu sa potrebama industrije. Mogu se razlikovati prema IP zaštiti, pogonu, kao i po konstruktivnim rešenjima ili manipulativnim mogućnostima. Trenutno je u svetu mali broj proizvođača UGV robota. Prvenstveno su projektovani za primenu u vojne svrhe (1930-ih, SSSR je razvio Teletenk, tenk naoružan mitraljezom koji je daljinski kontrolisan preko radija iz drugog tenka. Korišćeni su u Zimskom ratu (1939-1940) protiv Finske i na početku Istočnog fronta nakon što je Nemačka napala SSSR 1941.), ali je danas trend primene sve veći u industriji, poljoprivredi pa čak i u domaćinstvu. Pored pomenutih primena, naš UGV robot je pogodan i kao dostavni robot po ugledu na kompaniju iz San Franciska - "STARSHIP" [3]. Svedoci smo sve veće "robotizacije" društva i roboti bivaju sve prihvatljiviji u svim sferama života ljudi. UGV robot može imati set senzora ili kameru za posmatranje okoline i može samostalno donositi odluke o svom ponašanju ili će preneti informacije ljudskom operateru na drugoj lokaciji koji će kontrolisati samog robota.

II. PRINCIP RADA I PRAKTIČNA REALIZACIJA UGV ROBOTA

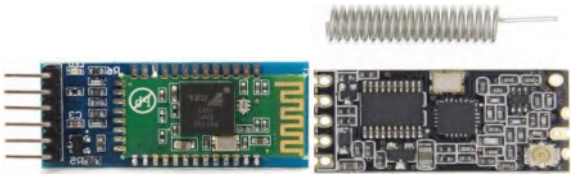
Kontrola pogonskih DC motora realizovana je uz pomoć dva kontrolera [4] ATMEGA 328P (slika 1) i bluetooth modula za komunikaciju (slika 2). Kao što gore navodimo, akcenat je na jednostavnosti upravljanja samim robotom, a trenutni domet upravljanja je oko 150 metara. Povećanje dometa do 1000m se može jednostavno realizovati korišćenjem modula HC12 (slika 2).

V. Aleksić i A. Aleksić, su učenici ETŠ "Nikola Tesla" u Nišu, Aleksandra Medvedeva 18, 18000 Niš, Srbija, E-mail: info@etstesla.rs.

P. Šubarević je mast. inž. mehatronike i upravljanja ETŠ "Nikola Tesla" u Nišu, Aleksandra Medvedeva 18, 18000 Niš, Srbija, E-mail: info@etstesla.rs.



Slika 1. Kontroler ATMEGA 328p (Arduino NANO).



Slika 2. Bluetooth modul HC05 (levo), HC12 (desno).

UGV robot - SCORPION (slika 3) razvijamo protekle dve godine. Izrađen je od čelika i malim delom od ABS plastike. Sama konstrukcija je od čeličnih profila, a oklop je izrađen od čeličnog lima debljine 1,2 mm što obezbeđuje dovoljnu čvrstinu samog UGV-a da nosi na sebi ili vuče manje terete. Ukupna nosivost UGV-a je testirana na 130 kg. Oslonjen je na 4 točka ukupne nosivosti testirane na 640 kg. Pogone ga dva DC motora sa obrtnim momentom od 120 kg/cm u većoj brzini i 150kg/cm u manjoj brzini. U toku kretanja imamo mogućnost izbora dve brzine i to kretanje manjom brzinom od 0 – 3 km/h i većom brzinom od 0 - 6,5 km/h. Svaka od brzina se kreće u svojim granicama zahvaljujući drajverima za brzinu motora tipa BTS7960 maksimalne struje od 40 A. Promena brzine kao i precizno podešavanje brzine se takođe vrši pomoću dela aplikacije za kretanje robota. Za prenos pogona sa motora na točkove koristili smo lance i lančanike radi što preciznijeg rada. Sva četiri točka imaju pogon što mu omogućava lakše savladavanje prepreka na teškim terenima. Promena smera kretanja samog UGV-a vrši se suprotnim okretanjem leve i desne strane pogonskih točkova. Ovom metodom smo postigli okretanje robota oko svoje ose. Većina konkurentnih UGV-a su skretanje rešili slično mehanizmu tenka i skretanje vrše jednostavnim kočenjem jedne pogonske strane. Ovom metodom se zahteva više prostora kod rotiranja samog robota za 180°. Dimenzije samog robota su strogo planirane tako da mu omogućuju primenu u raznim granama industrije. Težina osnovnog modela našeg UGV-a bez dodatnih modula je 27 kg, dužina 610 mm, a širina 630 mm (ovu širinu biramo radi lakšeg probijanja robota kroz poljoprivredne zasade jer je standardna širina sadilica u poljoprivredi 75-85 cm). Visina je 280 mm. Ovu visinu namenski biramo radi lakšeg podvlačenja robota ispod vozila ukoliko se koristi za kontrolu vozila. Težište samog robota je postavljeno u preseku dijagonala krajnjih spoljnih tačaka osovina. Time smo postigli izuzetnu stabilnost samog robota na terenu.



Slika 3. UGV robot – SCORPION.

Pregled situacije na terenu omogućen je HD kamerom. Naš osnovni model UGV robota na sebi nosi jednu IP kameru u HD rezoluciji, a ima softversku mogućnost pogleda na dve IP kamere ukoliko je to potrebno. Kamere mogu biti iste ili kombinacija HD kamere i kamere za noćno izviđanje. HD kamere mogu biti fiksirane na bilo kom delu UGV-a ili na predviđenoj pokretnoj platformi, na vrhu robota, sa koje ima mogućnost rotiranja 360°. Rotaciju kamere vrši servo motor [5] MG996R (slika 4).

MG996R Servo Motor - karakteristike:

- Radni napon je 5 V-7,2 V
- Struja: 2.5 A (6 V)
- Maksimalna torzija: 9,4 kg/cm (na 4,8 V)
- Maksimalni obrtni momenat zaustavljanja: 11 kg/cm (6 V)
- Radna brzina je 0,17 s/60°
- Materijal reduktora: Metal
- Rotacija : 0°-180°
- Težina motora : 55 g



Slika 4. Servo motor MG996R.

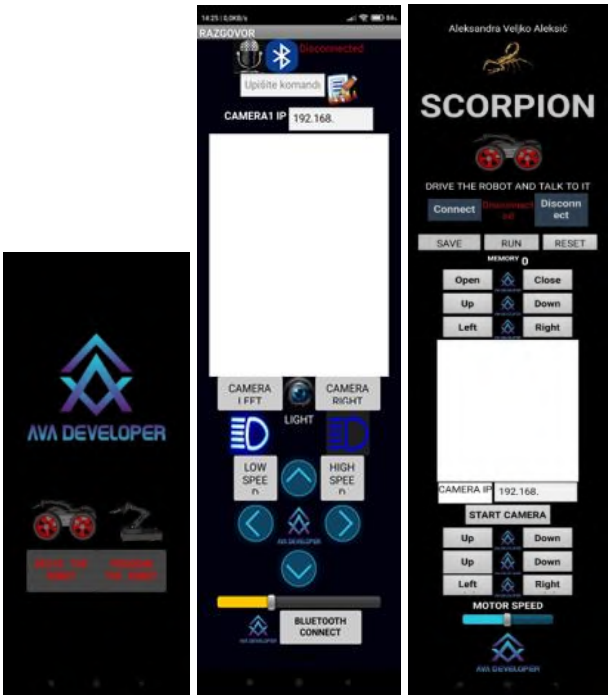
Sa prednje i zadnje strane poseduje LED svetla u vidu trake koja omogućuju korisniku nesmetano kretanje u noćnim uslovima. Najisturenije tačke sa svih strana našeg UGV robota su točkovi što nam pomaže u savladavanju mnogih prepreka. Pored prevazilaženja zahtevnih terena i prepreka, ovakva konstrukcija štiti UGV robota od eventualnih udara oklopa o tlo prilikom pada ili spuštanja padobranom. Osnovna verzija programa u kontrolerima sadrži kretanje UGV robota (glasovne pisane i manualne komande), promenu brzine, rotaciju kamere i uključivanje svetla. Sa zadnje strane predviđeno je kućište sa kontaktima za punjenje baterije (slika 5).



Slika 5. Kućište sa kontaktima za punjenje baterije.

Instalirana baterija obezbeđuje napajanje od 12 V i 20 A. Ovakva snaga baterija omogućava autonomiju rada UGV robota preko 3 sata. U punoj brzini, ovaj UGV robot je u stanju da prevali dužinu puta od preko 20 km sa jednim punjenjem baterije ili da detaljno pretraži teren od 12000 m^2.

Instaliranjem našeg aplikativnog softvera (slika 6), *smart* telefon ili tablet uređaj postaje upravljačka jedinica samog UGV robota sa dodatim programatorom robotske ruke ili upravljačkim sistemom nekog drugog modularnog oruđa ili uređaja prema želji korisnika ili nameni robota.



a) Naslovna stranica, b) Kontrola vozila c) Robotska ruka

Slika 6. Aplikativni softver.

Pored ručne manipulacije robotska ruka (kao modul) prepoznaje i ljudski govor, pa se putem naše aplikacije može njen rad kontrolisati i govorom ili pisanim komandama. U stanju je da odgovara na postavljena pitanja koja su vezana za njene mogućnosti i sam rad.

Usled nedostatka memorijskog prostora i broja izlazno/ulaznih pinova na jednom kontroleru, koristili smo dva zasebna kontrolera Arduino Nano ATmega328P. Po prvi put koristimo jedan bluetooth modul za kontrolu dva zasebna kontrolera sa različitim programima.

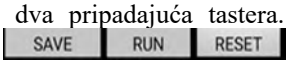
III. OPIS RADA

Ovaj UGV robot je krajnje jednostavan za korišćenje. Što se tiče samog UGV-a, potrebno je da korisnik startuje vozilo pritiskom na prekidač instaliran na samom vozilu. Ostale funkcije su obezbeđene jedinstvenom aplikacijom. Trenutna verzija UGV robota na sebi ima modul u vidu robotske ruke i za nju je takođe obezbeđena podrška kao deo aplikacije.

Aplikativni softver koji smo napisali za ovaj UGV robot je jedinstven i nije moguće ni robotu ni eventualnim modulima pristupiti ili ih kontrolisati bilo kojim drugim aplikativnim softverom, a trenutno sadrži tri stranice. Aplikacija je trenutno na engleskom jeziku ali može biti izmenjena na zahtev korisnika u bilo koji drugi jezik. U mogućnosti smo da uradimo i višejezični softver ukoliko korisnik ima za tim želju ili potrebu. Naslovna stranica sa jasno vidljivim našim logotipom (slika 6) je vodič ka ostalim stranicama i omogućava nam da lako prelazimo sa modula na modul. Konkretno u ovoj situaciji to je stranica ka modulu robotske ruke ili ka upravljačkom delu aplikacije. Odabirom tastera “DRIVE THE ROBOT”  prelazimo na stranicu koja ima funkciju da kontroliše samo vozilo (slika 6 b)).

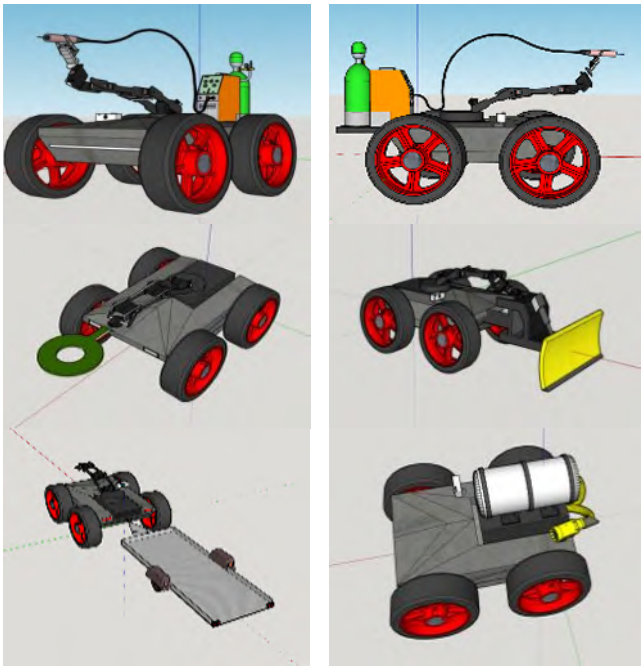
U gornjem delu prozora vidimo tastere u vidu simbola funkcije koju nose. Simbol  je taster za konekciju sa bluetoot modulom. Simbol mikrofona je za glasovne komande ka vozilu, a simbol  sa pripadajućim “text box”-om obezbeđuje pisane komande ka vozilu. Ispod ovih tastera sledi “text box” polje sa individualnom ip adresom kamere, a nakon toga i prozor kamere. Tasteri  obezbeđuju zaokret pokretne kamere u punom krugu, bilo to na levu ili desnu stranu. Sledeći nivo tastera  zadužen je za osvetljenje. Tasteri “Low speed” i “High speed”  su zapravo vrsta menjača i regulišu stepen prenosa na vozilu. Kursorski tasteri simbolično opisuju kretanje UGV robota. Na dnu aplikacije se nalazi slajder  kojim precizno povećavamo ili smanjujemo brzinu kretanja samog UGV robota. Odabirom tastera na naslovnoj stranici “PROGRAM THE ROBOT” prelazimo na stranicu aplikativnog softvera za kontrolu i programiranje trenutno implementiranog modula (slika 6 c)), a to je robotska ruka.

Ovo je samo jedan od mogućih modula naše izrade i ne bismo se oko njega puno zadržavali u opisu. Ovde prvo uočavamo ime ovog UGV-a - “SCORPION”. Odlučili smo se za ovaj naziv asocirani fizičkim izgledom samog UGV-a

kada je na njemu robotska ruka. I u ovom slučaju su svi tasteri imenovani i lako je korisniku za korišćenje. Nakon povezivanja sa bluetooth modulom slede tasteri za programiranje same robotske ruke. Robotska ruka ima 6 osa i svaka od osa ima svoja dva pripadajuća tastera. Počevši sa vrha vidimo tastere 

za pokretanje programa robotske ruke i memoriju. Niže, tasteri 

regulišu stisak klešti sa vrha robotske ruke (prva osa robotske ruke), pa tako redom svaki par tastera naniže regulišu sledeću osu po redu, sve do zadnje šeste ose. U središnjem delu prozora je opet prozorčić IP kamere i tu pratimo rad robotske ruke. Ovo je samo jedan od mogućih modula implementiran trenutno na robotu. Pored ovog modula korisnik može odabrati i druge module po svojoj želji ili potrebnoj nameni. Na slici vidimo više naših predloga modula (slika 7). Može biti zavarivački robot, minotragač, grtalica za sneg na trotoarima, kosilica, prskalica u poljoprivredi, kopačica u poljoprivrednim zasadima , za dekontaminaciju terena, dostavni robot itd.



Slika 7. Mogući modeli UGV robota.

IV. ZAKLJUČAK

Proverom svih trenutnih mogućnosti u laboratorijama ETŠ "Nikola Tesla" u Nišu zaključujemo da UGV robot ispunjava očekivane rezultate. Cena ovakvog modela je oko 700 evra, a ona je zanemarljivo mala u odnosu na eminentne svetske proizvođače čiji su roboti daleko skromnijih mogućnosti. Cena konkurentnih mašina se kreće u opsegu od 5000 hiljada evra pa i do više desetina hiljada evra. Cilj nam je prikupiti sredstva ili sponzore za serijsku proizvodnju robota domaće izrade. Kodiranje kontrolera izvršeno je u Arduino IDE-u. Svaki od kontrolera ima zaseban kod, ali podjednako učestvuju u komunikaciji sa UGV robotom.

ZAHVALNICA

Najveću zahvalnost za bezuslovnu podršku dugujemo mentoru prof. Predragu Šubareviću i direktoru škole prof. Nebojši Sokoloviću. Posebnu zahvalnost na podršci dugujemo: zameniku direktora prof. Tatjani Šubarević, prof. Nataši Tanasković, prof. Sonji Šumonji, prof. Ivani Dinić Živanović, prof. Biljani Stefanović, prof. Radmili Mitić, prof. Marini Božilović, prof. Predragu Jovanoviću, prof. Veri Stojanović, prof. Mileni Nikolić, prof. Jeleni Radovanović, prof. Ivani Janković, prof. Snežani Rančić, prof. Jeleni Božilović. Zahvaljujemo se i ostalim radnicima škole na podršci, koja nam je puno značila prilikom realizacije ovog projekta.

LITERATURA

[1] "Unmanned Ground Vehicle Technology", *Proceedings of SPIE--the International Society for Optical Engineering*, Society of Photo-optical Instrumentation Engineers, Izdavač: International Society for Optical Engineering, Univerzitet u Mičigenu 2001. god.
[2] IDTechEx Dostupno na: https://www.idtechex.com/en/research-report/agricultural-robotics-market-2022-2032/837.
[3] https://www.starship.xyz/company
[4] https://www.arduino.cc/
[5] https://components101.com/servo-motor-basics-pinout-datasheet

Ispitivanje stabilnosti probojnog napona gasnih odvodnika tipa *Littelfuse*-300 V

Lazar Šipić

Sadržaj - U ovom radu prikazano je istraživanje u oblasti gasnih odvodnika prenapona. U radu je opisan deo eksperimentalnih podataka dobijenih merenjem probojnog napona komercijalnog gasnog odvodnika proizvođača *Littelfuse* na radnom naponu 300 V. Odabrano je deset različitih uzoraka i određen njihov statički probojni napon dinamičkom metodom. Eksperimenti su odrađeni u Laboratoriji za vakuumsku i gasnu tehniku na Elektronskom fakultetu. Kako je probojni napon fizička veličina koja ima stohastički karakter, izvršena je i statistička analiza dobijenih eksperimentalnih vrednosti.

I. UVOD

U tehnici se pod prenaponom podrazumeva pojava da je potencijal jedne tačke neke komponente, uređaja ili voda u odnosu na neku drugu tačku ili tačku nultog potencijala veći od dozvoljenog. Prenaponi iznad određene vrednosti mogu dovesti do ugrožavanja bezbednosti ljudi koji rukuju uređajima kao i do oštećenja samih uređaja. Prenaponi iznad dozvoljenih vrednosti mogu dovesti do trajnog ili privremenog oštećenja pojedinih elektronskih komponenti i uređaja i do pojave šuma u prenosnim signalima [1].

U izvore prenapona spadaju atmosferska pražnjenja, elektrostatička pražnjenja, komutacioni prenaponi, radarski impulsi i elektromagnetni impulsi nuklearne eksplozije. Od svih vrsta prenapona, atmosfersko pražnjenje je najopasnije jer se ne može uticati na uzrok njegovog nastanka. Atmosfersko pražnjenje se javlja između oblaka čija su naelektrisanja različitog znaka (munja) ili između oblaka i Zemlje (grom). Ove vrste pražnjenja značajno utiču i na telekomunikacione vodove preko kojih oštećuju komponente, odnosno uređaje.

Elementi za zaštitu od prenapona se prema načinu primene napona na njihovim krajevima, pri povećanju struje kroz njih mogu podeliti na linearne i nelinearne. Linearni elementi za zaštitu od prenapona su električni filtri, čiji su najosetljiviji elementi kondenzatori.

Nelinearni elementi za zaštitu od prenapona se mnogo češće koriste od linearnih i oni se prema tehnologiji izrade i principu rada mogu podeliti u tri grupe. To su prenaponske diode, metaloksidni varistori i gasni odvodnici prenapona.

L. Šipić, student master studija modula Elektronika i miksrosistemi, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: lazarworkout@elfak.rs.

II. GASNI ODVODNICI PRENAPONA

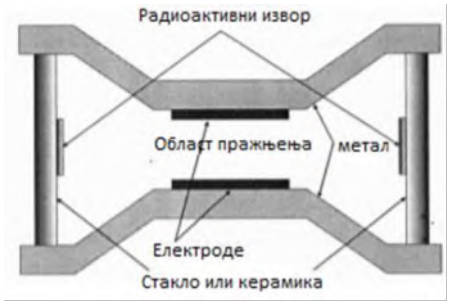
Gasni odvodnici prenapona (slika 1) imaju široku primenu u prenaponskoj zaštiti. Rade na principu električnog proboja gasa i mogu se koristiti za zaštitu od prenapona intenziteta od nekoliko desetina volti do nekoliko desetina hiljada volti. U pogledu energetske izdržljivosti oni su najizdržljiviji, a u pogledu cene najjeftiniji elementi za zaštitu od prenapona. Prednost gasnih odvodnika prenapona nad ostalim elementima za zaštitu od prenapona su u opsegu zaštitnih nivoa, sposobnosti provođenja velikih impulsnih struja i snazi disipacije. Nedostaci su nešto manja brzina reagovanja što se u kombinovanoj zaštiti nadoknađuje poluprovodničkim elementima i nemogućnost rednog vezivanja. Ova vrsta odvonika prenapona sastoji se od dve identične metalne elektrode zatopljene u keramičko ili stakleno kućište. Međuelektrodno rastojanje je veličine 1 mm.



Slika 1. Izgled gasnog odvodnika.

III. PRINCIP RADA GASNIH ODVODNIKA

Rad gasnih odvodnika prenapona se zasniva na principu proticanja električne struje kroz gas, tj. na električnom pražnjenju gasova. Gasni odvodnici prenapona spadaju u nelinearne elemente koji se isključivo koriste za zaštitu od prenapona. Oni se sastoje od dve ili tri elektrode koje su zatopljene u keramičko ili stakleno kućište (slika 2) [1,2].



Slika 2. Presek gasnog odvodnika prenapona sa dve elektrode.

Elektrode su ravne i paralelne kako bi se u međuelektrodnom prostoru obezbedilo približno homogeno električno polje. Rastojanje između elektroda je reda dela milimetra i smeštene su u kućištu koje se sastoji od keramike i metala. Kao izolacioni materijal koriste se plemeniti gasovi ili njihove smeše pod pritiscima.

Gasni odvodnici prenapona imaju niz prednosti u odnosu na druge elemente zaštite, a to su:

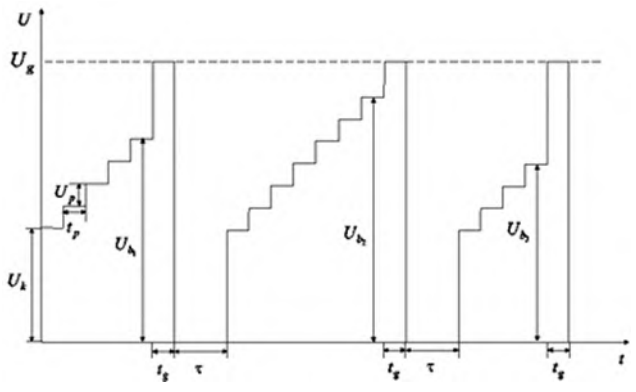
- Velika izdržljivost,
- Najveći opseg zaštitnog nivoa,
- Mala vrednost unutrašnje kapacitivnosti (od 1 pF do 10 pF),
- Velika i nepromenljiva otpornost.

Postoji i niz nedostataka:

- Mala brzina odziva,
- Veliko odstupanje vremena odziva za različite odvodnike,
- Problem gašenja odvodnika,
- Postojanje struje produženog delovanja u impulsnom režimu,
- Odstupanje probojnog napona od nazivne vrednosti date u tehničkoj dokumentaciji za +/- 20%.

IV. OPIS METODE MERENJA PROBOJNOG NAPONA

Za realizaciju eksperimenta korišćen je sistem za merenje i akviziciju podataka. Na slici 3 [1] je dat šematski prikaz postupnog povećanja napona do proboja. Ovde U_p je korak povećanja napona, t_p je trajanje vremenskog koraka, t_g je vreme pražnjenja koje se postiže održavanjem konstantne vrednosti napona U_g, U_k početni napon, τ je vreme relaksacije i U_{b1}, U_{b2},… su vrednosti probojnog napona. Kao što se može videti na slici 3, na gasnu komponentu se primenjuje fiksni napon U_k, koji je znatno niži od očekivanog probojnog napona U_b. Zatim se napon povećava za unapred određene naponske korake U_p, pri čemu je trajanje naponskog koraka t_p unapred podešeno.

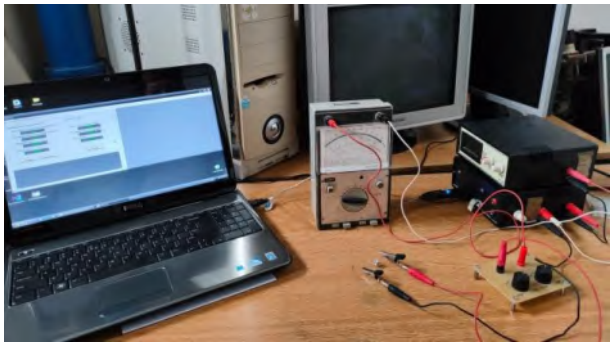


Slika 3. Princip rada sistema za akviziciju probojnog napona: U_k – početni napon znatno manji od očekivanog napona proboja; U_g - vrednost do koje raste napon.

Odnos $k=U_p/t_p$ predstavlja brzinu porasta napona. Kada dođe do proboja, zadata vrednost struje pražnjenja i_g je obezbeđena naponom U_g tokom vremena pražnjenja t_g, kako bi se uspostavila stacionarna koncentracija naelektrisanih i pobuđenih čestica u gasu. Napon na elektrodama gasne komponente se tada isključuje za vreme τ= 1 s. Kada istekne postavljeno vreme relaksacije, proces počinje iz početka.

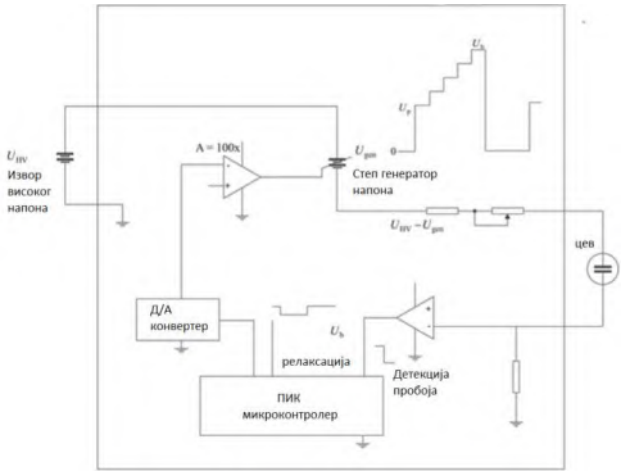
V. OPIS EKSPERIMENTA

Na slici 4 prikazana je sama postavka eksperimenta u laboratoriji. Sistem za merenje napona korišćen pri eksperimentalnoj realizaciji prikazan je na slici 5. Sastoji se od izvora visokog napona, stepenastog generatora signala, gasnog odvodnika čiji se probojni napon određuje, uređaja za detekciju električnog proboja gasa korišćene komponente i kontrolnog bloka. Visokonaponski izvor napajanja i stepenasti naponski generator su povezani u suprotnosti, kao što je prikazano na slici 4. Napon primenjen na gasni odvodnik jednak je razlici napona (U_{HV}-U_{gen}). Generator stepenastog napona je projektovan kao pojačavač napona sa pojačanjem od 100.



Slika 4. Fotografija mernog sistema u laboratoriji.

Upravljački blok se sastoji od mikroprocesora i digitalno-analognog (D/A) konvertora. Izlazni napon generatora je direktno proporcionalan naponu iz D/A konvertor, odnosno U_{gen}=100$U_{D/A.}$ Napon na gasnoj komponenti raste u koracima od naponskog nivoa U_k, a kada se pojavi proboj, naponski signal sa otpornika R_{sense} se prenosi preko pojačavača A2 do procesora. Napon pri kojem se javlja proboj na gasnoj komponenti se čuva u računaru koji je preko USB porta povezan sa ovim sistemom. Tokom vremena relaksacije određenog impulsom napona koji generiše mikroprocesor, napon na komponenti je jednak nuli. Napon U_k koji odgovara prvom koraku stepenastog generatora (maksimalni napon na stepenastom generatoru) može se promeniti modifikacijom napona iz izvora visokog napona. Na ovaj način, pored pouzdanije detekcije proboja, dobija se i mogućnost kontrole trajanja vremena pražnjenja t_g, što ovaj sistem čini povoljnijim od prethodno korišćenog sistema za merenje probojnog napona gasnih komponenata.



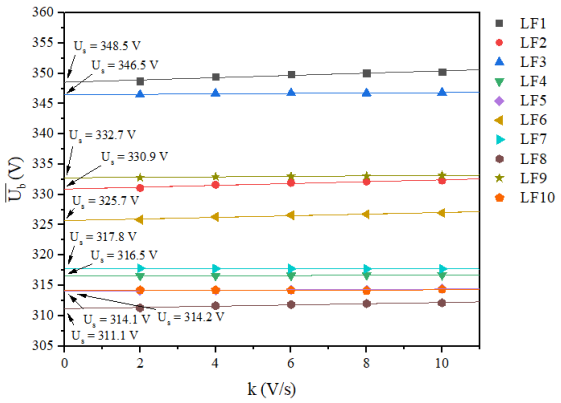
Slika 5. Blok šema sistema za merenje napona.

VI. EKSPERIMANTALNI REZULTATI

Statički probojni napon se može definisati kao najveći radni napon U_w za koji je verovatnoća električnog proboja jednaka nuli, tj:

$$\lim_{P\to 0} U_w = U_s. \tag{1}$$

Statički probojni napon U_s, za razliku od probojnog napona U_p, jednoznačno je određena veličina. Procena tačnog probojnog napona se vrši na osnovu merenja različitih parametara jer ga nije moguće tačno odrediti. Najbolja procena se, pored nekoliko metoda, može izvršiti iz zavisnosti srednje vrednosti probojnog napona od brzine porasta napona k. Fitovanjem eksperimentalnih rezultata pravom linijom i njenim presekom sa osom, koja odgovara vrednosti k=0, procenjuje se vrednost tačnog probojnog napona U_s. Na narednom grafiku (slika 6) i Tabeli I. prikazani su dobijeni rezultati merenja za deset različitih gasnih odvodnika proizvođača *Littelfuse* radnog napona 300 V.

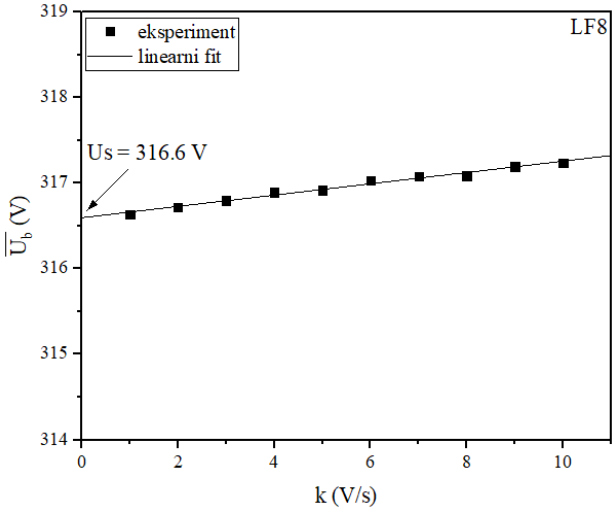


Slika 6. Prikaz statičkih probojnih napona korišćenih gasnih odvodnika.

TABELA I
STATIČKI PROBOJNI NAPONI KORIŠĆENIH KOMPONENTI

LF	U_s (V)
1	348.5
2	330.9
3	346.5
4	316.5
5	314.1
6	325.7
7	317.8
8	311.1
9	332.7
10	314.2

Nakon izvršenih merenja i nacrtanih grafika, određena je detaljna analiza dobijenih eksperimentalnih rezultata gasnog odvodnika sa najnižim statičkim probojnim naponom koji je najpribližniji vrednosti radnog napona iz tehničke dokumentacije [6]. U našem slučaju to je gasni odvodnik LF8 čiji je statički probojni napon procenjen na 311.1 V. Analiza je rađena do sada sa 100 podataka u jednom setu merenja. Za LF8 odrađen je ponovljen eksperiment pri čemu je za svaku brzinu porasta napona broj izmerenih vrednosti probojnog napona bio 400, što omogućuje realizaciju detaljnije statističke analize.



Slika 7. Gasni odvodnik LF8 čiji je statički probojni napon procenjen na 316.6 V.

Na grafiku prikazanom na slici 7 i u tabeli II dati su rezultati merenja za gasni odvodnik LF8 proizvođača Littelfuse radnog napona 300 V. U Tabeli 2 predstavljene su srednje vrednosti i standardne devijacije probojnog napona za svaku brzinu porasta probojnog napona za gasni odvodnik korišćen u eksperimentu.

TABELA II
SREDNJA VREDNOST I STANDARDNA DEVIJACIJA ZA SVAKU BRZINU PORASTA NAPONA ZA GASNI ODVODNIK OZNAČEN LF8.

k (V/s)	$\overline{U_b}$ (V)	σ_{Ub} (V)
1	316.6	2.5
2	316.7	2.3
3	316.8	2.3
4	316.9	2.2
5	316.9	2.2
6	317.0	2.2
7	317.1	2.1
8	317.1	2.1
9	317.2	2.1
10	317.2	2.1

VII. ZAKLJUČAK

U ovom radu analizirane su raspodele probojnog napona gasnih odvodnika prenapona. Analiza je izvršena za 10 komercijalnih gasnih odvodnika, proizvođača *Littelfuse* na radnom naponu 300 V. Nakon toga izabran je gasni odvodnih sa najmanjim probojnim naponom i na njemu je izvršena detaljnija analiza. Cilj daljih istraživanja je da se ista merenja izvrše sa sličnim gasnim komponentama radi uređivanja rezultata i pri novim eksperimentalnim uslovima u prisustvu zračenja [7]. Pokazan je značaj primene dinamičke metode za procenu statičkog probojnog napona, i značaj njegovog određivanja. Utvrđeno je da se za svaku od korišćenih eksperimentalnih uzoraka dobijala drugačija vrednost, ali je sve provereno u tehničkoj dokumentaciji i potvrđena je stabilnost napona i pouzdanost ovog tipa gasnog odvodnika.

ZAHVALNICA

Rad je realizovan u laboratoriji Elektronskog fakulteta kao deo rada na studijskom programu Elektronika i mikrosistemi. Autor se zahvaljuje mentoru prof. dr Emiliji Živanović na pruženom znanju i pomoći.

LITERATURA

[1] Momčilo M. Pejović, Gasne elektronske komponente – Uvod u električna gasna pražnjenja i proboj gasa, Elektronski fakultet, Univerzitet u Nišu, 2008.
[2] Emilija Živanović, Marija Živković, Milić Pejović, "The evolution of breakdown voltage and delay time under high overvoltage for different types of surge arresters", Facta Universitatis, Series: Electronics and Energetics, Vol. 34, No. 2 pp. 307-322, 2021.
[3] Momčilo M. Pejović, Milić M. Pejović, Električni proboj gasova- merni sistemi i eksperimenti i eksperimentalna istraživanja, Elektronski fakultet, Univerzitet u Nišu, 2009.
[4] P. Osmokrović, I. Krivokapić, D. Matijasević and N. Kartalović, "Stability of the gas-filled surge arresters characteristics under service conditions"
[5] Gas Discharge Tubes dostupno na: /https://www.bourns.com/docs/technical-documents/technical-library/gas-discharge-tubes/publications/bourns_gdt_short_form.pdf
[6] Tehnička dokumentacija gasnog odvodnika: https://es.datasheetq.com/CG145-doc-Littelfuse-2013
[7] Emilija Živanović, Marija Živković, Sandra Veljković, "Study of Breakdown Voltage Stability of Gas-Filled Surge Arresters in the Presence of Gamma Radiation", Electronics, Vol. 11, p. 2447, 2022.

Sistem za prijavljivanje zasnovan na prepoznavanju lica

Dajana Milošević

Sadržaj – U ovom radu je prikazana implementacija sistema za prijavljivanje zaposlenih koji se zasniva na prepoznavanju lica. Opisana je identifikacija potencijalnih zloupotreba ovako kreiranog sistema, kao i načini za njihovo sprečavanje. U praktičnom delu rada je predložena arhitektura i prototip sistema za prijavljivanje koji koristi dve kamere u cilju sprečavanja lažnog prijavljivanja korišćenjem slike.

I. UVOD

Detektovanje i prepoznavanje lica kao tehnike se sve češće koriste u savremenom svetu, obično kao delovi većih, kompleksnijih sistema. Kriterijumi za izbor algoritama jesu pouzdanost i brzina izvršenja. Cilj ovog rada je proučiti mogućnost implementacije sistema za prijavljivanje zaposlenih koji se zasniva na prepoznavanju lica. Ukoliko sistema detektuje da je reč o pravom licu, nakon toga ulazi u analizu prepoznavanja lica. Ovakav sistema bi imao određenu bazu sa licima koje je sistema naučen da prepozna. Ideja je da se ovakav sistema nalazi na ulazu kompanije, ili nekog drugog objekta, i da kontroliše ulaz/izlaz. Ovaj sistem ima mogućnost da sa kamera, koje snimaju u realnom vremenu, detektuje sva lica koja se pojavljuju u kadru kamera i da ih prepozna ukoliko se nalaze u bazi. Sistem je implementiran putem dve pomoćne kamere koje snimaju prostor, a svako lice koje želi da se prijavi mora biti snimljeno putem ovih kamera. Sistem, putem odgovarajućeg ekrana, obaveštava o tome da li je lice uspešno prepoznato i kao povratnu informaciju vraća ime tog zaposlenog i vreme prijavljivanja. U protivnom, sistem će izbaciti adekvatnu poruku da prijavljivanje nije mogće. Jedan od potencijalnih problema, kod sistema implementiranog na ovaj način, može biti i to da neko pokuša prijavljivanje sa tuđom slikom, da sistem detektuje da se radi o stvarnoj prijavi tog lica i da omogući ulaz licu koje nema pravo na to. U ovu svrhu, bezbednosti i zaštite sistema od neovlašćenog upada, kreira se mapa dubine sa slike, obrađuje se ulazni snimak i određuje se na osnovu dubine slike da li je reč o stvarnoj osobi, ili neko neovlašćeno pokušava da se prijavi korišćenjem slike zaposlenog lica. Takođe se pomoću dve kamere prate pozicije glave, da li je položaj glave različit u odnosu na obe kamere, a kao pomoćni deo ovog sistema se koristi i neuronska mreža koja je obučena tako da prepoznaje lažna i prava lica.

D. Milošević, master inženjer elektrotehnike i računarstva, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: dajana.milosevic@gmail.com.

II. DETEKTOVANJE I PREPOZNAVANJE LICA

Detekcija lica je računarska tehnika koja funkcioniše na način da se koristi klizajući prozor kojim se izvlače regioni od interesa, a zatim se region propusta kroz klasifikator pomoću kog se određuje da li se u njemu nalazi lice ili ne. Otkrivanje lica predstavlja neophodan prvi korak svih algoritama koji se na neki način bave analizom lica i njihovom obradom. Otkrivanje lica, u generalnom smislu, jeste problem lociranja i lokalizacije jednog ili više lica na ulaznoj slici ili snimku koji se prosleđuje na analizu. Nakon detekcije sledi proces prepoznavanja lica. Proces prepoznavanja lica se sastoji od četiri koraka:

1. Detekcija lica – Postoji više načina za detektovanje lica, u ovom sistemu je iskorišćen HOG (histogram orijentisanih gradijenata). [1]
2. Lice je uspešno izolovano. Nakon toga potrebno je pronaći 68 karakterističnih tačaka lica
3. Kodiranje lica – Rešenje je obučiti konvolucionu neuronsku mrežu da generiše 128 merenja za svako lice. Ovu mrežu je potrebno obučiti samo jednom. [2][3]
4. Nalaženje imena osobe – U ovu svrhu je korišćen SVM klasifikator [4][5]

III. TEHNIKE KOJE SE KORISTE SA CILJEM SPREČAVANJA ZLOUPOTREBA SISTEMA

Tehnike koje će biti opisane, a koje se koriste sa ciljem sprečavanja lažnog prijavljivanja jesu: kreiranje mape dubine, procena položaja glave u odnosu na kameru, kao i binarna konvoluciona neuronska mreža, obučena na način da prepozna lažna prijavljivanja. Arhitektura sistema bi se sastojala od dve kamere, koje bi uz prethodnu kalibraciju, jednoznačno mogle odrediti validnost prijavljivanja.

Svaka od ovih tehnika na pouzdan način određuje validnost prijavljivanja, a kombinacija istih to jednoznačno određuje. Pre korišćenja ovih tehnika, neophodno je izvršiti kalibraciju kamere. Kalibracija kamere jeste proces dobijanja parametara sočiva i senzora slike. Kalibracija preslikava koordinate stvarnog sveta u koordinate piksela. Kalibracijom dobijamo dve vrste parametara: spoljašnje, koji su potrebni za transformaciju koordinatnog sistema kamere u globalni koordinatni sistem i unutrašnje, gde spadaju žižna daljina, pozicija optičkog centra kamere itd.

Prvi korak kod ovog procesa jeste kreiranje slika šahovske table iz različitih uglova. Šahovska tabla se koristi iz razloga što ima jasno naglašene ivice i uglove. Parametri dobijeni na ovaj način se smeštaju u okviru xml fajla koji se kasnije učitava prilikom analize snimaka.

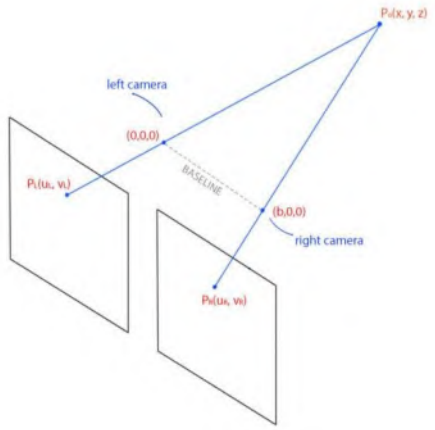
A. Percepcija dubine pomoću kamere

Prva tehnika koja je korišćena u cilju sprečavanja zloupotreba sistema jeste percepcija dubine pomoću kamere. Za ovu tehniku je potrebna ili stereo kamera ili kombinacija 2 kamere koje bi hvatale prostor iz različitih uglova i na osnovu ovoga se dobija željena slika. Izdvajaju se dva suštinska zahteva za procenu dubine (3D strukture) date scene:

1. korespondencija tačaka
2. relativna pozicija kamera

Ukratko: odgovarajuće tačke su pikseli različitih slika koje su projekcije iste 3D tačke. Pronalaženje korespondencije tačaka za sve 3D tačke snimljene na obe slike para stereo slika daje gustu korespondenciju, koja se može koristiti za pronalaženje mape guste dubine i percepciju 3D sveta.

Često je izazovno pronaći gustu korespondenciju, te se tu se ogleda lepota i efikasnost epipolarne geometrije u smanjenju prostora za pretragu tačaka korespondencije. Paralelne epipolarne linije dodatno pojednostavljuju problem, pošto su odgovarajuće tačke povezane horizontalnim pomeranjem. Ovo pomeranje se naziva disparitet. Dve kamere snimaju istu scenu, a epipolarna geometrija opisuje odnos između dva rezultujuća pogleda (slika 1) [6][8].



Slika 1. Formiranje slike u stereo postavci.

Dve kamere gledaju u tačku P_o, P_L I P_R su centri simetrije sočiva dve kamere. S obzirom da sistem koristi horizontalnu stereo kameru, odgovarajuće tačke za ispravljeni par stereo slike bi imale istu *y* koordinatu. Postavlja se pitanje na koji način pronaći odgovarajuće tačke. Može se zamisliti naivan pristup da se jednostavno uporede vrednosti piksela u istom redu para stereo slike. Međutim, ovo nije dobar pristup. Više piksela koji odgovaraju različitim slikama mogu imati isti intenzitet piksela. Štaviše, s obzirom na praktična ograničenja kao što su razlika u senzorima slike ili vrednosti ekspozicije, odgovarajući pikseli mogu imati nejednake vrednosti piksela. Ključni problem koji se ovde javlja jeste taj da isti piksel može biti posmatran kao dva različita piksela iz razloga što imaju različite vrednosti kada se radi o inenzitetu piksela, a da, s druge strane, dva različita piksela sa istom vrednošću intenziteta piksela, butu posmatrani kao jedan piksel. Ovaj problem je potrebno prevazići, i u ovu svrhu je u nastavku detaljnije opisan algoritam za podudaranje blokova. Bolji pristup je da se uzmu u obzir i neki susedni pikseli. To je pristup koji je korišćen u algoritmu za podudaranje blokova. Red koji je skeniran za stereo korespondenciju naziva se linija skeniranja.

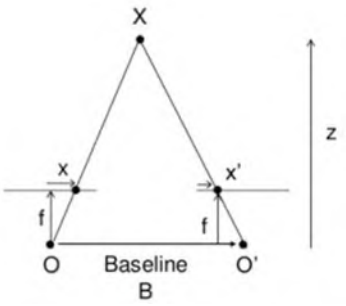
Postoji više metrika koje bi odredile podudaranje blokova, prilikom implementacije sistema je korišćen SAD, odnosno zbir apsolutnih razlika. Na primer, par sa najnižim SAD rezultatom se najbolje podudara jer je ukupni zbir u pikselima najmanji. OpenCV obezbeđuje implementaciju algoritma za podudaranje blokova – klasu StereoBM. Slike u sivim tonovima koje su dobijene su samo karte dispariteta, a ne mape dubine [7][9].

$$disparity = x - x' = \frac{Bf}{Z} \tag{1}$$

$$Bf = M \tag{2}$$

$$depthmap = \frac{M}{disparity} \tag{3}$$

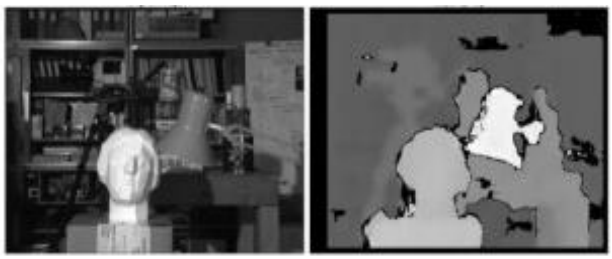
Na osnovu jednačine (3) može se zaključiti da je dubina tačke u sceni obrnuto proporcionalna razlici u udaljenosti odgovarajućih tačaka slike i njihovih centara kamere. Dakle, sa ovim informacijama može se izvesti dubina svih piksela na slici (slika 2) [9].



Slika 2. Vizuelni prikaz informacije o dubini.

Na ovaj način moguće je kreirati mapu dubine od prosleđene ulazne slike. Slika dobijena na ovaj način označava regione slike koji se nalaze na različitim

udaljenostima od kamere na način da su obojeni različitim nijansama sive boje. Regioni obojeni svetlijom bojom se nalaze bliže kameri, a oni tamniji su dalji kameri (slika 3).



Slika 3. Mapa dubine (desno) na osnovu slike (levo).

B. Procena položaja glave u određenom trenutku

Druga tehnika koja je korišćena jeste procena položaja glave u određenom trenutku. Položaj objekta se odnosi na njegovu relativnu orijentaciju u odnosu na kameru. Pre početka analize, potrebno je locirati lice u kadru. Proces dobro funkcioniše sa licima ka kameri, problem kada je lice pod uglom. Dodatno tome: neke karakteristike se ne vide zbog pomeranja glave. Proces se sastoji od tri koraka:

a) Detektovanje lica,
b) Detekcija karakteristika na licu,
c) Određivanje poze.

Potrebne su sledeće informacije:

a) 2D koordinate nekoliko tačaka na slici - potrebne su 2D lokacije nekoliko tačaka na slici, to su u ovom slučaju vrh nosa, brada, levi deo levog oka, desni deo desnog oka, levi deo usana i desni deo usana,
b) 3D lokacije istih tačaka, u ovu svrhu korišćen je generički 3D model,
c) Pretpostavka da je kamera prethodno kalibrisana.

Problem procene poze koji je korišćen u ovom radu je PNP (problem perspektive i tačke). Cilj je pronaći pozu objekta kada je kalibrisana kamera, a poznate su lokacije n 3D tačaka na objektu i njihove odgovarajuće 2D projekcije na slici. Generalno, ovaj proces se sastoji od sledećih koraka: [14]

a) Pronaći 6 karakterističnih tačaka lica,
b) Uzeti standardne 3D koordinate svih karakteristika lica i proceniti rotacione i translacione vektore na vrhu nosa,
c) Za tačnu procenu uneti unutrašnje parametre kamere,
d) Nakon dobijanja vektora, projektovati te 3D tačke na 2D površinu koju predstavlja ulazna slika,
e) Ovako dobijeni uglovi su po x i y osi u odnosu na normalan položaj, na osnovu kojih je moguće zaključiti da li je reč o pravom licu.

C. Neuronske mreže

Treća tehnika koju je neophodno pomenuti, u procesu detektovanja lica sa ulazne kamere jeste i tehnika koja se oslanja na primenu konvolucionih neuronskih mreža. Postoje dve kategorije neuronskih mreža: veštačke i biološke neuronske mreže. Predstavnik bioloških neuronskih mreža je nervni sistem živih bića. Veštačke neuronske mreže su po strukturi, funkciji i obradi informacija slične biološkim neuronskim mrežama, ali se radi o veštačkim tvorevinama. Neuronske mreže se sastoje od mnogo neurona između kojih se prosleđuju informacije, a ovaj tok infomacija može se odvijati na dva načina:

a) Kada uči – informacije ulaze u mrežu putem ulaznog sloja, pobuđuju skrivene slojeve, završavaju u izlaznom (Feedforward),
b) Kada uči da prepoznaje i klasifikuje pojmove – mora da postoji povratna informacija. Ovo se odnosi na poređenje izlaza koji mreža generiše sa izlazom koji je trebalo da generiše (Backpropagation) [11].

Konvolucione neuronske mreže se po arhitekturi mogu svrstati u duboke neuronske mreže. Dobile su ime po konvoluciji, operatoru koji se koristi u obradi slika i signala. Važna osobina neuronskih mreža jeste sposobnost da uče na ograničenom skupu podataka. Mreža se može obučiti na način da prepoznaje objekte na osnovu trening skupova koje mi proizvoljno zadajemo. Ova osobina je iskorišćena prilikom implementacije sistema. Dobra strana kod neuronskih mreža je to da neuronska mreža ne pamti samo određene podatke, nego da „generalizuje“ i uči stvarni problem i obrasce koji su sa njima povezani. Kada je kreiran skup podataka, potrebno je smestiti ih u odgovarajće foldere, gde će svaki od njih označavati zasebnu kategoriju. Najveći problem je što sve ove slike nisu iste veličine. Prvenstveno je potrebno promeniti veličine svim ulaznim fotografijama. Bitna stvar u ovom koraku jeste to da podaci budu izbalansirani odnosno uravnoteženi. Pod uravnoteženim se podrazumeva to da postoji isti broj primera za svaku kategoriju. Ako nije balansiran, potrebno je proslediti težine klasa modelu, tako da može da meri grešku na odgovarajući način, ili uravnotežiti uzorke tako što je potrebno smanjiti veći skup da bude iste veličine kao manji skup, a ovaj pristup je korišćen u analizi podataka [10].

Ako nije balansiran, model će u početku naučiti da je najbolje predvideti samo jednu klasu, koja god je najčešća. Zatim, potrebno je izmešati podatke, odnosno da klasifikator nauči da podjednako predviđa sve kategorije. Konvoluciona neuronska mreža je stekla popularnost korišćenjem podataka o slici i trenutno je najsavremenija za

otkrivanje šta predstavlja slika ili šta je sadržano u slici. Osnovna struktura CNN-a je sledeća:

a) Konvolucija,
b) Objedinjavanje,
c) Konvolucija,
d) Objedinjavanje,
e) Potpuno povezani sloj,
f) Izlaz.

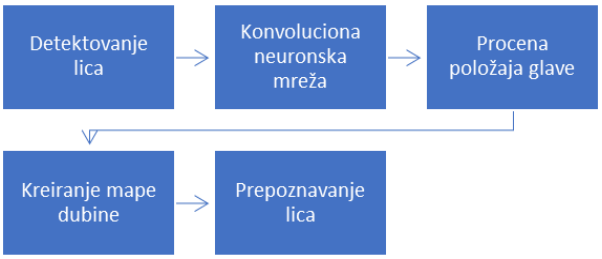
Konvolucija je postupak preuzimanja originalnih podataka i pravljenja mapa karakteristika od njih. Ulaz se konvertuje u piksele. Radi lakšeg opisa izabranog metoda, pretpostavka je da je svaki kvadrat piksel. Zatim, za korak konvolucije se uzima određeni prozor i pronalaze se karakteristike unutar tog prozora. Karakteristike tog prozora su sada samo jedna karakteristika veličine jednog piksela u novoj mapi obeležja, ali u stvarnosti se dobijaju više slojeva mapa obeležja. Zatim se povlači taj prozor i nastavlja se dalje sa procesom. Doći će do preklapanja, ali je bitno da se ne preskače nijedan piksel [12].

Sada se nastavlja ovaj proces dok se ne pokrije cela slika, i tako dobije mapa obeležja. Obično je mapa karakteristika samo više vrednosti piksela, samo veoma pojednostavljena. Odavde se radi udruživanje. Najčešći oblik objedinjavanja je „maksimalno objedinjavanje“, gde se jednostavno uzima maksimalna vrednost u prozoru, a to postaje nova vrednost za taj region. Svaki korak udruživanja je skriveni sloj. Posle ovoga sledi potpuno povezani sloj, a zatim i izlazni sloj. Potpuno povezani sloj je tip sloja tipične neuronske mreže (višeslojni perceptron), a isto je i sa izlaznim slojem. Nakon optimizacije, potrebno je sačuvati ovako kreiran model. Ovaj model se može kasnije učitavati i koristiti za prepoznavanje ulaza koji se naknadno prosleđuju. Mrežu je dovoljno obučiti samo jednom nad određenim skupom podataka, svaki naredni put se pri korišćenju koristi samo njegov kreiran model podataka [13].

IV. Implementacija sistema i primena pomenutih tehnika

Kada je reč o sistemu za prijavljivanje, on se može odnositi na prijavljivanje u okviru neke prostorije, kompanije ili nekog drugog objekta. U konkretnoj realizaciji, cilj ovog sistema je da omogući prijavljivanje na ulazu neke kompanije. Ovaj sistem bi radio na način da identifikuje i prepozna sve zaposlene iz kompanije na čijem ulazu je sistem postavljen. Ukoliko je lice uspešno prepoznato, sistem će dozvoliti otključavanje vrata i ulaz u kompaniju.

U konkretnoj realizaciji projekta je implementiran hipotetički sistem za prijavljivanje koji ima mogućnost da otvara vrata kompanije ili nekog drugog objekta u slučaju da je to lice zaposleno u kompaniji ili da iz drugog razloga ima dozvoljen ulaz u kompaniji ili nekom drugom objektu gde ovaj sistem kontroliše ulaz. Ako se lice nalazi u bazi podataka svih lica kojima je dozvoljen ulaz, vrata će se otključati, u suprotnom, neće, i ovom licu će biti zabranjen ulaz. Arhitektura sistema se sastoji od 5 logičkih celina (slika 4). Grafički prikaz arhitekture sistema dat je na sledećoj slici:



Slika 4. Arhitektura sistema.

Programski jezik koji je korišćen jeste Python. Kao pomoć pri traženju i prepoznavanju lica iskorišćena je biblioteka OpenCV. Biblioteke koje su takođe korišćene jesu numpy, matplotlib, pickle, tensorflow, math, os, random, tqdm. U bazi podataka su dodata sva lica koja predstavljaju zaposlene kompanije, a lice koje se prepoznaje sa ulazne kamere treba da se poklapa sa jednim od ovih lica iz baze. Pored ovih biblioteka, takođe su iskorišćene i biblioteke face_detector, face_landmarks i face_recognition i njihove funkcije koje omogućuvaju učitavanje željene slike, kodiranje određene slike, pronalaženje lokacije lica, upoređivanje dva lica, nalaženje sličnosti i razlika između dva lica.

Biblioteka za prepoznavanje lica radi u više koraka za prepoznavanje lica. U prvom koraku, potrebno je proslediti slike poznatih osoba da bi se obučio model, a biblioteka će koristiti HOG da pronađe lica prisutna na datim slikama.

Ovaj sistem pamti sve zaposlene u okviru dictionary-ja na način da se kao ključ za svakog zaposlenog pamti ID zaposlenog, a kao vrednost se pamti informacija o imenu zaposlenog i o tome da li je prijavljen tog dana. Ukoliko je lice tog dana prethodno detektovano, to znači da se zaposleni prijavio na ulazu kompanije tog dana i da ponovnom detekcijom želi da se odjavi iz firme. Ukoliko je sistem uspešno prepoznao zaposlenog na ekranu, na displeju će biti prikazano da li zaposleni pokušava da se prijavi ili odjavi, ime tog zaposlenog i tačno vreme kada se zaposleni prijavio/odjavio. Nakon uspešnog detektovanja, sistem će nakon par sekundi ponovo ući u režim u kom može da detektuje novog zaposlenog koji dolazi odnosno odlazi sa posla.

Veliki problem koji može da nastane kod sistema implementiranog na ovaj način jeste to da neko pokuša da se prijavi korišćenjem slike zaposlenog, pokušavajući na ovaj način da zavara sistem i da se uspešno prijavi i odobri mu se ulaz u objekat, iako nema prava na to. Iz ovog razloga se kreira mapa dubine, koja će napraviti razliku između stvarnog lica, koje kamera uživo snima, i fotografije pomoću koje neko druge lice pokušava da se prijavi. Iz ovog razloga su potrebne dve kamere za realizaciju sistema koje će određivati dubinu, i na ovaj

način razgraničiti pravo lice od lica sa fotografije koje je podmetnuto. Ova tehnika je detaljnije objašnjena u prethodnom poglavlju. Radi prevazilaženja ovog problema, za implementaciju sistema su korišćene 3 tehnike:

1. Kalibracija kamera radi kreiranja mape dubine,
2. Obučavanje neuronske mreže da prepozna lažna prijavljivanja,
3. Detektovanje položaja glave pomoću dve kamere.

Određivanju dubine prethodi kalibracija kamera. Potrebne su dve kamere, po mogućnosti kamere istog modela, koje bi snimale iz različitih uglova lice i na ovaj način kreirale mapu dubine čiji sadržaj može ukazivati na objekte koji su bliži odnosno dalji kameri, što može predstavljati jedan od ključnih faktora pri detektovanju stvarnog lica.

Prvi korak prilikom kalibracije kamere jeste kreiranje slika šahovske table iz različitih uglova pomoću obe kamere. Često korišćena metoda kalibracije, koja je korišćena i u ovom radu je kalibracija trodimenzionalnom rekonstrukcijom scene pomoću šahovske table. Takva kalibracija sa slika pokušava da zaključi o trodimenzionalnim podacima sa scene, a postupak se svodi na slikanje šahovske table sa poznatim brojem polja u redovima i kolonama sa više pozicija i obradu navedenih slika algoritmima računarskog vida. Za implementaciju kompjuterskih algoritama korišćena je već pomenuta OpenCV biblioteka. Šahovska tabla se koristi uglavnom zbog toga što ima jako izražene ivice crnih i belih polja. Nad svakom od slika se primenjuje određeni algoritam računarskog vida koji prvo prepoznaje ivice i uglove polja, za šta se koristi funkcija cv2.findChessboardCorners() iz OpenCV biblioteke. Ideja ovog dela postupka je da se koordinatni sistem postavi u izabrani ugao šahovske table na slici i određivanje transformacije njenog koordinatnog sistema u sistem kamera. Za određivanje opisane transformacije treba znati stvarnu dužinu ivica šahovske table.

U svrhu detektovanja stvarnih i lažnih lica, kreirana je i obučena konvoluciona neuronska mreža. Ovo je binarna neuronska mreža koja kao izlaz može da vrati jedan od moguća dva izlaza, a to su: da li je reč o pravom licu, ili se radi o lažnom prijavljivanju. Ova mreža je obučena na skupu slika sa pravim licima i slika sa podmetnum licima na kojima je prikazan pokušaj logovanja putem fotografije ili lica sa telefona. Na ovaj način, mreža je obučena tako da prepozna pokušaj logovanja putem podmetnutog lica i na ovaj način omogućava zaštitu objekta i neovlašćeni pristup. Za ovaj pristup je kreiran odgovarajući skup podataka sa stvarnim licima i lažnim prijavljivanjem, konkretno dva foldera koja bi sadržala primere iz pomenutih grupa.

Određivanje položaja glave pomoću dve kamere takođe može biti jedan od najvažnijih pokazatelja da li je reč o pravom licu ili ne. Svaka od kamera će detektovati pod kojim uglom se lice nalazi u odnosu na kameru i kao povratne vrednosti vratiti određene uglove po x i y osi. Ako je reč o fotografiji koja se podmeće radi detektovanja lica ili o slici sa telefona, položaj glave će biti isti. Ukoliko se radi o stvarnom licu, položaj glave će se razlikovati u zavisnosti pod kojim uglom se lice nalazi u odnosu na kameru. Ukoliko su ovi uglovi različiti, to je jasan pokazatelj da je reč o pravom licu.

Kao što je ranije pomenuto, postoje tri kriterijuma koja moraju biti ispunjena kako bi sistem smatrao da je lice u kadru kamera stvarno. Kriterijumi koji moraju biti zadovoljeni jesu sledeći:

a) prethodno obučena binarna konvoluciona neuronska mreža kao rezultat obrade ulaznog snimka vraća informaciju,
b) procena položaja glave sa obe kamere se analizira i vraća informaciju o tome da je reč o pravom licu,
c) kreirana mapa dubine na jasan i pregledan način daje informaciju o tome šta je bliže kamerama, a šta dalje, odnosno konturom lica jasno označava lice koje je u kadru.

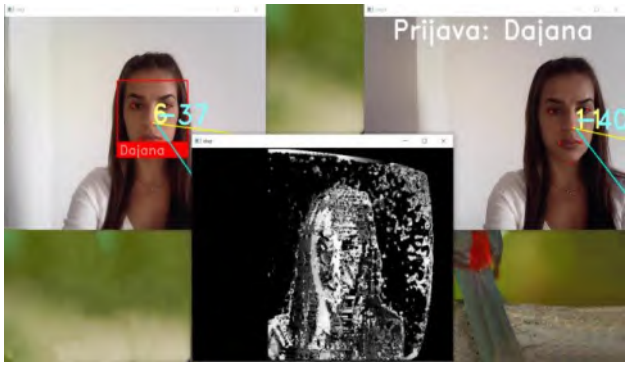
Ukoliko je svaki od kriterijuma ispunjen, odgovarajući fleg je setovan. Potrebno je da svi flegovi budu setovani kako bi se lice smatrali detektovanim. Ukoliko nijedan od flegova nije postavljen, dobija se jasna informacija da je reč o pokušaju prijavljivanja putem fotorafije ili telefona. U određenim slučajevima, sistem može pogrešno protumačiti ulazni snimak sa kamera, pa neki od flegova može biti pogrešno setovan ili ne. U ovim slučajevima, kada svi flegovi ne daju jednoznačno informaciju, potrebno je ostati u fokusu kamere dok sistem ne utvrdi rezultat prijave. Korisnik će u zavisnosti od analize i rezultata obrade, na ekranu dobiti odgovarajuće poruku:

a) prijava je regularna, pa se ispisuje ime osobe koja se prijavljuje, i da li je reč o prijavi ili odjavi. U ovom slučaju se razmatra da li se lice prethodno prijavilo, ovaj podatak se iščitava iz baze u kojoj se za sva lica, kojima je dozvoljen ulaz, pamti i informacija o tome da li su se tog dana već prijavili ili ne,
b) prijava nije regularna, i na ekranu se prikazuje poruka da prijavljivanje nije moguće,
c) poruka da je potrebno ostati u kadru još par trenutaka dok sistem jednoznačno ne odredi da li je prijavljivanje moguće.

Celokupnom postupku prethodi proces kalibracije kamere. Nakon fiksnog postavljanja kamera u određeni položaj, kreirane su slike šahovske table pomoću obe kamere, a kasnije je pokrenut i proces kalibracije kamera u tom položaju. Ceo proces detektovanja, a i kasnijeg prepoznavanja lica se ponavlja na određeni broj fremova koji je unapred zadat.

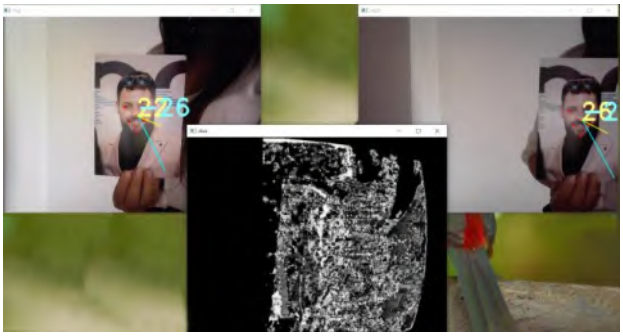
Važna stvar, koju je neophodno napomenuti, jeste ta da je kod ovakvog sistema adekvatno prijavljivanje sa velikom stopom tačnosti (iznad 90 %) moguće samo onda kad se prijavljivanje radi individualno, odnosno kada je

samo jedno lice u kadru. Prilikom pokretanja aplikacije, biće prikazana tri prozora, dva prozora koja predstavnjaju snimke sa leve, odnosno desne kamere, i treći, koji predstavlja mapu dubine dobijenu kombinacijom frejmova sa leve i desne kamere (slike 5 i 6). Nad kamerama je izvršen proces kalibracije. U nastavku su prikazani prozori koji se otvaraju nakon pokretanja aplikacije.



Slika 5. Prikaz prozora nakon pokretanja aplikacije.

Crvenim tačkicama su označene osnovne karakteristike na licu kao što su oči, nos, levi i desni deo usana i brada. Ove karakteristike mogu pomoći u procesu određivanja položaja glave, a kasnije i u procesu prepoznavanja lica sa slike. Žutom i plavom bojom su označeni uglovi pomoću kojih se određuje pozicija glave. Što su kamere udaljenije jedna od druge, to će ovi uglovi biti različitiji, što jeste jasan pokazatelj lica u prostoru. U trećem prozoru je prikazana mapa dubine. Na njoj se jasno može videti lice u prostoru koje je označeno belom bojom i pozadina koja je predstavljena crnom bojom. Na ovom snimku je prikazana jasna percepcija dubine. U drugom slučaju, ukoliko je pokušano prijavljivanje korišćenjem fotografije, dobiće se sledeći prikaz prozora:



Slika 6. Prikaz prozora nakon pokretanja aplikacije.

Ukoliko lice nije prepoznato kao pravo, ono neće uopšte ući u proces prepoznavanja. Na ovoj slici takođe se može zaključiti da nema mnogo informacija iz mape dubine, odnosno da sistem ne može da odredi dubinu iz ovih snimaka. Informacije iz mape dubine se koriste na način da se dobijena mapa upoređuje sa modelom idealne mape dubine kada se na njoj nalazi lice u realnom prostoru. Zatim, nalazi se sličnost između dobijene mape dubine i idealnog modela. Ukoliko je sličnost veća od unapred zadatog praga, to znači da je reč o stvarnom licu. U zavisnosti od rezolucije kamere, položaja kamera, udaljenosti između kamera i složenosti pozadine, potrebno je u nekim slučajevima naknadno podesiti parametre vezane za kreiranje mape dubine, kao što su veličina bloka, tip pre - filtera, veličina pre – filtera itd.

Neuronska mreža pozadinski određuje da li je reč o pravom licu. Ovu mrežu nije neophodno uvek obučavati prilikom korišćenja aplikacije, već je potrebno obučiti je samo jednom, a kasnije učitavati tako kreirani model. Mreža je obučena na način da jasno prepoznaje stvarna lica i ona na fotografijama ili telefonu, i kao rezultata vraća jednu od ove dve informacije. Skup podtaka koji se koristi u svrhu obučavanja mreže treba da bude dovoljno veliki tako da mreža može jasno prepoznati i klasifikovati ove dve grupe. Preporuka je da oba skupa sadrže isti ili približno jednak broj test primera kako bi mreža bila podjednako dobro obučena da prepozna oba tipa, a tako da ne bude prenaučena za određeni skup trening podataka.

Prepoznavanje lica sledi tek onda kada je lice uspešno detektovano i ispuni sve prethodno pobrojane kriterijume, odnosno kada su svi flegovi uspešno setovani. Vrši se kodiranje lica sa kamere i upoređuje se sa licima iz baze. Odgovarajuće podudaranje je znak da je to lice iz baze, a svako kodiranje je vezano određenim imenom, i kao povratna vrednost podudaranja dobijamo ime osobe sa ulaznog snimka. U slučaju kada je reč o pravom licu, ali lice se ne nalazi u bazi, na ekranu će biti prikazana poruka da je lice nepoznato odnosno “Unknown”.

V. Zaključak

Sistem implementiran u ovu svrhu može olakšati prepoznavanje lica, a može se iskoristiti kao sistem za prijavljivanje na ulazu kompanija, ili za pristup nekim objektima ili zaštićenim prostorijama. Sistem zasnovan na prijavljivanju samo putem lica, implementiran na ovaj način, pruža ogromnu sigurnost i zaštitu od neovlašćenog i neželjenog upada. Kako sistem koji je opisan u ovom radu dostiže veliku tačnost (iznad 90%), ovakvi sistemi daju temelje za razvoj ozbiljnih sistema koji bi se mogli komercijalno prodavati, a spajanjem postojećih algoritama, dobija se primena koja je izuzetno široka. Pored velike tačnosti, infrastruktura koja je potrebna za realizaciju ovakvog sistema nije previše ekonomski zahtevna, što takođe omogućava primenu na svetskom nivou.

Dalje unapređenje sistema je moguće u cilju poboljšanja performansi i brzine izvršenja programa. Ukoliko se sistem koristi u objektima gde je ulaz dozvoljen većem broju lica, jedan od problema može biti i način skladištenja i čuvanja celokupne baze. Jedan od predloga za dalje unapređenje sistema može biti korišćenje eksterne baze podataka ili prebacivanje sadržaja aplikacije u Cloud kako bi se smanjilo zauzeće skladišnog prostora na uređaju, kao i omogućilo lakše ažuriranje sadržaja aplikacije.

Skup trening podataka, koji se koristi za obučavanje neuronske mreže, moguće je proširiti na način da bude dovoljno veliki tako da su svi slučajevi prilikom detektovanja pokriveni. Ovo se ogleda u načinu detektovanja različitih izraza lica, posebno onih koji su praćeni okluzijama, lošim osvetljenjem, rezolucijom, čudnim izrazima i slično.

LITERATURA

[1] Wikipedia, Histogram of oriented gradients, https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients [poslednji pristup 07.08. 2022.]

[2] Ashish Arya, Key Facial points detection and face recognition, https://medium.com/ai-techsystems/key-facial-points-detection-and-face-recognition-9bce9bf84750 [poslednji pristup 27.08. 2022.]

[3] Adrian Rosebrock, Face recognition with OpenCV, Python and deep learning, https://pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/ [poslednji pristup 25.08. 2022.]

[4] Wikipedia, Support-vector machine, https://en.wikipedia.org/wiki/Support-vector_machine [poslednji pristup 15.07. 2022.]

[5] José Augusto Cadena Moreano, Nora La Serna Palomino, Alex Christian Llano Casa, Facial recognition techniques using SVM: A comparative analysis, https://www.redalyc.org/journal/5722/572261719008/html/ [poslednji pristup 15.07. 2022.]

[6] Baeldung, Disparity map in Stereo Vision, https://www.baeldung.com/cs/disparity-map-stereo-vision [poslednji pristup 05.10. 2022.]

[7] Wikipedia, Sum of absolute differences, https://en.wikipedia.org/wiki/Sum_of_absolute_differences [poslednji pristup 02.09. 2022.]

[8] Satya Mallick, Camera calibration using OpenCV, https://learnopencv.com/camera-calibration-using-opencv/ [poslednji pristup 15.08. 2022.]

[9] Kaustubh Sadekar, Depth perception using a stereo camera, https://learnopencv.com/depth-perception-using-stereo-camera-python-c/ [poslednji pristup 07.06. 2022.]

[10] Velibor Ilić, Neuronske mreže, http://solair.eunet.rs/~ilicv/neuro.html [poslednji pristup 28.06. 2022.]

[11] Casper Hansen, Neural Networks: Feedforward and Backpropagation Explained & Optimization, https://mlfromscratch.com/neural-networks-explained/#/ [poslednji pristup 23.08.2022.]

[12] Manav Mandal, Introduction to Convolutional Neural Networks(CNN), https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/ [poslednji pristup 22.06. 2022.]

[13] TopCoder, Convolutional neural networks in Pytorch, https://www.topcoder.com/blog/convolutional-neural-networks-in-pytorch/ [poslednji pristup 19.10. 2022.]

[14] Wikipedia, Perspective-n-Point, https://en.wikipedia.org/wiki/Perspective-n-Point [poslednji pristup 20.10. 2022.

Primena dronova za sejanje drveća

Bogdan Stanković, Predrag Šubarević

Sadržaj − Cilj ovog projekta je da se u što kraćem vremenskom roku i minimalnoj potrošnji energije posadi što je više moguće novih stabla.

I. UVOD

Drveće je glavni izvor prerađenog kiseonika u atmosferi. Ono predstavlja pluća naše planete. Drveće doslovno udiše ugljen-dioksid kroz tu svoju veliku površinu listova, nakon što ga mi izdahnemo kao biološki otpad, i izdiše kiseonik kao svoj biološki otpad. Bez drveća, naša atmosfera bi postala otrovna za nas.

Bez šumskog zemljišta koje pumpa milione tona vode u atmosferu, bilo bi manje vlage koja bi se oslobodila u atmosferu da se kondenzuje u oblake i da kasnije da kišu. Rezultat je da tamo gde je nekada bila šuma, a danas je razgolićen teren, kiše dugo ne padaju i počinje proces formiranja pustinje. To se dešava u većem delu severne i istočne Afrike, dovodeći do velikih gladi. Dok ovo čitamo, više od 600 hektara kopna postaje pustinja širom sveta na svakih sat vremena, velikim delom zbog uništenja šuma. Ukupna količina kišnihšuma koja je preostala na našoj planeti je otprilike kao površina kontinentalnog dela SAD-a, i svake godine površina veličine države Florida se poseče i trajno uništi [1].

Skoro trećini svetskih vrsta drveća preti izumiranje, dok su stotine na rubu nestanka, pokazuje novi izveštaj. Značajna studija, koju je objavila organizacija Botanic Gardens Conservation International (BGCI), pokazala je da je oko 17.500 vrsta drveća, ili 30 posto od ukupnog broja, u opasnosti od izumiranja, dok 440 vrsta ima manje od 50 primeraka u divljini [2].

II. PRINCIP RADA DRONA

Prototip drona (slika 1.) se sastoji od četiri motora i četiri elise koji su zaduženi da nose letelicu kroz vazduh. U zavisnosti od težine baterije, količine i težine semena koje nosi može imati i više od četiri motora i četiri elisa.

Da bi dron bio stabilan u vazduhu koristi žiroskop sa tri ose (X, Y, Z). Dron poseduje kameru sa prenosom uživo preko koje lice koje upravlja letelicom što omogućava precizniji rad i na većim udaljenostima. Kamera ima izdvojeno napajanje zbog boljeg kvaliteta snimka. Prednost izdvojenog napajanja je duzi vremenski rok beleženja podataka u slučaju nezgode. Glavni deo ove letelice je mehanizam za izbacivanje semena (slika 2.).

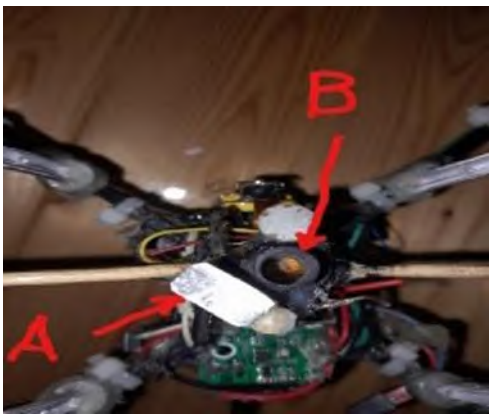


Slika 1. Prototip drona.

U zavisnosti od tipa zemljišta i samog semena mogu se primeniti dva tipa mehanizma:

1) sa ubrzanjem zemljine teže (slobodan pad),
2) sa dodatnim ubrzanjem.

Kod ovog prototipa je zastupljena stavka pod brojem 1) tj. izbacivanje se vrši pomoću slobodnog pada. Naime, sa donje strane drona nalazi se kapsula u kojoj se skladišti seme. Da se sadržaj kapsule ne bi prosuo, ispod kapsule nalazi se zatvarač.



Slika 2. Mehanizam za izbacivanje semena.

Zatvarač funkcioniše tako što se zarotira u jednu stranu i samim tim ceo sadržaj kapsule se prepušta

B. Stanković je učenik trećeg razreda ETŠ „Nikola Tesla“, Aleksandra Medvedeva 18, Niš, Srbija, E-mail: stankovicbogdan969@gmail.com.

P. Šubarević, mast. inž. Mehatronike i upravljanja, profesor u ETŠ „Nikola Tesla“, Aleksandra Medvedeva 18, Niš, Srbija, E-mail: p.subarevic92@gmail.com.

slobodnom padu. Da bi se klapna zarotirala, potrebno je da se pošalje signal sa mobilnog uređaja do čipa na kome je povezan elektro motor koji pokreće klapnu. Nakon izvršene radnje, elektro motor se automatski isključuje. Takođe postoji funkcija manuelnog gašenja motora u slučaju kvara. Pomoću upravljača (slika 3.) se kontroliše kretanje drona. Leva palica ima funkciju kretanja gore, dole i rotaciju u oba smera, dok desna palica ima funkciju kretanja napred, nazad, levo i desno. Poseduje i taster preko kojeg se reguliše brzina drone.

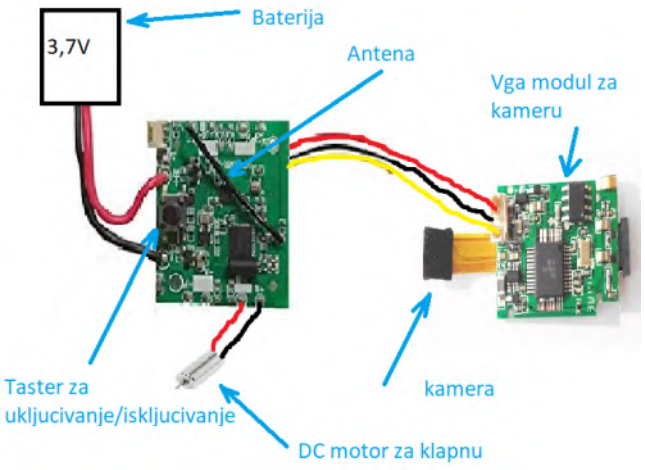


Slika 3. Upravljač.

III. Realizacija projekta

Na osnovu svih gore navedenih činjenica realizovan je dron čija je šema prikazana na slici 4. Za realizaciju ovog projekta je upotrebljeno:

1) baterija za napajanje kamere i zatvarača (3,7 V, 250 mAh),
2) baterija za napajanje četiri pogonska motora i osvetljenja (3,7 V/mAh),
3) dve crvene i dve led diode (2 V, 20 mA),
4) kamera 1080 p (30 fps),
5) čip za upravljanje dronom (DCH-330),
6) čip za upravljanje kamerom i zatvaračem (KK8),
7) jedna prijemna i dve predajne antene (2,4 GHz),
8) četiri prenosa sa pogonskim i gonjenim zupčanikom,
9) četiri DC motora 3,7 V (17x20 mm),
10) jedan DC motor 3,7 V (7x16 mm),
11) četiri elise.



Slika 4. Šema veze modula kamere i otvarača klapne.

IV. Zaključak

Prednost ove letelice je to što može da stoji u mestu za razliku od aviona tj. mnogo je preciznije i lakše ispustiti seme na pravo mesto.Dron je ispunio sva očekivanja. Na visini od 50 m uspešno je ispustio seme u krugu od 3 m. Korišćenje drona u svrhu sađenja drveća pozitivno utiče na svest ljudi o očuvanju životne sredine i pospešuje da se ono uspešno i efikasno sprovodi.

Literatura

[1] http://sumapeva.rs/content/zna%C4%8Daj-%C5%A1uma-i-posledice-njihovog-uni%C5%A1tenja
[2] https://www.podunavlje.info/dir/2021/09/02/bgci-alarmantno-stanje-cak-trecini-vrsta-drveca-na-svetu-preti-izumiranje

Statistička analiza probojnog napona i vremena odziva gasnog odvodnika proizvođača *EPCOS* 350 V

Mikica Radenković

Sadržaj – U ovom radu prikazani su rezultati ispitivanja probojnog napona i vremena odziva gasnog odvodnika proizvođača *EPCOS*. Ispitivanja su vršena u kontrolisanim laboratorijskim uslovima, primenom i analizom raspodela vremena kašnjenja električnog proboja. Eksperimentalni podaci srednje vrednosti probojnog napona su dobijeni za brzine porasta napona od 1 do 10 V/s. Analiza podataka izvršena je dinamičkom diskretizovanom metodom.

I. UVOD

Zaštita od prenapona je veoma bitna kod sofisticirane opreme od koje se očekuje visokog stepen pouzdanosti u radu, kao na primer u nuklearnim postrojenjima, vojnoj industriji, avionima. Najveću primenu u zaštiti uređaja od prenapona imaju zaštitne komponente za koordinaciju izolacije na niskonaponskom nivou. Elementi prenaponske zaštite na niskonaponskom nivou mogu se podeliti na nelinearne i linearne. U nelinearne elemente spadaju prenaponske diode, varistori i gasni odvodnici prenapona, dok u linearne spadaju različite vrste električnih filtara.

Gasni odvodnici prenapona (eng. *Gas Filled Surge Arresters*) pripadaju grupi najčešće korišćenih nelinearnih elemenata za zaštitu od prenapona. Rade na principu električnog proboja gasova pri niskom pritisku i mogu se koristiti za zaštitu od prenapona intenziteta od nekoliko desetina volti do nekoliko desetina hiljada volti.

Princip rada gasnih odvodnika prenapona zasniva se na električnom proboju u gasovima. Proučavanje proboja u gasovima i formiranje plazme je važno zbog njihove primene u tehnici dobijanja tankih filmova, poluprovodničkih procesa (nagrizanje i spaterovanje), elektronskih cevi, izvora svetlosti, displeja i analize materijala.

Elektronsko pražnjenje je pojava protoka naelektrisanja kroz neku sredinu pod dejstvom električnog polja. Gas je pod normalnim uslovima izolator tj. ne provodi električnu struju. Međutim, kada je izložen dejstvu električnog polja dovoljne jačine, gas postaje i počinje proces koji se naziva električno pražnjenje kroz gas.

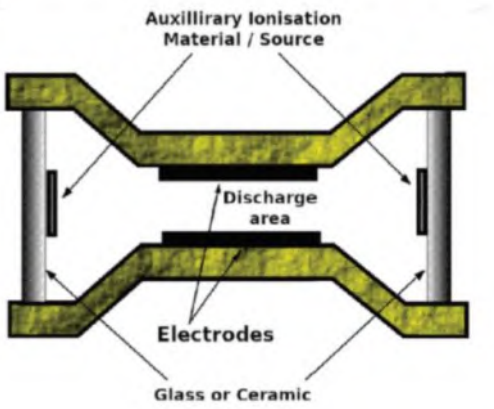
II. GASNI ODVODNIK *EPCOS* 350 V

Gasni odvodnici [1] prenapona spadaju u grupu nelinearnih elemenata koji se koriste za zaštitu od prenapona. Konfiguracija gasnih odvodnika može se sastojati od dve ili tri elektrode koje su smeštene u keramičko ili stakleno kućište (Slika 1).



Slika 1. Gasni odvodnici a) dvopinski i b) tropinski.

Gasni odvodnik *EPCOS* 350 V se sastoji od dve elektode, odnosno u pitanju je dvopinski gasni odvodnik. Elektrode, su ravne i paralelne kako bi se u međuelektrodnom prostoru obezbedilo približno homogeno električno polje, se nalaze u kućištu koje se sastoji od keramike i metala. Rastojanje između elektroda je reda milimetra ili dela milimetra. Kao izolacioni materijal koriste se ili plemeniti gasovi (argon, neon, kripton ili ksenon) ili njihove smeše na pritiscima od 100 Pa do 70 kPa. Na Slici 2 prikazan je poprečni presek strukture jednog gasnog odvodnika.



Slika 2. Poprečni presek gasnog odvodnika prenapona.

Tehničke specifikacije gasnog odvodnika prenapona *EPCOS* 350 V su: probojni napon 350 V DC sa tolerancijom od 20 %, izolatorska otpornost 10 GΩ na 100 V i kapacitivnost 1.5 pF. Detaljni pregled tehničkih karakteristika prikazan je na Slici 3.

M. Radenković je student master studija modula Elektronika i mikrosistemi na Katedri za mikroelektroniku, Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: mikica.radenkovic@elfak.rs.

Electrical specifications

DC spark-over voltage [1] [2]	350 ± 20	V %
Impulse spark-over voltage at 100 V/µs - for 99% of measured values - typical values of distribution at 1 kV/µs - for 99% of measured values - typical values of distribution	 < 700 < 650 < 900 < 800	 V V V V
Service life [8]		
10 operations 50 Hz; 1 s	20	A
1 operation 50 Hz; 0.18 s (9 cycles)	100	A
10 operations 8/20 µs	20	kA
1 operation 8/20 µs	25	kA
1 operation 10/350 µs	2.5	kA
300 operations 10/1000 µs	200	A
Insulation resistance at 100 V_{DC}	> 10	GΩ
Capacitance at 1 MHz	< 1.5	pF
Arc voltage at 1 A Glow to arc transition current Glow voltage	~ 15 ~ 0.5 ~ 60	V A V
Weight	~ 2.5	g
Operation and storage temperature	-40 ... +90	°C
Climatic category (IEC 60068-1)	40/ 90/ 21	
Marking, blue negative	**EPCOS** 350 YY O 350 - Nominal voltage YY - Year of production O - Non radioactive	

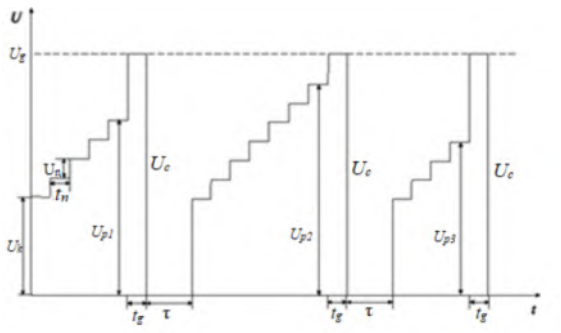
[1] At delivery AQL 0.65 level II, DIN ISO 2859
[2] In ionized mode

Slika 3. Tehnička specifikacija gasnog odvodnika EPCOS 350 V [1].

III. PROBOJNI NAPON GASNOG ODVODNIKA

Napon pri kojem se javlja električni proboj gasa naziva se probojni napon, predstavlja jedan od važnih parametara za praćenje proboja gasova. Električni proboj gasa može biti statički ili dinamički zavisno od oblika primenjenog napona [2]. Dinamički proboj nastaje kada je brzina promene napona jednaka ili veća od brzine promene elementarnih procesa u gasu. Statički proboj se javlja kada je brzina promene napona mnogo manja od brzine promene elementarnih procesa koji dovode do proboja.

Slika 4 prikazuje šematski prikaz stepenastog porasta napona do proboja. Napon U_n predstavlja naponski korak za čiju vrednost se povećava fiksna vrednost napona, pri čemu se unapred zadaje vreme trajanja naponskog koraka t_n. Brzina porasta napona se definiše odnosom k=U_n/t_n . Kada nastupi proboj, naponom U_c se obezbeđuje zadata vrednost struje pražnjenja i_g za vreme pražnjenja t_g. Bitna veličina je vreme relaksacije τ koju je potrebno unapred zadati. To je vremenski interval tokom koga je napon na elektrodama gasne komponente isključen.



Slika 4. Stepenasti porast napona na gasnoj komponenti za seriju od tri merenja.

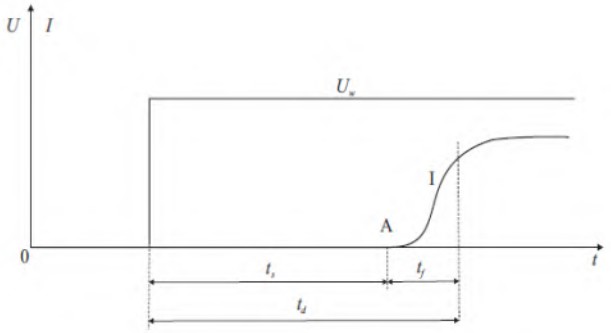
Nakon isteka vremena relaksacije proces počinje od početka. Poželjno je, radi što preciznijeg određivanja napona U_p parametre U_n, t_n, i_g i τ održavati konstantnim. τ se takođe definiše kao vreme od momenta isključivanja napona U_c do ponovnog priključivanja napona U_k na gasnu komponentu.

IV. VREME ODZIVA GASNOG ODVODNIKA

Vreme kašnjena električnog proboja najčešće podrazumeva vreme koje protekne od momenta priključenja radnog napona U_w na elektrode gasne komponente, koji je veći od statičkog probojnog napona U_s, do nastanka proboja, tj. dostizanja zadate vrednosti struje pražnjena i_g. Gasovi imaju karakteristiku veće inertnosti pri proticanju električne struje u odnosu na provodnike. Proboj se ne dešava u trenutku priključivanja napona, čak i onda kada je inicijalni elektron oslobođen sa katode i priključeni napon veći od statičkog probojnog napona U_s. Proboj nastaje posle određenog vremena koje se naziva vreme kašnjenja električnog proboja. Ova pojava je svojstvena za sve gasove [2].

Na Slici 5 je prikazan porast struje od neke male vrednosti do saturacione. Vreme kašnjenja električnog proboja sastoji se iz statističkog vremena kašnjenja t_s i vremena formiranja pražnjenja t_f, dok A predstavlja trenutak pojavljivanja inicijalnog elektrona [3].

$$t_d = t_s + t_f \quad (1)$$

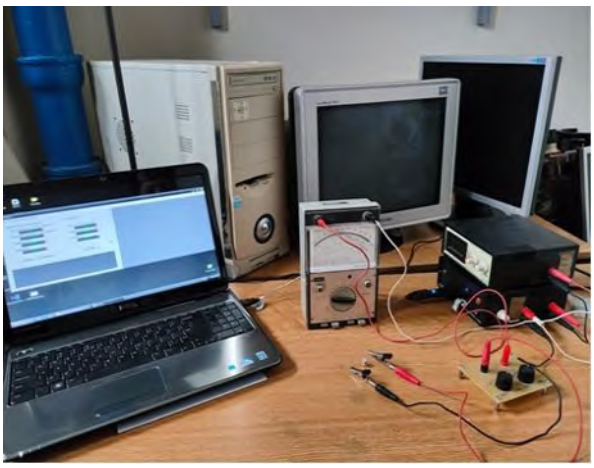


Slika 5. Šematski prikaz vremena kašnjenja električnog proboja t_d.

V. EKSPERIMENT

Merenja probojnog napona i vremena kašnjenja gasnog odvodnika *EPCOS* 350V vršena su kontrolisanim laboratorijskim uslovima. Na Slici 6 prikazano je radno okruženje i realizacija sistema za merenje probojnog napona i vremena kašnjenja gasnog odvodnika.

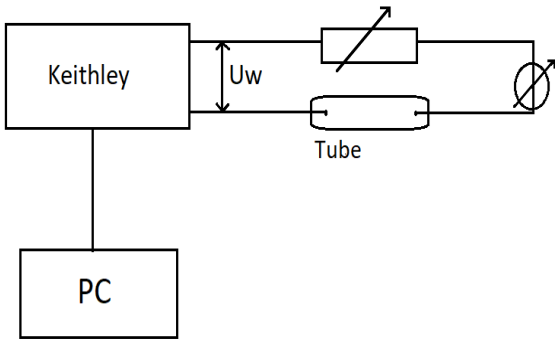
Ispitivanje gasnog odvodnika vrši se sledećim redosledom. Prvobitno je potrebno izabrati model gasnog odvodnika koji će biti korišćen, tj. u ovom slučaju *EPCOS* na radnom naponu 350 V.



Slika 6. Radno okruženje i realizacija sistema za merenje probojnog napona i vremena kašnjenja gasnog odvodnika.

Zatim sledi povezivanje gasnog odvodnika pomoću odgovarajućih konektora na izvor napajanja i računar koji će beležiti podatke za različite vrednosti koje je ispituju. Analize su izvršene sa vremenima relaksacije za komponentu *EPCOS* 350 V u opsegu [1 – 30 000] ms. Broj merenja u jednom ciklusu pri identičnim eksperimentalnim uslovima je postavljen na 100.

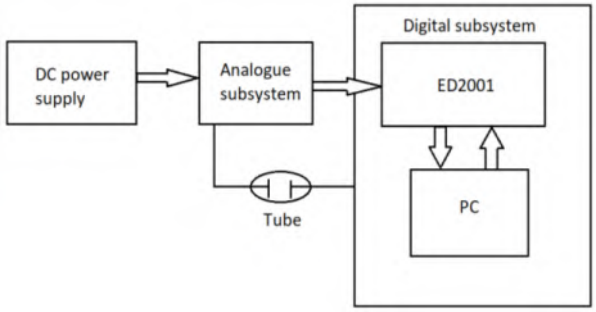
Akvizicija podataka probojnog napona vršena je računarom koji sadrži program za podržavanje ovih merenja. Uređaj ima mogućnost da za zadato vreme poveća napon čija se vrednost programski zadaje. Na Slici 7 data je blok šema uređaja za merenje probojnog napona [2].



Slika 7. Blok šema uređaja za merenje probojnog napona.

Merenje probojnog napona vršeno je na gasnom odvodniku *EPCOS* 350 V. Merenja su vršena na sobnoj temperaturi, pri različitim brzinama porasta napona i to od k=1 V/s do k=10 V/s. Korišćenjem dinamičke metode za određivanje statičkog napona određeno je da on iznosi U_s=444 V za gasni odvodnik *EPCOS* 350 V. Isti gasni odvodnik, istih karakteristika je kasnije korišćen za merenje vremena kašnjenja.

Princip merenja vremena kašnjenja električnog proboja se obavlja u računarski vođenom eksperimentu, što podrazumeva da se pomoću računara zadaju vrednosti nekih parametara vezanih za režim rada (vreme pražnjenja i vreme relaksacije gasa), obavlja se eksperiment (izvršava dati broj merenja u ciklusu), meri vreme kašnjenja, prikupljaju podaci realizovanih merenja i smeštaju u odgovarajuće baze podataka. Korišćenjem softvera koji naizmenično prikazuje uneto vreme relaksacije i vreme kašnjenja, vršeno je merenje vremena kašnjenja električnog proboja za gasni odvodnik *EPCOS* 350 V. Na Slici 8 prikazan je sistem za merenje vremena kašnjenja [2].



Slika 8. Sistem za merenje vremena kašnjenja gasnog odvodnika.

Pri merenjima vremena kašnjenja za različite vrednosti vremena relaksacije korišćen je prethodno procenjen statistički probojni napon U_s= 444 V. Merenja su vršena pri radnom naponu U_w=110%U_s, U_w=115%U_s i U_w=120%U_s, struji pražnjenja i_g= 0.5 mA, vremenu pražnjenja t_g=1 s. Izvršeno je po 100 merenja. Vreme relaksacije τ se kretalo u opsegu od 1 do 30000 ms.

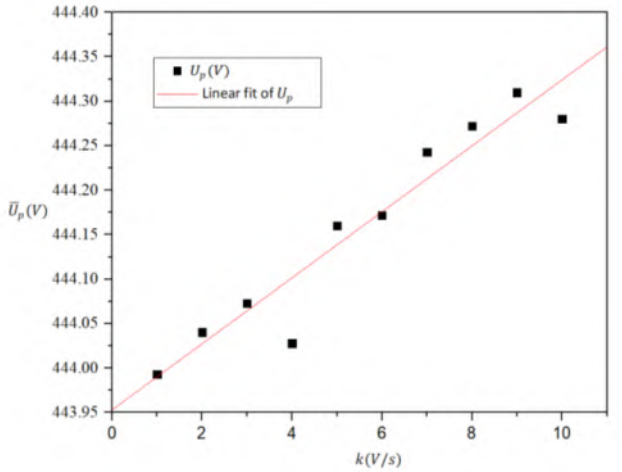
VI. REZULTATI

Dinamička metoda merenja probojnog napona [2] vrši se u uslovima linearnog porasta napona u seriji ponovljenih merenja pri istim uslovima i izmerene vrednosti su različite i podležu određenoj raspodeli. Ovo nije slučaj samo za velike brzine porasta napona, već i pri malim brzinama porasta napona (npr. 0.1 V/s), što znači da se ni tada ne može reći da se radi o statičkom probojnom naponu, već o determinističkoj veličini. Dinamičkom metodom procena statičkog probojnog napona gasnog odvodnika *EPCOS* 350 V) se dobija fitovanjem U_P i presekom sa osom kao što je prikazano na Slici 9.

Vrednost statičkog probojnog napona U_S dobija se presekom fitovane krive sa U_P osom (za k=0) i u ovom slučaju iznosi približno 444 V, za gasni odvodnik *EPCOS* 350 V. Maksimalna vrednost priključenog napona koja je korišćena u eksperimentu iznosi 533 V, 20% veća vrednost u odnosu na statički probojni napon, veće vrednosti nisu korišćene iz bezbednosnih razloga.

Na Slici 10 prikazane su memorijske krive, odnosno zavisnost srednjih vrednosti vremena kašnjenja od vremena relaksacije, za tri različite vrednosti radnog napona gasnog odvodnika *EPCOS* 350 V. Dobijeni rezultati pokazuju da kada se na *EPCOS* 350 V priključi vrednost radnog napona U_w=1.2 U_s, gasni odvodnik najbrže reaguje tj. ima najkraće vreme kašnjenja, dok kada je priključen na radni napon od

U_w=1.1 U_s gasni odvodnik najsporije reaguje odnosno ima najduže vreme kašnjenja.



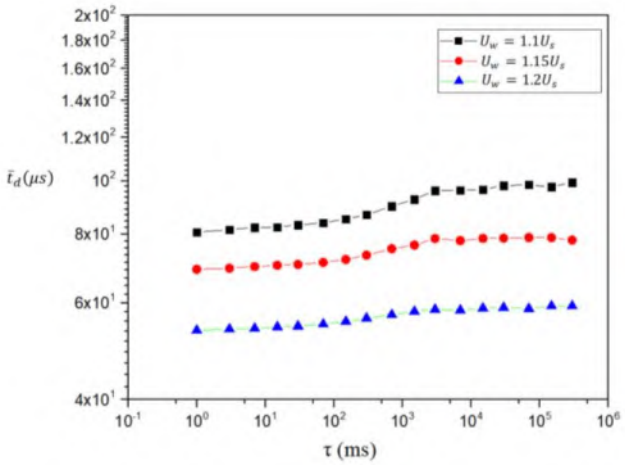
Slika 9. Srednja vrednost probojnog napona U_P funkciji brzine porasta napona k za komponentu *EPCOS* 350V.

TABELA I

PRIKAZ BRZINE PORASTA NAPONA *K* I SREDNJA VREDNOST I STANDARDNE DEVIJACIJE PROBOJNOG NAPONA ZA EPCOS 350 V

k(V/s)	Up(V)	σ (V)
1	444	0.7
2	444.1	0.6
3	444.1	0.6
4	444	0.7
5	444.1	0.6
6	444.2	0.6
7	444.2	0.6
8	444.3	0.7
9	444.3	0.6
10	444.3	0.7

Statističko vreme kašnjenja električnog proboja predstavlja veličinu koja zavisi od verovatnoće oslobađanja elektrona u međuelektrodnom prostoru cevi i od verovatnoće P da slobodni elektron dovede do proboja. Na niskim pritiscima, ako se pretpostavi da je $P \approx 1$, statističko vreme kašnjenja električnog proboja zavisi od brzine oslobađanja inicijalnih elektrona. Dakle, odnos brzine oslobađanja elektrona sa katode i srednje vrednosti statističkog vremena kašnjenja električnog proboja predstavlja vrlo bitan podatak [4]. Vreme kašnjenja električnog proboja predstavlja zavisnost srednje vrednosti vremena kašnjenja od vremena relaksacije, ta zavisnost predstavlja memorijske krive. Statističko vreme kašnjenja električnog proboja predstavlja veličinu koja zavisi od verovatnoće oslobađanja elektrona u međuelektrodnom prostoru cevi i od verovatnoće P da slobodni elektron dovede do proboja.



Slika 10. Prikaz vremena kašnjenja za tri različite vrednosti radnog napona gasnog odvodnika *EPCOS* 350V.

ZAHVALNICA

Zahvaljujem se prof. dr Emiliji Živanović na saradnji i izvanrednoj stručnoj pomoći i podršci koju mi je pružila tokom celog studiranja, kako prilikom izrada projekata, tako i prilikom izrade diplomskog i master rad. Veliku zahvalnost dugujem i celokupnoj Katedri za mikroelektroniku.

LITERATURA

[1] Tehnička dokumentacija za gasni odvodnik EPCOS 350V, dostupno na: https://www.tdk-electronics.tdk.com/inf/100/ds/EC350X-X0810S102.pdf

[2] Momčilo M. Pejović, Milić M. Pejović, „Električni proboj gasova. Merni sistemi i eksperimentalna istraživanja“, Elektronski fakultet u Nišu, Niš, 2009

[3] Emilija Živanović, Marija Živković i Milić Pejović. “The evolution of breakdown voltage and delay response under high overvoltage for different types of surge arresters“, *Facta Universitatis, Series: Electronics and Energetics*, Vol. 34, No. 2 pp. 307-322, 2021.

[4] Momčilo M. Pejović, „Uvod u električna gasna pražnjenja. Gasne elektronske komponente“, Elektronski fakultet u Nišu, Niš, 2008.

Poređenje karakteristika tipova litijum-jonskih baterija

Aleksandar Vučković

Sadržaj – Ovaj rad se bavi istraživanjem karakteristika tipova litijumskih baterija u pogledu kapaciteta, vrednosti nominalnog napona i slično. Osnovni cilj jeste da se pronađe najpodobnija primena ovih baterija i da se istraže njihove osnovne razlike.

I. UVOD

Cilj ovog istraživačkog rada jeste da uporedi karakteristike dva tipa baterija izrađenih od litijuma kao osnovnog materijala i to sa katodom izrađenom od litijum-kobalt-oksida [1] i sa katodom izrađenom od litijum-gvožđe-fosfata [1] u pogledu ciklusa punjenja i pražnjenja. Obzirom da su olovne baterije postale standardan tip baterija pogotovo u autoindutriji, različiti tipovi litijumskih baterija su zauzeli tržište uglavnom za manje kućne aparate, razne akumulatorske alate, zatim baterije u mobilnim telefonima itd. Međutim, sada se postavlja pitanje da li je moguće koristiti litijumske baterije za industrijske uređaje kao što su semafori, svetlosni bilbordi, razni tipovi uređaja kojima je potrebno napajnje iz baterije a koriste standardne olovne baterije tj. uređaje kod kojih se bitno menja dimenzija i masa u zavisnosti od tipa baterije koju koriste. Osnovne prednosti litijumskih baterija [2] jesu pražnjenje gotovo celog kapaciteta, veliki broj ciklusa punjenja i praženja bez gubitka korisnog kapaciteta baterije kao i smanjene dimenzije u odnosu na olovne baterije. Štaviše, jedna litijumska baterija ima duži životni vek i ne zahteva održavanje kao olovna baterija [3] jer poseduje elektrolit koji je u čvrstom stanju dok je kod olovne u pitanju tečni elektrolit.

II. OSNOVNE KARAKTERISTIKE BATERIJA

Razlog za upoređivanje karakteristika litijumskih baterija sa katodom od litijum-kobalt-oksida (u daljem tekstu litijum-kobalt-oksid baterija) i tipa litijumskih baterije koji koristie leguru litijum-gvožđe-fosfata (u daljem tekstu litijum-gvožđe-fosfat baterija) kao materijal za katodu je taj što su ovo tipovi litijumskih baterija koje se najčešće sreću u slobodnoj prodaji na tržištu. Ova dva tipa baterija se porede prvenstveno zbog različitih vrednosti napona i struje koje mogu da daju, pa tako je nominalni napon za litijum litijum-gvožđe-fosfat bateriju 12,8 V dok za litijum-kobalt-oksid bateriju taj napon iznosi 10,8 V. Obzirom da se baterije sastoje iz više pojedinačnih ćelija, litijum-gvožđe-fosfat baterija je sastavljena od 3x4 redno vezanih ćelija nominalnog napona 3,2 V (3,65 V kada je napunjena) dok je litijum-kobalt-oksid baterija sastavljena od 10x3 redno vezanih ćelija nominalnog napona 3,6 V (4,2 V kada je napunjena).

Uz to, litijum-gvožđe-fosfat baterije mogu da podnesu veće vrednosti struje što će se kasnije pokazati u rezultatima merenja. Kapaciteti oba tipa baterija iznose 24,3 Ah.

A. Vučković je master inženjer elektrotehnike i računarstva za Elektoniku i mikrosisteme, Elektronski fakultet Niš, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: aleksandar.vuckovic@elfak.rs.

III. OSNOVNE METODE ZA UPOREĐIVANJE

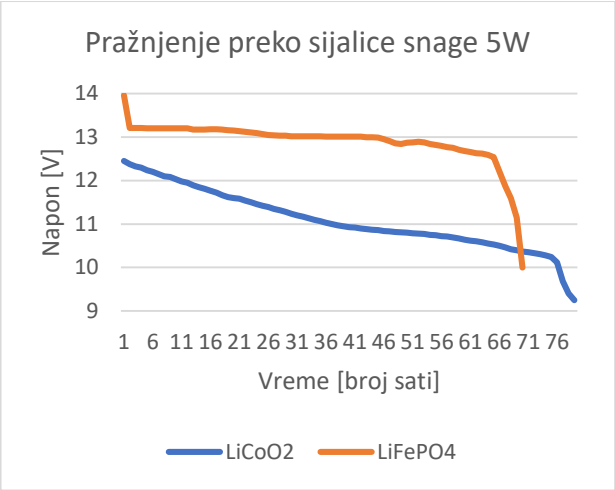
Baterije će biti testirane punjenjem i pražnjenjem pri čemu će se vrednosti napona i struje meriti na svakih sat vremena. Ovom metodom se može proceniti stanje napunjenosti odnosno stanje ispražnenosti baterije pri čemu se može pretpostaviti koji deo kapaciteta baterije je koristan.

Punjenje baterija se vrši pomoću specijalizovanih namenskih punjača za svaki tip litijumskih baterija. U ovom slučaju su potebna dva tipa punjača jer se istraživanje vrši nad dva tipa različitih baterija. Punjači pune baterije u dva režima – konstantna struja i konstantan napon. Režim konstantne struje nastupa prvi gde se baterija puni strujom čija je vrednost približno konstantna u vremenu sve dok napon na bateriji ne pređe neku definisanu vrednosti. Nakon toga se prelazi u režim konstantnog napona gde se napon održava konstantnim a vrednost struje se menja što utiče na vrednost napona na bateriji. Režim je takav da punjač podesi vrednost napona koju treba dostići, a kada je ta vrednost dostigne poveća je za određeni korak sve do ciljane vrednosti napona. Ove vrednosti napona su definisane naponom napunjenosti pojedinačnih ćelija tako da ove vrednosti iznose 12,6 V za litijum-kobalt-oksid bateriju i 14,6 V za litijum-gvožđe-fosfat bateriju što se kasnije može videti u rezultatima. Kada punjač detektuje ove napone, on prestaje da puni bateriju što znači da je baterija puna.

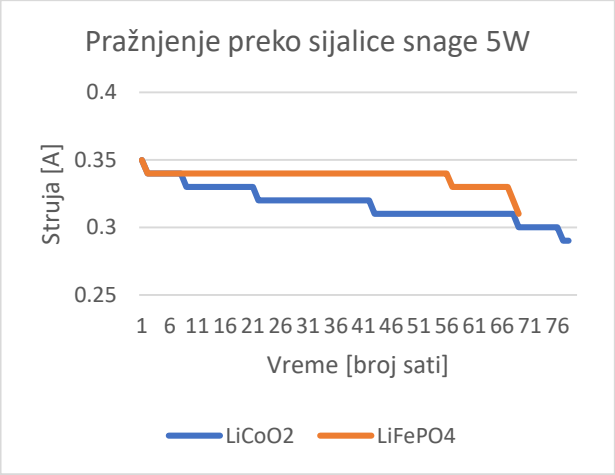
Kada govorimo o pražnjenju, baterije će biti priključene na opterećenja snage 5 W i 12 W. Opterećenje snage 5 W predstavlja sijalicu sa užarenim vlaknom dok opterećenje snage 12 W predstavlja keramički otpornik. Merenje napona se vrši pomoću unimera koji se postavlja paralelno i što bliže izvodima baterije dok se merenje struje izvodi takođe unimerom tako što se on postavi redno u kolo sa opterećenjem i baterijom.

IV. PRIKAZ I ANALIZA REZULTATA ISTRAŽIVANJA

Poređenje litijum-kobalt-oksid i litijum-gvožđe-fosfat baterija se otpočinje sa punim baterija tj. baterije se prvo prazne preko sijalice od 5 W. Na Slici 1 se mogu videti promene napona, a na Slici 2 se mogu videti promene struja pražnjenja za oba tipa baterija.



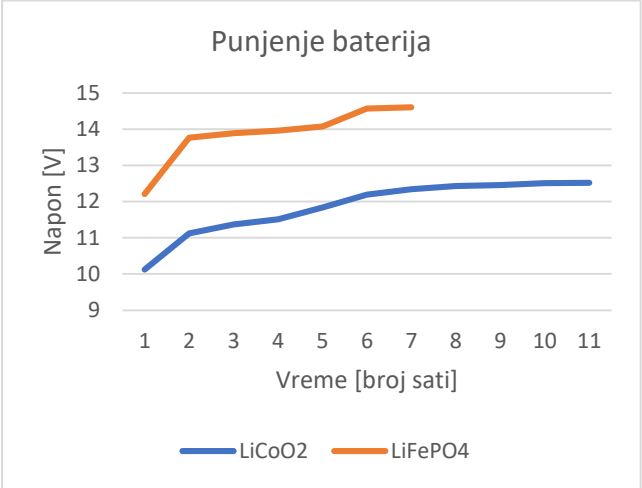
Slika 1. Naponska karakteristika snimljena pražnjenjem baterija.



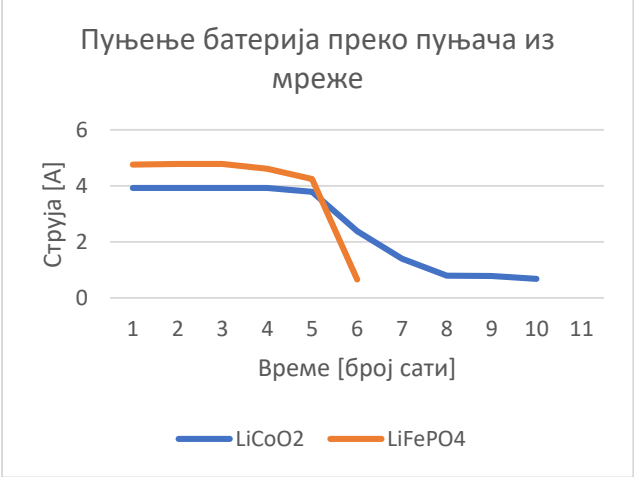
Slika 2. Strujna karakteristika snimljena pražnjenjem baterija.

Ono što se najviše zapaža na Slici 1 jeste kriva napona litijum-gvožđe-fosfat baterije koja se vrlo malo menja tokom vremena u odnosu na promenu napona litijum-kobalt-oksid baterije. Ova sličnost se može uočiti i kod promene struje pražnjenja tokom vremena data na Slici 2. Odavde se može zaključiti da litijum-gvožđe-fosfat baterija na svom izlazu daje približno konstantne vrednosti napona i struje tokom vremena sve do određenih vrednosti napona i struje posle kojih ove vrednosti počinju drastično da opadaju. Što se tiče litijum-kobalt-oksid baterije, kod nje se vrednosti napona i struje smanjuju sa vremenom tj. nisu približno konstantne kao kod drugog tipa litijumske baterije. Zbog toga, ovaj tip baterije može duže da snabdeva energijom potrošača koji ne zahteva konstantnu vrednost snage odnosno onog koji ne zavisi previše od promena napona i struje. Na osnovu merenja struje tokom vremena, može se proceniti koji je kapacitet baterija ispražnjen (koji može biti koristan) što za litijum-gvožđe-fosfat bateriju iznosi 23 Ah (amper-sati) a za litijum-kobalt-oksid iznosi 23,39 Ah. Na osnovu ovih podataka se može zaključiti da se ova dva tipa litijumskih baterija mogu isprazniti više od 90 % svog nominalnog kapaciteta što im proširuje mogućnosti za primenu u različitim granama industrije jer su efikasnije od drugih tipova baterija.

Kada govorimo o punjenju ovih baterija, jasno se vidi da se litijum-gvožđe-fosfat baterija napuni brže i da može da podnese veće struje punjenja nego litijum-kobalt-oksid baterija. Ovde se takođe mogu videti režimi punjenja baterija – režim konstantne struje nastupa prvi pa se na Slici 4 vidi da je struja punjenja kod oba tipa baterija približno konstanta dok se nakon određene vrednosti napona koji se može videti na Slici 3, značajno smanjuje jer sada nastupa režim konstantnog napona.



Slika 3. Naponska karakteristika snimljena punjenjem baterija.



Slika 4. Strujna karakteristika snimljena punjenjem baterija.

V. ZAKLJUČAK

Obzirom da se trenutno u svetu litijum dosta eksploatiše, pogotovo za električna vozila, vrlo je verovatno da će nastati još veći broj različitih tipova litijumskih baterija nego danas sve sa ciljem da se unapredi efikasnost

istih. Pored toga, biće potrebno rešiti i problem sa karakteristikama baterija na niskim temperaturama i temperaturama [4] ispod 0 stepeni Celzijusove skale. Ovo danas predstavlja problem pa su naučnici razvili nekoliko metoda [5] koji uključuju zagrevanje baterija na temperaturama iznad 0°C kada je spoljna temperature ispod ove vrednosti. Ova metoda je trenutno neophodna kako bi se litijumske baterije mogle koristiti u širem temperaturnom opsegu bez degradacije njihovih karakteristika.

Ovaj rad može biti unapređen istraživanjem jednostavnog načina grejanja baterija ili primenom nekog od drugih metoda zagrevanja koji su već potvrđeni kao podobni. Takođe, mogu se istražiti karakteristike baterije u pogledu napona i struje kojom baterija može da napaja potrošača na različitim temperaturama kao i u kojoj meri to zavisi od njene temperature.

ZAHVALNICA

Autor ovog rada se zahvaljuje Katedri za mikroelektroniku na Elektronskom fakultetu, Univerziteta u Nišu, posebno prof. dr Danijelu Dankoviću na pruženoj pomoći i svakodnevnoj dostupnosti, kao i u davanju smernica prilikom realizacije ovog projekta.

LITERATURA

[1] Nathalie Lyczko, Ange Nzihou, Patrick Sharrock, Alain Germeau, Claudia Toussaint. “Characterization of LiFePO4/C Cathode for Lithium Ion Batteries”, Industrial and engineering chemistry research, American Chemical Society, 2012.

[2] Rushendra Paravasthu „Synthesis and characterization of lithium-ion cathode materials in the system (1-x-y) LiNi1/3Mn1/3Co1/3O2 · xLi2MnO3 · yLiCoO2“, Department of Mechanical Engineering, Colorado State, University Fort Collins, Colorado, 2012.

[3] Elena M. Krieger, John Cannarella, Craig B. Arnold, „A comparison of lead-acid and lithium-based battery behavior and capacity fade in off-grid renewable charging applications”, Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ 08544, USA, 2013.

[4] Kevin L. Gering, „Low-temperature Performance Limtations of Lithium-ion Batteries, Idaho National Laboratory, Idaho Falls, USA, 2006.

[5] Xiaosong Hu, Yusheng Zheng, David A. Howey, Hector Perez, Aoife Foley, Michael Pecht, „Battery warm-up methodologies at subzero temperatures for automotive applications: Recent advances and perspectives”, Progress in Energy and Combustion Science, Queen's University Belfast, 2019.

Ispitivanje pouzdanosti gasnih odvodnika tipa *Littelfuse*- 300 V

Ljubica Krivokapić

Sadržaj − Cilj ovog rada je odrediti statički probojni napon gasnog odvodnika proizvođača *Littelfuse* koji ima radni napon 300 V. U tu svrhu iskorišćena je dinamička metoda. Nakon toga snimane su memorijske krive, tačnije zavisnosti srednje vrednosti vremena kašnjenja električnog proboja od vremena relaksacije za različite vrednosti prenapona. Utvrđeno je njihovo nepostojanje, tačnije potvrđena pouzdanost u radu ovih gasnih odvodnika.

I. UVOD

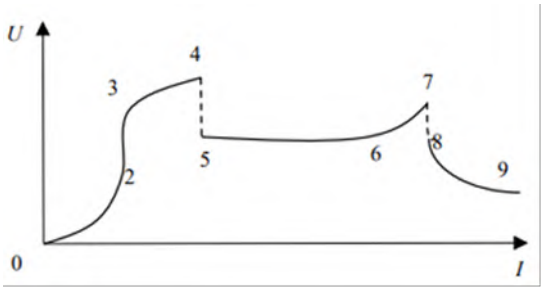
Gasni odvodnici prenapona su nelinearni elementi koji se koriste za zaštitu od prenapona. Označavaju se kao SVP (eng. *Surge Voltage Protector*), češće kao GDT (eng. *Gas Discharge Tube*), a najčešće se koristi oznaka GFSA (eng. *Gas Filled Surge Arresters*), mada nemački naziv osigurač sa plemenitim gasom (nem. *Edel gas sicherung*) najviše odgovara njihovoj funkciji. Rade na principu električnog proboja gasa pri niskom pritisku i mogu se koristiti za zaštitu od prenapona intenziteta od nekoliko desetina volti do nekoliko desetina hiljada volti. U pogledu energetske izdržljivosti oni su najizdržljiviji, a u pogledu cene najjeftiniji elementi za zaštitu od prenapona na niskonaponskom nivou [1].

Prednost gasnih odvodnika prenapona nad ostalim elementima za zaštitu od prenapona su: mogućnost provođenja velikih struja (do 60 kA), niskoj sopstvenoj kapacitivnosti (reda veličine 1 pF što ih čini podesnim za primenu u području visoke frekvencije) gde kapacitivnost potiče od kondenzatora koga čine elektrode u stanju bez struje i ona je zanemarljiva u odnosu na kapacitivnost ostalih elemenata prenaponske zaštite (kod prenaponskih dioda ona iznosi 12000 pF, a kod metaloksidnih varistora i do 15000 pF) i ogromnoj i nepromenljivoj otpornosti (pri naponima nižim od napona paljenja ona iznosi 108 Ω – 1010 Ω, a u stanju provođenja je reda 0.1 Ω). Funkcionisanje gasnog odvodnika vezano je za Taunzendov mehanizam proboja gasova.

II. PRINCIP RADA GASNIH ODVODNIKA

Na primeru elektronske cevi punjenje plemenitim gasom (pritiska 10^{-6} bar do 0.08 bar) koji je obično neon, argon, helijum ili njihova smeša, može se analizirati princip rada gasnog odvodnika prenapona. Nastanak električnog pražnjenja u gasu može se objasniti po Taunzendovoj teoriji.

Posmatraćemo sliku 1. Deo karakteristike od tačke 0 – 4 predstavlja oblast Taunzendovog pražnjenja. U tački 4 dovoljan broj elektrona dostiže jonizacionu energiju za samoodržavanje procesa, napon na cevi opada što pokazuje da je za održavanje procesa jonizacije gasa dovoljan manji napon nego za njegovo započinjanje. Napon u tački 4 se zove napon paljenja cevi, a cev se osvetli svetlošću koja je različita za različite vrste gasa, smeše gasa, pritiska gasa i rastojanja elektroda. Proces između tačaka 5 i 6 naziva se procesom tinjavog pražnjenja. U tački 7 počinje proces lučnog pražnjenja kada napon na cevi opadne do vrednosti jonizacionog potencijala za dati gas koji može biti 15 V– 20 V pri strujama od nekoliko stotina ili hiljada ampera. Problem u kolima prenaponske zaštite predstavlja činjenica da je za održavanje lučnog pražnjenja dovoljna struja intenziteta 0.1 A, pa se gašenje cevi može postići jedino smanjenjem struje na zanemarljive vrednosti ili njenim prekidanjem [1-3].

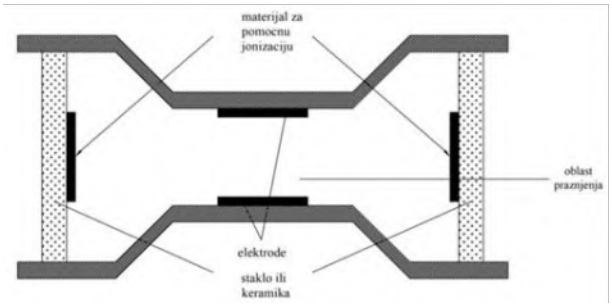


Slika 1. U-I karakteristika cevi punjene inertnim gasom i šema za njeno napajanje.

III. KONSTRUKCIJA GASNIH ODVODNIKA

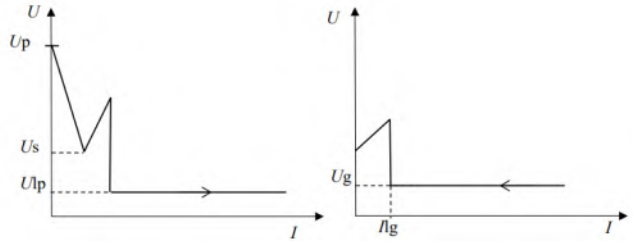
Gasni odvodnik prenapona sastoji se od dve identične metalne elektrode zatopljene u keramičko ili stakleno kućište. Međuelektrodno rastojanje je veličine 1 mm. Na unutrašnjem zidu kućišta se u nekim slučajevima nanose radioaktivni izotopi radi povećanja brzine odziva (α-zračenje). Izolacioni medijum je plemeniti gas (neon, helijum, argon,...) ili smeša plemenitih gasova na pritisku od 0.01 bar do 1 bar. Kućište odvodnika mora biti od neporoznog materijala pošto plemeniti gasovi kao što je poznato, imaju popunjenu poslednju atomsku orbitu i oni se javljaju isključivo u atomskom obliku, malih dimenzija i lako difunduju kroz porozne sudove. Karakterističan presek jednog odvodnika prenapona dat je na slici 2 [2].

Lj. Krivokapić je student master studija modula Elektronika i miksrosistemi, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: krivokapiccc@gmail.com.



Slika 2. Karakterističan presek odvodnika prenapona.

Na slici 3 prikazane su tipične UI karakteristike gasnog odvodnika prenapona. Uočavaju se sledeće karakteristične vrednosti: U_p – napon paljenja (vrednost napona pri kome dolazi do porasta gasnog prenapona (probojni napon)), U_{sp} – napon svetlog pražnjenja (napon pri kome dolazi do emisije svetlosti talasne dužine iz oblasti vidljivog dela spektra iz međuelektrodnog prostora) odvodnika prenapona, U_{lp} – napon pri kome dolazi do paljenja električnog luka, U_g – napon gašenja električnog luka (napon pri kome dolazi do prekida lučnog pražnjenja) i I_{lg} – struja gašenja električnog luka [1].

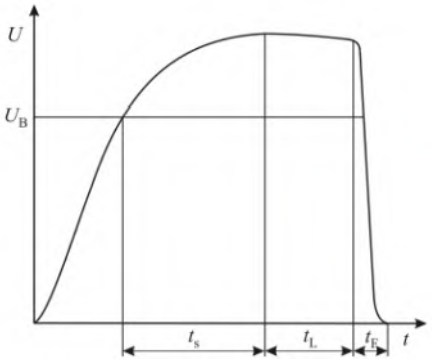


Slika 3. Tipična naponsko-strujna karakteristika gasnog odvodnika.

IV. VREME KAŠNJENJA ELEKTRIČNOG PROBOJA

Najčešće korišćena definicija za vreme kašnjenja je da je to vreme koje protekne od momenta priključenja radnog napona na gasnu komponentu, koji je veći od probojnog napona, pa do momenta pojave proboja, tj. postizanja struje kroz gas koji je moguće detektovati [4]. Podrazumeva se da je vreme porasta napona tokom njegovog priključenja zanemarljivo malo u odnosu na vreme kašnjenja. Osim toga potrebno je da unutrašnja otpornost izvora bude zanemarljivo mala, a na red sa gasnom komponentom vezana je omska otpornost. Vreme kašnjenja električnog proboja se označava sa t_D i predstavlja zbir intervala statističkog vremena kašnjenja t_S, vremena formiranja lavine t_L i vremena formiranja pražnjenja t_F , odnosno $t_D = t_S + t_L + t_F$ (slika 4.) [2].

Vreme kašnjenja električnog proboja zavisi od više parametara kao što su: vrsta i pritisak gasa, vrsta materijala katode i kvalitet njene površine, prenapon, temperatura gasa, međuelektrodno rastojanje. Jedan od važnih parametara koji utiče na vreme kašnjenja električnog proboja je vreme relaksacije τ (posleprobojni period) koje predstavlja vreme proteklo od momenta prekida pražnjenja kroz gasnu komponentu do momenta ponovnog uključivanja napona koji je veći od probojnog napona.



Slika 4. Karakterističan tok impulsnog napona kojim je ostvaren proboj.

U praksi se zbog stohastičkog karaktera vremena kašnjenja električnog proboja koristi srednja vrednost ove veličine za datu vrednost vremena relaksacije. Ako se meri ukupno vreme kašnjenja električnog proboja pri proučavanju jednog dela, drugi deo se mora učiniti malim, kao na primer ako se želi informacija o procesu oslobađanja inicijalnih elektrona, tada uslovi eksperimenta moraju biti tako izabrani da je $t_F << t_S$, pa se t_S može smatrati približno jednakim t_D. Obrnuto, ako se želi informacija o sekundarnim procesima u gasu, t_S se mora svesti na najmanju meru ili bar učiniti mnogo manjim od t_F.

V. MEMORIJSKI EFEKAT U GASOVIMA

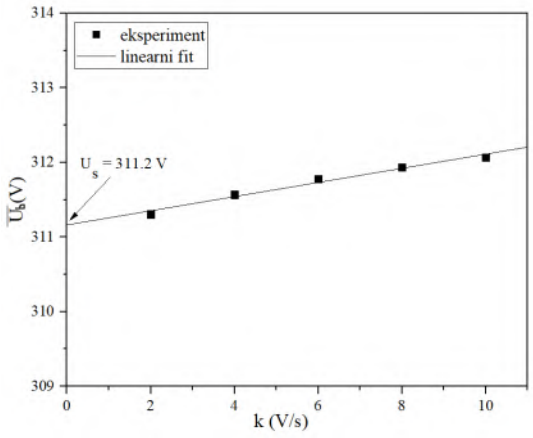
Posle određenog vremena relaksacije koje zavisi od pritiska i vrste gasa, vrste materijala katode i ozračenosti gasa, gasna komponenta dolazi u „stacionarno“ stanje u kojem je verovatnoća proboja određena intenzitetom spoljašnjeg zračenja u kojem je gasna komponenta izložena.

Tada vrednost vremena kašnjenja proboja t_D neznatno fluktuira oko neke srednje vrednosti koja zadržava približno konstantnu vrednost, naročito kada je prisutno jonizujuće zračenje. U praksi se zbog stohastičkog karaktera vremena kašnjenja električnog proboja koristi srednja vrednost ove veličine za datu vrednost vremena relaksacije. Ovaj postupak se izvrši za različite vrednosti vremena relaksacije da bi se dobila kriva zavisnosti srednje vrednosti vremena kašnjenja električnog proboja od vremena relaksacije koja je poznata pod nazivom „memorijska kriva“ („kriva pamćenja“) [4]. Na osnovu ponašanja ove krive može se pratiti opadanje koncentracije pozitivnih jona i onih neutralnih aktivnih čestica koje sa katode mogu osloboditi sekundarne elektrone, odnosno može se pratiti memorijski efekat usled prisustva ovih čestica do tako niskih koncentracija, kada je verovatnoća

oslobađanja inicijalnih elektrona ovim česticama porediva sa verovatnoćom izazavanom kosmičkim zračenjem i radioaktivnim zračenjem okoline [4, 5].

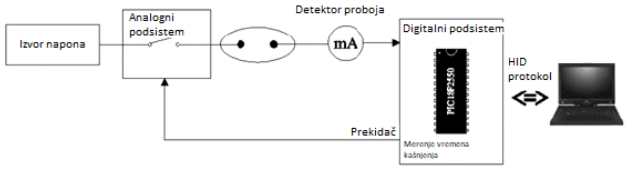
VI. OPIS METODE MERENJA

Naime, dinamička metoda za procenu statičkog probojnog napona koja je korišćena u ovom radu zasnovana je na snimanju zavisnosti srednje vrednosti probojnog napona od brzine porasta napona *k*. Linearnim fitovanjem zavisnosti podataka dobijenih eksperimentom $\overline{U_b} = f(k)$ možemo dobiti vrednost statičkog probojnog napona. Ta vrednost se očitava na preseku prave sa $\overline{U_b}$-osom (za *k*=0). Srednja vrednost probojnog napona dobijena je na osnovu izmerenog seta od 100 podataka. Na slici 5 prikazan je eksperimentalno dobijen rezultat, kao i linerni fit na osnovu koga je procenjena vrednost statičkog probojnog napona iznosila 311.2 V za brzine porasta napona od 2, 4, 6, 8 i 10 V/s.



Slika 5. Primena dinamičke metode za procenu statičkog probojnog napona gasnog odvodnika.

Nakon procene statičkog probojnog napona pristupilo se merenju vremena kašnjenja električnog proboja gasa ove komponente za različite vrednosti primenjenog napona. Šema eksperimenta prikazana je na slici 6 [5].



Slika 6. Blok dijagram eksperimenta za merenje vremena kašnjenja.

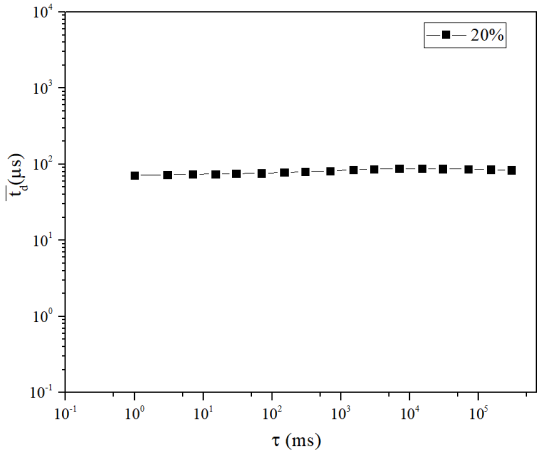
Merenja vremena kašnjenja električnog proboja kod ovih komponenata izvršena su preko softvera koji naizmenično prikazuje svako uneto vreme relaksacije i vreme žarenja. Pojava memorijske krive predstavljaju neželjeni efekat kod gasnih odvodnika. Vremena relaksacije kretala su se od 1 μs pa do 30 s. Broj merenja je postavljen na 100, a naredbom 2 počinje merenje. U našim eksperimentima koristili smo seriju sa stotinu merenja vremena kašnjenja za različita vremena relaksacije. Vremena relaksacije izabrana su prema logaritamskoj skali.

Merenje se vrši na sledeći način. Serije impulsa konstantnog napona su primenjivani na komponentu, dok je proteklo vreme između primene impulsa napona zadato i vreme kašnjenja električnog proboja je izmereno. Izmerena vremena kašnjenja su sačuvana u memoriji računara, a tok napona je održavan na komponenti za vreme t_g da bi se održali isti eksperimentalni uslovi za svako merenje. Nakon toga, gasna komponenta je isključena tokom zadatog vremena relaksacije τ. Postupak je ponovljen željeni broj puta, za različite periode relaksacije.

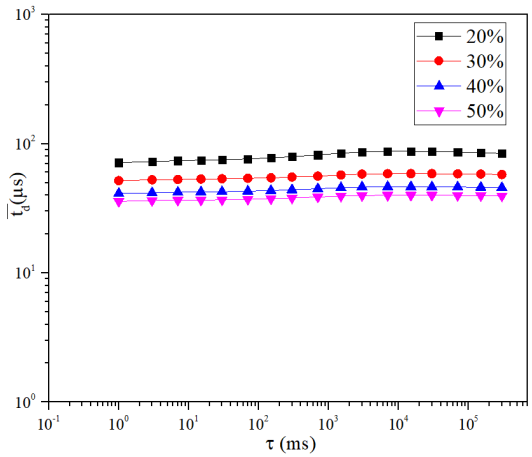
VII. EKSPERIMENTALNI REZULTATI I DISKUSIJA

Postojanje memorijske krive je nepoželjno kada se proučava pouzdanost gasnih odvodnika prenapona. Pošto proizvođači GFSA ne daju tačnu specifikaciju sastava gasa u tehničkoj dokumentaciji, može se naći samo da se radi o plemenitim gasovima, a u tu svrhu najčešće se koriste argon i neon. Tokom samog eksperimenta, primećuje se crvenkasto-narandžasti sjaj, što je nagoveštaj da se radi o plemenitim gasovima. Eksperimentalno dobijene memorijske krive za *Littelfuse* GFSA mogu biti u uporedene sa prethodno dobijenim za argon i neon. Razlika je što se memorijska kriva u istraživanim slučajevima ne javlja. Naime, primećuje se konstantnost vremena kašnjenja električnog proboja prilikom povećanja vremena relaksacije. Pored toga rezultati pokazuju da dolazi do smanjenja vremena kašnjenja sa povećanjem prenapona, odnosno primenjenog napona na ispitivani gasni odvodnik.

Na slikama 7 i 8 prikazane su zavisnosti srednje vrednosti vremena kašnjenja u funkciji vremena relaksacije za prenapon od 20 %, kao i za 20, 30, 40 i 50 % redom.



Slika 7. Zavisnosti srednje vrednosti vremena kašnjenja u funkciji vremena relaksacije za prenapon od 20 %.



Slika 8. Zavisnosti srednje vrednosti vremena kašnjenja u funkciji vremena relaksacije za različite prenapone.

TABELA I
REZULTATI ZA PRENAPON OD 20 %

τ (ms)	t_d (μs)	σ_{td} (μs)
1	71.47917	2.38611
3	72.73833	2.87375
7	73.96833	2.96999
15	74.63417	3.13198
30	75.30833	3.19352
70	76.36667	3.32841
150	77.78417	3.62157
300	79.42	4.08033
700	81.995	5.03223
1500	84.58667	6.43987
3000	86.55667	7.80251
7000	87.42917	8.54858
15000	87.41417	8.66593
30000	86.94	8.48124
70000	86.30833	8.28192
150000	85.3325	7.97131
300000	84.00583	7.69211

Na osnovu analize dobijenih rezultata dolazi se do sledećih zaključaka. Kao što se vidi iz Tabele I, za vrednost prenapona od 20 % tip gasnog odvodnika *Littelfuse* radi pouzdano, za vreme kašnjena od 100 μs. Sa porastom primenjenog napona na elektrode gasnog odvodnika dolazi do smanjenja vremena odziva gasnog odvodnika. Vreme odziva se kreće između 40 i 50 μs za prenapone od 30 do 50 %. Raniji rezultati testiranih gasnih komponenata pokazuju slično ponašanje. Analiza rezultata dobijenih za ovaj tip gasnog odvodnika pri različitim prenaponima ukazuje da radi pouzdano. Pored toga, rezultati ukazuju da vrednosti prenapona slabo utiču na vreme odziva.

VIII. ZAKLJUČAK

Na osnovu rezultata prikazanih u radu, kao i na osnovu ostalih rezultata dobijenih tokom opsežnih ranijih eksperimentalnih ispitivanja, može se zaključiti da je moguće novim konstruktivnim rešenjima optimizovati karakteristike gasnih odvodnika prenapona. Raniji rezultati pokazuju da je u cilju dugovremenske stabilizacije radne tačke gasnog odvodnika prenapona (minimalni probojni napon za određenu vrednost proizvoda pritiska i međuelektrodnog rastojanja) potrebno koristiti elektrode oblika Rogovskog, a proizvod pritiska i međuelektrodnog rastojanja locirati u blizini odgovarajućeg Pašenovog minimuma. Ovakav sistem ima i bolji dinamički odziv od sistema sa cilindričnim elektrodama i čistim plemenitim gasom kao izolacionom sredinom. Međutim, tokom ispitivanja se pokazalo da na dinamički odziv gasnog odvodnika prenapona (pošto je brzina odziva gasnog odvodnika ujedno i njegov najveći nedostatak) presudno utiče gustina i energetski spektar gasa slobodnih elektrona. Pokazano je u tom smislu da se odziv gasnog odvodnika prenapona poboljšava ugrađivanjem radioaktivnog izvora u njega, i to tako da se Bragov maksimum tog α zračenja nalazi u centru međuelektrodnog prostora.

LITERATURA

[1] Dragan V. Brajović, "Optimizacija dinamičkih karakteristika gasnih odvodnika prenapona na niskonaponskom nivou", doktorska disertacija, Elektrotehnički fakultet, Beograd, 2014.
[2] Momčilo M. Pejović, "Uvod u električna gasna pražnjenja. Gasne elektronske komponente", Elektronski fakultet, Niš, 2008.
[3] Mirko N. Stojkanović, "Statička i dinamička sinergija izolacionih osobina gasne smeše SF6/N2", doktorska disertacija, Elektrotehnički fakultet, Beograd, 2013.
[4] M. M. Pejović, G. S. Ristić, J. P. Karamarković, "Electrical Breakdown in Low Pressure Gases", *J. Phys. D: Appl. Phys*., 35, 10, pp. R91-R103, 2002.
[5] E. Živanović, S. Veljković, M. Živković and M. Pejović, "Reliability of various type of gas-filled surge arresters under DC discharge", In Proceedings of the 31st International Conference on Microelectronics, Niš, Serbia, 2019, pp. 113-116.

Generisanje neograničeno dugih klavirskih sintetičkih melodija različitih žanrova upotrebom kreiranih stohastičnih algoritama

Maja Milović

Sadržaj – U radu je predstavljen proces kreiranja modela za računarsko generisanje neograničeno duge klavirske melodije. Princip rada modela zasniva se na nasumičnom biranju tona koji se naredni reprodukuju na osnovu skupa koji sadrži sve dostupne tonove. Na taj način, melodija je uvek različita. Računar dostupne tonove određuje pomoću teorije muzike. Reprodukovani klavirski ton ima podršku akorda koji se u određenoj formi takođe reprodukuje u pozadini.

I. UVOD

Cilj računarskog generisanja muzičkih melodija u ovom radu jeste dobijanje klavirske melodije koja se menja sve dok se njen tok prinudno ne zaustavi. Tok melodije se određuje nasumičnim izborom pravila iz predefinisanog skupa pravila koji je baziran na teoriji muzike. U zavisnosti od žanra koji se odabere, algoritam pronalazi četiri akorda iz istog tonaliteta koji se zauvek ponavljaju i reprodukuju po nasumično određenom šablonu kroz čitavu melodiju. Tonalitet u muzici predstavlja sistem koji nam govori kako se koji ton odnosi prema glavnom. On određuje tonsku osnovu kompozicije. Na primer, muzički komadi koji zvuče „veselo“ su durskog tonaliteta, a oni „tužniji“ su molskog. Takođe, oni mogu da budu „viši“ i „niži“. Akord predstavlja zvuk koji se dobija kada se tri ili više tona odsviraju u isto vreme. Niz akorda koje generiše program predstavlja pozadinsku muziku koja služi kao osnova za glavnu melodiju. Glavna melodija nikada nije ista.

Doprinos ovog rada je specifikacija, implementacija, kao i evaluacija algoritma za generisanje melodije neograničene dužine. Melodija se ne čuva nigde, već postoji da bi se uvek menjala i razvijala. Ako se program zaustavi i iznova pokrene, gotovo je nemoguće da se prethodna melodija ponovi.

II. METOD

A. Teorija muzike

Muzička lestvica predstavlja niz od 8 tonova pri čemu je osmi ponovljen prvi, ali za oktavu viši. Može se posmatrati kao matematički skup S = {C, D, E, F, G, A, H}. Skup S je uređen sa relacijom „pre” tako da je C pre D, D pre E, itd. Takođe je, analogno, D „posle” C, E posle D pa tako kroz čitav skup. Elementi ovog skupa predstavljaju nazive muzičkih nota koje, respektivno, ulaze u sastav najjednostavnije lestvice u muzici, C-dura. Polustepen u muzici je najmanje rastojanje između dva tona. Jedan stepen se sastoji iz dva polustepena. Durska lestvica je takva da se između svaka dva stupnja nalazi čitav stepen, osim između 3. i 4, kao i 7. i 8, između kojih je polustepen. Osnovni akord svake lestvice je onaj koji se nalazi na prvom stupnju. Moguće je izgraditi akord na svakom stupnju jedne lestvice. Akord u lestvicama kojima se bavi ovaj naučni rad predstavlja niz od 3 note. Najniža od te tri se naziva osnovnom, a sledeće dve su one koje su, redom, dva i četiri elementa „posle” osnovne. Osnovni akord, to jest C-dur, na taj način predstavljao bi skup od 3 tona, C, E, i G, koji su podskup skupa S. Međutim, iako je tonalitet durski, ne znači da takav mora biti svaki akord u lestvici. Štaviše, durski akordi se u durskoj lestvici nalaze samo na 1, 4. i 5. stupnju, a na ostalima su molski, osim na sedmom, gde je umanjeni. Molski akordi zvuče tužno, sumorno. Umanjeni zvuče nelagodno i neprirodno i iz tog razloga ne ulaze u sastav rada. Kako skup S predstavlja skup od 7 elemenata a u muzici je potrebno koristiti daleko više od toga, neophodno je shvatiti da posle sedmog dolazi ponovljen prvi ton, a posle njega ponovljen drugi. Ta činjenica može biti posmatrana sa strane matematičke kongruencije. Ako su dva tona iz iste skale kongruentna po modulu koji predstavlja broj tonova u toj skali, oni su onda isti ton. Dakle, se gradi akord sa 6. stupnja C-dur lestvice, dobija se A-mol, koji se sastoji od nota A, C i E. Kako su svi fajlovi sa klavirskim notama svrstani u niz i poređani kao dirke na klavijaturi, računarskom programu je skala predstavljena kao podniz tog niza: počinje nasumičnom notom, i u njega svrstava svaku drugu notu, osim kod treće i sedme, jer je između 3. i 4. stupnja polustepen, kao i između 7. i 8.

B. Generisanje jednog klavirskog tona

Zvučni talasi koji nastaju prilikom pritiskanja dirki na klaviru nisu trivijalnog oblika i razlikuju se u zavisnosti od dubine, tj. visine tona koji se svira. Pri nižim frekvencijama, talasi su nalik kvadratastim, a pri višim gotovo savršene sinusoide. Boja zvuka klavira je jedna od najkompleksnijih od svih muzičkih instrumenata. Moduliranje zvučnih talasa u pokušaju da se dobije „sintisajzer“ koji dovoljno dobro

M. Milović, učenica prirodno-matematičkog smera gimnazije „Svetozar Marković” u Subotici, Šandora Petefija 1, 24000 Subotica, Srbija. Polaznica Istraživačke stanice Petnica, E-mail: maja.milovic05@gmail.com.

reprodukuje tonove koje stvara klasični klavir nikome do sada nije donelo zadovoljavajuće rezulate [1].

Iz tog razloga, optimalni način za dobijanje zvuka koji proizvodi klasičan klavir je preuzimanje i reprodukcija audio uzoraka svih 88 tonova koje proizvode klavirske dirke. Pronađena je bibilioteka koja pruža profesionalno snimljene uzorke visokog zvučnog kvaliteta koji su potrebni za svrhe ovog rada. Model klavira u pitanju jeste *Steinway & Sons* klasični koncertni klavir, model B [2]. Kako su audio datoteke bile AIFF formata (*Audio Interchange File Format*) koje programski jezik *ChucK* ne podržava, te datoteke su bez oštećenja prevedene u WAV format (*wave*). Datoteke su trajale približno 40 sekundi, od kojih se zvuk klavirskih dirki gotovo nije čuo poslednjih 30 sekundi. Zbog toga, svaka je isečena tako da su ostali audio fajlovi koji sadrže prvih 8 sekundi svake datoteke iz pomenute biblioteke. Na taj način se izdvojio najkorisniji deo datoteka bez značajnih gubitaka, a bafer koji učitava audio datoteke je mogao efikasnije da manipuliše njima nakon što su bile pet puta kraće.

Projekat je implementiran u programskom jeziku *ChucK*. Algoritam može da generiše melodije različitih muzičkih žanrova. Žanrovi u pitanju su klasični pop, džez i lofi. Džez je muzički žanr čiji predstavnik je saksofon. Iz tog razloga, umesto reprodukovanja pomenutih klavirskih fajlova u WAV formatu, za glavnu melodiju u džezu je iskorišćena klasa *UGen* koja se bavi jediničnim generatorma. Iz nje je nasleđena subklasa sa alatkama za sintezu zvuka pod nazivom *STK Instuments* [3]. On je u sklopu programskog jezika *ChucK*. Pomoću nje, modelovanjem zvučnih talasa proizvode se zvuci koji podsećaju na izvesne muzičke instrumente. Međutim, podešavanjem određenih parametara kojima se moglo pristupiti preko biblioteke za duvačke instrumente, zvuk najbliži saksofonu je i dalje delovao sintetisano, veštačko i ni nalik pravom instrumentu. Iz tog razloga, opcija za saksofon je odbačena, a džez melodija je ostala klavirski orijentisana.

C. Sastavljanje melodije od tonova

Melodija je lista uređenih četvorki. Te četvorke su tempo, tonalitet, visina i notne vrednosti. Tempo melodije, kao i njen tonalitet, su komponente od izrazite važnosti. Međutim, visina i notne vrednosti tonova najviše utiču na kvalitet melodije.

Muzički tonalitet melodija koja se generiše u programu je durski. Durski tonalitet je karakterističan za Zapadnu muziku. Visina osnovnog tona je određena nasumično i definisana je kao pomeraj. Pomeraj je bilo koji broj između 18 i 29, i predstavlja udaljenost od najdublje dirke na klaviru, izraženu u polustepenima. Polustepen je najmanje rastojanje između dve note u lestvici. Bilo koji muzički ton se ponavlja nakon 12 polustepena - zvuči gotovo isto ali je za oktavu viši. Iz tog razloga je izabran raspon od 12 brojeva od kojih računar nasumično određuje jedan - pomeraj. Prilikom učitavanja bilo koje klavirske note iz niza koji ih sadrži sve, pravilno poređane, konstantna vrednost pomeraja se dodaje kako bi tonalitet uvek bio usklađen.

Muzička kompozicija neograničene dužine podrazumeva glavnu melodiju koja se neprestano menja, a bazirana je na četiri akorda koji se ponavljaju. Prvi od ta četiri je osnovni, zato što se na taj način stiče utisak stabilnosti i pouzdanosti. Ostala tri se određuju nasumično, s tim što su programu data određena uputstva: ne smeju da se ponavljaju, i nijedan od njih ne sme da bude izgrađen na 7. stupnju, jer je taj umanjeni. Na taj način, dobijaju se četiri akorda od kojih se proizvodi pozadinska melodija u određenim šablonima. Ona je dublja od glavne melodije i sve što je nasumično jeste koji šablon se izabere, i on se ponavlja sve dok nije zaustavljen. Na primer, može da bude u obliku svih nota koje ulaze u sastav akorda „odsviranih“ istovremeno dva puta u taktu; ili, iste te note raščlanjene, tako da se smenjuju četiri puta u taktu kako bi stvorile *appregio* efekat. Definisano je deset različitih šablona, i svaki na autentičan način doprinosi atmosferi celokupne kompozicije (Slika 1).

```
fun void levaRuka1()              //funkcija za reprodukovanje jednog od ,,šablona" za pozadinsku melodiju
{

    while(true)
    {
        for(0 => int m; m<position.cap(); m++)      //ponavlja se toliko puta koliko ima akorda
        {

            note[position[m] + offset] => buf1.read;        //učitavanje note u bafer
            38000 => buf1.pos;              //krece od 3.8 sekunde zbog pauze na početku audio fajla
            1 => env1.keyOn;                //primenjivanje ranije definisanih ADSR parametara

            note[position[m] + offset + left_hand[m*4+1]] => buf2.read;
            1 => env2.keyOn;
            38000 => buf2.pos;

            note[position[m] + offset + left_hand[m*4+2]] => buf3.read;
            1 => env3.keyOn;
            38000 => buf3.pos;

            beat/2 => now;      //vreme ,,napreduje", tj. reprodukuju se fajlovi koji stoje u baferu

            38000 => buf1.pos;      //vraćanje audio fajla na početnu poziciju
            38000 => buf2.pos;
            38000 => buf3.pos;

            beat/4 => now;

            snare => snareb.read;   //zvuk bubnja

            beat/4 => now;
        }
    }
}
```

Slika 1. Deo računarskog koda koji prikazuje funkciju za reprodukovanje jednog od šablona za pozadinsku melodiju u programskom jeziku *ChucK*.

Tempo muzičkih melodija koje se neograničeno dugo komponuju je konstantan i nasumičan, kao i tonalitet. Raspon u kom se on kreće zavisi od žanra u pitanju. Na primer, uopšteno govoreći, pop kompozicije su brže od lofija. Međutim, ono što se nasumično određuje jeste dužina trajanja jednog takta. Akordi se menjaju na tom nivou, a samim tim se u ovom projektu uvek koristi ista vrsta takta – najjednostavnija i najrasprostanjenija, četiri četvrtine. To znači da je moguće da jedan takt u sebi sadrži jednu celu notu, a jedinica brojanja je četvrtina. Osim cele note i četvrtine, notna vrednost može da bude i polovina – ona traje jednu polovinu takta, polovina sa tačkom – traje tri četvrtine takta, četvrtina sa tačkom, koja traje tri osmine, kao i osmina. U svakom taktu, iznova se nasumično biraju notne

vrednosti koje će da čine taj takt sve dok u zbiru ne iznose tačno jednu celu notu, i onda iznova.

Za svaku notnu vrednost, potrebno je odrediti koja nota će se izvoditi. To je ono što daje kvalitet muzičkoj melodiji i ono na šta se obraća najviše pažnje. Melodija je u bliskoj vezi sa akordima – note koje čine akord uvek zvuče dobro kada se sviraju uz njega. Međutim, na taj način se gubi ,,tok" melodije, jer ima puno skokova. Ostale note iz skale zvuče prijatno kada se koriste kao prelazne. Iz tog razloga, programu su date sledeće smernice: obavezno je da prva nota koju svira u taktu bude jedna od onih koji ulaze u sastav akorda koji se pušta u pozadini, a ostale mogu da budu bilo koje iz skale. On pamti koju je notu poslednju pustio, i na osnovu nje odlučuje koju će sledeću, jer ima okvire u kojima sme da se kreće. Ti okviri su prostor za „skokove" melodije, kako bi se stekao utisak kontinuiteta. Zbog određenog stepena slobode koji ima računarski program korišćen za implementaciju ovog modela, on je stohastičan. Početna tačka sa koje počinje melodija je za dve oktave više od pozadinske, one koja svira šablone akorda, jer je to najčešće slučaj u klavirskoj muzici.

D. Definisanje parametara za različite žanrove

Gore opisan proces je poprilično univerzalan i predstavlja tipičan šablon za komponovanje. Na taj način funkcioniše klasičan pop. Međutim, džez je drugačiji, jer on ima posebne skale i progresije akorda koje su zastupljene. Ritam je, takođe, sasvim neobičan. Umesto da je akcenat na prvoj noti u taktu, kod njega je na polovini takta. Retkost je u džezu naići na klasične durske i molske akorde – on koristi „sedmice", koji pored tri pomenuta stupnja sadrže i sedmi, koji im daje utisak „nečistoće." U džezu su dozvoljene nepravilnosti – štaviše, poželjne su. Zbog toga program ima više slobode i dozvoljeno mu je da, prilikom komponovanja džeza, izađe iz okvira skale tonaliteta u kom se nalazi. Ali, sa druge strane, ima manje slobode po pitanju nizanja akorda. Za razliku od klasičnog popa, program koji komponuje džez melodiju bira jednu od osam progresija akorda koje su karakteristične za džez izvedbe.

Džez koji se generiše je više vrsta ,,koktelskog" džeza nego nekog njegovog kompleksnijeg podžanra, poput bi-bapa i svinga. Služi za opuštanje i stvaranje prijatne atmosfere. Pored klavirskih tonova, on može da ima dodat zvuk malog bubnja koji se pušta jednom ili dva puta u taktu, u zavisnosti od šablona za akorde koji se odabere. Ili, može isto to da uradi tako da zvuči kao pucketanje prstiju, a ne mali bubanj. To doprinosi utisku da je žanr kompozicije džez, jer naglašava ritam koji ga odlikuje. Fajlovi koji sadrže ta dva zvuka su takođe u WAV formatu i puštaju se preko bafera, isto kao klavirske note.

Lofi (*low fidelity*) je muzički žanr koji određuje niska tačnost. On se sastoji od džez instrumentala, kombinacije basova i bubnjarskog ritma i namerne produkcijske greške (šuštanje, galama, kiša, škripa i sl.). Iz tog razloga, kao njegova osnova je od džez melodije uzet proces dobijanja melodije, ali je ritam kao kod klasičnog popa, sa akcentom na prvoj noti u taktu. Prilikom generisanja te melodije, nasumično se bira jedan od pet bubnjarskih ritmova karakterističnih za lofi i usklađuje se sa tempom same kompozicije, pa se sve pušta istovremeno. Uz to, moguće je dodati efekat kiše, a uvek se čuje blago šuštanje koje ide u prilog produkcijskoj nesavršenosti koja karakteriše ovaj žanr.

E. Neograničeno duga reprodukcija

Kako bi se muzička melodija neprestano proizvodila i usput menjala, nije dovoljno staviti čitav program u petlju kojom bi se sve iznova izvršavalo. Neki parametri ostaju isti. Štaviše, većina se ne menja. Tempo, ofset, visina osnovnog tona, niz akorda, šabloni u kojima se oni ,,sviraju", tonalitet, a u slučaju džeza i lofija i tip ritma koji se u obliku bubnjeva ili pucketanja prstiju pušta u pozadini – sve je predefinisano i nasumično se određuje samo prilikom pokretanja programa, i nijednom više.

Ono što se menja jeste glavna melodija. Promenljive koje nju određuju jesu upravo skok između dva tona i podatak da li je na više ili niže, kao i njihove notne vrednosti. Ne može se predvideti kojim putem će ići melodija i da li će se kretati ka višim ili nižim tonovima. Zbog toga ona, iako se pozadina nikada ne menja, slušaocu ne može lako dosaditi.

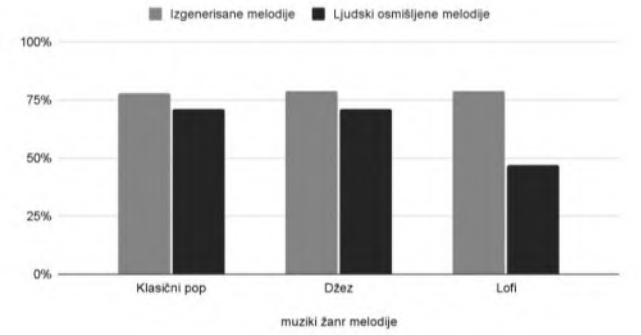
III. REZULTATI I DISKUSIJA

Priroda rada je takva da metriku predstavlja subjektivan osećaj ljudi koji im stvara muzika koju slušaju. Iz tog razloga, napravljen je eksperiment u kom je velik ljudi ocenjivao melodije koje su puštane, bez znanja o tome da li je muziku proizvodio program ili je ljudski izvedena. Upitnik je popunilo 108 ispitanika. Oni su, za svaku od 6 melodija koje su imali priliku da čuju, imali zadatak da pretpostave da li je melodiju generisao računar ili ne, da je na skali od 1 do 10 ocene u skladu sa subjektivnim užitkom u melodiji i da je potom svrstaju u neki muzički žanr.

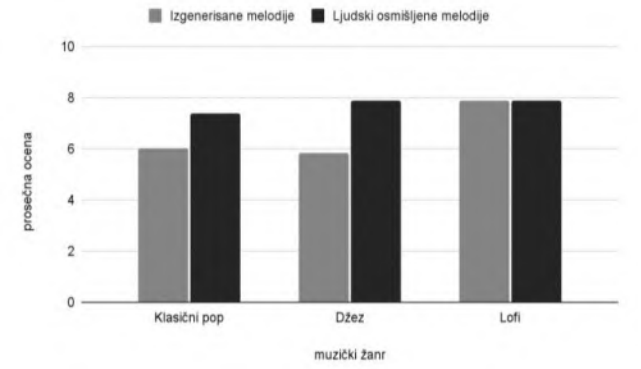
U proseku, ispitanici su ispravno prepoznali da li je muziku izveo čovek ili računar u 69,3% slučajeva. Na grafiku (Slika 2) prikazuje da su melodije lofi žanra koje nisu izvedene od strane računarskog programa konstruisanog za svrhe ovog rada imale značajno manji procenat tačnih odgovora. Veoma je verovatno da su ispitanici posumnjali u izvor poslušane melodije zbog studijskih šumova koji su obavezna karakteristika lofi muzičkog žanra, a koji prosečnom slušaocu mogu delovati kao ,,greška" u programu.

Ocene za računarski generisane melodije su u proseku imale za 0,735 nižu vrednost od onih koje su ljudski komponovane. Na grafiku (Slika 3) su prikazane dodeljene ocene u odnosu na muzički žanr. Primećuje se da je stepen uživanja u melodijama kod ispitanika bio najveći prilikom slušanja lofi muzike dok su pop i džez dobili slične ocene. Za analiziranje podataka dobijenih iz ocena ispitanika takođe je korišćena Fleissova kapa [4]. Fleissova kapa je statistički metod za merenje stepena složnosti između više

od dva ocenjivača prilikom dodeljivanja kategoričkih ocena. Izražava se numerički na skali od -1 do 1, gde 1 predstavlja potpunu složnost, 0 složnost koja bi se dobila slučajnim raspoređivanem, a -1 potpunu neslogu. Računarski izgenerisane melodije su u proseku imale vrednost koeficijenta od -0.012, a ljudski komponovane -0.007. Obe vrednosti su bliske vrednosti složnosti za koju je najveća verovatnoća da bi nastala slučajnošću. Ukusi za muziku se razlikuju u zavisnosti od pojedinca, pa se toj pojavi može prepisati velika raznolikost u dobijenim odgovorima od strane ispitanika.



Slika 2. Grafički prikaz prosečne vrednosti ispravnog prepoznavanja izvora melodije.

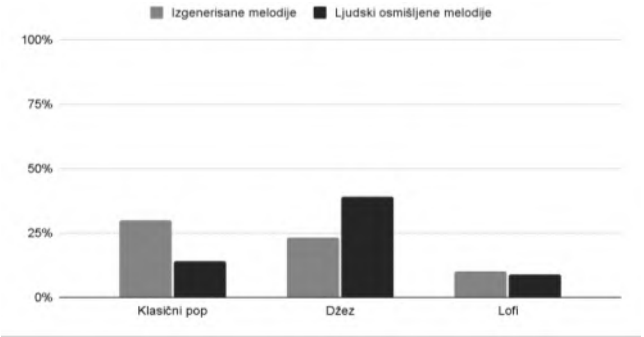


Slika 3. Grafički prikaz prosečne ocene muzičkih melodija u odnosu na žanr kom pripadaju.

Na grafiku (Slika 4) je procentualno prikazan broj tačnih odgovora za pripadnost melodija svakom od tri žanra pojedinačno u odnosu na ukupan broj odgovora. Podatak da nijedan broj ne prelazi vrednost od 40% implicira da prosečan stepen upućenosti ispitanika u osnovne karakteristike pojedinih muzičkih žanrova nije visok, iz razloga što je svaka od nesintetičkih melodija posedovala osnovna svojstva žanra kom pripada. Međutim, džez se ističe po tačnosti, što je verovatno zbog njegove izrazite osobenosti. Sa druge strane, lofi je prepoznat uz najnižu ispravnost. To je moguće iz razloga što je on žanr koji je nastao u drugoj polovini dvadesetog veka, dok su klasična muzika, pop i džez u to vreme već bili izrazito popularni i razvijeni muzički žanrovi.

IV. ZAKLJUČAK

Iz rezultata dobijenih popunjavanjem upitnika od strane više od stotinu ispitanika može se zaključiti da je kvalitet generisanih melodija bio zadovoljavajuć. Razlike između ljudski komponovanih i kompjuterski generisanih muzičkih melodija su primetne, mada ne i velike u odnosu na ukupni opseg skala u pitanju.



Slika 4. Grafički prikaz procentualno izraženog broja melodija ispravno raspoređenih u žanrove kojima pripadaju.

Iako je dobijena muzika dovoljno dobra prema subjektivnim ocenama ispitanika, ima dosta prostora za dalji rad. Prilikom smenjivanja jedne datoteke .wav formata drugom unutar bafera, čuje se blago pucketanje usled naglog prekida koje, ako se slušalac koncentriše na melodiju, može sasvim pokvariti užitak. Rešavanje tog problema uključilo bi povećanje broja bafera ili eksperimentisanje sa ponašanjem datoteka različitog formata u istim uslovima. Međutim, još uvek je neispitana uspešnost bilo koje od te dve metode za rešavanje pomenutog problema. Iako postoje delovi algoritma koji bi se mogli poboljšati, glavni cilj rada je ispunjen i uspešno je kreiran model koji dovoljno dobro proizvodi beskonačne muzičke melodije za komercijalne potrebe.

LITERATURA

[1] Bank, B., Chabassier J., “Model-Based Digital Pianos: From Physics to Sound Synthesis”, *IEEE Signal Processing Magazine*, 36(1), 2019, pp. 103-114.

[2] Univerzitet u Ajovi [Internet]. "Studio za električnu muziku, koncertni klavir Stainway & Sons Model B. UIOWA."s [ažurirano 3. jula 2017, pristupljeno 10. oktobra 2022]. Dostupno na: https://theremin.music.uiowa.edu/MISpiano.html

[3] Ge Wang, Perry Cook [Internet]. "Unit generators." Univerzitet Stanford; [ažurirano 18. maja 2021, pristupljeno 10. oktobra 2022]. Dostupno na: https://chuck.stanford.edu/doc/program/ugen_full.html#StkInstrument

[4] Fleiss, J. L., B. Levin, M. C. Paik, Statističke Metode za Ocene i Proporcije. Treće izdanje. Njujork: *Wiley*, 2003.

Poređenje tačnosti modela random forest i hibridne neuronske mreže na klasifikaciju teksta u tipove ličnosti po Keirsey modelu

Marko Milenković

Sadržaj − Na (MBTI) Myers-Briggs Personality Type Dataset setu podataka odrađen je preprocessing sa *spaCy* bibliotekom nakon čega su podaci klasifikovani korišćenjem random forest metode i hibridnog modela neuronskih mreža (CNN+LSTM). Pre treniranja hibridnog modela je odrađen word embedding unapred istreniranim *tok2vec* alatom koji pruža *spaCy*. Zbog poređenja je takođe kao referentni model uzet hibridni model koji nije koristio *spaCy word embedding* nego *tensorflow word embedding layer*. Random forest model je imao najveći prosečan *f1-score* 70 % sa 10000 stabla, a najmanji prosečan *f1-score* 57 % sa 100 stabla. Hibridni model sa *spaCy word embeddingom* je imao najveći prosečan *f1-score* 65 %, a referentni model je imao maksimalni prosečan *f1-score* 75 %. U radu se pokazalo da je hibridni model precizniji kao i da unapred istreniran *spaCy word embedding* nije adekvatan za primenu na objave koje sadrže puno internet žargona.

I. Uvod

Temperament, kao i ostale klasifikacije ličnosti, se obično određuju ispunjavanjem formulara. Ovo je nepraktično jer zahteva puno vremena i ne može se očekivati od svakoga da izdvoji ponekad i sat vremena za to. Takođe, takav način podrazumeva samoprocenjivanje, što nije uvek precizno. Poslednjih godina je procena ličnosti postala popularna i zato je interesantna tema za mnogo istraživača. Motivacija iza ovog rada je da se pronađe pouzdan način aproksimacije vrste temperamenta koju ima neki pojedinac samo analizom njegovih objava, odnosno, načina pisanja. Da bi se to postiglo mogu se primeniti NLP (*natural language processing*) tehnike [1] i machine learning algoritmi. Za klasifikaciju temperamenta je korišćena podela Davida Keirseya [2] zbog njene praktičnosti i kompatibilnosti sa MBTI klasifikacijom ličnosti [3].

RF (*random forest*) model [4] je supervizovani algoritam za mašinsko učenje [5] koji se koristi za klasifikaciju i regresiju. Prednost je što se jako lako implementira i lako menja.

M. Milenković je učenik 4. razreda IT odeljenja Gimnazije „Svetozar Marković”, Petefi Šandora 1, 24000 Subotica, Srbija; polaznik seminara računarstva u Istraživačkoj stanici Petnica, Selo Petnica, 14104 Valjevo, Srbija, E-mail: mmarkomile@gmail.com.

Ali, zbog svoje jednostavnosti se ne može očekivati da daje najpreciznije rezultate kada su u pitanju komplikovani zadaci. RF modeli se koriste u raznim situacijama - predviđanje berze, dijagnostika u zdravstvu, psihologija, sistemima za preporuku i dr. Neuronske mreže [6] se takođe mogu koristiti u jako puno različitih situacija, pri čemu se koriste različite vrste neuronskih mreža. Tako su i počeli nastajati hibridni modeli u kojima se različite vrste neuronskih mreža sekvencijalno spajaju kako bi se poboljšala njihova preciznost. Hibridni model koji se u ovom radu koristio je spoj CNN (*convolutional neural network* - konvoluciona neuronska mreža) [7] i LSTM (*long short term memory*) neuronske mreže [8]. Razlog zašto je to dobra kombinacija je jer CNN dobro izdvaja bitne delove rečenica, a LSTM efektivno povezuje te delove i pronalazi kontekst.

U prošlim radovima su predstavljeni razni modeli koji uglavnom koriste NLP metode namenjene za istraživanje i naučne radove ili metode koje su prevaziđene [9]-[12],[17]. U ovom istraživanju je korišćena spaCy biblioteka [13] koja je namenjena da bude laka za implementiranje i dostupna svim programerima. Nakon preprocesiranja teksta iz dataseta, tekst se lematizovao alatom iz spaCy biblioteke i onda se od njega pravio *bag of words* u slučaju RF modela. (lematizacija i *bag of words* su objašnjeni u narednom delu rada) U slučaju hibridnog modela se koristio unapred istreniran tok2vec alat iz spaCy biblioteke koji pretvara reči u jedinstvene 300-dimenzionalne vektore koji ih predstavljaju, nakon čega se na njima trenirao i evaluirao model. Cilj je da se proveri da li spaCy biblioteka pruža dovoljan nivo preciznosti i primene, kao i da se poredi RF model sa hibridnim modelom baziranim na neuronskim mrežama.

II. Metode za klasifikaciju

A. Dataset

Korišćen je (MBTI) *Myers-Briggs Personality Type Dataset* [14] koji sadrži preko 8600 redova podataka. Svaki red sadrži, u jednoj koloni tip ličnosti (jedan od 16 MBTI tipova ličnosti), a u drugoj koloni poslednjih 50 objava koju je ta osoba objavila, razdvojenih sa „|||” (3 uspravne crte). Podaci su uzeti sa *PersonalityCafe foruma*.

B. Preprocesiranje

Pošto korisnici društvenih mreža i foruma često koriste neformalni vokabular i različite simbole koji mogu biti korisni za klasifikaciju ličnosti, ali stvaraju velike poteškoće modelima mašinskog učenja i zauzimaju mnogo memorije potrebno je očistiti dataset. Kako bi se pročistio dataset niz operacija je izvršeno na njemu pre nego što je korišćen za treniranje algoritama. Operacije koje se izvršavaju su uklanjanje uspravnih crta, zamena linkova za njihov domen, uklanjanje viška razmaka, uklanjanje objava koje sadrže samo brojeve ili linkove, uklanjanje specijalnih i akcentovanih karaktera, uklanjanje stopwords-a, uklanjanje brojeva, pretvaranje u mala slova, sklanjanje ponavljanih karaktera, uklanjanje reči dužine 1 karaktera i uklanjanje najčešćih i najređih reči. Takođe, izdvojeni su podaci koji bi mogli biti korisni RF modelu: broj reči napisanih velikim slovima (u celini), broj stopwords-a, broj linkova, broj reči, prosečna dužina reči.

Zato što je korišćen dataset namenjen za MBTI klasifikaciju, bilo je potrebno mapirati MBTI klase u *Keirsey* model. „ISTJ" „ISFJ" „ESFJ" „ESTJ" su mapirani u 0 (*Guardian*), „ISFP" „ISTP" „ESFP" „ESTP" su mapirani u 1 (*Artisan*), „INFJ" „INFP" „ENFP" „ENFJ" su mapirani u 2 (*Idealist*), „INTJ" „INTP" „ENTP" „ENTJ" su mapirani u 3 (*Rationalist*).

C. Primena spaCy biblioteke

spaCy je *natural language processing* biblioteka za *Python*. Namenjena je za programere i napravljena je tako da je jednostavna za korišćenje. Korišćena je za lematizaciju [15] reči nakon prvog dela preprocesiranja. Lematizacija reči je proces u kojem se reči pretvaraju u svoj koren (lemu). To je korisno jer drastično smanjuje količinu podataka i gube se nepotrebne razlike između reči koje u osnovi nose isto značenje, ali su različitog oblika. Korišćenjem alata koji pruža biblioteka preprocesirane reči se zamenjuju njihovim lemama.

D. Random forest model

Kako bi se uzeo ujednačen uzorak sve 4 klase, nasumično se odabralo 450 objava od svake klase koje će se koristiti za treniranje modela. Da bi objave koje se nalaze u *datasetu* mogle da se koriste za učenje RF modela, prvo se pravi *bag of words* [16] svih objava. Ovako se pravi vektor celih brojeva dužine broja jedinstvenih reči u svim objavama koji predstavljaju da li i koliko puta se neka reč nalazi u objavi. *Bag of words* vektor se spaja sa ostalim izdvojenim podacima dobijenim tokom preprocesiranja i dobijeni dataset se koristio za treniranje i evaluaciju. Model je treniran sa 100, 200, 500, 1000 i 10000 stabla, a za treniranje je korišćeno 80% odabranih objava. Ograničenja dubine nisu postavljena.

E. Hibridni deep learning model

Hibridni model koji je evaluiran je hibridni model pokazan u istraživanju *A Hybrid Deep Learning Technique for Personality Trait Classification From Text* [17]. Prvi sloj modela u pomenutom istraživanju je *embedding layer*. Njegova uloga je da pretvori reči u vektore koji mogu da se koriste u narednim delovima modela. On se iterativno poboljšava kao i neuronske mreže da bi bolje predstavio reči vektorima. *Embedding layer* tog modela je zamenjen unapred istreniranim *spaCy tok2vec* alatom. *Tok2vec* od reči u stringu (tokena) pravi 300-dimenzionalne vektore koji predstavljaju tu reč u vektorskom prostoru. Ovom metodom svaka reč iz *spaCy* vokabulara može da se predstavi jedinstvenim vektorom. Da bi se dobijena lista vektora mogla koristiti za treniranje odrađen je *padding* (na kraju svake liste dodat je niz nula vektora) kako bi sve liste bile iste dužine. Odrađen je i *one hot encoding* labela u 4 dimenzionalni vektor.

Takođe, za razliku od spomenutog modela, korišćen je Adam [18] optimizer jer je pokazao malo bolje rezultate od Adamax [18]. LSTM mreža je trenirana sa 120 jedinica.

Kao referentni model uveden je i model koji ne koristi *spaCy* biblioteku, nego sadrži *embedding* sloj u sebi (ovaj model je skoro identičan kao u ranije spomenutom istraživanju).

F. Evaluacija modela

Za evaluaciju tačnosti modela gledao se *f1-score*, *preciznost* i *recall* za svaku od klasa. Ovo je bilo potrebno zbog toga što nisu u svim slučajevima korišćeni uravnoteženi datasetovi i zbog toga što sve te metrike zajedno daju kompletniju sliku performansi modela.

TABELA I
RANDOM FOREST MODEL (UZORAK OD 450)

broj stabla	f1-score (%)				recall (%)				preciznost (%)			
	0	1	2	3	0	1	2	3	0	1	2	3
100	56	59	53	6o	53	61	53	61	60	57	53	59
200	59	61	58	64	54	66	56	68	65	58	61	60
500	67	65	61	67	61	68	59	72	75	62	64	62
1000	69	68	66	65	60	73	67	69	82	64	66	62
2000	72	68	69	69	63	74	69	71	83	63	69	67
10000	71	68	69	68	63	73	69	70	80	63	69	66

III. REZULTATI I DISKUSIJA

U rezultatima RF modela (Tabela I) se vidi da se povećavanjem broja stabla, povećava i *f1-score*. *F1-score*

od 70% se računa kao prosek sa 10000 stabla i predviđen je kao gornja granica mogućnosti RF modela.

Hibridni model je ispitan u 2 okolnosti. U prvom slučaju je nasumično odabrano 450 uzoraka od svake klase i to je korišćeno kao set za treniranje i validaciju. U drugom slučaju korišćen je ceo dataset. Ovo služi da poredimo koliko je bitna ravnomerna raspodela klasa. Po rezultatima hibridnog modela (Tabela II i Tabela III) možemo videti da su modeli imali bolji *f1-score* kada se koristi ceo *dataset*. Takođe se primećuje značajno veća preciznost modela koji ne koristi *spaCy* biblioteku (Tabela III) u oba slučaja. Za sve slučajeve neuronska mreža dostiže svoju maksimalnu preciznost oko 10 epoha, nakon čega počinje da opada ili stagnira zbog overfitovanja. Ovo se možda može povezati sa ograničenjima *dataseta*. Maksimalni prosečni *f1-score* RF modela je 70% sa 10000 stabla pri korišćenju uzoraka od 450 redova. Maksimalni prosečni *f1-score* neuronske mreže je iznosio 75% u modelu bez *spaCy* biblioteke i 65% u modelu sa spaCy bibliotekom, u oba slučaja korišćenjem celog *dataseta*.

Bitno je napomenuti da se govori o *weighted average* načinu računanja proseka. To podrazumeva proporcije labela u *datasetu*. U slučaju *macro average* načina, koji ne uvažava proporcije labela rezultati su drugačiji. Ovo se da primetiti u modelu sa *spaCy* bibliotekom koji je koristio ceo *dataset* (Tabela II).

TABELA II
HIBRIDNI MODEL + SPACY

dataset	f1-score (%)				recall (%)				preciznost (%)			
	0	1	2	3	0	1	2	3	0	1	2	3
uzorci od 450	42	38	51	47	49	31	53	46	36	48	49	49
ceo dataset	0	1	72	66	0	1	81	68	0	33	65	64

Ne gledajući na proporcije, ovaj model izgleda kao da ima jako loš prosek (35%) jer je zbog neravnomerno raspoređenih klasa u datasetu skoro sasvim zanemario klase 0 i 1.

IV. ZAKLJUČAK

U rezultatima se može videti da je najbolji metod hibridni model bez korišćenja *spaCy* biblioteke. Ovo je najverovatnije slučaj zbog toga što internet žargon sadrži jako širok spektar reči koje se ne nalaze u *spaCy* vokabularu, pa ne može napraviti odgovarajuće vektore.

Takođe je bitno napomenuti da je dataset prilično ograničen. Sadrži samo oko 8600 osoba što je veoma malo za potrebe neuronskih mreža. Veliki broj objava ne sadrži značajne podatke što još više smanjuje preciznost modela. Veoma su neujednačeni uzorci ličnosti jer se klasa 0 pojavljuje samo oko 450 puta, a klasa 3 čak oko 4500 puta, zbog ovoga su modeli naklonjeni ka češćim klasama.

To je i bio razlog za biranje uzoraka od 450 u testiranju. Pokazalo se da je ipak bolje koristiti ceo *dataset*, bez obzira na to što su klase neravnomerno raspoređene, ali treba obratiti pažnju da model ne prestane razmatrati one klase koje se retko pojavljuju.

U daljem istraživanju bi se mogao sličan ogled odraditi na malo boljem datasetu. Iako su očigledna ograničenja dataseta, unapred istreniran *spaCy tok2vec* alat je pokazao da nije adekvatan za primenu na objave na internetu koje sadrže veliku količinu internet žargona. *spaCy* je pored toga bio sasvim adekvatan kao alat za preprocesiranje teksta, odnosno pretvaranja reči u njihove leme.

TABELA III
HIBRIDNI MODEL BEZ SPACY

dataset	f1-score (%)				recall (%)				preciznost (%)			
	0	1	2	3	0	1	2	3	0	1	2	3
uzorci od 450	66	55	47	64	69	66	34	64	63	47	72	63
ceo dataset	50	50	82	76	58	42	88	70	43	63	76	83

LITERATURA

[1] Chowdhary, K., Natural language processing. *Fundamentals of artificial intelligence*, 2020, pp.603-649.

[2] Keirsey, D. & Bates, M.M., 1978. *The sixteen types*, Prometheus Nemesis Book Co.

[3] Myers, I.B., The Myers-Briggs Type Indicator: Manual 1962.

[4] Breiman, L., Random forests. *Machine learning*, 2001, 45(1), pp.5-32.

[5] Kotsiantis, S.B., Zaharakis, I. and Pintelas, P., Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 2007, 160(1), pp.3-24.

[6] Bishop, C.M., 1994. Neural networks and their applications. *Review of scientific instruments*, 65(6), pp.1803-1832.

[7] Albawi, S., Mohammed, T.A. and Al-Zawi, S., 2017, August. *Understanding of a convolutional neural network*. In *2017 international conference on engineering and technology (ICET)* (pp. 1-6). IEEE.

[8] Sherstinsky, A., 2020. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena*, *404*, p.132306.

[9] Arnoux, P.H., Xu, A., Boyette, N., Mahmud, J., Akkiraju, R. and Sinha, V., 2017, May. 25 tweets to know you: A new model to predict personality with social media. In Eleventh international AAAI conference on web and social media.

[10] Keh, S.S. and Cheng, I., 2019. Myers-Briggs personality classification and personality-specific language generation using pre-trained language models. arXiv preprint arXiv:1907.06333.

[11] Lima, A.C.E. and de Castro, L.N., 2019. TECLA: A temperament and psychological type prediction framework from Twitter data. Plos one, 14(3), p.e0212844.

[12] Wang, X., Sui, Y., Zheng, K., Shi, Y. and Cao, S., 2021. Personality classification of social users based on feature fusion. Sensors, 21(20), p.6758.

[13] Honnibal, M., Montani, I., Van Landeghem, S. and Boyd, A., 2020. spaCy: Industrial-strength natural language processing in python.

[14] J, M., 2017. (MBTI) Myers-Briggs personality type dataset. *Kaggle*. Available at: https://www.kaggle.com/datasets/datasnaek/mbti-type [Accessed September 25, 2022].

[15] Balakrishnan, V. and Lloyd-Yemoh, E., 2014. Stemming and lemmatization: A comparison of retrieval performances

[16] Zhang, Y., Jin, R. and Zhou, Z.H., 2010. Understanding bag-of-words model: a statistical framework. *International journal of machine learning and cybernetics*, *1*(1), pp.43-52.

[17] Ahmad, H., Asghar, M.U., Asghar, M.Z., Khan, A. and Mosavi, A.H., 2021. A hybrid deep learning technique for personality trait classification from text. *IEEE Access*, *9*, pp.146214-146232.

[18] Kingma, D.P. and Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Korišćenje LSTM neuronske mreže za prepoznavanje varanja prilikom ciljanja u video igri CS:GO

Miša Stefanović

Sadržaj − U ovom radu je ispitana pouzdanost LSTM neuronske mreže za prepoznavanje varanja prilikom ciljanja u video igri CS:GO [2]. Za prepoznavanje su korišćeni podaci o položaju igračeve kamere, položajima za koje igrač treba da pomeri kameru da bi naciljao protivnika. kao i da li igrač trenutno gađa. Skup podataka se sastojao od 50000 uzoraka poštenih igrača i 50000 igrača koji varaju. Tokom rada su rađeni razni eksperimenti gde su isprobavane razne arhitekture mreže, kao i razne obrade *dataseta*. Najbolji model je koristio 200 neurona, nije imao nikakve naknadne obrade *dataseta* i imao je preciznost od 82.26 % i F1 *score* od 46 %.

I. UVOD

Varanje predstavlja svako kršenje određenih pravila, u cilju ostvarivanja nekih prednosti. U današnjem vremenu, varanja ima u svim sferama, tako da ni industrija video igara nije izuzetak. U video igrama se varanje izvršava pomoću namenskih hardvera i softvera ili pomoću već postojećih grešaka u kodu igrice.

Counter Strike : Global Offensive je video igra koju je napravila kompanija *Valve.* Igra je akcionog tipa, i u njoj se igrači bore da eliminišu protivnički tim ili ispune zadatak. Međutim, jedan od vodećih problema sa kojima se igrica bori jeste veliki broj igrača koji varaju.

Najpopularniji način za varanje u igri jeste pomoću *aimbota. Aimbot* je tip softvera za varanje (eng. *cheat*), koji pomaže igraču da lakše nacilja na protivnika tako što cilja umesto igrača simuliranjem pomeranja miša.

Igra trenutno ima 2 glavna načina odbrane protiv varanja: njihov softver zvani VAC (koji je neefektivan), kao i program zvani *Overwatch*, gde igrači mogu da gledaju meč osumnjičenog igrača, te da mu „presude" da li je stvarno varao.

Ovaj rad je pokušaj da se napravi treći način, koji će pomoću neuronske mreže odlučivati o legitimnosti igračevih mečeva.

II. METOD

Implementacija je rađena u programskom jeziku *Python* [3], i za nju su korištene biblioteke *NumPy* [4], *TensorFlow* [5], *Keras* [6], *Numba* [7] i *scikit-learn* [8].
Prva faza rada je bila prikupljanje podataka.

M. Stefanović je polaznik seminara računarstva u Istraživačkoj stanici Petnica, Omladinskih Brigada 10/13, 19250 Majdanpek, E-mail: mikmik1011@proton.me.

Budući da je za dobar potrebno mnogo uzoraka, a prikupljanje istih u igrici proces koji iziskuje mnogo vremena i ljudi, odustalo se od ideje ručnog skupljanja i iskorišćen je [9] preuzet sa interneta.

Podaci u njemu su bili u obliku četvorodimenzionalne matrice, raspoređeni po sledećoj hijerarhiji:
igrač -> uzorci -> vremenski trenuci -> parametri

Parametri u *datasetu* su: vrednosti vertikalnog (u daljem tekstu: *pitch*) i horizontalnog (u daljem tekstu: *yaw*) ugla kamere napadača, vrednosti *pitch*-a i *yaw*-a za koje napadač treba da pomeri kameru da bi naciljao na protivnika i *firing*, tj. da li napadač gađa iz oružja. Iako ih ima malo, ovi parametri se pojavljuju u skoro svim video igrama tog žanra, te se rezultati ovog rada mogu primeniti na većinu sličnih igara.

Ti parametri su beleženi 32 puta po sekundi, te se svaki uzorak sastoji od 6 sekunde igre. Od tih 6 sekundi, 5 zabeleženih sekundi su pre, a zadnja sekunda posle pogotka. Autor *dataseta* je grupisao te uzorke po igraču, i to za svakog igrača po 30 uzoraka.

U drugoj fazi je pristupljeno pripremi *dataseta*. Pre svega je razbijeno grupisanje uzoraka po igračima, pa je svaki uzorak postao nezavisan jedan od drugog. Tako je dobijeno 300000 uzoraka poštenih igrača i 60000 uzoraka varanja. Pošto su uzorci varanja činili samo jednu šestinu svih podataka, za dalje korišćenje je odabrano samo prvih 60000 uzoraka poštenih igrača, te je tako *dataset* postao ravnomerno izbalansiran. Nakon toga je *dataset* podeljen na train i test set, sa 80% podataka u *train* setu i 20% u test setu, za evaluiranje mreže.

U trećoj fazi je izvršeno testiranje *dataseta*. Algoritmi korišćeni za testiranje su *Zero Rule* i *One Rule*. *Zero Rule* algoritam je pokazao preciznost od 50 % (što je i bilo očekivano, budući da je *dataset* bio prethodno izbalansiran). Za njegovu implementaciju je korišćen *DummyClassifier* iz *scikit-learn* biblioteke. *One Rule* je pokazao najveću preciznost od 57 %. On je implementiran tako što su svi vremenski trenuci zajedno spojeni u istu dimenziju sa parametrima, pa je tada *dataset* iz trodimenzionalnog postao dvodimenzionalan.

U četvrtoj fazi je napravljen odabir arhitekture i hiperparametara mreže, kao i biblioteka za njenu implementaciju. Kako bi se izabrala glavna arhitektura mreže koja će se koristiti za dalje oglede, vršena su merenja preciznosti i F1 - *scorova* različitih arhitektura. Merene arhitekture su imale po 32, 64, 96, 128, 160 i 200 LSTM ćelija. Odabrana je arhitektura koja se sastoji od

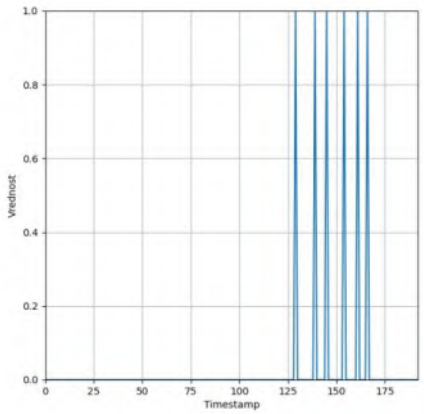
jednog LSTM sloja sa 200 neurona, i sloja sa jednim *output* neuronom koji je imao *Sigmoid* aktivacionu funkciju.

U poslednjoj fazi su vršeni različiti ogledi za koje su vršene dodatne obrade *dataseta*, kako bi se metrike potencijalno poboljšale i kako bi se probao izvući maksimum iz podataka. Tokom ovih ogleda sve vreme je merena preciznost mreže, kao i F1-*score*.

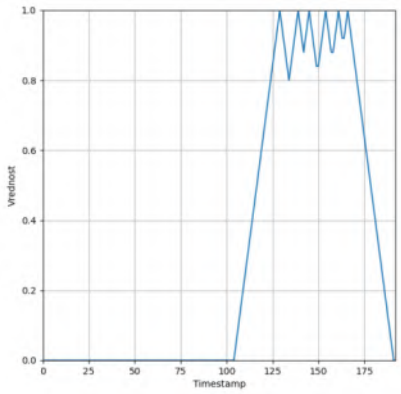
Prvo je izvršena relativizacija *dataseta*. *Pitch* i *yaw* vrednosti u svim vremenskim trenucima u uzorku su oduzeta od vrednosti u prvom trenutku, pa se tako dobio *patern* pomeranja *pitch*-a i *yaw*-a.

Zatim je izvršena min-max normalizacija. Od vrednosti svakog parametra je oduzeta najmanja vrednost tog parametra u uzorku, pa je ta razlika podeljena sa razlikom najveće i najmanje vrednosti tog parametra, pa su se tako dobile vrednosti između 0 i 1.

Takođe, u jednom ogledu je izvršen i *smoothing firing* parametra. Budući da je vrednost *firing*-a uvek 0 ili 1, ovim putem je interpolirano postepeno povećanje i smanjivanje vrednosti kako se približava trenutak gađanja (slika 1).



(a)



(b)

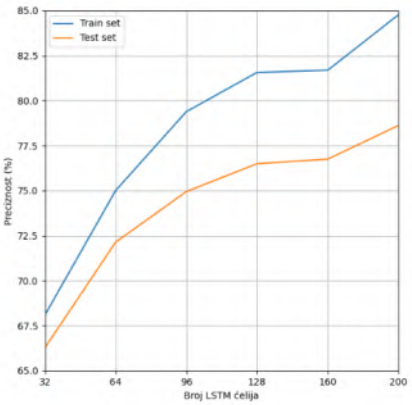
Slika 1. *Firing* parametar (a) bez *smoothinga* i (b) sa smoothingom.

Prilikom tog ogleda je postepeno menjana razlika između 2 interpolirane vrednosti. Te razlike su: 1 (tj. bez interpolirane vrednosti između), 0.5, 0.33, 0.25, 0.2, 0.14, 0.1, 0.07, 0.05, 0.04, 0.03, 0.02 i 0.01.

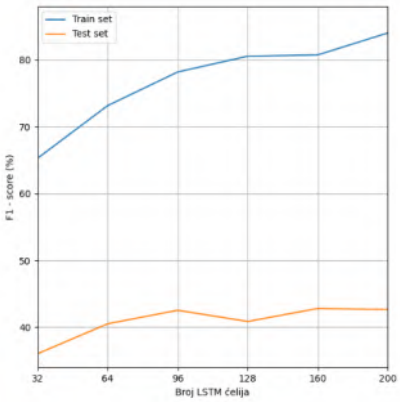
Na kraju su postepeno bili dodavani uzorci poštenih igrača sve dok nisu svi dostupni uzorci bili ubačeni i balans mreže kompletno izbačen iz ravnoteže. Najpre je broj uzoraka poštenih igrača bio dodavan u skokovima od po 5000 između 50000 i 80000 uzoraka, a nakon toga je vrednost dodavana u skokovima od po 30000. Razlog tome je što bi se mreža u praksi susrela sa mnogo većim brojem fer uzoraka nego varanja. Pošteni igrači su samo dodavani u train setu, test set je i dalje ostao uravnotežen.

III. Rezultati

Merenje arhitektura sa različitim brojem LSTM neurona je pokazalo da preciznost i *train* i test seta rastu srazmerno sa brojem neurona, ali da takođe i njihova razlika raste, što znači da se model lakše *overfituje* sa većim brojem ćelija. (slika 2, tabela I). Ono što je takođe interesantno je što F1 *score* na test setu kreće da oscilira već nakon 64 neurona.



(a)



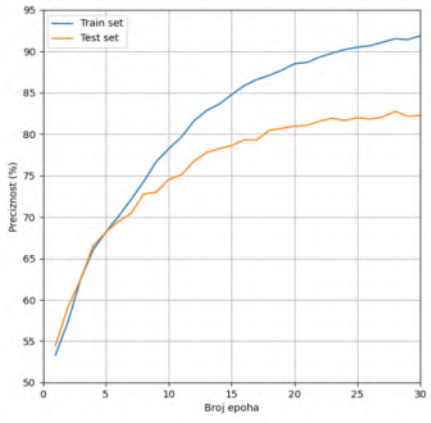
(b)

Slika 2. Grafikon odnosa broja LSTM ćelija i (a) preciznosti i (b) F1-score-a.

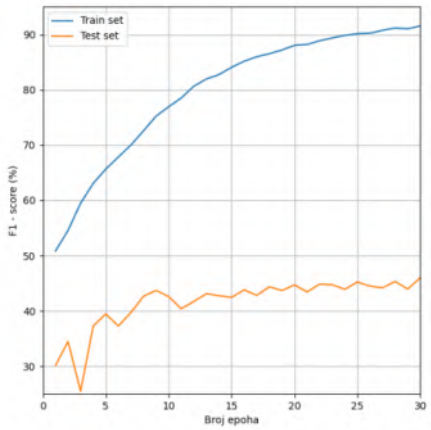
Treniranjem odabrane arhitekture pomoću 30 epoha se vidi da već nakon 15 epoha F1 *score* kreće da oscilira za 1-2 %, a da preciznost tada jako usporava, i da se za 5 epoha povećala za samo 2.36 %, a da nakon toga stagnira, te je za preostalih 10 epoha skočila samo 1.26 %.

TABELA I
POREĐENJE METRIKA U ODNOSU NA BROJ LSTM NEURONA

Broj LSTM neurona	Preciz-nost na train setu (%)	Preciz-nost na test setu (%)	F1 - score na train setu (%)	F1 - score na test setu (%)
32	68.10	66.28	65.28	36.02
64	75.01	72.14	73.16	40.50
96	79.39	74.95	78.20	42.53
128	81.57	76.50	80.57	40.85
160	81.69	76.75	80.76	42.80
200	84.77	78.62	84.05	42.64



(a)

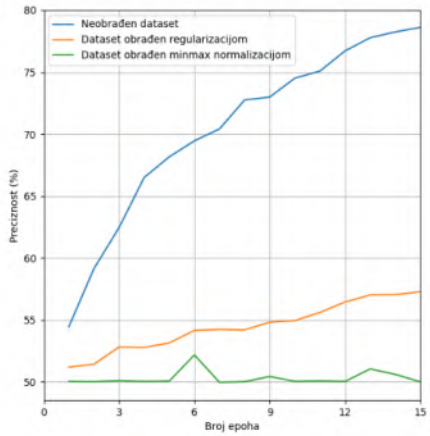


(b)

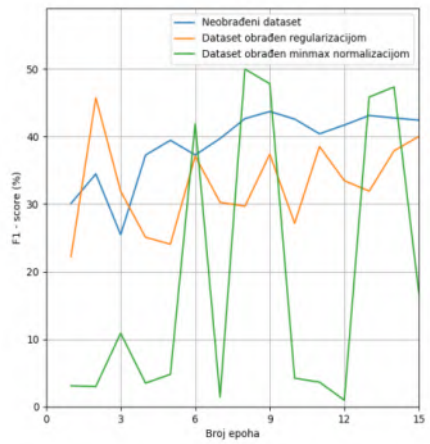
Slika 3. Grafikon (a) preciznost i (b) F1- scora u odnosu na broj epoha.

Relativizacija i normalizacija nisu uopšte pomogli da se poboljša bilo koja metrika, čak naprotiv, sve metrike su se znatno pogoršale.

Ono što možemo da primetimo kod min-max normalizacije je da je preciznost ista kao i prilikom bacanja novčića (tj, 50 %), a da F1-*score* haotično skače i pada, pa se može doći do zaključka da mreža tada nasumično predviđa odgovor.



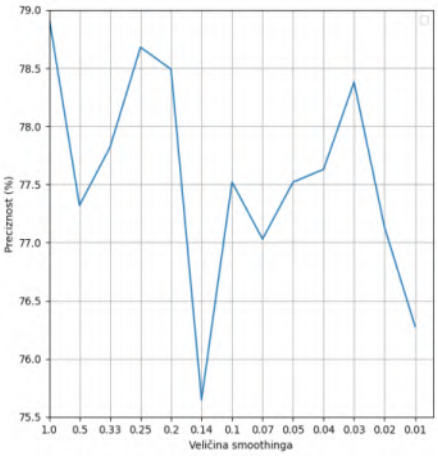
(a)



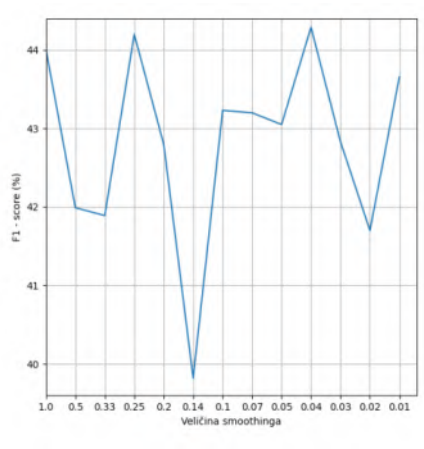
(b)

Slika 4. Grafikon odnosa (a) preciznosti i (b) F1 - *scora datasetova* obrađenih različitim metodama i broja epoha.

Smoothing je, pak, bio nešto uspešniji, ali nije došlo do poboljšanja metrika.



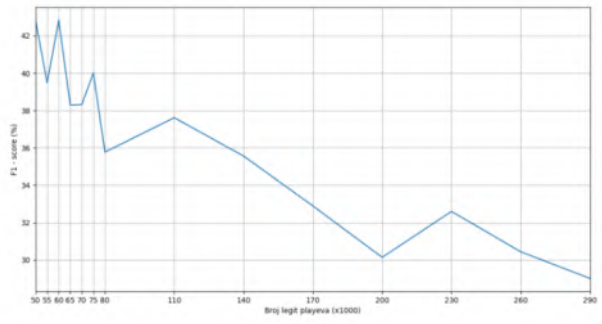
(a)



(b)

Slika 5. Grafikon odnosa preciznosti i (a) veličine smoothinga i (b) F1 – *scora*.

Iako je od povećanog *dataseta* bilo očekivano da će u nekom momentu biti po metrikama bolji od klasičnog, ni to se nije desilo.



Slika 6. Grafikon odnosa F1-scora i broja uzoraka fer igrača.

IV. Zaključak

Iz svega ovoga možemo videti da je najbolje rezultate dala mreža od 200 epoha, sa *datasetom* nad kome nisu rađene dodatne obrade, nakon 30 epoha. Tada je model imao preciznost od 82.26 % i F1 *score* od 46 %, iako su metrike na *train* setu veće, zbog *overfitovanja*.

Uprkos tome što je 82 % značajno veća preciznost od 57 %, koliko je dobijena pomoću *One Rule* algoritma, mreža i dalje nije dovoljno razvijena da bi se mogla zapravo primeniti u praksi. Uzrok toga je sam *dataset*. Tu je najveći problem nedovoljan broj parametara, jer mreža nije mogla lepo da utvrdi povezanost. Smatram da bi mreža mnogo bolje radila kada bismo imali razne podatke o igračima, kao što su njihova pozicija, brzina, tim, itd. Takođe, nije ni poznato kakav skup veština poseduju igrači koji nisu varali, niti tačan način ponašanja *aimbota*. To u mnogome utiče na vrednost samih parametara koji su ovde iskorišćeni, pa bi mešanje različitih vrsta *aimbotova* i igrača moglo otežati posao mreži, iako je u realnosti takav tip podataka nešto sa čime bi se mreža i susretala.

Literatura

[1] S. Hochreiter and J. Schmidhuber, “Long short-term memory”, *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[2] ITN Team, Opis : Counter-Strike: Global Offensive, 2016. https://web.archive.org/web/20161024094407/http://www.itnetwork.rs/Opis-Counter-Strike-Global-Offensive-article-5412.htm

[3] G. Van Rossum and F. L. Drake, Python 3 Reference Manual. Scotts Valley, CA: CreateSpace, 2009.

[4] C. R. Harris et al., “Array programming with NumPy”, *Nature*, vol. 585, pp. 357–362, 2020, doi: 10.1038/s41586-020-2649-2.

[5] M. Abadi et al., “Tensorflow: A system for large-scale machine learning”, in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 265–283.

[6] F. Chollet and others, “Keras”, 2015. https://github.com/fchollet/keras

[7] S. K. Lam, A. Pitrou, and S. Seibert, “Numba: A llvm-based python jit compiler”, in *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, 2015, pp. 1–6.

[8] F. Pedregosa et al., “Scikit-learn: Machine learning in Python”, *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[9] E. Laiho, “CSGO cheating dataset”, 2022. https://www.kaggle.com/datasets/emstatsl/csgo-cheating-dataset

Kontrola rada DC motora pomoću Arduino okruženja i TTP223-BA6 *touch* senzora

Petar Zdravković, Luka Nikolić, Vukašin Arsić

Sadržaj – U ovom radu je prikazan princip kontrole DC motora korišćenjem *Arduino UNO* razvojnog okruženja. Uključivanje i isključivanje motora kontroliše se korišćenjem dva tač (*touch)* senzora povezanih na *Arduino UNO* razvojno okruženje.

I. UVOD

Veliki broj uređaja koje danas vidimo oko sebe sadrže električni motor [1]. Električni motori su elektronski uređaji čiji rad se zasniva na principu pretvaranja energije. Oni uzimaju električnu energiju i pretvaraju je u mehaničku. Električni motori se koriste za kontrolu raznih uređaja koje koristimo u svakodnevnom životu. Često su deo uređaja koji se koriste u domaćinstvima, i neizostavni deo u skoro svim segmentima industrije. Takođe, primenu mogu imati i u računarskoj opremi, automobilima, itd. Električni motori su klasifikovani u dve različite kategorije: motori jednosmerne struje (DC) i motori naizmenične struje (AC). DC motori se uglavnom prave za manje snage, jer bi za veće snage oni morali da imaju veće i jače izvore jednosmerne struje. Električni motori uglavnom rade na principu elektromagnetne indukcije, mada postoje i elektromotori koji koriste druge elektromehaničke fenomene, kao što su elektrostatička sila i piezoelektrični efekat. Kada se kaže elektromotor, uglavnom se misli na rotacioni elektromotor tj. motor koji stvara obrtno kretanje.

Za kontrolu uključivanja i isključivanja motora u ovom radu iskorišćeni su tač (*touch*) senzori umesto mehaničkih tastera. Razlog za to je što su manje podložni fizičkim oštećenjima zato što nemaju mehaničke delove, te traju duže. Pri izradi ovog sistema konkretno su korišćeni *TTP223-BA6 touch* senzori.

II. PRINCIP RADA SISTEMA

Kontrola upravljanja DC motorom je ostvarena pomoću tastera osetljivog na dodir kapacitivnog tipa. Kapacitivni modul (slika 1) senzora dodira zasnovan je na integrisanom kolu TTP223 [2]. Modul sadrži jednu integrisanu oblast osetljivu na dodir površine 11x10,5 mm pri čemu je opseg osetljivosti senzora <5 mm. LED dioda na kapacitivnom modulu će dati vizuelnu indikaciju kada se senzor aktivira.

P. Zdravković, L. Nikolić i V. Arsić su učenici trećeg razreda prirodno-matematičkog smera gimnazije „Bora Stanković“, Vožda Karađorđa 27, 18000 Niš, Srbija, E-mail: petarzdravkovic73@gmail.com.



Slika 1. TTP223 kapacitivni senzor za dodir [3].

Imajući u vidu da pomenuti taster poseduje jedan izlazni pin, za potrebe detekcije nivoa signala od strane *Arduino*-a neophodno je povezati ga na neki od pinova koji su opšte namene, odnosno GPIO tipa. Nakon pritiska tastera i detekcije od strane mikrokontrolera na *Arduino* razvojnom okruženju, preduzima se generisanje odgovarajućeg nivoa signala putem GPIO pina prema MOSFET tranzistoru IRF510 [4] u cilju uključivanja DC motora. U ovom slučaju MOSFET se ponaša kao prekidač prilikom uključivanja i isključivanja DC motora.

Dioda dozvoljavaja struji da teče samo u jednom smeru (od anode ka katodi). Kada se isključi motor, dobija se negativan naponski impuls, koji može oštetiti *Arduino* ili tranzistor. Dioda štiti ostatak kola od negativnog napona tako što se strujno kolo u tom slučaju zatvara preko diode koja postaje direktno polarisana.

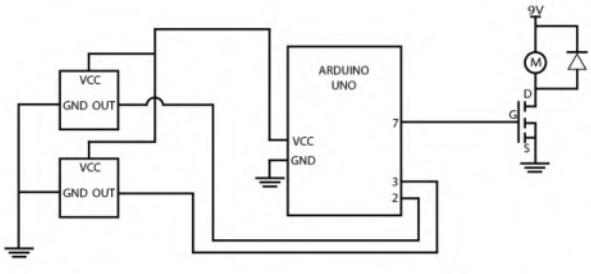
Imajući u vidu da je izlazna struja GPIO pina na *Arduino* razvojnom okruženju reda veličine 20 mA i napon napajanja 5 V, što je nedovoljno za potrebe rada motora, koristi se dodatni izvor napajanja u vidu baterije od 9 V.

III. REALIZACIJA SISTEMA

A. Električna šema sistema

Na električnoj šemi se nalaze dva TTP223-BA6 tač senzora koji se koriste za uključivanje i isključivanje motora. VCC i GND pinovi tastera su adekvatno povezani na napon od 5 V i 0 V *Arduino*-a redom, dok su OUT pinovi povezani na pinove 2 i 3 *Arduino*-a preko kojih se detektuju promene stanja na senzorima.

Na pin 7 *Arduino*-a vezan je gejt (*gate*) pin tranzistora, sors (*source*) pin tranzistora povezan je na masu. Drejn (*drain*) pin tranzistora vezan je za jedan kontakt motora, dok je drugi kontakt motora povezan na plus kraj baterije. Minus kraj baterije vezan je na masu. Skicirana šema prikazana je na slici 2.



Slika 2. Električna šema sistema.

B. Kod za Arduino okruženje

Za realizaciju koda korišćen je softver *Arduino IDE* što je prikazano na slici 3.

```
proba_1 | Arduino 1.8.19 (Windows Store 1.8.57.0)
File  Edit  Sketch  Tools  Help
Upload
proba_1 §
#define taster1 2
#define taster2 3
#define motor 7
void setup() {
  // put your setup code here, to run once:
  pinMode(taster1,INPUT);
  pinMode(taster2,INPUT);
  pinMode(motor,OUTPUT);
  digitalWrite(motor,LOW);
  Serial.begin(9600);

}

void loop() {
  // put your main code here, to run repeatedly:
  if(digitalRead(taster1)==HIGH){
    digitalWrite(motor,HIGH);
    Serial.println("Motor je ukljucen");
    delay(100);
    }
  if(digitalRead(taster2)==HIGH){
    digitalWrite(motor,LOW);
    Serial.println("Motor je iskljucen");
    delay(100);
    }

}
```

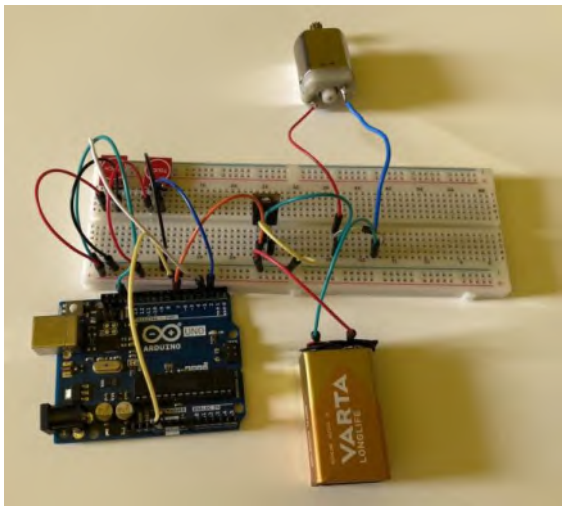
Slika 3. Kod u Arduino okruženju.

C. Praktična realizacija

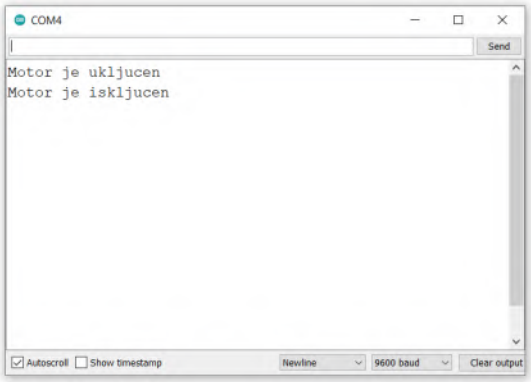
Sistem je nakon skiciranja praktično realizovan sa realnim komponentama, kod je otpremljen na *Arduino*, a potom je sistem uspešno pokrenut i ponašao se kao što je očekivano. Praktično realizovan sistem prikazan je na slici 4, a rad sistema na slici 5.

IV. Zaključak

U ovom radu je prikazan princip kontrole rada DC motora koji se uključuje i isključuje korišćenjem dva tač senzora povezanih na *Arduino UNO* razvojno okruženje. Ovaj jednostavan sistem može pronaći svoju primenu u uređajima koji se koriste u domaćinstvu poput ventilatora, blendera i sl. Dalje unapređenje ovog sistema ogledalo bi se u dodatnim prekidačima koji bi podešavali brzinu ili smer obrtanja motora.



Slika 4. Izgled povezanog sistema.



Slika 5. Sistem u radu.

Zahvalnica

Ovaj rad nastao je kao rezultat realizacije projekta „Let STEM visit again IEEESTEC“, finansiranog u okviru 2022 IEEE Pre-University STEM Portal Grant Program-a. Autori se zahvaljuju Milanu Stojanoviću, asistentu sa Katedre za mikroelektroniku na Elektronskom fakultetu u Nišu, na podršci i pruženoj pomoći pri izradi rada.

Literatura

[1] Tehnička dokumentacija za DC motor dostupna na: https://sr.jf-parede.pt/what-is-dc-motor-basics
[2] Tehnička dokumentacija za *TTP223-BA6 touch sensor* dostupno na: https://www.alldatasheet.com/datasheet-pdf/pdf/1132198/ETC1/TTP223-BA6.html
[3] Fotografija tač senzora sa: https://cutt.ly/wBzbrVX
[4] Tehnička dokumentacija za MOS tranzistor IRF510. Dostupnona: https://www.vishay.com/docs/91015/irf510.pdf

Tsp biblioteka

Milan Stojanović, Aleksandar Panić, Ana Jovanović, Bojan Ćirić

Sadržaj – Ovaj rad predstavlja pregled veb aplikacije za digitalizaciju biblioteke. Aplikacija ima mogućnost komunikacije i manipulacije bazom podataka sa podacima o knjigama, magazinima i novinama i bazom podataka koja sadrži podatke o svim članovima biblioteke.

I. Uvod

Usred prevelikog broja knjiga i zastarelog načina rada u bibliotekama dolazi do prenatrpanosti i sporog rada. Iako je primena digitalnih knjiga sve veća, nije moguće digitalizovati sve knjige, a i da jeste trajalo bi predugo. Kako bi i biblioteke išle u korak s vremenom, i samim tim održale tradicionalni način čitanja i smanjile potrebu za digitalnim knjigama, naš tim razvio je sistem koji olakšava i ubrzava rad biblioteka.

Ovaj sistem stvoren je s ciljem digitalizacije rada biblioteke, izmena starih članskih kartica i primene QR kodova u okviru bržeg prepoznavanja učenika, kreiranje baze podataka sa svim knjigama koje biblioteka ima i njihovo trenutno stanje – izdate, na policama odnosno vraćene ili još uvek nisu vraćene uprkos isteku roka. Sistem je inovativan, spaja tradicionalno sa modernim, i što je najvažnije veoma je praktičan i jednostavan za korišćenje, takođe se lako implementira u okviru bilo koje biblioteke.

II. Princip rada i praktična realizacija

Sistem je krajnje jednostavan i za početak rada neophodno je samo popuniti bazu podataka knjigama koje biblioteka poseduje, to je moguće obaviti pomoću alata koji su implementirani u sistem.

Sistem automatski prepoznaje sve knjige u bazi podataka i prikazuje ih u tri kategorije: izdate, na policama odnosno vraćene ili još uvek nisu vraćene uprkos isteku roka. Moguće je dodavanje novih knjiga i nakon pokretanja sistema (slika 1.) kao i uklanjanje knjiga koje su već unete u bazu podataka. Knjige koje su dodate u bazu dele se na umetnička dela, magazine i novine.

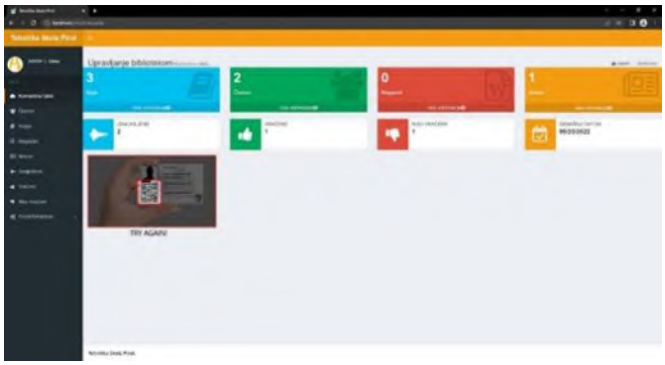
Za svaku knjigu koju dodajete potrebno je upisati ISBN, naziv knjige književnu vrstu, ime autora, količinu, datum dodavanja, ediciju, nabavnu cenu, broj strana knjige i izdavača, to je prikazano na slici 2.

Još jedan od alata je brza pretraga koja omogućava brzo pronalaženje bilo kog podatka iz baze, sve podatke bilo da su rezultat pretrage ili se radi o svim podacima moguće je filtrirati po bilo kom parametru.

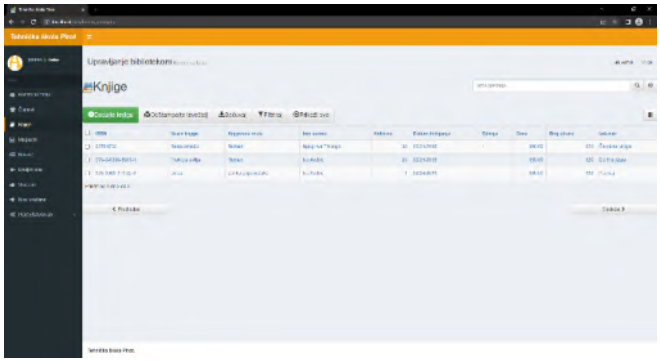
M. Stojanović, A. Panić, A Jovanović su učenici Tehničke škole Pirot, Takovska 22, 18300 Pirot, Srbija.

B. Ćirić je profesor u Tehničkoj školi Pirot, Takovska 22, 18300 Pirot, Srbija, E-mail: bojan.ciric@tsp.edu.rs.

Zatim sledi dodavanje članova biblioteke u bazu podataka, za to je neophodna veb kamera koja ujedno služi za slikanje i očitavanje QR kodova, osim slike člana neophodni su i osnovni podaci kao što su ime, prezime, pol, broj članske karte… Moguća je modifikacija neophodnih podataka za člansku kartu. Nakon toga sistem automatski generiše grafički izgled članske karte a osnovni dizajn pravi administrator. QR kod koji je implementiran u svaku člansku kartu služi za brzo prepoznavanje svakog člana biblioteke jer sadrži identifikacioni broj koji je upisan u bazu podataka. Postoji mogućnost posete biblioteci kao gost ali nije moguće iznajmiti ili vratiti knjigu.



Slika 1. Početna stranica aplikacije.

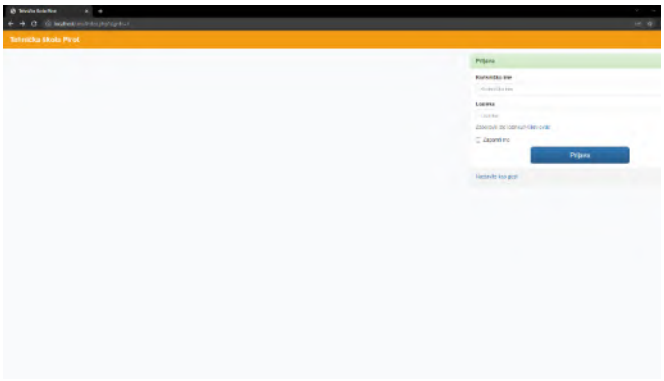


Slika 2. Kontrolna stranica za rad sa knjigama.

Grafički intefejs omogućava laku komunikaciju i manipulaciju bazom podataka. Automatski upravlja knjigama u bazi i njihovim stanjem, takođe automatski obaveštava da li su knjige vraćene na vreme ili je reč o kašnjenju. Moguće je takođe sačuvati fajl sa svim podacima iz baze i ubaciti ga u bilo koji drugi interfejs ovog tipa, ovo omogućava lako čuvanje podataka, pravljenje rezervne kopije ili prebacivanje podataka iz jedne biblioteke u drugu.

Sistem za prijavu razlikuje dva tipa korisnika, člana biblioteke i admina. Prijava se vrši preko korisničkog imena i lozinke. Ova dva parametra se za članove definišu prilikom

registracije i mogu se izmeniti kasnije, za razliku od parametara za admina koji se definišu direkno u bazi podataka, odnosno ne postoji grafički interfejs za registraciju admina, slika 3.



Slika 3. Interfejs za prijavu korisnika i admina.

III. MOGUĆE PRIMENE SISTEMA

Ovaj sistem je lako primenjiv u sve vrste biblioteka, bilo da su one školske, gradske ili univerzitetske. Takođe sistem je veoma fleksibilan pa uz male izmene može se primenjivati i u administracijama i arhivima za rad sa fajlovima i dokumentima. Pogodan ja i za državne organe kao što su policija i sud jer uz manje promene u tipu baza podataka koje sistem podržava on može raditi sa dokumentima o slučajevima i suđenjima koji se nalaze u arhivima.

IV. MOGUĆE NADOGRADNJE SISTEMA

Sistem se može nadograditi upotrebom *WebAR* tehnologije tako što će u sklopu QR koda na članskim karticama biti implementiran *WebAR* hologram koji može prikazivati biblioteku za koju je članska karta namenjena, kontakt u slučaju da se kartica izgubi i hiljadu drugih mogućnosti. Ova nadogradnja omogućava dodatnu funkcionalnost članske kartice kao i slobodu izbora svakoj biblioteci da ovu mogućnost iskoristi na svoj način, prikazano na slici 4.



Slika 4. Model rada sa *WebAR*-om unutar QR koda.

V. ZAKLJUČAK

Sistem modernizacije biblioteka je relativno jeftin, brz za izradu, praktičan i ima široku primenu, a takođe je zbog svoje fleksibilnosti podložan promenama i adaptaciji na sve vrste organizacija sličnim bibliotekama. Sve u svemu ovaj sistem je moderan i inovativan, lako podnosi promene i otvoren je za mnoga unapređenja što ga čini široko primenjivim.

LITERATURA

[1] PHP. Dostupno na: https://www.w3schools.com/php/
[2] MySql. Dostupno na: https://www.mysqltutorial.org/

Analiza metoda genetskog algoritma u izvođenju formule funkcije sa zadatog grafika

Dušan Korizma

Sadržaj – U ovom radu su istraženi metodi genetskog algoritma na problemu izvođenja formule funkcije pomoću grafika. Cilj algoritma je da se funkcija poklapa sa grafikom van onog grafika koji je dat. Gledana je uspešnost algoritma uz date mutacije, ukrštanje i fitnes funkcije. Algoritam određuje formule prostih funkcija, dubine 2 ili manje, sa minimalnim greškama na tačkama van zadatog grafika. Malo je manje uspešniji na komplikovanijim funkcijama, dubine 3 ili više, gde se dešava da ne uspeva da dovoljno približi funkciju zadatom grafiku. Problem se dešava zbog previše blagih mutacija ili zbog manjak resursa da se izvrši algoritam sa velikim brojem jedinki. Zaključak jeste da algoritam odlično izvodi funkcije malih dubina dok dolazi do komplikacija na malo dubljim funkcijama.

I. UVOD

Genetski algoritam je heuristički algoritam koji traži tačno ili približno rešenje zadatog problema. On pripada klasi evolutivnih algoritama koji koriste mehanizme viđene u prirodi kao metode rada. Genetski algoritam koristi mutaciju, ukrštanje i prirodnu selekciju koja je viđena kod razmnožavanja ćelija [1]. Genetski algoritmi se koriste za rešavanje problema koje nemaju tačno definisano rešenje. Korišćeni su za rešavanje optimizacionih problema i rešavanje problema pretrage objekata.

Genetski algoritam (*Genetic algorithm* ili GA) počinje sa nasumično generisanom populacijom koja je sačinjena od velikog broja jedinki. Jedinka predstavlja jedno moguće rešenje zadatog problema. Svakoj jedinki se dodeljuje fitnes, vrednost koja predstavlja koliko je data jedinka ispunila zahtev zadatka. Kada se celoj populaciji, generaciji, izračuna fitnes oni su rangirani prema njihovim fitnesima. Sledeća generacija se pravi reprodukcijom najuspešnijih jedinki iz prošle generacije. Što jedinka ima bolji fitnes ona ima veću verovatnoću da se reprodukuje. Mutacijom i ukrštanjem tih izabranih jedinki čini da sledeća generacija ima veće fitnes vrednosti. Ovaj proces se vrši sve dok krajnji zahtev nije ispunjen [2]. Jedinke su često prikazane kao stringovi ili kao brojevi, u ovom algoritmu jedinke su prikazane preko binarnih stabla iskaza.

Binarna stabla iskaza su tip binarnih stabala koji se koriste u prikazivanju matematičkih iskaza. Binarna stabla su korišćena u genetskim algoritmima za dizajniranje logičkih kola [3]. Već postojeći radovi su istraživali metode mutacija i ukrštanja na binarnim stablima iskaza [4], zaključujemo da su oni odlična struktura podataka za dati problem.

Cilj ovog rada jeste da proveri svaku metodu genetskog algoritma, reprodukciju, selekciju, mutaciju i ukrštanje, i da ukaže na moguće greške i na dobre strane algoritma. Upoređivanjem rezultata različitih instanci nalazimo načine da unapredimo delove algoritma.

D. Korizma, Istraživačka stanica Petnica, 14104 Valjevo Srbija, E-mail: dulekorizma@gmail.com.

II. METODE

A. Struktura jedinki

Jedna jedinka predstavlja jednu funkciju. Funkcije su predstavljene binarnim stablima iskaza, gde listovi sadrže jednu od osnovnih funkcija: konstantna, promenljiva, logaritamska, eksponencijalna, n-ti koren i trigonometrijska funkcija. Unutrašnji čvorovi sadrže operaciju koja se izvršava između njene dece (+, -, *, /, o). Ovakvom strukturom moguće je moguće je opisati svaku kombinaciju osnovnih funkcija.

Vrednosti za y, vrednost funkcije, mogu biti realni brojevi ili nedefinisani. Realni brojevi se nalaze na intervalu $[-10^9, 10^9]$, ako se vrednost funkcije ne nalazi u tom intervalu njena vrednost se tretira kao nedefinisana.

B. Ulazni podaci

Ulazni podaci predstavljaju tačke sa grafika tražene funkcije. Cilj našeg GA jeste da pronađe funkciju kojoj taj grafik odgovara.

Grafik koji opisuje traženu funkciju predstavlja niz uređenih parova (x, y), gde je x parametar iz skupa realnih brojeva, a y odgovarajuća vrednost funkcije u tački x. Vrednost funkcije je realan broj, ili nedefinisan za slučaj kada parametar x ne pripada domenu funkcije. Tačke su za svaku funkciju izabrane na isti način: uzete su tačke na intervalu [-200, 200] sa korakom od 0.5 (-200, -199.5, ….., 199.5, 200) .

C. Kreacija populacije

Populacija je kreirana pomoću algoritma *pola pola* [5]. Polovina populacije je generisana koristeći *ful* algoritam [5], koji generiše potpuno drvo do zadate dubine, tako da svaki list i čvor sadrže nasumično izabrane funkcije i operacije. Druga polovina je generisana *rast* algoritmom [5], koji nasumično odlučuje da li nastavlja da generiše čvorove. Svaki put kada se generiše novi čvor algoritam bira da li je

sledeći član čvor ili list. Verovatnoća da sledeći član bude list jeste 50%. Tako generisana drveća ne prelaze maksimalno definisanu dubinu [6].

Pokazano je da *grow* neće generisati drveća uniformno, već da mala drveća i beskonačno velika drveća imaju najveću verovatnoću da se generisu [7]. GA radi sa veoma malim dubinama (maksimalno 5) tako da na dubinsku raznovrsnost početne populacije nece imati veliki uticaj mana *grow* algoritma. Algoritam *full* ispravlja bilo koje nepravilnosti *grow* algoritma zato što se dubine jedinki generisanih pomoću *full* algoritma mogu kontrolisati.

Nova populacija je generisana iz dva dela. Prvi deo populacije su novo-generisana nasumična stabla, oni čine 10% celokupne populacije. Ostatak populacije se dobija mutacijom i ukrštanjem jedinki iz prošle generacije. Biranje roditelja za nove jedinke se vrši tako što se svakoj jedinki dodeljuje verovatnoća da se izabere kao roditelj po formuli:

$$Verovatnoća(k) = (1-p)k - 1 * p \quad (1)$$

gde je *k* rang jedinke po fitness, a *p* parametar verovatnoće koji iznosi 0.5.

Svaki odabran roditelj ima veovatnoću da se ukrsti sa drugim roditeljem i verovatnoću da mutira po formulama:

$$Verovatnoća\ ukrštanje = 0 + sd\ /\ 50 * 0.2 \quad (2)$$

$$Verovatnoća\ mutacija = 1 - sd\ /\ 50 * 0.07 \quad (3)$$

gde *sd* predstavlja standardnu devijaciju fitnesa svih jedinki iz prethodne generacije.

D. Fitnes funkcija

Vrednost fitnes funkcije je obrnuto srazmerna sa približnošću funkcije, npr. što je vrednost fitnesa manja to je jedinka bolja.

Iz ulaznih podataka se uzimaju x vrednosti i određuju y vrednosti jedinke J (jedinka za koju se računa fitnes). Ako označimo y0 kao vrednosti tražene funkcije, a y1 kao vrednosti jedinke J onda za svaki par tih vrednosti važi:
1. Ako su y0 i y1 definisani brojevi F odgovara formuli:

$$F = min(|y0 - y1|, max_odstupanje) \quad (4)$$

2. Ako su y0 i y1 nedefinisani brojevi F odgovara formuli:

$$F = 0 \quad (5)$$

3. Ako je jedan od y0, y1 nedefinisan broj F odgovara formuli:

$$F = max_odstupanje \quad (6)$$

gde *max_odstupanje* predstavlja najveće moguće odstupanje tačke od grafika.

Odstupanje tačke od grafika je ograničeno kako bi smo mogli da brojčano predstavimo nedefinisane vrednosti, i kako bismo ograničili fitnes funkciju.

Za svaki par (x, y) određen je F pa je fitnes određen po formuli:

$$Fitnes(J) = \sqrt{\frac{\sum_1^k F_i^2}{k}} \quad (7)$$

E. Ukrštanje

Ukrštanje dva stabla kreira dva nova stabla tako da ona imaju istu ili manju dubinu od svojih roditelja. Time što dubina stabla ne prelazi dubinu roditelja povećavamo efektivnost GA [6]. Algoritam za ukrštanje funkcioniše tako što izabere po nasumično podstablo od oba roditelja i zameni ih. Time se generišu dve nove jedinke, dok roditelji ostaju u opticaju za sledeća ukrštanja.

F. Mutacija

Mutacija parametra

Svaki parametar funkcije se uveća za određenu vrednost, u stvari svaki list jedinke se promeni. To uvećanje je izabrano preko normalne raspodele brojeva gde je $\mu = 4$, a $\sigma = 3$. Trigonometrijske funkcije imaju drugačiji tip parametara ("*sin*" je parametar) pa njih mutiramo na drugačiji način. Datu trigonometrijsku funkciju promenimo u drugu nasumičnu trigonometrijsku funkciju.

Mutacija operacija

Za svaku operaciju postoji verovatnoća da se promeni u drugu nasumično izabranu operaciju.

Mutacija podstabla

Nasumično se izabere čvor iz stabla i ispod njega se generiše novo podstablo tako da novodobijeno stablo ne prelazi maksimalnu dubinu. Maksimalna dubina je konstanta koja je unapred definisana.

Mutacija podstabla kao nove jedinke

Nasumicno izabrano podstablo se uzima kao nova jedinka.

G. Biranje nasumičnih čvorova u stablu

Tokom ukrštanja i mutacija nasumično biramo čvorove u datim stablima. Algoritam prolazi kroz stablo nasumičnim brojem koraka, na svakom čvoru postoji verovatnoća da sledeći izabran čvor bude levo dete i verovatnoća da sledeci izabran čvor desno dete. Formule za izračunavanje verovatnoći se nalaze u formulama:

$$Verovatnoća\ levo = \frac{broj\ čvorova\ ispod\ levog\ deteta}{broj\ čvorova\ ispod\ datog\ čvora} \quad (8)$$

$$Verovatnoća\ desno = \frac{broj\ čvorova\ ispod\ desnog\ deteta}{broj\ čvorova\ ispod\ datog\ čvora} \quad (9)$$

III. REZULTATI

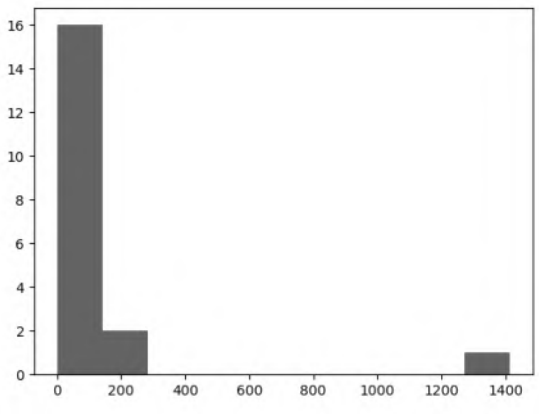
A. Fitnes funkcija

Fitnes funkcija proverava koliko je dobijena funkcija blizu grafika samo na jednom odsečenom delu. Računanjem fitnesa funkcije van tog dela grafika dobijamo uspešnost našeg algoritma. Vidimo koliko je naš algoritam pogodio funkciju izvan datih podataka. Za fitness van grafika posmatrane su tačke [-300, -200) ∪ (200,300] sa korakom od 0.1, posmatrane tačke su: -300, -299.9, -299.8…. 299.8, 299.9, 300.

Uspeh funkcije možemo da povežemo sa brojem definisanih tačaka na grafiku uz pomoć koga algoritam izračunava fitnes, gledamo procenat definisanih tačaka na datom grafiku.

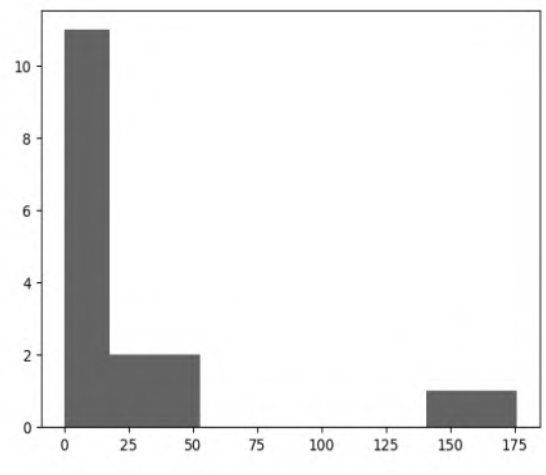
B. Fitnes van grafika

Prosečan fitnes van grafika (FVG) na 20 različitih instanci GA je *87.145*, dok je na grafiku (FNG) bio *76.236*. Ako bismo isključili najmanju i najveću fitnes vrednost, prosek FVG bi bio *18.261*, dok je FNG prosek bio *28.648*.



Slika 1. Fitnes van grafika svih instance predstavljen preko histograma.

U 3 instance je FVG bio 0 dok FNG je bio znatno veci (na prvom *223.735*, na drugom *244.758*, na trećem *0.318*). FVG je bio nula zato što data funkcija nije bila uopšte definisana na tim tačkama i time je dobila perfektan fitnes rezultat. Ako bi smo posmatrali prosečni FVG bez njih dobija se *21.913*, a FNG je *3.136*. Time razlika između proseka FVG i FNG jeste *18.777*. Ta razlika predstavlja apsolutnu grešku fitnesa algoritama van zadatog grafika.



Slika 2. Fitnes van grafika bez najmanje i najveće instance preko histograma.

C. Nedefinisane tačke

U svakoj instanci broj nedefinisanih tačaka prve jedinke i tražene funkcije je bio isti. Time zaključujemo da alogritam 100 % pogađa domen tražene funkcije, sledi da je metoda ograničavanja odstupanja od grafika dobro 'kažnjavala' nedefinisane vrednosti. Funkcije koje su na zadatom grafiku imale preko 50 % definisanih tačaka su imale prosečni FVG od 7.101, dok funkcije sa manje od 50 % definisanih tačaka su imale prosečni FVG od 179.409. Zaključujemo da grafik treba imati barem 50 % definisanih tačaka.

D. Proste funkcije

Algoritam veoma lako pronađe oblik prostijih funkcija, funkcije sa dubinom 0, 1 i 2. Kada tražena funkcija sadrži parametarske funkcije, mutacija parametara pomaže pri aproksimaciji parametara tražene funkcije, što znači da direknto smanjuje fitnes tražene funkcije.

Za funkcije koje nemaju parametre, npr. trigonometrijske funkcije i promenljive, kada algoritam nađe oblik našao je i samu funkciju. Sve instance gde su funkcije bile dubine 2 ili manje su imale manji FVG od 5. Time zaključujemo da algoritam ima veoma dobru preciznost na prostim funkcijama.

E. Komplikovane funkcije

Algoritam je tražio funkcije dubine 3, 4 ili 5 u sedam različitih instanci. U tri od sedam pomenutih instanci algoritam nije uspeo naći približnu formulu funkcije, već je fitnes najboljih jedinki bio veći od 200. U preostale četiri instance fitnes nije prelazio broj 20. Time zaključujemo da je algoritam veoma loše pronalazio dublje funkcije.

IV. ZAKLJUČAK

Algoritam odlično izvodi formule funkcija male dubine, dok ima problema pri izvođenju formule funkcija veće dubine. Razlog lošeg uspeha jeste loša selekcija tačaka za grafik, koji čini ulazne podatke. Mutacije nisu dovoljno značajne da bi pomogle tim dubljim funkcijama.

LITERATURA

[1] Mitchell, Melanie. "An introduction to genetic algorithms mit press." Cambridge, Massachusetts. London, England 1996 (1996).

[2] Kumar, Manoj, Dr Husain, Naveen Upreti, and Deepti Gupta. "Genetic algorithm: Review and application." *Available at SSRN 3529843* (2010).

[3] Coello, Carlos A., Alan D. Christiansen, and Arturo Hernández Aguirre. "Using genetic algorithms to design combinational logic circuits." *Intelligent Engineering through Artificial Neural Networks* 6, no. 0 (1996): 391-396.

[4] Poli, Riccardo, and William B. Langdon. "Schema theory for genetic programming with one-point crossover and point mutation." *Evolutionary Computation* 6, no. 3 (1998): 231-252.

[5] Koza, John R. "Evolution of subsumption using genetic programming." In *Proceedings of the first European conference on artificial life*, pp. 110-119. Cambridge, MA, USA: MIT Press, 1992.

[6] Whigham, P. A., and G. Dick. "GP and Bloat: Absorbing boundaries and spatial structures." In *Proceedings of the Third Asian-Pacific workshop on Genetic Programming*, pp. 1-12.

[7] Luke, Sean. "Two fast tree-creation algorithms for genetic programming." *IEEE Transactions on Evolutionary Computation* 4, no. 3 (2000): 274-283.

Scikit-learn biblioteka i metode mašinskog učenja za obradu neuroslika

Miljan G. Jeremić

Sadržaj − Za analizu podataka neuroslika se sve više koriste metode statističkog mašinskog učenja. Glavna vrlina je njihova sposobnost da modeliraju visokodimenzionalne skupove podataka, na primer, multivarijantnu analizu aktivacionih slika ili vremenske serije stanja mirovanja. Učenje pod nadzorom se obično koristi u postavkama dekodiranja ili kodiranja kako bi se slike mozga povezale sa ponašanjem ili kliničkim zapažanjima, dok učenje bez nadzora može otkriti skrivene strukture u skupovima slika (npr. MRI u stanju mirovanja) ili pronađu podpopulacije u velikim kohortama. Ako se uzmu u obzir različite funkcionalne aplikacije koje rade sa neuroslikama, ilustruje se kako *scikit-learn*, *Python* biblioteka za mašinsko učenje, može koristiti za obavljanje nekih ključnih koraka analize. *Scikit-learn* sadrži veliki skup statističkih algoritama za učenje, kako nadziranih tako i bez nadzora.

I. Uvod

Interesovanje za primenu statističkog mašinskog učenja za analizu podataka primenom neuroslika je dugi niz godina u sve većem porastu [1]. Cilj ovog rada je da prikaže vezu između mašinskog učenja i obradu neuroslika demonstracijom kako alat za mašinsko učenje opšte namene, *scikit-learn*, može da obezbedi najsavremenije metode za analizu neuroslika dok je programski kod jednostavan i razumljiv. Ovde je fokus na softveru; za konceptualniji uvod u metode mašinskog učenja u fMRI analizi (Pereira i dr., 2009) ili (Mur et al., 2009), dok (Hasti et al., 2019) pruža dobru referencu o mašinskom učenju [2]-[4]. Korišćenje *scikit-learn* bibloteke sa alatima je referentno mašinsko učenje, i ima još i niz algoritama koji odgovaraju određenim paketima, ali i zato što je implementiran u *Python*-u, a samim tim se lepo uklapa u *Python*-ov ekosistem obrade neuroslika. Ovaj rad istražuje nekoliko primena statističkog učenja za rešavanje uobičajenih potreba neuroslika, detaljno navodeći odgovarajući kod, izbor metoda i osnovne pretpostavke. Ne govori se samo o rezultatima predviđanja, već i interpretabilnost rezultata, što navodi da se istraži interni model raznih metoda. U ovom radu cilj nije predstavljanje specifičnih biblioteka neuroslika, već i delovi koda koji se odnose na *scikit-learn*. *Nilearn* biblioteka [18], kao softverski paket, nastoji da pojednostavi upotrebu *scikit-learn* za obradu neuroslika. Rad je organizovan na sledeći način. Nakon uvođenja alata za *scikit-learn* biblioteku, pokazuje se kako pripremiti podatke za primenu *scikit-learn* rutine.

M. G. Jeremić je profesor Knjaževačke gimnazije, Karađorđeva 16, 19350 Knjaževac, Srbija, E-mail: miljan.jeremic@gmail.com.

II. Korišćeni alati: Scikit-Learn i Python ekosistem

A. Osnovni PYTHON alati za obradu neuroslika

Python je jedan od boljih alata za analizu podataka neuroslika pomoću alata [5]-[6] kao što su Nipy ili Nipype. Naučne *Python* biblioteke koje se koriste u ovom radu su:

• NumPy: većina naučnih *Python* biblioteka, uključujući *scikit-learn*, koristite NumPy nizove kao ulazne i izlazne podatke;

• SciPy: ona je povezana sa kompajliranim bibliotekama koje obezbeđuju visoke performanse (BLAS, Arpack i MKL za linearnu algebru i matematičke operacije). NumPy i SciPy zajedno pružaju robusno naučno okruženje za numeričko računanje i koriste se u svim algoritmima.

• Matplotlib: biblioteka za crtanje u *Python*- u. Nudi kvalitet objavljivanja slika u različitim formatima i koristi se za generisanje slika u radu [7].

• Nibabel: biblioteka za pristup podacima u formatima datoteka neuroslika. Koristi se na početku svih skupova podataka.

B. Scikit-learn i ekosistem mašinskog učenja

Scikit-learn je biblioteka opšte namene mašinskog učenja napisana u *Python*-u [8]. Obezbeđuje efikasne implementacije najsavremenijih algoritama, dostupnih za stručnjake za ne-mašinsko učenje i može se ponovo koristiti u naučnim disciplinama i drugim oblastima primene. Takođe se koriste prednosti *Python*-ove interaktivnosti i modularnosti za brzu i laku izradu prototipa. Postoji i niz drugih paketa za učenje. Na primer, u *Python*-u, PyBrain je najbolji u neuronskim mrežama i pristupu učenja sa potkrepljivanjem, ali su njegovi modeli ne odgovarajući za potrebe da se tumače rezultati [9]. Izvan *Python*-a, program *Weka* je bogato okruženje napisano u *Java* programskom jeziku, međutim, više je orijentisano na pretragu podataka [10]. Neki okviri višeg nivoa obezbeđuju kompletan sistem prenosa za primenu tehnike mašinskog učenja do neuroslika. PyMVPA je Python paket koji priprema podatke, za učitavanje i analizu, kao i vizuelizacija rezultata [11]. Izvodi multivarijantnih analiza obrazaca i može da koristi spoljne alate kao što su R, *Scikit-learn* ili *Shogun* [12]. PRONTo je napisan u *MATLAB*-u i može lako da se povezuje sa SPM-om, ali ne predlaže mnogo algoritama mašinskog učenja [12]. Ovde, umesto potpune obrade analize neuroslika sistemom prenosa, raspravlja se o obrascima nižeg nivoa

koji se kvare kako se podaci neuroslika unose u *scikit-learn* i kasnije kroz program obrađuju.

C. Scikit-learn koncept

U *scikit-learn* biblioteci, svi objekti i algoritmi prihvataju ulazne podatke u oblik 2-dimenzionalnih nizova veličine uzoraka × karakteristika uzorka. Ova konvencija čini ga generičkim i nezavisnim od domena. *Scikit-learn* objekti dele jedinstven skup metoda koji zavisi od njihovih svrha: procenitelji mogu da uklope modele iz podataka, prediktori mogu da naprave predviđanja o novim podacima i transformatori pretvaraju podatke iz jednog načina predstavljanja u drugi.

• Procenjivač- Interfejs ili sprega za procenjivanje, jezgro biblioteke, izlaže odgovarajući metod za učenje parametara modela iz skupa podataka o obuci (*train* podaci). Svi algoritmi učenja pod nadzorom i bez nadzora (npr. za klasifikaciju, regresiju ili grupisanje) su dostupni kao objekti koji implementiraju ovaj interfejs.

• Prediktor- Prediktor je procenjivač sa metodom predviđanja koji uzima ulazni niz X_test i pravi predviđanja za svaki uzorak u njemu. Ovaj ulazni parametar označavamo "X_test" kako bi se naglasilo da se predviđanje generalizuje na nove podatke. U slučaju procenjivača nadgledanog učenja, ovaj metod je tipično vraća predviđene oznake ili vrednosti izračunate iz procenjenog modela.

• Transformator- Uobičajeno je menjati ili filtrirati podatke pre nego što se ubace u algoritam za učenje, ili neke procene, imenovani transformatori implementiraju metod transformacije. Prethodnom obradom, kao i izborom karakteristika i smanjenja dimenzionalnosti, svi algoritmi su obezbeđeni kao transformatori unutar biblioteka. Ako se transformacija može obrnuti, onda postoji i metod zvani inverzna transformacija, koji takođe postoji. Prilikom testiranja estimatora ili postavljanja hiperparametara, potrebna je pouzdana metrika za procenu njegovog učinka. Korišćenje iste veličine podataka za obuku i testiranje nije prihvatljivo jer dovodi do preterano samouverene performanse modela, što je takođe poznat fenomen za prenamenu. Unakrsna validacija je tehnika koja omogućava da se pouzdano proceni procenjivač na datom skupu podataka. Sastoji se u iterativno prilagođavanje procenitelja na delu podataka, tzv. skup za obuku i njegovo testiranje na izostavljenim nevidljivim podacima, koji se nazivaju test podaci. Postoji nekoliko strategija za particionisanje podataka. Na primer, K-fold unakrsna validacija se sastoji u podeli (nasumično ili ne). uzoraka u k-podskupova: svaki podskup se zatim koristi jednom kao skup za testiranje dok se ostali k − 1 podskupovi koriste za obuku procenitelja. Ovo je jedna od najjednostavnijih i najčešće korišćenih unakrsnih validacija strategije. Parametar k se obično postavlja na 5 ili 10. Druga strategija, koja se ponekad naziva unakrsna validacija *Monte-Carlo*, koristi mnogo nasumičnih particija u podacima. Za dati model i neku fiksnu vrednost hiperparametara, rezultati na različitim skupovima testova mogu se usredsrediti da bi se dobio kvantitativni rezultat da proceni koliko je model dobar. Maksimiziranje ovog skora unakrsne validacije nudi principijalan način za postavljanje hiperparametara i omogućava izbor između različitih modela. Ovaj postupak je poznat kao izbor modela. U *scikit-learn* bibloteci, podešavanje hiperparametara se može lako obaviti pomoću *GridSearchCV* procene. Uzima se kao ulaz procenjivač i skup kandidata hiperparametara. Zatim se izračunavaju rezultati unakrsne validacije, paralelno sve kombinacije hiperparametara, po redu da nađenog najboljeg.

III. PRIPREMA PODATAKA: OD MR SKUPA PODATAKA DO PODATAKA MATRIX

Pre primene statističkog učenja na podatke neuroslika, standardno mora se primeniti prethodna obrada. Za fMRI, ovo uključuje korekcija kretanja, korekcija vremena preseka i normalizaciju na zajednički šablon kao i MNI (*Montrealski neurološki institut*) ako je potrebno. Referentni softveri za ove zadatke [13]-[14] su SPM i FSL. Dostupan je *Python* interfejs za ove alate u Nipype *Python* biblioteci [15]. Biće reči o oblikovanju prethodno obrađenih podataka u format koji se može uneti u *scikit-learn* bibiloteku. Za podešavanja mašinskog učenja potrebni su nam podaci u obliku matrica, koju ćemo označiti X, i opciono ciljnu promenljivu predvideti kao y.

A. Prostorno ponovno uzorkovanje

Podaci o neuroslikama često dolaze kao Nifti datoteke, 4-dimenzionalni podaci (3D skeniranja sa vremenskim serijama na svakoj lokaciji ili vokselu) zajedno sa matricom transformacije (nazvana afina) koja se koristi za izračunavanje lokacija voksela od indeksa niza do globalnih koordinata [1]. Pri radu sa više subjekata, svaki pojedinačni podatak se registruje na zajedničkom šablon (MNI, Talairah...), dakle na zajedničkoj afini, tokom prethodnih obrada. Afina matrica može izraziti anizotropiju podataka, kada je rastojanje između dva voksela nije isto u zavisnosti od pravca. Ovu informaciju koriste algoritmi koji se oslanjaju na prostorno strukturu podataka, na primer reflektor. SciPy rutina *scipy.ndimage.affine_transform* može se koristiti za obavljanje ponovnog uzorkovanja slike: menjanje prostorne rezolucije podataka. Ovo je interpolacija i menja podatke, zato ga treba pažljivo koristiti. Smanjenje uzorkovanja koje je uobičajeno koristi se za smanjenje veličine podataka za obradu. Tipične veličine su 2 ili rezolucijom od 3 mm, ali se prostorna rezolucija skeniranja povećava sa napredkom u fizici elementarnih čestica (MR fizici). Afina matrica može kodirati kaliranje faktora za svaki pravac.

B. Čišćenje signala

Zbog svog složenog i indirektnog procesa akvizicije, podaci neuroslika često imaju nizak odnos signal-šum. Oni sadrže trendove i artefakte koji se moraju ukloniti kako bi se

osigurala maksimalna efikasnost algoritama mašinskog učenja [1].

Čišćenje signala uključuje:

• Uklanjanje trenda uklanja linearni trend tokom vremenske serije svakog od njih kao voksela. Ovo je koristan korak kada se proučavaju fMRI podaci, kada intenzitet voksela sam po sebi nema nikakvog značaja i potrebno ga je proučiti varijacijom i korelacijom sa drugim vokselima. Detrendiranje može biti urađeno zahvaljujući SciPy (*scipy.signal.detrend*).

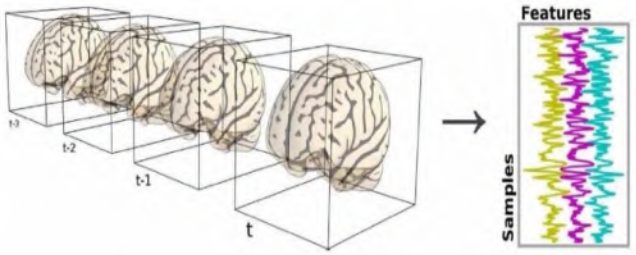
• Normalizacija se sastoji u postavljanju varijanse vremenske serije na 1. Ovo usklađivanje je neophodno za neko mašinsko učenje, jer su algoritmi osetljivi na različite opsege vrednosti.

• Frekventno filtriranje se sastoji u uklanjanju visoke ili niske frekvencije signala. Niskofrekventni signali u fMRI podacima su uzrokovani fiziološkim mehanizmima ili pomeranja skenera. Može se izvršiti filtriranje zahvaljujući Furijeovoj transformaciji (*scipy.fftpack.fft*) ili korišćenjem Buttervortovog filtra (*scipy.signal.butter*).

C. Od 4-dimenzionalnih slika do 2-dimenzionalnog niza: maskiranje

Podaci obrade neuroslika su predstavljeni u 4 dimenzije: 3 prostorne dimenzije i jednu dimenziju za indeksiranje vremena ili ispitivanja [1]. *Scikit-learn* algoritmi, s druge strane, prihvataju samo 2-dimenzionalne uzorke × matrice karakteristika (videti odeljak II.3). U zavisnosti od postavka, vokseli i vremenske serije mogu se smatrati osobinama ili uzorcima. Na primer, u analizi nezavisnih prostornih komponenti (ICA), vokseli su uzorci.

Proces redukcije sa 4D slika na vektore obeležja dolazi sa gubitkom prostorne strukture (slika 1).



Slika 1. Konverzija skeniranja mozga u dvodimenzionalne podatke.

To međutim omogućava odbacivanje neinformativnih voksela, kao što su oni van mozga. Takvi vokseli koji nose samo šum artefakte skenera bi smanjio SNR i uticao na kvalitet procene.

Odabrani vokseli formiraju moždanu masku. Takva maska se često daje zajedno sa skupovima podataka ili se može izračunati pomoću alata kao što su FSL ili SPM.

Primena maske je olakšana naprednim indeksiranjem NumPy koristeći logičke nizove. 2D maskirani podaci biće nazivani X da bi se pratile konvencije *scikit-learn*:

```
mask = nibabel.load('mask.nii').get_data()
func_data = nibabel.load('epi.nii').
get_data()
# Ensure that the mask is boolean
mask = mask.astype(bool)
# Apply the mask, X = timeseries * voxels
X = func_data[mask].T
# Unmask data
unmasked_data = numpy.zeros(mask.shape,
dtype=X.dtype)
unmasked_data[mask] = X
```

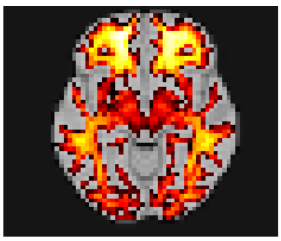
D. Vizualizacija podataka

U svim primerima, vokseli od interesa su predstavljeni kao aksijalni deo mozga. Neke transformacije originala matričnih podataka su potrebni da odgovaraju formatu podataka za *matplotlib* biblioteku. Sledeći deo pokazuje kako se učitava i prikazuje aksijalni presek prekriven aktivacionom mapom [1]. Pozadina je anatomsko skeniranje i njegovi najviši vokseli se koriste kao sintetički aktivacije.

```
# Load image
bg_img = nibabel.load('bg.nii.gz')
bg = bg_img.get_data()
# Keep values over 6000 as artificial
activation map
act = bg.copy()
act[act < 6000] = 0.
# Display the background
plt.imshow(bg[..., 10].T,
origin='lower',
interpolation='nearest', cmap='gray')
# Mask background values of activation
map
masked_act = np.ma.masked_equal(act,
0.)
plt.imshow(masked_act[..., 10].T,
origin=
'lower', interpolation='nearest',
cmap='hot')
# Cosmetics: disable axis
plt.axis('off')
plt.show()
```

Nakon izvršetka dela programskog koda dobija se slika 2.

Treba imati na umu da je pozadina potrebna za prikaz delimičnih mapa. Preklapanje dve slike može da se uradi zahvaljujući naredbi *numpy.ma.masked_array*. Postoji nekoliko opcija za poboljšanje ukupne aspekt fabule. *Nipy* paket pruža funkciju *plot_map* koja je podešena da prikazuje aktivaciju mape (pozadina je čak obezbeđena ako je potrebno). Retka inverzna kovarijansa može izdvojiti funkcionalnu strukturu interakcije iz vremenske serije fMRI koristeći procenu *graF-Lasso* [16].



Slika 2. Prikaz voksela u okviru mozga.

Za ovo istraživanje korišćeno je programsko okruženje u *Python*-u zvano *Anaconda*, kao i skup neuro slika koji je pribavljen sa zvaničnog sajta [17].

IV. REZULTATI

Ovi rezultati odgovaraju neuronaučnom znanju kako bi se istakli regioni visokog nivoa ventralnog vizuelnog korteksa koji je poznato da sadrži vizuelne oblasti specifične za kategoriju. Samo pet linija u programu su potrebni za pokretanje *scikit-learn* klasifikatora. Kako svi linearni modeli dele isti interfejs, zamenjujući *Support Vector Classification* drugim linearnim modelom, npr. kao *ElasticNet* ili *LogisticRegression*, zahteva promenu samo jedne linije programa.

V. ZAKLJUČAK

U ovom radu jednostavnim primerima ilustrovano kako tehnike mašinskog učenja mogu se primeniti na fMRI podatke koristeći *scikit-learn Python* alate kako bi predstavili neuronaučnim problemima. Kodiranje i dekodiranje se može osloniti na nadgledano učenje da poveže slike mozga sa stimulansima. Učenjem bez nadzora može da se izdvaja struktura kao što su funkcionalne mreže ili regioni mozga iz podataka o stanju mirovanja. Prateći *Python* kod sa zadacima mašinskog učenja su jednostavni. Poteškoće leže u primeni pravilne predobrade podataka, odabir pravog modela za probleme tumačenje rezultata. Rešavanje ovih poteškoća dok naučnicima pruža jednostavan i čitljiv kod zahteva izgradnju biblioteke specifične za domen, posvećene primeni *scikit-learn* biblioteke do podataka o neuroslikama. *Nilearn* ima za cilj da olakša korišćenje *scikit-learn* biblioteke gde se uči o podacima neuroslika.

Primeri obrađeni u ovom radu samo zagrebu površinu primene statističkog učenja na neuroslika. Moderna analiza podataka obrade neuroislika podrazumeva uklapanje modela na ograničene količine podataka. Ovo su problemi visokodimenzionalne statistike koje zahtevaju tehnike statističkog učenja.

ZAHVALNICA

Autor se zahvaljuje 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] Alexandre Abraham, Fabian Pedregos, Michael Eickenberg, Philippe Gervais, “Machine learning for neuroimaging with scikit-learn”

[2] Pereira, F., Mitchell, T., and Botvinick, M. (2009). “Machine learning classifiers and fMRI: a tutorial overview. Neuroimage 45”, S199–S209.

[3] Mur, M., Bandettini, P. A., and Kriegeskorte, N. (2009). “Revealing representational content with pattern-information fMRI–an introductory guide” Soc. Cogn. Affect. Neurosci. 4, 101–109.

[4] Hastie, T., Tibshirani, R., and Friedman, J. J. H. (2001). “The Elements of Statistical Learning”, Vol. 1. NY: Springer.

[5] Millman, K. J., and Brett, M. (2007). “Analysis of functional magnetic resonance imaging in python”. *Comput. Sci. Eng*. 9, 52–55.

[6] Gorgolewski, K., Burns, et al. (2011). “Nipype: a flexible, lightweight and extensible neuroimaging data processing framework in python”. Front. Neuroinform. 5:13.

[7] Hunter, J. D. (2007). “Matplotlib: a 2d graphics environment”. *Comput. Sci. Eng*. 9, 90–95.

[8] Pedregosa, F., Varoquaux, G., Gramfort, A. Michel, V., Thirion, B., Grisel, O., et al. (2011). “Scikit-learn: machine learning in python”. J. Mach. Learn. Res. 12, 2825

[9] Schaul, T., Bayer, J., Wierstra, D., Sun, Y., Felder, M., et al. (2010). PyBrain. J. Mach. Learn. Res. 11, 743–746

[10] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). “The weka data mining software: an update”. ACM SIGKDD *Explor. Newsl*. 11, 10–18

[11] Hanke, M., Halchenko, Y. O., Sederberg, P. B., Hanson, S. J., Haxby, J. V., and Pollmann, S. (2009). “PyMVPA: a python toolbox for multivariate pattern analysis of fMRI data.” *Neuroinformatics* 7, 37–53.

[12] Sonnenburg, S., Rätsch, G. Henschel, S. Widmer, C., Behr, J., Zien, A., et al. (2010). “The SHOGUN machine learning toolbox”. J. Mach. Learn. Res. 11, 1799

[13] Friston, K. (2007). “Statistical Parametric Mapping: The Analysis of Functional Brain Images”. *London: Academic Press*.

[14] Smith, S. M., Jenkinson, M., Woolrich, M. W., Beckmann, C. F., Behrens, T. E., Johansen-Berg, H., et al. (2004). “Advances in functional and structural MR image analysis and implementation as FSL”. Neuroimage 23, S208–S219.

[15] Gorgolewski, K., Burns, C. D., Madison, C., Clark, D., Halchenko, Y. O., Waskom, M. L., et al. (2011). “Nipype: a flexible, lightweight and extensible neuroimaging data processing framework in python” Front. Neuroinform. 5:13.

[16] Varoquaux, G., and Craddock, R. C. (2013). “Learning and comparing functional connectomes across subjects” Neuroimage 80, 405

[17] Dostupo na: http://www.github.com/AlexandreAbraham/frontiers2013

[18] Dostupo na: http://nilearn. github. io

Sistem za dimovanje RGB diode

Strahinja Dimitrijević, Vukašin Nikolić, Nemanja Nedeljković

Sadržaj − Cilj ovog rada je bio da se projektuje sistem za dimovanje RGB (*Red, Green i Blue*) diode, tako da se očitava vrednost napona sa tri potenciometra i da se ta vrednost ispisuje na OLED (*Organic Light-Emitting Diode*) displeju. Detaljan opis korišćenih komponenata i demonstracija primene će biti detaljnije objašnjeni u nastavku rada.

I. UVOD

Jedan od prilično često zastupljenih sistema boja je poznat i kao RGB (*Red, Green i Blue*). U okviru ovog sistema se apsolutno svaki piksel određuje sa ukupno 3 informacije, a svaka od njih označava vrednost pomenute tri boje, koje su ujedno i osnovne. Kada je u pitanju RGB sistem boja, reč je o takozvanom aditivnom sistemu mešanja boja. Zapravo se u tom slučaju kombinje svetlost svake od ovih boja, a osnovna primena ovakvog načina mešanja boja je na prvom mestu vezana za finalizaciju određene fotografije ili drugog sadržaja, a kako bi on bio prikazan na televizijskom ekranu ili monitoru kompjutera, tableta ili nekog drugog uređaja. S obzirom na to da je potrebno ukombinovati navedene tri boje, a kako bi se dobile sve ostale, to je i sasvim jasno da se njihova koncentracija kreće u rasponu od 0 do 100 procenata, a prevedeno u računarski zapis to podrazumeva od 0 do 255, što je vrednost jednog bajta, tako da svaka komponenta RGB diode prati zapis x/y/z i spada u tom opsegu. Što se tiče označavanja, recimo bela boja se označava kao 255/255/255, dok je 0/0/0/ oznaka za crnu boju [5].

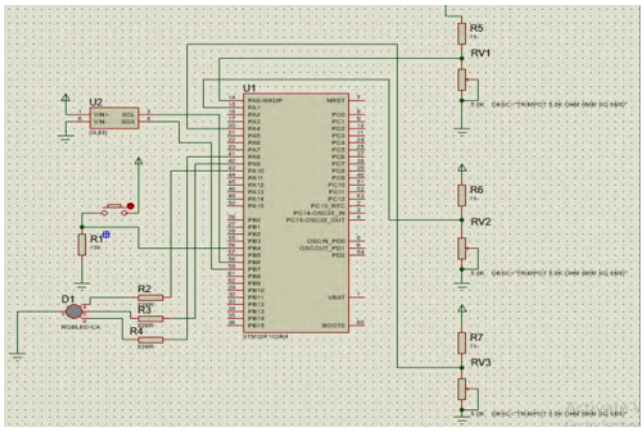
U zavisnosti od tri dobijene vrednosti sa potenciometra potrebno je bilo modulisati širinu impulsa koja se isporučuje pinovima RGB diode, tako da ona svetli odgovarajućim sjajem od 0 do 255. Analogne i digitalne vrednosti napona sa svakog od tri potenciometra su prikazane na OLED SSD1306 displeju (*Organic Light-Emitting Diode*). Za svaku od tri grupe analognih i digitalnih vrednosti korišćene su odgovarajuće boje (crvena, zelena ili plava) za ispisivanje. Dalje, pritiskom na taster, menja se prikazivanje na displeju i onda se na displeju prikazuju vrednosti faktora ispune svakog od impulsa koji se prosleđuje RGB diodi i vreme trajanja visokog naponskog nivoa. Ispisivanje odgovarajućih grupa vrednosti vrši se različitim bojama za svaku od grupa i različitih u odnosu na prvobitne tri. Zatim, ponovnim pritiskom na taster, ispisivanja displeja se vraćaju u prvobitna stanja.

Kao razvojno okruženje za realizaciju projekta korščen je Blue Pill. Ideja je bila da se korišćenjem okruženja prati položaj potenciometara, čija se vrednost pomoću AD konverzije koju pruža mikrokontroler, koristi da bi se poslao odgovarajući PWM (*Pulse Width Modulation*) signal RGB diodi. Za adekvatno korišćenje AD konverzije iskorišćen je tajmer TIM3, a za prosleđivanje PWM signala tajmer TIM1. Korišćen je princip globalnog interrupt-a da bi AD konverzija bila uspešna. Istovremeno, korišćenjem OLED displeja, ispisuju se željene vrednosti. OLED je povezan na okruženje poštujući principe I2C protokola (*Inter-Integrated Circuit*) [3] [7].

II. ELEKTRIČNA ŠEMA I KORIŠĆENE KOMPONENTE SA OBJAŠNJENJEM

Električna šema sistema prikazana je na slici 1. Spisak korišćenih komponenata:

- razvojno okruženje *Blue Pill*;
- OLED SSD1306 displej;
- 3 potenciometra;
- 3 otpornika od 10 kΩ;
- 1 taster;
- 1 otpornik od 10 kΩ;
- RGB diode;
- 3 otpornika od 220 Ω;
- protoploča;
- konektori (žice);



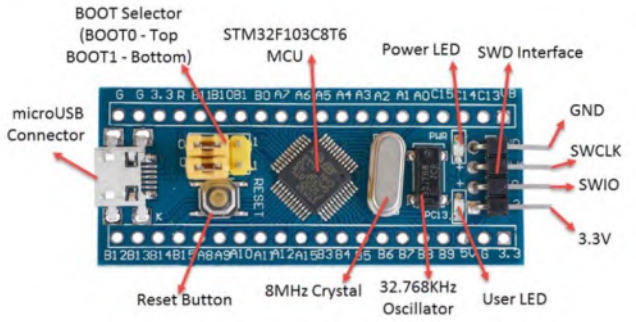
Slika 1. Električna šema u Proteus-u.

S. Dimitrijević, V. Nikolić i N. Nedeljković su studenti sa Katedre za mikroelektroniku, Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: straxi271@gmail.com, neca.nedeljkovic@gmail.com, vukastudent@elfak.rs.

A. Blue Pill

Blue pill (slika 2) je nadimak koji je dat najpopularnijoj, ultra jeftinoj i kompaktnoj STM32F103 ploči. *Blue Pill* je ARM Cortex-M3 32-bitna razvojna ploča mikrokontrolera visokih performansi koja radi na maksimalnoj radnoj frekvenciji od 72 MHz.

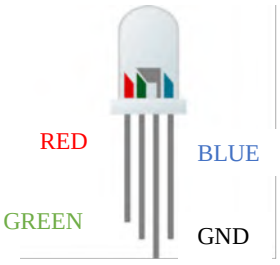
Poredeći se sa Arduino Uno pločom ima više ulaznih/izlaznih pinova, 12-bitnih ADC-a, više eksternih prekidačkih pinova, veću brzinu PWM-a i mnogo više funkcija. *Blue Pill* ploča ima više od jednog I2C, UART i SPI porta, što pomaže u komunikaciji sa drugim uređajima mnogo brže od Arduino UNO ploče. Ima mnogo I/O pinova koji olakšavaju povezivanje više senzora, modula i drugih elektronskih komponenti na ploču [3] [4].



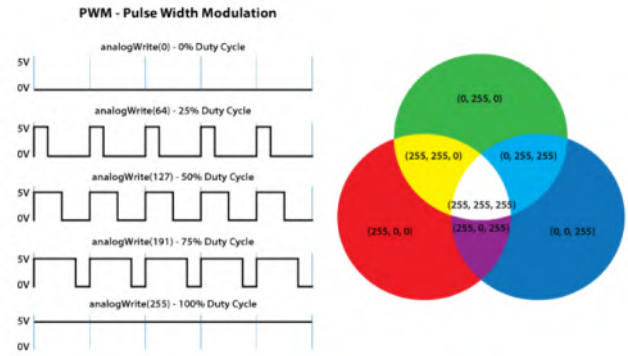
Slika 2. Prikaz *Blue pill* razvojne ploče.

B. RGB diode

RGB dioda predstavlja poseban tip LE diode koja može biti sa zajedničkom anodom ili katodom. RGB dioda korišćena u ovom projektu je sa zajedničkom katodom. Ovaj tip diode poseduje četiri izvoda; tako da je jedan vezan za zajedničku katodu, dok su ostala tri izvoda na anodi pomoću kojih se kontrolišu pinovi. Prikaz RGB diode je na slici 3. Dovođenjem odgovarajućeg napona na pinove RGB diode, ona svetli različitom bojom. Takođe, moguće je dobiti i druge boje sem osnovnih metodom PWM-a, odnosno prosleđivanjem naponskog impulsa sa promenljivom širinom, čime se postiže privid određene boje koja se dobija kombinacijom osnovnih tri (slika 4) [6].



Slika 3. Izgled RGB diode.



Slika 4. Vrednosti PWM-a za dobijanje određenih boja.

C. OLED SSD1306 displej

OLED SSD1306 modul predstavlja moderniji tip displeja za primenu u embedded sistemima, pre svega zbog svojih malih gabarita. Njima nije potrebno pozadinsko osvetljenje, tako da su tanji i efikasniji u poređenju sa LCD ekranima kojima je potrebno belo pozadinsko osvetljenje. Na displeju OLED-a se mogu bojiti pikseli iz mreže piksela. Na taj način dobija se utisak poruke ili slike. Pikseli su uglavnom jednobojni, mada postoje varijacije displeja koje omogućavaju da pikseli svetle u dve boje [1].

OLED displej korišćen u ovom projektu prikazan je na slici 5. Povezuje se preko samo 4 pina, korišćenjem I2C komunikacije, što rasterećuje MCU [7]. Koristi SSD1306 drajver koji omogućava veliki broj podešavanja. Serijski protokoli omogućavaju povezivanje pomoću manjeg broja pinova. I2C protokol podrazumeva komunikaciju između master uređaja i do 27- 1 slejv uređaja, korišćenjem linije za takt (*SCL - Serial Clock*) i linije za podatke (*SDA- Serial Data*) [1] [2].

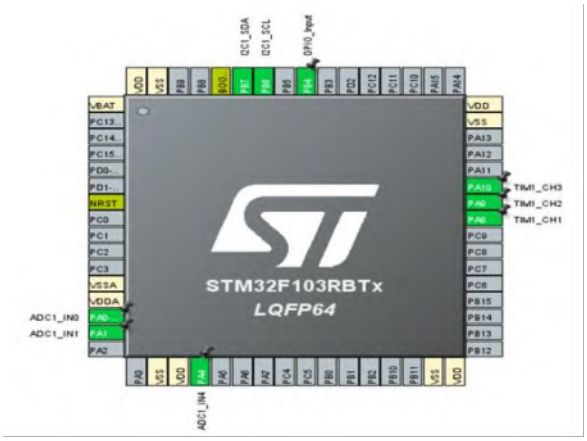
Princip rada ostalih komponenata (potenciometara, otpornika, tastera…) prilično je jednostavan i dobro poznat, pa neće biti detaljnije objašnjen [9] [10] [11].



Slika 5. OLED modul SSD1306.

III. Podešavanja u CubeMX-u

Podešavanja i inicijalne konfiguracije pinova okruženja *Blue Pill* prikazan je na slici 6.



Slika 6. Inicijalne konfiguracije pinova.

Korišćeni pinovi PB4 kao izlazni pin za registrovanje promena na tasteru, zatim PB6, PB7 kao pinovi za komunikaciju sa OLED-om, PA0, PA1, PA4 pinovi za ADC konverziju (očitavanje vrednosti sa potenciometra), i PA8, PA9, PA10 pinovi za slanje PWM signala na RGB diodu [8].

IV. Programski kod u Keil-u

Da bi smo krenuli sa pisanjem koda potrebno je prvo uključiti pojedine biblioteke. Biblioteku za rad sa strignovima, biblioteku koja podrzava I2C komunikaciju kod OLED displeja.

```
#include "fonts.h"
#include "ssd1306.h"
```

Naredni korak jeste inicijalizacija promenljivih, koje će biti korišćene dalje u kodu. Po uslovu zadatka koristili smo jedan taster za promenu prikazivanja vrednosti: digitalne, analogne vrednosti, faktora ispune i vreme trajanja logičke jedinice. Tačnije na OLED dipleju inicijalno biće prikazivane digitalne i analogne vrednosti PWM signala, pritiskom na taster biće prikazini faktor ispune i vreme trajanja PWM signala i tako do narednog pritiska tastera. Funcija koja ispunjava ovaj uslov:

```
int funkcija(taster)
{
    if(taster) //HAL_GPIO_ReadPin(GPIOB,
GPIO_PIN_4
    {
            stanje=!stanje;
    }
    else if(!taster)
    {
            stanje=stanje;
    }
    return stanje;
}
```

Vidimo da je funcija tipa *integer*, njena povratna vrednost jeste parametar stanje, ukoliko je taster pritisnut stanje uvek uzima surotnu vrednost u odnosu na prethodu. Ukoliko nije pritisnut stanje zadržava prethodnu vrednost. Funkcuju zovemo u okviru beskonačne petlje.

```
while(1)
{
…
//digitalne vrednosti adcVal[0], adcVal[2], adcVal[3]
sa potenciometara
// pretvaranje digitalne vrednosti u analognu
// analogne vrednosti sa potenciometara
```

U okviru beskonačne petlje radimo AD konverziju za sve tri vrednosti potenciometra. Digitalne i analogne vrednosti smeštamo u različite stringove koje ćemo koristiti za prikaz na displeju.

```
//generisanje PWM signala koji se šalje RGB diodi sa
potenciometara u opcegu od 0-255

DTC1 = (255*adcVal[0])/2047;
DTC2 = (255*adcVal[1])/2047;
DTC3 = (255*adcVal[2])/2047;
htim1.Instance -> CCR1 = DTC1;
htim1.Instance -> CCR2 = DTC2;
htim1.Instance -> CCR3 = DTC3;

//formule za očitavanje fakora ispune i vreme trajanja
visokog naponskog nivoa
F1= (adcVal[0]*255)/4095;
F2= (adcVal[1]*255)/4095;
F3= (adcVal[2]*255)/4095;
FaktIsp1=F1/255;
FaktIsp2=F2/255;
FaktIsp3=F3/255;
T1=FaktIsp1*T;
T2=FaktIsp2*T;
T3=FaktIsp3*T;
```

Takođe potrebno je napraviti string za smeštanje faktora ispune PWM signala i vreme trajanja logičke jedinice.

```
//pozivanje funkcije
tasterpritisnut=HAL_GPIO_ReadPin(GPIOB,
GPIO_PIN_4);
zamena =funkcija(tasterpritisnut);
```

Ovim delom koda pozivamo funkciju za rad sa tasterom i njenu vrednost upisujemo u promenljivu “zamena”, na osnovu toga da je ova promenljiva na logičkoj nuli ili jedinici biće prikazivane određenje vrednosti signala.

```
if (zamena == 1)
{
SSD1306_Fill(0);
SSD1306_GotoXY(10, 1);
SSD1306_Puts(TxData1, &Font_7x10, 1);
SSD1306_GotoXY(10, 15);
SSD1306_Puts(TxData2, &Font_7x10, 1);
SSD1306_GotoXY(10, 30);
SSD1306_Puts(TxData3, &Font_7x10, 1);

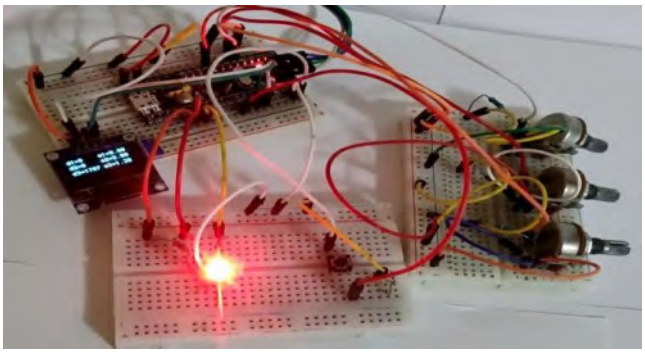
SSD1306_GotoXY(65, 1);
SSD1306_Puts(txPodatak1, &Font_7x10, 1);
SSD1306_GotoXY(65, 15);
SSD1306_Puts(txPodatak2, &Font_7x10, 1);
SSD1306_GotoXY(65, 30);
SSD1306_Puts(txPodatak3, &Font_7x10, 1);

SSD1306_UpdateScreen();
HAL_Delay(50);
}
if (zamena == 0)
{
…
}
```

V. PRAKTIČNA REALIZACIJA

U zavisnosti od stanja potenciometra, tj. njihovim podešavanjem inicijalno će se ispisivati odgovarajuće digitalne i analogne vrednosti na OLED displeju. Tako da će diode svetleti odgovarajućom bojom.

Pritiskom na taster, promeniće se ispis na ekranu OLED displeja, tako da će se sada ispisivati faktor ispune i vreme trajanja visokog naponskog impulsa za sva tri potenciometra. Ponovnim pritiskom na taster ispis na ekranu se vraća u prvobitno stanje, tj. na ispis očitanih analognih i digitalnih vrednosti. Prikaz praktične realizacije dat je na slici 7.



Slika 7. Prikaz povezivanja komponenti i same realizacije projekta.

VI. ZAKLJUČAK

U radu je prikazan sistem za dimovanje RGB diode uz pomoć 3 potenciometra što je i na kraju praktično realizovano kroz povezivanje komponenata na protoploči. Projekat kao takav se može dosta nadograditi i unaprediti po želji projektanta. Može se iskoristiti bilo koji displej za ispisivanje željenih podataka (TFT, LCD, …).

S obzirom da se koristi PWM signal, ovo kolo može se iskoristiti za drajvovanje uređaja koji rade upravo na tom principu (DC motori, razni audio sistemi, invertori…).

ZAHVALNICA

Autori rada se neizmerno zahvaljuju Katedri za mikroelektroniku za pomoć u vidu pužanja adekvatne opreme za praktičnu realizaciju projekta, kao i asistentu Nikoli Mitroviću na nesebičnoj pomoći i sugestijama tokom samog perioda izrade projekta.

LITERATURA

[1] Tehnička specifikacija za OLED SSD1306 modul, Vishay. Dostupno na: https://www.vishay.com/docs/37902.pdf
[2] Tehnička specifikacija za SSD1306 kontroler. Vishay. Dostupno na: https://cdnshop.adafruit.com/datasheets/SSD1306.pdf
[3] „Discovering the STM32 Microcontroller Geoffrey Brown", University of Indiana. Dostupno na: https://legacy.cs.indiana.edu/geobrown/book/pdf
[4] Tehnička specifikacija za NucleoF103. Mbed platform. Dostupno na: https://os.mbed.com/platforms/ST-Nucleo-F103RB/
[5] RGB diode, Dostupno na: https://www.oxfordwebstudio.com/da-li-znate/sta-je-rgb-sistem-boja.html
[6] M. Loi et al., „Challenges and Opportunities of Light-Emitting Diode (LED) as Key to Modulate Antioxidant Compounds in Plants". *A Review*. 2021. Dostupno na: https://www.mdpi.com/2076-3921/10/1/42
[7] „Fast and Effective Embedded Systems Design - Second Edition". Rob Toulson, Tim Wilmshurst. Elsevier, 2017. Dostupno na: https://www.perlego.com/book/1830703/fast-and-effective-embedded-systems-design-applying-the-arm-mbed-pdf
[8] „Podešavanja u CubeMx-u za STM mikrokontrolere, Priručnik". Dostupano na: https://www.st.com/resource/en/user_manual/dm00104712-stm32cubemx-for-stm32-configuration-and-initialization-c-code-generation-stmicroelectronics.pdf
[9] Potenciometar objašnjenje,. Dostupno na: https://en.wikipedia.org/wiki/Potentiometer
[10] Otpornik objašnjenje. Dostupno na: https://en.wikipedia.org/wiki/Resistor
[11] Prekidač (taster) objašnjenje. Dostupno na: https://en.wikipedia.org/wiki/Electronic_switch

Projektovanje pametnog sistema za pešačke prelaze

Aleksandar Vučković

Sadržaj – Ovaj rad predstavlja projekat koji se odnosi na pametni pešački prelaz takav da bi mogao samostalno da kontroliše saobraćaj u zoni pešačkog prelaza. Osnovni cilj projekta jeste povećanje bezbednosti pešaka blagovremenim obaveštavanjem vozača da su pešaci u blizini prelaza.

I. UVOD

Sa porastom broja stanovnika u gradovima rastu i potrebe za novim bulevarima i ulicama a samim tim i za dodatnim prevoznim sredstvima kao što su metroi ili električni autobusi kao i ostali tipovi javnog prevoza. Uz to, raste i potreba za obezbeđivanjem raskrsnica, za projektovanjem održivih sistema koji mogu samostalno da ih kontrolišu čime može da se poveća bezbednost kako putnika u vozilu tako i pešaka, biciklista pa i dece koja mogu neoprezno da izlete na ulicu. Ovaj rad se bavi upravo takvim jednim sistemom čiji je osnovni cilj da poveća bezbednost pešaka na putu kao i na ulicama gde postoji velika frekvencija saobraćaja a pešački prelazi nisu regulisani semaforima, već pešačkom zebrom i znakovima. Kao takvi, mogu biti jako opasni ukoliko se na pešačkom prelazu sretnu pešaci i nesavesni vozač pa može doći do tragičnog ishoda.

Zbog navedenih razloga, ovakav sistem treba biti projektovan da bude pouzdan i da obezbedi siguran prelaz pešaka preko kolovoza. Osnovni zadatak jeste da blagovremeno upozori vozača da se na pešačkom prelazu očekuju pešaci i da treba što pre da prilagodi brzinu kretanja svog vozila i da se na taj način zaštiti pešak. Ovo je posebno važno u situacijama kada se ne očekuje kretanje pešaka, a to je najčešće noću kada se i vozila kreću brže. Zbog toga se mora povećati opreznost vozača sugerisanjem da nailazi na pešački prelaz. Ceo ovaj sistem treba da se sastoji od senzora koji detektuju pešake koji nailaze na pešački prelaz, kontrolne elektronike koja to obrađuje i nekog tipa svetlosnog saobraćajnog znaka. Ovaj znak treba tekstom ili svetlosnim signalom da upozori vozača i da mu da informaciju da se očekuje pešak na prelazu. Uz to, treba da pošalje informaciju do susednog saobraćajnog znaka da se upozore i vozači sa druge strane kolovoza. Na ovaj način bi pešački prelaz postao mnogo sigurniji.

II. SVETLOSNI SAOBRAĆAJNI ZNAKOVI

Svetlosni saobraćajni znakovi su postali veoma zastupljeni kako po autoputevima tako i po raznim magistralnim putevima, pa i po gradovima sve sa ciljem da se saobraćaj obezbedi bolje i da se izbegne što je više moguće nezgoda.

Standard za svetlosne saobraćajne znakove sa izmenljivom porukom EN12966 [1] definiše osnovna pravila po kojima treba da se vode proizvođači ovakvih znakova kako bi sve bilo uniformno na nivou cele Evrope. U pitanju je jako složen standard koji se poziva na dosta drugih standarda pa ga je potrebno veoma dobro razumeti. U samom standardu su definisani materijali za izradu kućišta znakova, naponi napajanja, fizičke karakteristike koje treba da podnesu u pogledu udara vetra, temperature, prodora vode i prašine. Kao svetleći medijum koji može da se koristi u znakovima jesu LE diode koje se mogu naći u velikom broju kućišta, različitih snaga, napona i struje vođenja pa o upotrebi takvih treba voditi računa.

Neka od jako bitnih pravila koja moraju biti zadovoljena, pogotovo za LE diode, jesu vizuelne performanse u koje spadaju:

- klasifikacija fotometrijskih parametara,
- boje,
- osvetljaj,
- odnos osvetljaja,
- širina snopa zraka,
- uniformnost,
- vidljivo treptanje.

Sve ove stavke su preuzete iz standarda EN12899 [2] na koji se nadovezuje gore navedeni standard.

III. SISTEM ZA PEŠAČKE PRELAZE

Ideja za projektovanje jednog ovakvog sistema je potekla iz očiglednih razloga da se u Republici Srbiji nedovoljno posvećuje pažnja putnim prelazima, bilo da su u pitanju pružni ili pešački prelazi. Uglavnom su nedovoljno obezbeđeni adekvatnom signalizacijom ili onom koja je dotrajala pa ne funkcioniše korektno što povećava nesigurnost kako za pešaka tako i za vozilo.

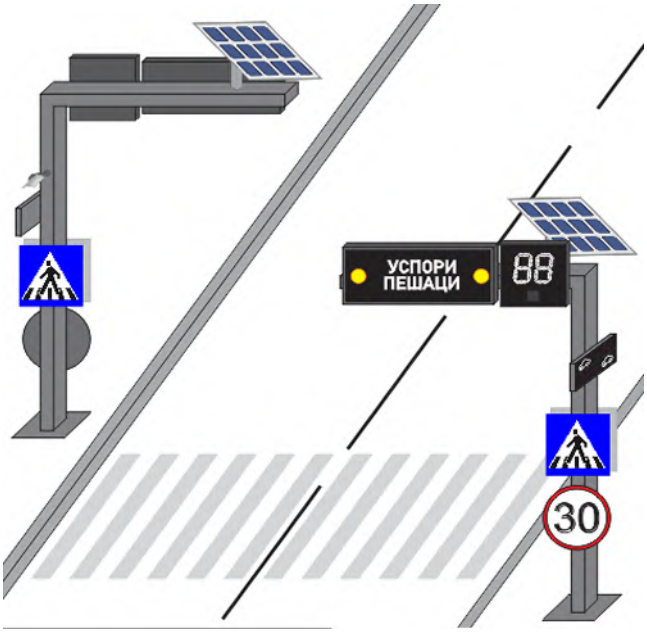
U ovom radu se razmatra nezavisan sistem koji ne zahteva prisustvo čoveka za nadzor prelaza, već može samostalno da funkcioniše. Na slici 1 data je ilustracija projekta koja je verodostojna ideji.

U pitanju je sistem koji detektuje vozilo kao i njegovu brzinu i nakon toga upozorava vozača da nailazi na pešački prelaz i da treba da prilagodi brzinu. Sa druge strane, i pešak dobija adekvatno upozorenje u vidu svetlosne signalizacije da se po kolovozu kreće vozilo i to sa koje strane nailazi na pešački prelaz. Pored toga, specijalizovani detektori pešaka stalno motre na površinu pešačkog prelaza iako nema vozila na kolovozu tako da se nailaskom pešaka na pešački prelaz

A. Vučković je dipl. inž. elektrotehnike i računarstva, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: aleksandar.vuckovic@elfak.rs.

aktivira znak koji upozorava vozača da se pešak kreće po prelazu. Sve ovo dosta povećava bezbednost na putu jer su upozorena i vozila na putu i pešaci koji imaju nameru da koriste prelaz.

Sistem se sastoji od konzolnog nosača koji se postavlja neposredno pre pešačkog prelaza a na njemu su postavljeni najbitniji delovi sistema. U suštini, ovaj sistem se može postaviti na bilo kakvom putu gde postoji mogućnost za postavljanje nosača jer su to delovi sistema koji zahtevaju najviše prostora. Kada govorimo o postavljanju na magistralnim putevima koji prolaze kroz naseljena mesta, ni to ne predstavlja prepreku jer ima dovoljnu visinu pa neće smetati visokim vozilima kao što su kamioni i autobusi.



Slika. 1. Ilustrovani sistem za pešačke prelaze.

IV. OSNOVNI DELOVI SISTEMA

Pošto se ovde govori o jednom sistemu koji ima dosta delova, svaki će biti opisan ponaosob. U osnovne delove sistema spadaju:

a. konzolni nosač,
b. piktogramski znak sa porukom,
c. radarski znak,
d. bočni znak sa indikacijom vozila,
e. kontrolna štampana ploča,
f. uređaj za detekciju pešaka,
g. solarni panel i baterija.

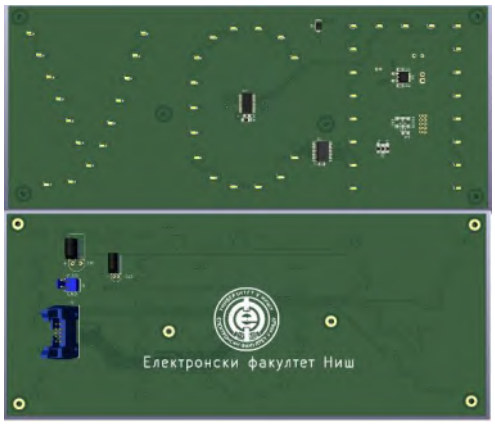
A. Konzolni nosač

Kao što je navedeno u prethodnom odeljku, ceo sistem se postavlja na konzolni nosač koji u suštini treba da bude izrađen od materijala koji ima dobru mehaničku čvrstoću i nosivost kako bi mogao da podnese masu znakova kao i različite vremenske prilike. Čelični profili se uklapaju u ove karakteristike, a pošto je čelik materijal koji može lako da korodira mora biti podvrgnut cinkovanju sa spoljne strane čime se štiti materijal. Ovaj nosač je projektovan po uzoru na one koji se već koriste u saobraćaju, bilo da je u pitanju autoput ili zona škole i kao takav se pokazao kao vrlo pouzdan na duže vremenske prilike. Konzolni nosač treba biti postavljen tako da se ne nalazi ni predaleko ni preblizu pešačkog prelaza što zavisi najviše od uređaja za detekciju pešaka koji mora biti usmeren ka pešačkom prelazu (u daljem tekstu će biti detaljno opisana funkcija ovog modula) kako bi što pre dao informaciju sistemu da se pešak nalazi na kolovozu.

B. Piktogramski znak sa porukom

Piktogramski znak sa porukom *„УСПОРИ, ПЕШАЦИ“* u vidu teksta je najbitniji znak koji daje informaciju vozačima da se očekuju pešaci na putu i da je potrebno da prilagode brzinu svog vozila. U pitanju je svetlosni saobraćajni znak gde delove teksta odnosno slova predstavljaju grupe svetlećih dioda koje su raspoređene u obliku slova i postavljene na štampanu ploču. Svako slovo ili grupa od nekoliko slova predstavlja jednu štampanu ploču koju je potrebno dizajnirati. Sve pomenute štampane ploče u ovom radu su projektovane pomoću softverskog alata KiCad EDA [3]. Rad LE dioda se kontroliše pomoću specijalizovanih integrisanih kola – LED drajvera koji su takođe kontrolisani od strane mikrokontrolera.

Na slici 2 su prikazane LE diode u obliku ćiriličnih slova sa ciljem da se prilikom dizajniranja štampanih ploča sa LE diodama i LED drajverima odrede lokacije na kojima će LE diode biti postavljene kako bi se ispratili oblici slova. Najbitnije je definisati veličinu pojedinih karaktera u tekstu kao i dimenzije treptača koji treba da svetle žutom bojom u impulsnom režimu i na taj način da upozore vozače na pešake. Treptači se projektuju tako da su diode raspoređene sa dovoljnim rastojanjem u krugu prečnika veličine jednog karaktera. Obzirom da u tekstu postoji dosta karaktera, projektovano je četiri štampanih ploča od kojih je prva ploča sa delom teksta *„УСП”* prikazana na slici 2.



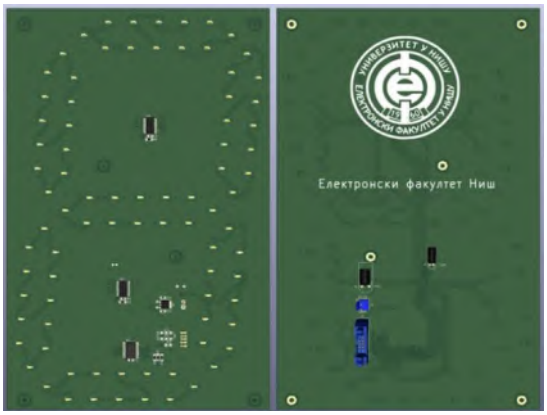
Slika. 2. Štampana ploča sa delom teksta *„УСП”*.

C. Radarski znak

Radarski znak je jako bitan u ovom projektu jer daje informaciju o brzini vozila koje se kreće po kolovozu i približava se pešačkom prelazu. Uloga mu je dvostruka:

- da upozori vozača ukoliko se kreće brzinom većom od dozvoljene i
- da obavesti kontrolnu ploču da se vozilo kreće po kolovozu.

U suštini, radarski znak se sastoji iz radar modula koji meri brzinu vozila i šalje podatak o brzini kontrolnoj ploči. Kontrolna ploča na osnovu primljenog podatka i putem SPI komunikacije šalje podatke LED drajverima u radarskom znaku na štampanim pločama sa LE diodama o tome koje grupe treba da budu uključene. LE diode su u radarskom znaku raspoređene u obliku 7-segmentnog displeja tako da se svaki segment sastoji iz jednakog broja LE dioda. Na slici 3 je prikazana projektovana štampana ploča sa LE diodama u obliku 7-segmentnih cifara.

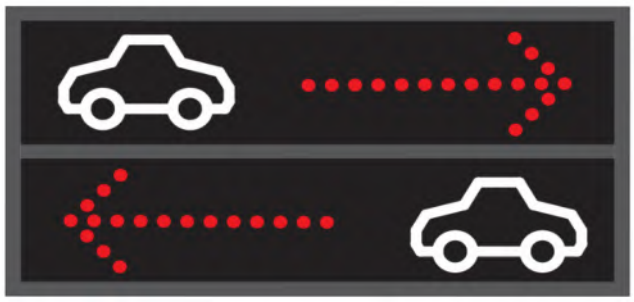


Slika. 3. Štampana ploča 7-segmente cifre.

Radarski modul koji će se koristiti za svrhu prepoznavanja brzine je *HLK-LD306-24G* [4]. Ovaj modul može kontinualno da daje podatke o brzini vozila preko serijske komunikacije što znači da može da vrši nadzor na putu sve vreme dok je uključen. Upravo je ovakav modul potreban celokupnom ovom sistemu jer će on vršiti detekciju vozila kao i brzine vozila i na osnovu tih podataka će se uključivati natpisi na znakovima.

D. Bočni znak sa indikacijom vozila

Osnovna uloga ovog znaka bi bila da pruži informaciju pešacima sa koje strane kolovoza nailazi vozilo i da li je bezbedno da kroče na pešački prelaz, ukoliko vozač nije na vreme prilagodio svoju brzinu. Znak je dovoljno velikih dimenzija tako da je lako uočljiv, a indikacija vozila je u obliku strelica koje pokazuju u kom smeru u odnosu na pešački prelaz nailazi vozilo. Izgled ovog znaka je prikazan na slici 4.



Slika. 4. Znak za indikaciju vozila.

E. Kontrolna štampana ploča

Kontrolna ploča predstavlja najbitniji deo ovog sistema jer je odgovorna za pravilno funkcionisanje svih pojedinačnih periferija. Radom kontrolne ploče upravlja 32-bitni mikrokontroler *STM32F103C8T3* [5] koji predstavlja mikrokontroler novije generacije sa mnoštvom funkcija u pogledu brzine, protokola za komunikaciju, načina programiranja, načina pristupa memoriji… Kada se govori o njegovim električnim karakteristikama može raditi pri naponima od 2-3,6 V, može se koristiti interni oscilator za davanje taktnog signala pored spoljašnjeg i spada u mikrokontrolere sa malom potrošnjom.

Kontrolna štampana ploča treba da obezbedi kontrolu nad celim sistemom na osnovu programskog koda i dodatnih integrisanih i elektronskih komponenata. Elektronika sa štampane ploče prikuplja podatke, obrađuje ih, a na osnovu svoje memorije odlučuje koji će saobraćajni znak biti uključen i u kom režimu će raditi. Kao i ostale štampane ploče i ova je projektovana tako da je raspored komponenata na ploči postavljen u skladu sa periferijama koje se povezuju na nju odnosno na ostale znakove i senzore. Obzirom da se radi o dva pametna sistema (po jedan za oba smera saobraćajne trake) oni međusobno komuniciraju bežičnim putem čime su komande sihnronizovane. Sama štampana ploča je smeštena u saobraćajni znak sa tekstom, a izgled ove ploče je dat na slici 5.



Slika. 5. Štampana kontrolna ploča.

F. Uređaj za detekciju pešaka

Uređaj za detekciju pešaka je neophodan element ovog sistema jer mu ovaj modul daje informaciju o prisutnosti pešaka na kolovozu ili trotoaru. Može se reći da je jednako bitan za pešake kao što je radarski modul bitan za vozila. U pitanju je piroelektrični senzor koji detektuje pešaka na osnovu infracrvenog zračenja koje on emituje i daje informaciju kontrolnoj ploči. Ovaj uređaj može da detektuje pešake na udaljenosti od 24 metara i širine do 2 metra. Ovo daje korisnu površinu za detekciju od 48 kvadratnih metara, a obzirom da postoje sa obe strane kolovoza to daje površinu od 96 kvadratnih metara u kojima će pešaci sigurno biti detektovani.

G. Solarni panel i baterija

Da bi ceo ovaj sistem bio potpuno održiv i samostalan zahteva baterijsko napajanje kao i solarni panel za dopunjavanje iste. Kada se govori o solarnom panelu, bitno je da bude snage od minimalno 150 W jer će se koristiti litijumska baterija kapaciteta 52 Ah po jednom konzolnom nosaču. Litijumska baterija koja će biti deo ovog sistema jeste baterija oznake *ReLion RB52-LT* [6] koja spada u grupu litijumskih baterija sa ćelijom izrađenom od materijala litijum-gvožđe-fosfat. Nominalni napon ove baterije jeste 12,8 V, a prilikom punjenja dostiže vrednosti od 13,5 V pri kom se smatra da je baterija puna. Kada baterija dostigne napon od 12,5 V smatra se da se ispraznila 90 % dok za napon od 12 V se smatra da je ostalo 5 % korisnog kapaciteta (95 % kapaciteta je ispražnjeno).

V. Princip rada sistema

Ono što nedostaje da bi ceo ovaj sistem bio „pametan“ jeste programski kod, koji ovde neće biti prikazan, već će biti opisan princip rada po kome bi programski kod trebalo napisati. Na početku rada, mikrokontroler treba da proveri da li je detektovano vozilo u oba smera i ukoliko jeste, treba da uključi odgovarajuću štampanu ploču strelice. Obzirom da se detekcija vozila zasniva na detekciji brzine tj. da li radarski modul očitava kretanje vozila određene brzine, očitanu vrednost brzine vozila je potrebno iskoristiti kao bi se ona ispisala na radarskom znaku odnosno na štampanim pločama cifre i to sa one strane kolovoza na kome je detektovano vozila. Ukoliko je detektovano sa obe strane kolovoza, potrebno je ispisati brzinu vozila na oba radarska znaka prema odgovarajućoj brzini vozila. Pošto je vozilo detektovano paralelno sa tim se proverava da li je uređaj za detekciju pešaka dao informaciju o prisutnosti pešaka i samim tim se sada uključuju oba znaka za tekstom kao i treptači. Ukoliko je vozilo detektovano samo sa jedne strane kolovoza a očekuju se pešaci na prelazu takođe se uključuju oba znaka sa tekstom. Kada su detektovane brzine vozila, proverava se da li u dozvoljenom opsegu, tj. da li je brzina vozila manja od 30 km/h i da li je veća od 3 km/h. Ukoliko je brzina vozila ispod 30 km/h treptači treba da se uključuju i isključuju jednom u sekundi (0,5 s su uključeni, 0,5 s su isključeni), a tekst treba da bude samo uključen. Kada je brzina vozila u drugom opsegu onda se brzine treptanja treptača povećava na 4 puta u sekundi dok sada tekst počinje da treperi dvaput u sekundi. Na ovaj način se vozačima daje upozoravajuća informacija da nailaze na pešački prelaz i da treba da prilagode brzinu svog vozila. Nakon toga se proverava da li su pešaci ili vozila detektovani i poslednjih deset sekundi na osnovu čega se ili produžava rad znakova ili se isključuju zbog uštede energije obzirom da se radi o baterijskom napajanju.

VI. Zaključak

Reflektujući znakovi koji se danas mogu sresti kod bilo kog obeleženog pešačkog prelaza nekada nisu dovoljni da se vozačima ukaže na to da nailaze na pešački prelaz što može dovesti do nesreća. Ovde se može zaključiti da je savest vozača jednako bitna kao i savest pešaka i da ni jedan ni drugi ne treba da očekuju da će neko drugi voditi računa o njihovoj bezbednosti. Upravo iz ovog razloga je proistekla ideja za projektovanje ovakvog sistema – povećanje bezbednosti na pešačkom prelazu. Pošto je sistem samostalan i zahteva minimalno održavanje idealan je za sredine gde je teško dopremiti električnu energiju jer poseduje sopstvenu bateriju pa može biti postavljen svuda gde mu mehaničke dimenzije dozvoljavaju.

Zahvalnica

Autor ovog rada se zahvaljuje Katedri za mikroelektroniku na Elektronskom fakultetu, Univerziteta u Nišu, posebno prof. dr Danijelu Dankoviću na pruženoj pomoći i svakodnevnoj dostupnosti, kao i u davanju smernica prilikom realizacije ovog projekta.

Literatura

[1] European committee for standardization, “Vertical road signs - Part 1: Variable message signs”, Technical Committee CEN/TC 226, EN 12966-1, March 2007
[2] European committee for standardization, “Fixed, vertical road traffic signs - Part 1: Fixed signs”, Technical Committee CEN/TC 226, EN 12899-1, November 2007
[3] KiCad EDA - Schematic Capture & PCB Design Software, dostupno na: https://kicad.org
[4] Hi-Link, “Millimeter wave radar Module”, HLK-LD306-24G datasheet, May 2020
[5] STMicroelectronics, “Medium-density performance line Arm® -based 32-bit MCU”, STM32F103C8T3 datasheet, Mar. 2022
[6] ReLion, “Lithium iron phosphate battery”, RB52-LT datasheet, Aug. 2021

Digitalni sat realizovan korišćenjem integrisanog kola DS1302 i ATMEGA328P mikrokontrolera

Hristina Todorović, Mirjana Ćirić, Aleksandra Trajković, Boris Jevtić

Sadržaj – Realizacija projekta digitalnog sata koji ima za cilj prikaz tačnog vremena na četiri sedmosegmentna displeja realizovana je pomoću softverskih alata: *Proteus*, *Altium Designer* i *Arduino IDE*.

I. UVOD

Cilj našeg projekta je da se korišćenjem mogućnosti koje pruža integrisano kolo DS1302 povezano na mikrokontroler ATMega328P prikazuje tačno vreme u 24 h formatu. Idejna šema je kreirana u softverskom alatu *Proteus*, koji takođe služi za testiranje funkcionalnosti uređaja i programskog koda pisanog u *Arduino IDE* softverskom alatu. Princip rada uređaja svodi se na prosleđivanje očitane vrednosti časova i minuta na mikrokontroler i prikaz njihovih vrednosti na sedmosegmentnim displejima.

II. KORIŠĆENE KOMPONENTE

A.Mikrokontroler ATMega328P

Mikrokontroler ATMega328P iz familije megaAVR, služi da primi podatke sa integrisanog kola DS1302 (*Real Time Clock*), da podatke obradi i prosledi pomeračkim registrima 74HC164 (*Serial in – Paralel out*), koji paralelnom komunikacijom sa dekoderima 74LS47 određuju cifre koje će biti ispisane na sedmosegmentnim displejima.

B. Pomerački registar 74HC164

Da bi se ostvarila paralelna komunikacija između mikrokontrolera i četiri sedmosegmentna displeja potrebno je 28 izlaznih pinova. Korišćenjem dekodera *74LS74* broj potrebnih pinova smanjen je na 16. Uvođenjem *Serial In – Paralel Out* pomeračkih registara broj potrebnih pinova na mikrokontroleru smanjen je na dva.

H. Todorović, M.Ćirić, A.Trajković i B. Jevtić su studenti na modulu Elektronske komponente i mikrosistemi, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: hristina.t@elfak.rs, mirjana.ciric@elfak.rs, trajkovic.aleksandra@elfak.rs, borisjevtic@elfak.rs.

C. BCD – 7 segmentni dekoder 74LS47

BCD u sedmosegmentni dekoder služi da informaciju koju dobije na ulaznim pinovima A, B, C i D prosledi na pinove od QA do QG koji respektivno aktiviraju LE diode na sedmosegmentim displejima i to čine nulom, jer je sam sedmosegmentni displej u konfiguraciji *Active Low*.

D. Otporni niz

Otporni niz sastoji se od 8 otpornika od po 220 Ω koji služe da ograniče struju kroz LE diode.

E. Sedmosegmentni displej

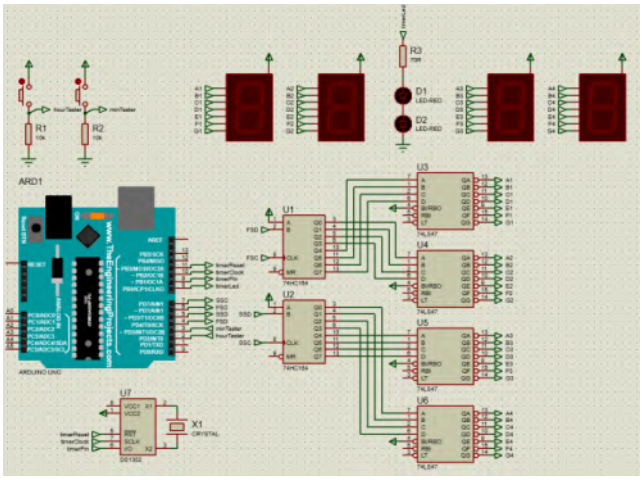
Konfiguracija sedmosegmentnog displeja je *Active Low*, jer su sve diode povezane na zajedničku anodu i samim tim imaju zajedničko napajanje, pa se aktivacija određene diode odvija dovođenjem nule na katodu željene diode.

F. Real Time Clock DS1302

RTC DS1302 je komponenta koja računa sekunde, minute i sate i ima mogućnost da pamti vreme kada uređaj nije priključen na napajanje uz korišćenje litijumske baterije. Kada je vrednost napajanja na pinu *VCC2* veća od 0,2 V koristi se dato napajanje, a kada je manja koristi se napajanje sa litijumske baterije. Komponenti se dodaje kristalni oscilator koji osciluje na 32,768 kHz.

III. ŠEMA U PROTEUS-U

Na slici 1 u softverskom alatu *Proteus* data je šema koja prikazuje način na koji se sa RTC-a DS1302 prosleđuje vreme na *Arduino* okruženje preko pina *dsTimer*. Zatim, *Arduino* putem pinova *firstShiftClock*, *firstShiftData*, *secondShiftClock* i *secondShiftData* daje informaciju konvertorima kodova 74LS47 koje LE diode na sedmosegmentima da uključe.



Slika 1. Šema uređaja u softverskom alatu *Proteus.*

Sve se odvija u realnom vremenu i to upload-ovanjem programskog koda koji kontroliše rad celog uređaja. Naravno, moguće je i manuelno podešavanje vremena preko tastera za sate i minute, koji su u *pull-down* konfiguraciji, kao i *reset*-ovanje čitavog sata.

IV. PROGRAMSKI KOD U ARDUINU

Na slici 2 u softverskom alatu *Arduino* prikazan je prvi deo koda u kome su definisani pinovi i u delu *setup* su dodata dva *ISR-a (Interrupt Service Routine)* za sate i minute. Funkcija *myRTC.setDS1302Time()* sadrži parametre za sekunde, minute, sate, dan u nedelji, dan u mesecu, mesec i godinu respektivno.

```
#include <virtuabotixRTC.h>
#include <TimerOne.h>

volatile const int firstShiftData=4, secondShiftData=5, firstShiftClock=6, secondShiftClock=7;
const int  timerPin=9, timerClock=10, minTaster=3, hourTaster=2, timerReset=11;
volatile int minutes=19, hours=12, ledStatus=HIGH, timerLed=8, sec;
volatile byte dataHour=0, dataMin=0;
virtuabotixRTC myRTC(timerClock, timerPin, timerReset);

void setup() {
  Serial.begin(9600);
  for (int i=4; i<11; i++){
    pinMode(i, OUTPUT);
  }
    pinMode(minTaster, INPUT);
    pinMode(hourTaster, INPUT);
  attachInterrupt(digitalPinToInterrupt(minTaster), minISR, RISING);
  attachInterrupt(digitalPinToInterrupt(hourTaster), hourISR, RISING);
  Timer1.initialize(500000);
  Timer1.attachInterrupt(timerISR);
  myRTC.setDS1302Time(sec, minutes, hours, 4, 13, 10, 2022); //sec, min, hour, day in the week, month, year

}
```

Slika 2. Prvi deo programskog koda u *Arduinu.*

U glavnom delu programa (slika 3) postoje dve funkcije: jedna iz biblioteke za *RTC* koja *update*-uje vreme i druga koja prikazuje vreme na sedmosegmentnim dispelijma i koja je defnisana zasebno. Nakon toga definisani su i *Interrupt* za minute, koji update-uje vreme na svakih 60 sekundi, kao i *Interrupt* za sate koji *update*-uje vreme na svaka 24 sata.

```
void loop() {
  myRTC.updateTime();
  showTime();
}

void minISR(){
  minutes++;
  if (minutes>59){
    minutes=0;
  }
  myRTC.setDS1302Time(0, minutes, hours, 4, 13, 10, 2022);
  myRTC.updateTime();
}

void hourISR(){
  hours++;
  if (hours>23){
    hours=0;
  }
  myRTC.setDS1302Time(0, minutes, hours, 4, 13, 10, 2022);
  myRTC.updateTime();
}
```

Slika 3. Drugi deo programskog koda u *Arduinu.*

U krajnjem delu (slika 4) programski kod je prilagođen prikazu od po dve odvojene cifre iz svakog od *Shift* registara i to konverzijom iz binarnog zapisa.

Postoji i tajmer koji na svakih 500 ms menja stanje naponskog nivoa pina *timerLed* i time indikuje prolazak polovine sekunde na *LE* diodama između četiri sedmosegmentna displeja.

```
void showTime(){
  hours=myRTC.hours;
  dataHour=hours;
  if(hours>=10){
    dataHour=hours+6; }
  if(hours>=20){
    dataHour=hours+12; }
    shiftOut(firstShiftData, firstShiftClock, MSBFIRST, dataHour);
  delay(30);
  minutes=myRTC.minutes;
  dataMin=minutes;
  if(minutes>=10){
    dataMin=minutes+6; }
  if(minutes>=20){
    dataMin=minutes+12; }
  if(minutes>=30){
    dataMin=minutes+18; }
  if(minutes>=40){
    dataMin=minutes+24; }
  if(minutes>=50){
    dataMin=minutes+30; }
    shiftOut(secondShiftData, secondShiftClock, MSBFIRST, dataMin);
  delay(30);

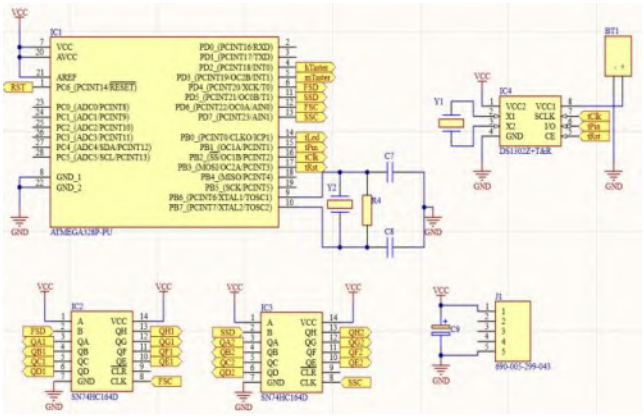
}
```

Slika 4. Treći deo programskog koda u *Arduinu*

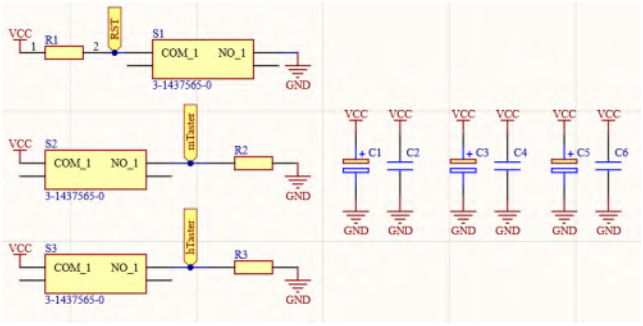
V. KREIRANJE PLOČICE U *ALTIUM*-U

Kako bi se adekvatno realizovao sat neophodno je isprojektovati štampanu ploču koja sadrži sve komponente koje doprinose radu istog. Na slici (slika 5) prikazan je blok mikrokontrolera na kome se nalaze *ATMEGA328P, DS1302 i 74HC164.*

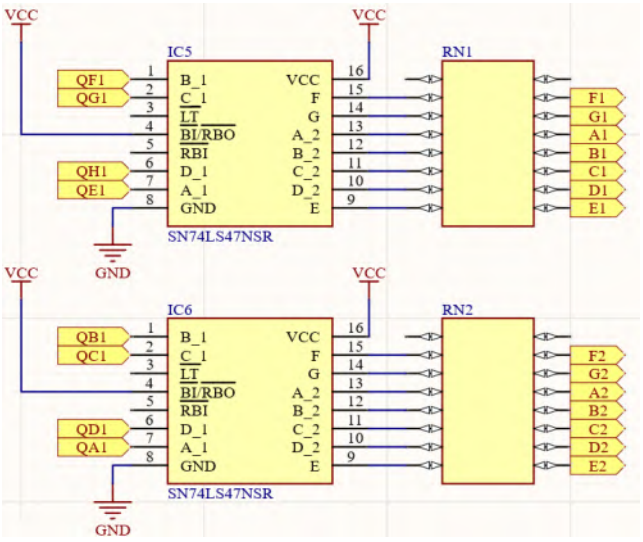
Dalje na slikama 6, 7 i 8 prikazani su tasteri za podešavanje sati i minuta, kao i taster za reset, zatim dekoderi za sedmosegmentne displeje, *74LS47* koji se povezuju na same displeje preko otporne mreže i to u *Active-Low* konfiguraciji.



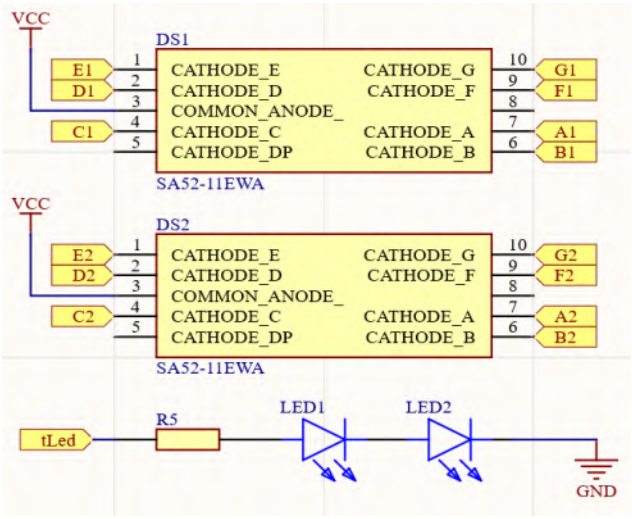
Slika 5. Blok mikrokontrolera u *Altium*-u.



Slika 6. Tasteri u *Altium*-u.

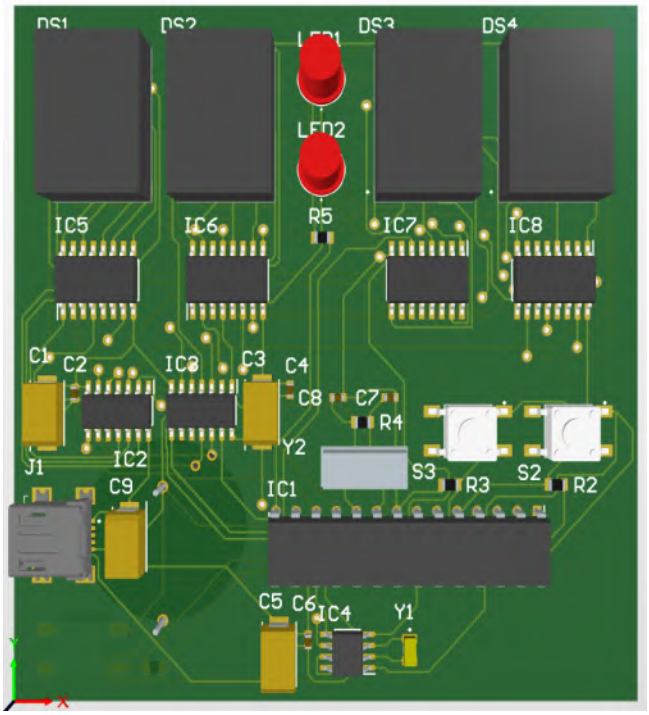


Slika 7. *BCD* u sedmosegmentni dekoderi *74LS47* sa svojom otpornom mrežom.

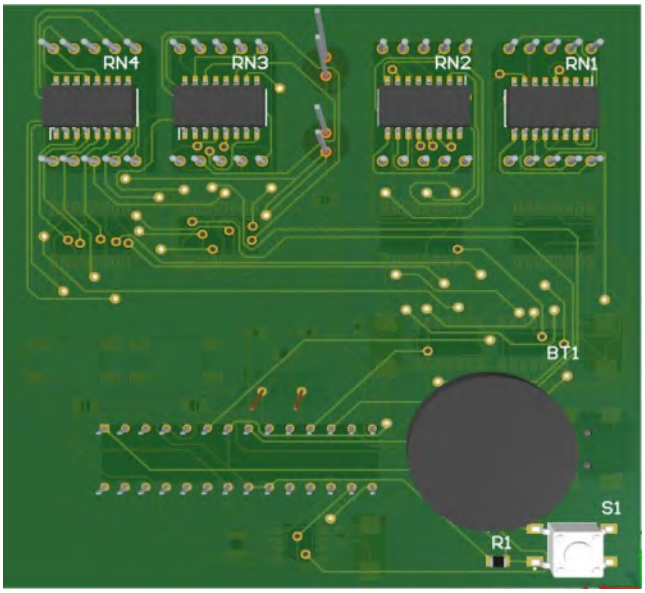


Slika 8. Sedmosegmentni displeji u konfiguraciji sa zajedničkom anodom.

Prikaz štampane pločice u 3D-u dat je na slikama 9a i 9b.



Slika 9a. 3D prikaz gornje strane uređaja.



Slika 9b. 3D prikaz donje strane uređaja.

VI. ZAKLJUČAK

U ovom radu realizovan je digitalni sat. Opisan je postupak odabira komponenata, date prikladne simulacije i projektovanja štampane ploče korišćenjem adekvatnih softverskih paketa.

ZAHVALNICA

Autori ovog rada se srdačno zahvaljuju profesorima i saradnicima sa Katedre za mikroelektroniku na Elektronskom fakultetu u Nišu. Posebna zahvalnost asistentkinji Aleksandri Stojković koja nam je obezbedila sav potreban materijal za izradu ovog projekta, kao i sugestije koje nam je uputila tokom izrade istog.

LITERATURA

[1] Zoran Prijić, Digitalna mikroelektronika, Pomerački registri Dostupno na: http://mikro.elfak.ni.ac.rs/wp-content/uploads/2017/10/Digitalna_Shift_registers.pdf
[2] Zoran Prijić, Digitalna mikroelektronika, Kombinaciona logička kola Dostupno na : http://mikro.elfak.ni.ac.rs/wp-content/uploads/2017/10/Digitalna_combinational_logic.pdf
[3] Tehnička dokumentacija pomeračkog registra 74HC164 Dostupno na : https://www.diodes.com/assets/Datasheets/74HC164.pdf
[4] Tehnička dokumentacija mikrokontrolera ATMEGA328P Dostpuno na: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf
[5] Tehnička dokumentacija dekodera HC74LS47.Dostupno na: https://pdf1.alldatasheet.com/datasheet-pdf/view/51080/FAIRCHILD/74LS47.html

Illumination Indicator Using an Arduino Development Environment

Nikola Stefanović

Abstract −This paper presents a multi-featured illumination indicator using a phototransistor to provide the Arduino Uno microcontroller board its input, which is then used to control an array of LED's and an LCD.

I. Introduction

An illumination indicator refers to a device that, when exposed to different intensities of light, produces different outputs reflecting the objective intensity of light that reaches the sensor. The information the sensor receives has to be processed so it can be used effectively, and for this paper, the Arduino Uno [1] microcontroller board is used.

The Arduino Uno is a beginner friendly, easy to use, microcontroller board, which uses an ATMEGA328P microcontroller for various input and output functions, in order to fulfil different goals.

II. Setup

This device uses a single phototransistor to provide the input signal, and is connected to the Arduino Uno board to analog pin A0, which gives it a range of values from 0 to 1023, or ten bits.

For the output, there are two types: an LED array, and an LCD screen. The LED [2], [3] array consists of 10 red LED's, set up in a way so that, depending on the value of the output from the Arduino Uno, or rather the voltage, the LED's light up sequentially, with none of them lighting up at zero volts, and all of them lighting up at five volts. Since the Arduino Uno can't produce an analogue output signal, a PWM pin is used since it can mimic an analogue output signal through clever manipulation of a digital signal. The LED's are each connected, over the anode, by one Schottky diode [4] which has a low forward voltage, ideal for this circuit.

The cathode of the LED is connected to ground through a resistor which regulates the current in that circuit branch so that its value is always between 10-30 mA. To ensure all the LED's light up sequentially, the resistors have values of 160, 82 and 43 Ohms. The circuit used for the LED array is shown in Fig. 1.

N. Stefanović, 3rd-grade student of Electronic components and microsystems at Faculty of Electronic engineering, University of Nis, Aleksandra Medvedeva 14, 18000 Nis, Serbia, E-mail: nikola.stefanovic@elfak.rs.

The LCD [5] is configured to show two values: on the first row the illumination in lux units, and on the second row, the corresponding environment for that illumination level. The LCD is connected to the Arduino Uno board through sixteen pins which control the contrast of the screen, background brightness, how it receives data etc [6].

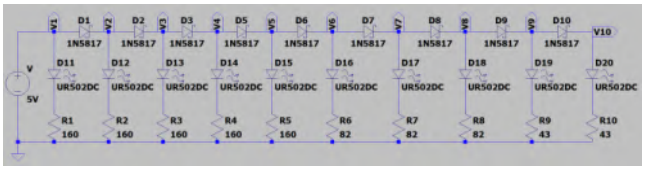


Fig. 1. The circuit used for the LED array.

The entire setup is shown below in Fig. 2.

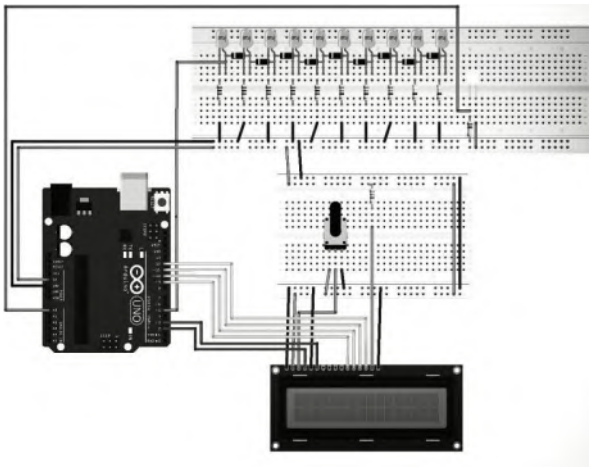


Fig. 2. The entire device setup pictured using Fritzing [7].

III. Measurements

In order to get an accurate representation of what illumination values correspond to what values of the input signal, measurements were made using two separate instruments: Mastech MS8209 multimeter with a lux setting [8], and a Peaktech 5086 digital luxmeter [9], shown here in Fig. 3.

Measurements were taken with the Arduino Uno board, and the specific instrument, both placed in a container which has an opening to manually adjust the light coming into the container, so that both would be exposed to the same light levels, and an accurate comparison could be made. The container setup is shown in Fig. 4.



Fig. 3. Instruments used in measuring the illumination in lux. Notice the different values displayed for the same light level.



Fig. 4. The setup used to measure illumination levels for the Arduino board and the instrument.

The measuring process was done using values from the input signal to indicate light levels inside the container. Then for the same light level, the reading from the instrument was written down. For lower values, measurements were made with smaller intervals, as well as for values approaching the sensitivity limit of the phototransistor. Only one instrument was used at a time in order to improve the accuracy of the measurement, and different measurements were made depending on how precise the measuring environment for the specific instrument could be set up. All the results are shown below in Table I.

Since it is harder to get a sense of how the data correlates, the values are graphically show below in Fig. 5 and Fig. 6. The change in illumination is an exponential function, and in Fig. 6, a very sharp rise in illumination is seen towards the high end of the Arduino Uno input signal. For clarity, this part is shown in Fig. 7.

TABLE I
MEASUREMENT VALUES

Arduino input	Multimeter (lx)	Luxmeter (lx)
5	14	0,55
11	14	0,59
20	14	0,63
50	15	0,73
100	18	0,91
160	36	8,20
200	41	23,6
250	56	46,1
300	72	71,8
350	85	100
400	95	132
450	116	153
500	132	177
550	202	194
600	229	212
650	266	217
700	284	234
750	337	251
800	449	268
850	498	286
900	544	312
950	579	340
970	-	545
980	-	585
990	-	750
991	-	791
992	-	823
993	-	900
994	-	1000
995	-	1100
996	-	1200
997	-	1300
998	-	1400
999	-	1500
1000	979	2200
1001	-	3000

Using this data, an exponential function that correlates these two data sets can be implemented. Unfortunately, this function is not implemented in this paper. Instead, a linear correlation was used, which is presented in the next section of this paper.

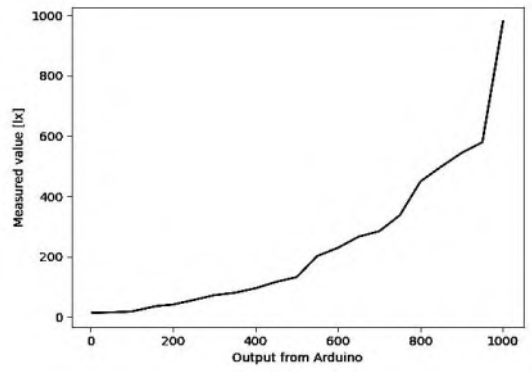


Fig. 5. Graph showing values in lux measured with the multimeter, showing an exponential dependency between measurements in lux and measurements from the Arduino Uno.

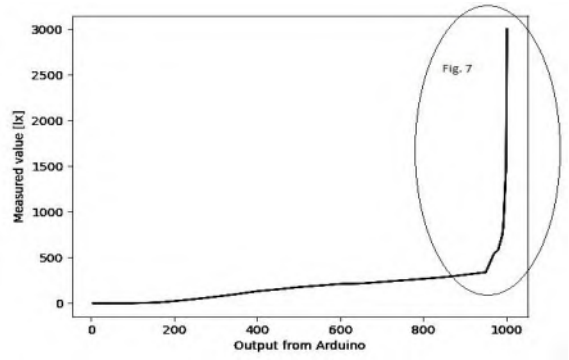


Fig. 6. Graph showing values in lux measured with the luxmeter, with a part highlighted.

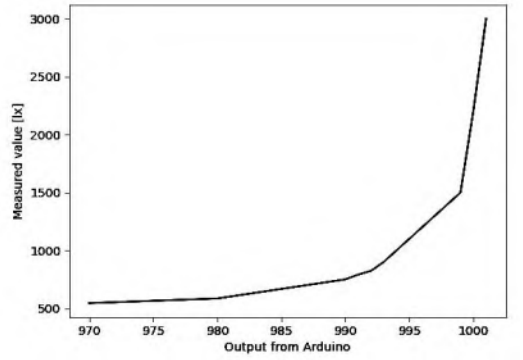


Fig. 7. Graph showing the highlighted part of Fig. 6 zoomed in, emphasizing the steep exponential growth of illumination in lux compared to the output from the Arduino.

IV. THE CODE

The Arduino developers provide their own Integrated Development Environment (IDE) simply called Arduino IDE, but the code for this project was written in PlatformIO [10]. PlatformIO is a VSCode [11] extension and it provides far more features than the other IDE.

In order to use an LCD with the Arduino Uno board, the relevant library [12] needs to be included. After the library is included, the function LiquidCrystal needs to be used in order to initialize the parameters of the display. This code needs to come before all other functions. Also in this section, all the variables that are used, are created.

Arduino processes its code using two main functions: void setup() and void loop(). Void setup() is used for code that will be executed only once. To start the LCD, its necessary to use the function lcd.begin(). To define the pins on the Arduino Uno board that are active and for what purpose, the function pinMode() is used. The void loop() function, as its name suggests, loops indefinitely as long as the Arduino Uno board has power. The main part of the program is written here.

Firstly, the Arduino Uno board gets the analogue input from the phototransistor and assigns it to a variable. It then uses this variable in the map function to map one set of values to another, since the PWM output of the Arduino only ranges from 0 to 255. It then writes this value to the pin which controls the LED array (Fig. 8).

```
18  void loop() {
19    // This code controls the LED array
20    int vrednost = analogRead(A0);
21    ledPin = map(vrednost, 0, 1023, 0, 255);
22    analogWrite(5, ledPin);
```

Fig. 8. Code that controls the LED array.

The second part of the code deals with writing to the LCD. Since there is no auto clear function, meaning anything on the screen that is not written over constantly will stay there, the screen needs to be cleared for every loop of the program using lcd.clear(). This necessitates the use of the delay() function in order to control the "framerate" of the screen so it doesn't flicker. The code that prints out on the LCD is shown in Fig. 9.

```
27  // This code prints the initial text on the LCD
28  lcd.home();
29  lcd.clear();
30  lcd.print("Lux: ");
31  lcd.print(lux);
32  lcd.setCursor(0, 1);
33  lcd.print("Amb: ");
34  lcd.print(ambijent[i]);
```

Fig. 9. Code that prints both lines on the LCD.

In order to print the guess of what environment the user might be in, or rather what environment ideally has that illumination level, a series of if functions determine which word from an array should be printed out. A sample of the code is shown in Fig. 10. This part works on a delay of five seconds so that the display will not respond for momentary changes in illumination levels. The reference for what illumination level in lux corresponds to what environment is shown in Fig. 11.

```
39  // Changes the ambient value which is written on the LCD
40  // Night
41  if (lux < 10 && millis() - trenutno_vreme > period) {
42    i = 1;
43    trenutno_vreme = millis();
44  }
```

Fig. 10. A sample of the if statement that determines what word should be printed.

Effective Lighting Levels

Illuminance (lux)	Activity	Area
100	Casual seeing	Corridors, changing rooms, stores
150	Some perception of detail	Loading bays, switch rooms, plant rooms
200	Continuously occupied	Foyers, entrance halls, dining rooms
300	Visual tasks moderately easy	Libraries, sports halls, lecture theatres.
500	Visual tasks moderately difficult	General offices, kitchens, laboratories, retail shops.
750	Visual tasks difficult	Drawing offices, meat inspection, chain stores.
1000	Visual tasks very difficult	General inspection, electronic assembly, paintwork, supermarkets.
1500	Visual tasks extremely difficult	Fine work and inspection, precision assembly.
2000	Visual tasks exceptionally difficult	Assembly of minute items, finished fabric inspection.

Fig. 11. Table showing effective lighting levels [13].

There are a total of ten environments: Night, Corridor, Room, Big room, Library, Kitchen, Store, Supermarket (written as SMarket because of space limitations), Overcast and Sunny.

V. Conclusion

This paper presents a working prototype for an illumination indicator with different output capabilities.

It enables this device to be used for different purposes including, but not limited to: smart home lighting, automatic street lighting, detection of outside conditions etc. Further upgrades to the device could include a better configuration for the LED array with more uniform currents and the implementation of a function which accurately maps values from the Arduino Uno board to the measured values from the instruments. The author invites others to use and improve upon the functionality of the device.

Acknowledgments

The author of this paper would like to express his deepest gratitude to prof. dr. Emilija Živanović for recommending to try and write a paper for this conference, and to prof. dr. Danijel Danković for providing consistent and insightful tips and advice which helped the author write this paper, and make it better than the author alone could manage.

References

[1] Arduino, https://www.arduino.cc/
[2] https://www.arduino.cc/documents/datasheets/LED (red).pdf
[3] A. Prijić, *Poluprovodničke komponente: Diode*, Avaliable on http://mikro.elfak.ni.ac.rs/predmeti/poluprovodnicke-komponente/
[4] https://www.onsemi.com/pdf/datasheet/1n5817-d.pdf
[5] https://www.arduino.cc/documents/datasheets/LCDscreen.pdf
[6] Arduino Projects Book, ed. S. Fitzgerald, M. Shiloh, T. Igoe, Arduino LLC, 2015. https://fritzing.org/
[7] https://www.walmart.com/ip/Mastech-MS8209-5-in-1-Multimeter-Lux-Sound-Level-Humidity-Temperature-True-RMS/978701616
[8] https://peaktech-rce.com/en/environmental-measures/463-peaktech-5086-digital-lux-meter-3-digit-40-400000-lux.html
[9] https://platformio.org/
[10] https://code.visualstudio.com/
[11] https://www.arduino.cc/reference/en/libraries/liquidcrystal
[12] https://www.savemoneycutcarbon.com/learn-save/brief-guide-to-lighting-levels/

Automatizacija parkinga sa PIC mikrokontrolerom

Bogdan Krstić

Sadržaj – U ovom radu opisana je realizacija automatskog sistema za parking korisćenjem PIC16F628-04. Program za mikrokontroler je pisan u programskom jeziku C i razvojnom okruženju MicroC for PIC.

I. UVOD

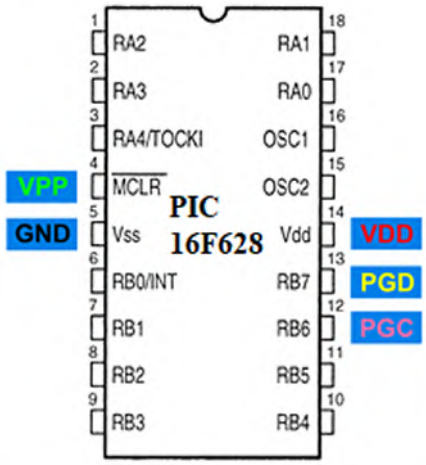
U ovom radu biće opisan sistem za automatizaciju parkinga pomoću PIC mikrokontrolera. Na današnjim parking servisima potrebno je prisustvo radnika koji će regulisati ulaz i izlaz vozila kao i broj slobodnih mesta na prakingu. Ovaj sistem, pomoću odgovarajućih komponenata (mikrokontrolera, displeja, senzora, dioda i pasivnih komponenata) obavlja ove radnje i poboljšava rad parking servisa.

II. PIC 16F28

PIC 16F628-04 spada u 8-bitne mikrokontrolere RISC arhitekture. PIC (*Peripheral Interface Controller*) mikrokontroleri su mikrokontroleri američke kompanije *Microchip Technolgy*. PIC1628-04 ima RISC arhitekturu koja predstavlja noviji koncept od fon-Nojmanove arhitekture a nastala je iz potrebe da se ubrza rad mikroprocesora. Kod fon-Nojmanove arhitekture postoji jedna magistrala za komunikaciju CPU sa programskom memorijom, memorijom za podatke i perifernim uređajima. Kod Harwardske arhitekture postoje tri magistrale: magistrala podataka (Data Bus), programska magistrala (Program Bus) i ulazno/izlazna magistrala podataka (Input/output Data Bus), čime je omogućen veći protok podataka kroz centralnu procesorsku jedinicu a time i veća brzina rada, odnosno efikasnost procesora. Na primer, dok CPU radi sa memorijom podataka, istovremeno se u CPU prenosi programska instrukcija

PIC16F84A-04 ima osamnaest nožica, odnosno pinova i izrađuje se u varijanti kućišta DIL18 i SMD kućištu. Skraćenica DIP ili DIL nastala je od je od početnih slova engleskih reči *Dual In-line Package* što u prevodu znači pakovanje sa nožicama sa obe strane. SMD je takođe skraćenica od početnih slova engleskih reči *Surface Mounted Devices* što u prevodu znači komponente za površinsku montažu, asocirajući da za lemljenje ove vrste komponenti nije potrebno imati rupe, na štampanoj ploči, kroz koje bi nožice prošle i bile zalemljene.

B. Krstić je student treće godine osnovnih akademskih studija na smeru Elektronske komponente i mikrosistemi na Elektronskom fakultetu u Nišu, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: bkrstic@elfak.rs.



Slika 1. Uloga pinova PIC 16F628.

Pinovi mikrokontrolera PIC16F628-04 imaju sledeće uloge:

Pin br.1 RA2 -Drugi pin porta A.

Pin br.2 RA3 -Treći pin porta A.

Pin br.3 RA4/T0CKI- Četvrti pin porta A. Ako se slobodni brojačTMR0 pobuđuje impulsima iz spoljnog izvora,tada se mpulsi dovode na ovaj pin.

Pin br.4 MCLR -Reset ulaz i ulaz Vpp napona programiranja mikrokontrolera (12 do 13,5V).

Pin br.5 Vss -Ulaz za masu napajanja.

Pin br.6 RB0/INT -Nulti pin porta B. Dodatna funkcija je interaptni ulaz.

Pin br.7 RB1 -Prvi pin porta B.

Pin br.8 RB2 -Drugi pin porta B.

Pin br.9 RB3 -Treći pin porta B.

Pin br.10 RB4 -Četvrti pin porta B.

Pin br.11 RB5 -Peti pin porta B.

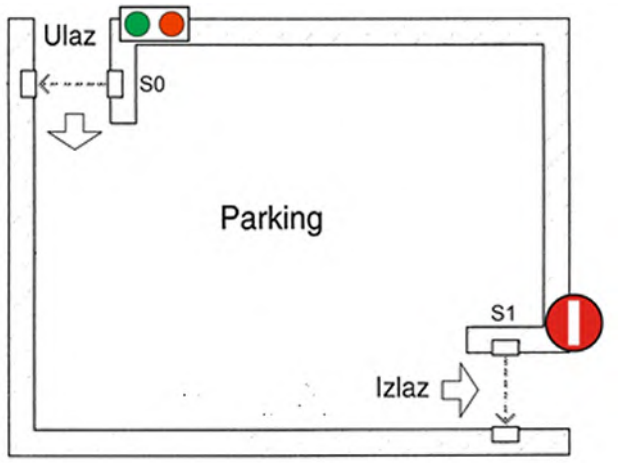
Pin br.12 RB6 -Šesti pin porta B. Dodatna funkcija je „Clock“ linija PGC u programskom modu.

Pin br.13 RB7 -Sedmi pin porta B. Dodatna funkcija je „Data“ linija PGD u programskom modu.

Pin br.14 Vdd -Ulaz pozitivnog pola napajanja (2V do 5,5V).

Pin br.15	OSC2 -Pin za povezivanje sa oscilatorom.
Pin br.16	OSC1 -Pin za povezivanje sa oscilatorom.
Pin br.17	RA0 -Nulti pin porta A.
Pin br.18	RA1 -Prvi pin porta A.

III. OPIS RADA



Slika 2. Šema parkinga.

Mikrokontroler je povezan sa ostalim elementima kao na slici 3 i registruje ulazak i izlazak vozila sa parkinga, kao na slici 2 kako bi odredio broj preostalih slobodnih mesta. Zbog bržeg testiranja programa u razvojnom sistemu, u ovom primeru kapacitet parkinga je 10 vozila.

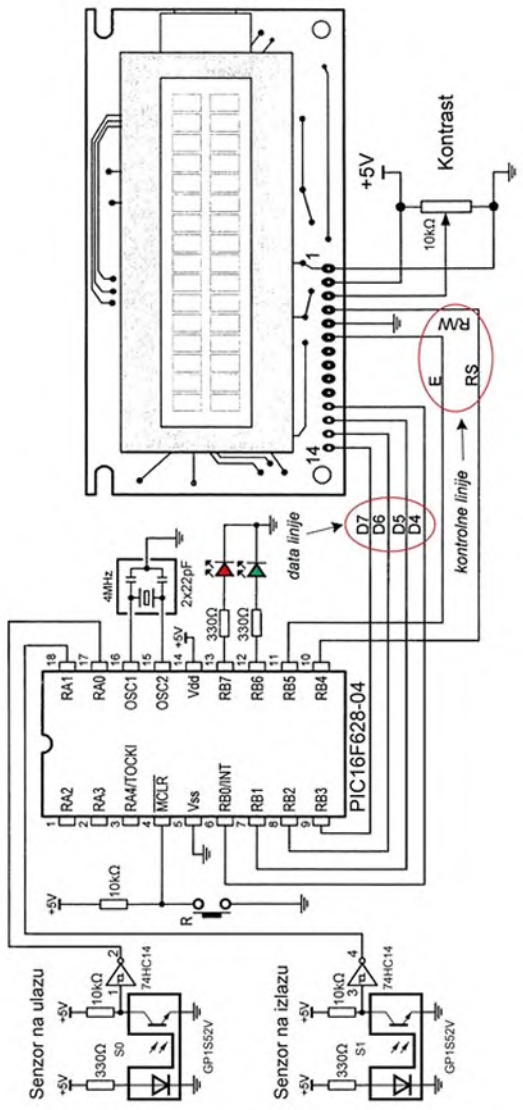
Za registrovanje ulaska i izlaska vozila, na ulazu i izlazu parkinga nalaze se senzori S0 i S1 , respektivno. Svi pinovi porta A su ulazni. Pin RA0 spojen je sa senzorom S0 a pin RA1 sa senzorom S1. Senzori su fotoelektrični, odnosno to su fotoelektrični prekidači sa LED diodom, kao predajnikom, i fototranzistorom, kao prijemnikom. U odsustvu vozila dioda osvetljava fototranzistor, koji je zbog toga provodan tako da je na kolektoru nivo logičke nule. Prisustvo vozila sprečava osvetljavanje tranzistora koji je zbog toga neprovodan tako da je na kolektoru nivo logičke jedinice. Zbog relativno male brzine kretanja vozila na ulazu i izlazu, naponi na kolektorima nisu pravougaoni. Dobijanje pravougaonih napona omogućavaju invertujuća Šmitova okidna kola. Nivo logičke jedinice na pinovima RA0 i RA1 označava odsustvo vozila na ulazu i izlazu, respektivno. Nivo logičke nule na pinovima RA0 i RA1 označava prisustvo vozila na ulazu i izlazu, respektivno.

Za određivanje broja slobodnih mesta, u programu se koristi promenljiva x2. U programu je definisano da se prilikom učitavanja programa u mikrokontroler, u memorijsku lokaciju, sa adresom 0x02 u EEPROM memoriji, učita početni sadržaj 10. Na početku izvršenja programa ova vrednost dodeljuje se promenljivoj x2.

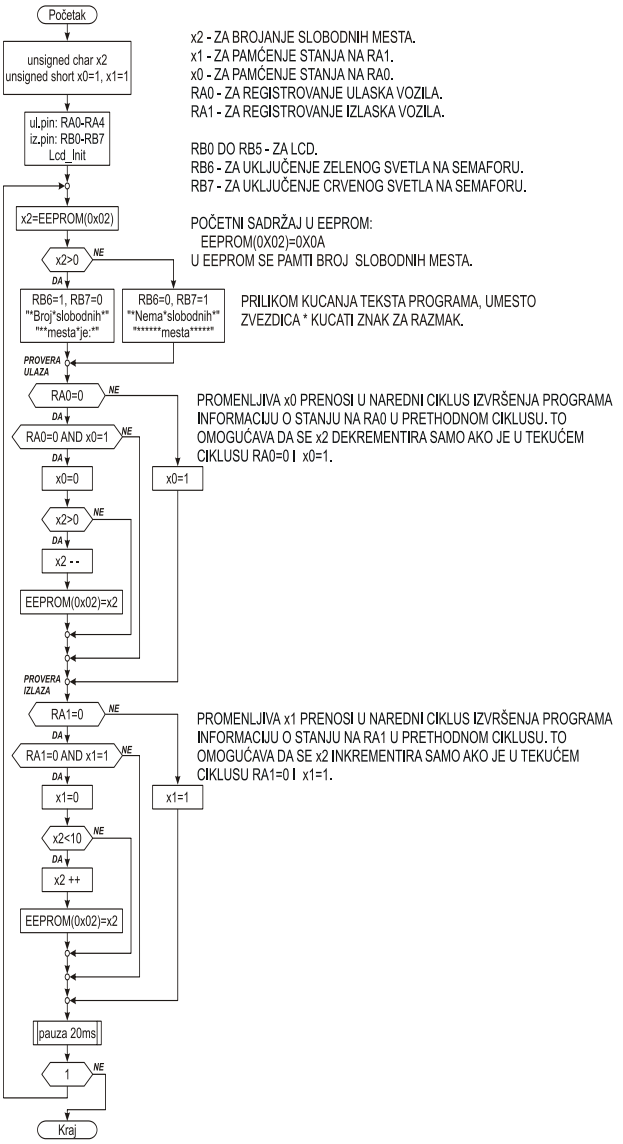
Svakim ulaskom vozila u parking, promenljiva x2 se dekrementira a svakim izlaskom vozila iz parkinga, promenljiva x2 se inkrementira. Svaki put kada se promeni, vrednost promenljive x2 bekapuje se u memorijsku lokaciju, sa adresom 0x02 u EEPROM memoriji, kako bi se sačuvala u slučaju prekida napajanja.

Promenljiva x0 prenosi u naredni ciklus izvrsenja programa informaciju o stanju na RA0 u prethodnom ciklusu. To omogucava da se x2 dekrementira samo ako je u tekucem ciklusu RA0=0 i x0=1.

Promenljiva x1 prenosi u naredni ciklus izvrsenja programa informaciju o stanju na RA1 u prethodnom ciklusu. To omogucava da se x2 inkrementira samo ako je u tekucem ciklusu RA1=0 i x1=1.



Slika 3. Električna šema.



Slika 4. Algoritam rada.

Kod u programskom jeziku C:

```
sbit LCD_RS at RB4_bit;
sbit LCD_EN at RB5_bit;
sbit LCD_D4 at RB0_bit;
sbit LCD_D5 at RB1_bit;
sbit LCD_D6 at RB2_bit;
sbit LCD_D7 at RB3_bit;
sbit LCD_RS_Direction at TRISB4_bit;
sbit LCD_EN_Direction at TRISB5_bit;
sbit LCD_D4_Direction at TRISB0_bit;
sbit LCD_D5_Direction at TRISB1_bit;
sbit LCD_D6_Direction at TRISB2_bit;
sbit LCD_D7_Direction at TRISB3_bit;
// End LCD module connections
unsigned char x2;
unsigned short x0,x1;
void main()
{
 TRISA=0b11111111;
CMCON = 0X07;
TRISB=0;
Lcd_Init();
Lcd_Cmd(_LCD_CLEAR);
Lcd_Cmd(_LCD_CLEAR);
Lcd_Cmd(_LCD_CURSOR_OFF);
do
  {
   x2=EEPROM_Read(0x02);
   if(x2>0)
    {
     PORTB.F6=1;
     PORTB.F7=0;
     Lcd_Out(1,1,"*Broj*slobodnih*");
     Lcd_Out(2,1,"**mesta*je:*");
     if(x2>9)Lcd_Chr(2,13,48+x2/10);
     else Lcd_Chr(2,13,' ');
     Lcd_Chr(2,14,48+x2%10);
    }
   else
    {
     PORTB.F6=0;
     PORTB.F7=1;
     Lcd_Out(1,1,"*Nema*slobodnih*");
     Lcd_Out(2,1,"******mesta*****");
    }
   if(PORTA.F0==0)
    {
     if(PORTA.F0==0 && x0==1)
      {
        x0=0;
        if(x2>0)
         {
          x2--;
          EEPROM_Write(0x02, x2);
         }
      }
    }
   else
    {
      x0=1;
    }
   if(PORTA.F1==0)
    {
     if(PORTA.F1==0 && x1==1)
       {
        x1=0;
        if(x2<10)
         {
          x2++;
          EEPROM_Write(0x02, x2);
         }
       }
    }
```

```
  else
    {
      x1=1;
    }
   Delay_ms(20);
  }
 while(1);
}
```

IV. ZAKLJUČAK

Korišćenjem PIC mikrokontrolera,LCD-a 2x16 karkatera kao i odgovarajućih komponenti otpornika, dioda, proto polče, tastera, optokaplera realizovan je automatski sistem za parking koji shodno EEPROM memorija PIC-a može registrovati $10 \cdot 10^6$ ulazaka i izlazaka vozila.

LITERATURA

[1] Nebojša Matić: PIC mikrokontroleri, MikroElektronika, Beograd, 2003.

[2] Milan Verle: PIC mikrokontroleri, MikroElektronika, Beograd, 2008

[3] Milan Verle: PIC microcontrollers Programming in C, MikroElektronika, Beograd, 2009.

[4] PIC16F628 Data Sheet, Microchip, 2001

Implementacija Linuks operativnog sistema na mikrokontrolerima sa ARM arhitekturom

Nikola Mitrović, Milan Đorđević

Sadržaj − U ovom radu prikazana je osnovna implenetacija Linuks operativniog sistema na mikrokontroler BCM2709-A7 sa ARM arhitekturom. Detaljnije je objašnjena čitava arhitektura postavke sistema, sa naglaskom na sam proces pokretanja sistema na odredišnom uređaju. Glavni korišćeni resursi su detaljno objašnjeni i parametri izbora resursa su detaljno prikazani. Primer rada Linuks kompajlera pokazan je na jednostavnom test primeru.

I. UVOD

Tendencija povećanja broja portabilnih elektronskih uređaja je u konstantnom porastu. Velika većina portabilnih elektronskih uređaja imaju u sebi ugrađena (eng. *embedded*) elektronska kola, pa se ovakvi uređaji mogu nazvati *embedded* sistemima. Ovakvi sistemi su zastupljeni od okruženja komercijalnih *bluetooth* zvučnika, pametnih svetla i malih kućnih aparata pa do mobilnih telefona, delova automobila i medicinskih uređaja. Može se smatrati da ovi sistemi objedinjuju sve elektronske sisteme koji nisu računari.

Jedan od glavnih razloga zbog kojih se računari ne smatraju kategorijom *embedded* sistema, pored toga što nisu portabilni, je ta što na matičnoj ploči imaju mikroprocesor, dok se kao centralni element *embedded* sistema uzima mikrokontroler. Razlika između mikroprocesora i mikrokontrolera je ta što mikroprocesor predstavlja samo procesor, dok se pod izrazom mikrokontroler podrazumeva uređaj koji, pored procesora (koji se najčešće u literaturi može sresti kao CPU - *Central Processor Unit*), ima i svoju memoriju (*flash*, ROM, RAM) i svoje periferijske jedinice (pinovi, tajmeri, dodatna kola za UART, SPI i slično) [1]. Zbog toga, mikroprocesoru (na primer, jednog računara) je potrebno eksterno dodati RAM i ROM memoriju i eksterno ga povezati sa periferijskim jedinicama, ali mikroprocesor to kompenzuje dosta većom brzinom i radnim taktom u odnosu na mikrokontroler. Analogno tome, pošto je u jedan čip mikrokontrolera potrebno smestiti i CPU i memorije, memorijski kapacitetti mikrokontrolera uglavnom su ograničeni i dosta manji nego memorije koje se obično povezuju na računar, odnosno povezuju sa mikroprocesorom računara.

N. Mitrović je student doktorskih studija pri Katedri za mikroelektroniku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: nikola.i.mitrovic@elfak.ni.ac.rs.

M. Đorđević je student na Akademiji Tehničko Vaspitačkih Strukovnih Studija – Odsek Niš, Aleksandra Medvedeva 20, 18000 Niš, Srbija, E-mail: milan94srb@gmail.com.

Ukupna memorija jednog prosečnog komercijalnog mikrokontrolera, na primer STM32F103 je 128 kB fleš memorije i 20 kB RAM memorije [2], dok su memorije najmanjih kapaciteta koje se povezuju sa mikroprocesorima računara reda veličine desetina GB. Zbog toga, primena mikrokontrolera u *embedded* sistemima je najčešće jednostavna i podrazumeva uključivanje ili isključivanje niza pinova, očitavanja nekih informacija sa pinova, prijem ili slanje određenih podataka i slično.

Ipak, razvojem i skaliranjem komponenata, postalo je moguće povećati internu memoriju mikrokontrolera. Tako na primer, razni mikrokontroleri, pre svega oni novijih serija, bazirani na ARM arhitekturi mogu imati memorijske kapacitete koji se mere redom veličine stotina megabajta. Povećanjem broja proizvođača, pre svega iz Azije, ovakvi mikrokontroleri su postali jako pristupačni tržištu i omogućili veliki razvoj, pre svega, industrije mobilnih telefona. Glavna prednost koju su ovi mikrokontroleri sa velikim kapacitetima memorije omogućili je ta da se usvoji jedna od glavnih prednosti računara, a to je da se na mikrokontroleru omogući implementiranje pojedinih operativnih sistema (u najjednostavnijem slučaju za mobilne telefone, operativni sistem *Android*). Pregled memorijskih zahteva pojedinih OS dat je u tabeli I [3].

TABELA I
POREĐENJE NEKIH OPERATIVNIH SISTEMA

operativni sistem	godina objavljivanja	minimalni memorijski zahtevi	preporučeni memorijski zahtevi
Linux Ubuntu	2004	256 MB	1+ GB
DS Linux	2008	8 MB	256 MB
Android 10	2019	1 GB	2+ GB
iOS 10	2016	2 GB	4+ GB
Windows 10	2015	2 GB	4+ GB

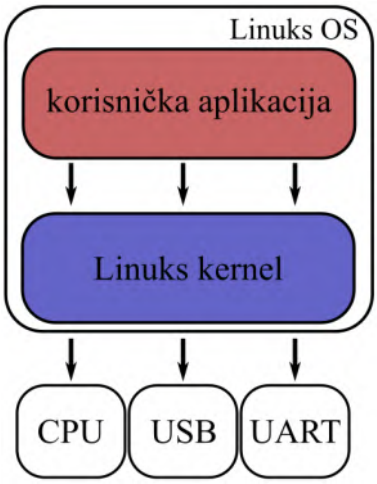
Zbog toga, mikrokontroleri koji imaju napredne memorijske kapacitete se u litearturi često mogu naći i pod nazivom mikroračunari [4]. Razvojem i drugih periferijskih jedinica, omogućen je rad sa implementiranim operativnim sistemima u realnom vremenu (bez velikog kašnjenja izvršenja instrukcija). U ovom radu detaljnije je prikazan proces implementacije jednog operativnog sistema na jedan mikroračunar, konkretno implementacija Linuks operativnog sistema na Broadcom BCM2709 sa ARM *Cortex* A7 jezgrom koji je glavni deo *Raspberry Pi* 2 okruženja.

II. KORIŠĆENI RESURSI

U ovom poglavlju biće detaljnije opisani korišćeni softverski i hardverski resursi.

A. Operativni sistem Linuks Ubuntu

Linuks je operativni sistem otvorenog tipa. Glavni deo Linuks operativnih sistema (OS) je Linuks kernel, napisan od strane Linusa Torvaldsa, koji je i bio preteča za projektovanje čitavog operativnog sistema [5]. Kernel sistem predstavlja podskup OS koji je zadužen direktno za komunikaciju sa hardverskim periferijskim jedinicima. Treba razlikovati Linuks distribucije OS (kao što su npr. *Ubuntu*, *Fedora*, *Nitrux* itd.) od Linuks kernela (deo svakog od distribucija Linuks OS, zadužen za komunikaciju sa periferijskim jedinicama). Blok šema koja prikazuje ovaj odnos prikazana je na slici 1.

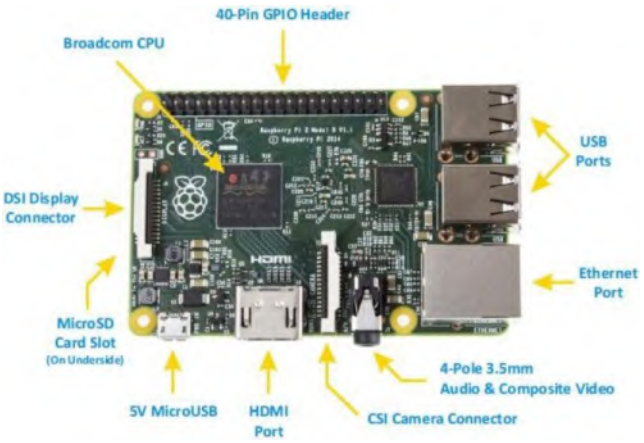


Slika 1. Blok šema principa rada Linuks OS.

U osnovnim slučajevima, kernel nije vidljiv korisniku OS i korisnik ne zadaje komande direktno kernelu, već preko odgovarajućih interfejsa i aplikacija višeg nivoa se, između ostalih delova OS, kada je to potrebno, aktivira i kernel. Karakteristika Linuks kernela, za razliku od nekih drugih OS (*Windows*, *Android*) je ta što korisnik može, relativno jednostavno, da menja način funkcionisanja samog kernela. Ovo je jedna od karakteristika otvorenih OS i glavna karakteristika koja je preporučila Linuks kernel, samim tim i Linuks OS, za primenu u *embedded* sistemima. Korisnik može direktno prilagoditi kernel konkretnom hardverskom uređaju, a onda, nakon pokretanja na odredišnom mikrokontroleru ili mikroračunaru, koristiti gotovo sve mogućnosti Linuks OS. Zbog toga se prilagođene verzije Linuks OS, sa adekvatno modifikovanim kernelom, u literaturi često sreću i pod imenom *Embedded Linux* [6].

B. Raspberry Pi 2 B

Raspberry Pi 2 je okruženje čii je glavni deo mikroračunar *Broadcom* BCM2709, podverzija BCM2836 sa ARM *Cortex* A7 jezgrom [7]. Ovo kolo ima 1 GB RAM memorije i može ostvariti radnu frekvenciju od čak 900 MHz. Sastavni delovi okruženja su i slot za *micro-SD* karticu, veliki broj GPIO (*General Purpose Input Output*) pinova, kao i brojni konektori za različite tipove komunikacija (USB, HDMI, *Ethernet* itd.). Radni napon okruženja je 5 V. Slika okruženja sa označenim sastavnim delovima prikazana je na slici 2. Centralno kolo *Broadcom* BCM2836 zasnovano je na ARM arhitekturi.



Slika 2. Okruženje *Raspberry Pi 2* na kome će biti implementiran Linuks OS.

ARM arhitektura, što znači *Advanced RISC Machine*, predstavlja tip moderne arhitekture, vrlo zastupljene u modernim mikrokontrolerima i mikroprocesorima. Glavna karakteristika ovih uređaja je ta da je na njima moguće izvršiti samo određeni ARM (redukovani) set instrukcija, ali su te instrukcije projektovane tako da mogu realizovati gotovo sve zadate komande. Ovakav koncept omogućio je unapređenje hardverskih realizacija ARM arhitekture pre svega u pogledu proširenja kapaciteta memorije, ali i unapređenju periferijskih jedinica, i samim tim ih približio karakteristima mikroračunara.

C. U-boot bootloader

Iako memorija mikroračunara na *Raspberry Pi* okruženju ima dovoljno kapaciteta da se na njoj pokrene Linuks, nije dovoljno samo prebaciti fajlove sa računara u memoriju BCM2836 da bi se izvršilo pokretanje OS. Kao i prilikom pokretanja OS na računarima, potrebno je prvo na ciljnom računaru pokrenuti odgovarajući program (*bootloader*), koji će omogućiti da se računar pokrene (*bootuje*) u odgovarajućem režimu koji će omogućiti dalji rad. Za ovu potrebu iskorišćen je alat koji se zove *U-boot* [8]. Ovaj alat je odabran pre svega jer ga je moguće koristiti sa više platforma (*cross-platform*). Ova

karakteristika omogućava da se korišćenjem *U-boot* alata mogu prebacivati razni OS preko raznih OS i to čak ne samo na mikroračunare, već i na druge uređaje.

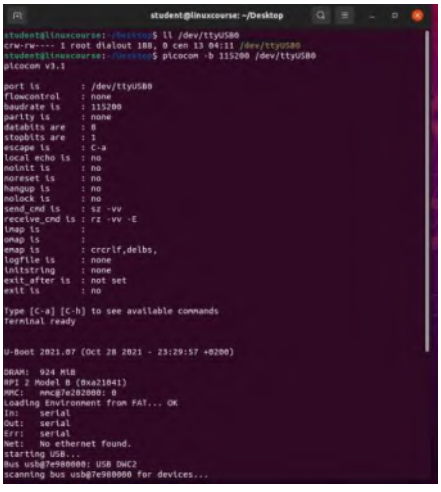
Uz korišćenje opisanih alata, podrazumeva se i korišćenje standardnih alata kao što su USB to UART adapter (u ovom slučaju korišćeno je FTDI kolo), kao i serijskog terminala (iskorišćen je *PicoCom*).

III. PROCES IMPLEMENTACIJE

Sam proces implementacije Linuks Ubuntu OS zahteva nekoliko koraka. Prvi korak je projektovanje odgovarajućeg kernela koji bi se prilagodio BCM2836 mikroračunaru. Ovo je jako komplikovan zadatak, pošto sam BCM2836 predstavlja jako složen mikroračunar. Ipak, zahvaljujući širokoj *open-source* Linuks zajedinici, kernel za ovaj rasprostranjeni mikroračunar već postoji i može se preuzeti sa zvanične *github* prezentacije *Raspberry* projekta [9].

Nakon preuzimanja, potrebno je napraviti odgovarajući *zImage* fajl, kao i kod pokretanja OS na drugim uređajima (na računarima ili telefonima). Ovo se ponovo radi pomoću eksternih softverskih alata, gde je konkretno iskorišćen alat *gparted*, dostupan na Linuks sistemima. U *zImage* fajl svakako treba implementirati i *bootloader U-boot*, koji će omogućiti opcije za pokretanje sistema. Kreirani *zImage* fajl se prebacuje na mikro SD karticu koja se povezuje na *Raspberry* (pošto na samom okruženju postoji slot za karticu) odakle će biti pokrenut prilagođeni kernel.

Nakon povezivanja (*mount*) SD kartice, potrebno je povezati razvojni uređaj (računar ili laptop) sa odredišnim uređajem (*Raspberry*) preko USB to UART kola na pinove 8 (TX) i 10 (RX) *Raspberry Pi* uređaja. Prilikom povezivanja naravno voditi računa o tome da se TX pin kola povezuje sa RX pinom odredišnog uređaja i obratno. Korišćenjem *PicoCom* serijskog terminala, može se videti prijem i slanje poruka preko serijskog porta.



Slika 3. Pokretanje *U-boot* alata na *Raspberry Pi 2* okruženju.

Kao što se može videti na slici 3, nakon prepoznavanja uređaja, pokreće se *bootloader U-boot*. Pritiskom tastera *Enter*, ulazi se u režim gde je moguće zadati komande. Pokretanjem navedenih komandi vrši se upis odgovarajućih fajlova sa SD kartice u internu memoriju odredišenog uređaja.

```
setenv bootcmd "tftp 0x01000000 zImage;
tftp 0x02000000 bcm2709-rpi-2-b.dtb;
bootz 0x01000000 - 0x02000000"
saveenv
```

Navedene linije će omogućiti da se nakon restarta pokrene sistem sa ovih memorijskih lokacija, i da se samim tim, prilikom narednog pokretanja pokrene i Linuks OS. Ovime je implementiran Linuks OS. Prema DTB (*Device Tree Binary*) fajlu koji je sastavni deo *zImagea*, kreiran je *root file* (korenski Linuks fajl), sa pratećim datotekama. U okviru ovog procesa, najčešće se vrši i instalacija *tftp* paketa koji omogućavaju povezivanje razvojne i odredišne mašine preko LAN kabla korišćenjem *Ethernet* protokola (umrežavanje računara). Ovo omogućava brz i jednostavan prenos podataka sa razvojne na odredišnu mašinu (brzina protoka podatka korišćenjem *Ethernet* protokola (100 MBps) je višestruko veća nego brzina protoka podatka korišćenjem UART komunikacije (1 MBps), što kod prenosa većih fajlova može napraviti značajnu razliku). Ukoliko se izvrše odgovarajuća podešavanja, *root* fajlu Linuks OS pokrenutog na odredišnoj platformi moguće je pristupiti iz *nfsroot* foldera, koji se nalazi kao deljeni folder u *root* folderu razvojne mašine.

IV. PRIMER RADA IMPLEMENTIRANOG SISTEMA

Glavna svrha celokupnog procesa implementacije je ta da se celokupan razvoj softverkog programa (osnovnog koda kao i pisanje i/ili preuzimanje i implementacija biblioteka), vrši na razvojnoj mašini, gde se može testirati i prilagođavati, a da se izvršavanje programa obavlja na odredišnoj platformi. Algoritam daljeg razvoja aplikacija koje će izvršavati na *Raspberry Pi* okruženju nije komplikovan, pre svega zbog postojanja odgovarajućih *cross-comile* alata. U nastavku je dat jedan vrlo jednostavan primer koda koji će preko serijskog terminala ispisivati poruku *Zdravo sa Raspberryja*.

```
#include<stdio.h>
main(){
   printf("Zdravo sa Raspberryja\n");
   return(0);
}
```

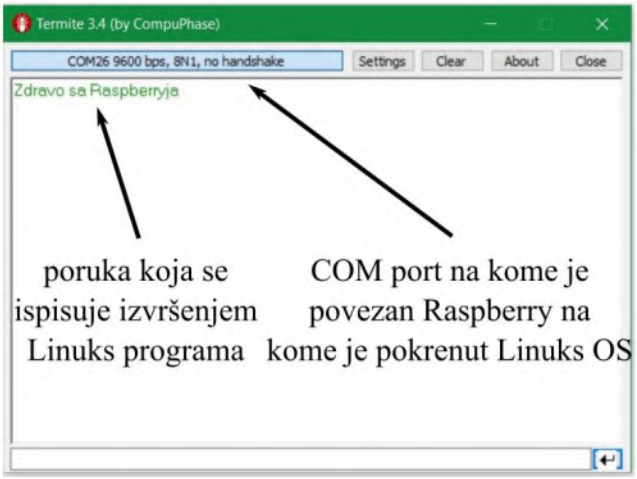
Program je u potpunosti razvijen preko razvojnog računara sa Linuks sistemom i sačuvan kao fajl *test.c*. Pisanjem odgovarajućeg *Makefilea* čiji je ključni deo pokretanje komande *gcc -o test test.c* vrši se kompajliranje i izvršavanje tog fajla pa se preko serijskog terminala ispisuje odgovarajuća poruka. Za izvršavanje ovog koda na odredišnoj *Raspberry* platformi, potrebno je prvo iskopirati

sve potrebne fajlove (u ovom slučaju samo jedan) u *nfsroot* direktorijum, ili u neki njegov poddirektorijum (za ovo je naravno potrebno da razvojna i odredišna mašina budu umrežene i povezane preko LAN kabla).

Nakon toga fajl je podvrgnut *cross-compiling* procesu (procesu unakrsnog kompajliranja). Izvršeno je kompajliranje i bildovanje identičnog programa kao i u prvom slučaju, ali ne za izvršavanje na razvojnoj mašini (računaru), već na odredišnoj mašini (*Raspberry*). Ovo se postiže dodavanjem sledećih komandi u *Makefile*:

```
export ARCH=arm
export CROSS_COMPILE=arm-linux-gnueabihf
make bcm2709_defconfig
make
```

Pokretanjem *Makefilea* opisane sadržine vrši se pokretanje istog tog programa, ali sa Linuks OS koji je pokrenut na okruženju *Raspberry Pi*. Identično ispisivanje, ali ovoga puta sa ulaznog porta može se videti na slici 4. Za prikaz poruke iskorišćen je softverski alat *Termite*, alat za softversku komunikaciju koji je dostupan samo preko *Windows* OS, kako bi se pokazalo da je serijska komunikacija moguća i na drugim OS.



Slika 4. Ispivanje poruke preko COM porta koji je povezan sa *Raspberry Pi* okruženjem sa pokrenutim Linuks OS.

Na prvi pogled, ovakav tip razvoja jako je sličan i bilo kom drugom načinu razvoja programa za mikrokontrolere. Ipak, ono što je drugačije je (pored fleksibilnosti i standardizacije *open-source* biblioteka) je i mogućnost debagiranja i „simulacije“ bez „spuštanja“ koda na čip. Najjednostavnije rečeno, ukoliko razvijeni program radi prema očekivanjima na razvojnoj platformi, radiće i na odredišnoj, jer platforme koriste identične OS.

V. Zaključak

U ovom radu prikazana je mogućnost pokretanja Linuks operativnog sistema na okruženju *Raspberry Pi*. Osnovni opisi alata ukazuju na to da pokretanje ovog sistema na moćnoj platformi kao što je *Raspberry Pi* nije komplikovan zadatak. Mogućnosti *cross-compilinga*, kao i dostupnost velikog broja *open-source* biblioteka kompatibilnim sa Linuks sistemima čine ovakav način projektovanja sistema odgovarajućim za zahtevne, komercijalne *embedded* aplikacije.

Zahvalnica

Autori rada se zahvaljuju koleginici Sandri Veljković na ustupanju hardverske opreme za realizaciju projekta kao i asistentima Katedre za mikroelektroniku na podršci pri realizaciji projekta.

Literatura

[1] Mikroprocesorska elektronika, predavanja, slajdovi. Dostupno na: https://www.elektronika.ftn.uns.ac.rs/mikroprocesorska-elektronika-mehatronika/wp-content/uploads/sites/ (pristupljeno 09.10.2022.)
[2] Tehnička specifikacija za kolo STM32F103. Dostupno na: https://www.st.com/resource/en/datasheet/stm32f103c8.pdf (pristupljeno 09.10.2022.).
[3] I. Odun-Ayo et al., “Comparative Study of Operating System Quality Attributes”, *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 107, p. 012061, 2021.
[4] V. Vasiljević et al., *Mikroračunari*, Viser, 2009.
[5] M. Garrels, *Introduction to Linux*, Garrels.be, 2008.
[6] K. Yaghmour et al., *Building Embedded Linux Systems*, O’Reilly, 2008.
[7] Tehnička dokumentacija za Raspberry Pi 2 B. Dostupno na: https://cdn-shop.adafruit.com/pdfs/raspberrypi2modelb.pdf (pristupljeno 09.10.2022.).
[8] *U-boot bootloader*. Dostupno na: https://github.com/u-boot (pristupljeno 09.10.2022.).
[9] Zvanična stranica github projekta. Dostupno na: https://github.com/raspberrypi/linux (pristupljeno 09.10.2022.).

Projektovanje veb aplikacije za grafički prikaz podataka u IoT sistemima

Milan Đorđević, Đorđe Mitrović, Nikola Mitrović

Sadržaj − U ovom radu je prikazan razvoj jednostavne veb aplikacije, koja kao deo *Internet of Things* sistema, u realnom vremenu vrši prikaz primljenih podataka. Prikazan je postupak projektovanja aplikacije, njen princip rada i detaljno su objašnjeni korišćeni alati, kao i sadržina ključnih fajlova za projektovanje. Nakon projektovanja, odrađena su osnovna testiranja, kako bi se potvrdio rad projektovane aplikacije.

I. UVOD

Akvizicija što većeg broja podataka iz okruženja predstavlja jednu od modernih tendecija savremene tehnologije. Trendovi idu ka tome da se omogući što veći priliv podataka, a da se istovremeno razvijaju tehnologije za adekvatan prikaz merenih, odnosno dobijenih podataka. Neretko, sastavni deo ovakvih tipova sistema su i aplikacije koje omogućavaju prikaz podataka, ili dela podataka, u realnom vremenu. Ovakve aplikacije najširu primenu našle su u *Internet of Things* (IoT) sistemima [1].

IoT sistemi su sistemi koji obuhvataju dvosmernu bežičnu komunikaciju između odgovarajućih hardverskih kola i serverske veb aplikacije. Odgovarajuća hardverska kola, u najosnovnijem smislu, podrazumevaju kola ili module koji mogu da dobiju određenu informaciju iz spoljašnjeg sveta (da izmere ili utvrde promenu koncentracije gasova, doze radijacije, jačine svetlosti ili temperature). Izmerena vrednost koja postaje podatak se onda, preko odgovarajućih kola, korišćenjem različitih tipova bežične komunikacije (*bluetooth*, Wi-Fi, GSM komunikacije i druge) prosleđuje veb aplikaciji koja je pokrenuta na serveru. Aplikacija vrši prijem podatka, i u zavisnosti od konkretne primene vrši njegovu dalju obradu i/ili prikaz.

U zavisnosti od konkretne primene, broja hardverskih kola, potrebne brzine prenosa i prikaza podataka, razlikuju se osnovni pristupi projektovanju potrebnih veb aplikacija. U ovom radu biće dat prikaz projektovanja veb aplikacije kod koje se primljeni podaci prikazuju odmah nakon prijema (u realnom vremenu). Rad će se skoncentrisati na slanje podataka korišćenjem Wi-Fi komunikacije, iako bi razmatranja za druge tipove bežičnih komunikacija bila jako slična.

M. Đorđević je student na Akademiji Tehničko Vaspitačkih Strukovnih Studija – Odsek Niš, Aleksandra Medvedeva 20, 18000 Niš, Srbija, E-mail: milan94srb@gmail.com.

Đ. Mitrović i N. Mitrović su studenti Elektronskog fakulteta, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija.

II. IZBOR METODOLOGIJE

Da bi se prenos podataka omogućio što brže, sa tendencijom da to bude u realnom vremenu, potrebno je odabrati odgovarajuću metodu za slanje podataka. Metodu bi bilo potrebno implementirati i sa strane hardverskog kola zaduženog za slanje prikupljenog podatka i sa strane veb aplikacije zadužene za prijem tog podatka. Samim tim, selekcija metode najčešće je ograničena mogućnostima hardverskih kola. Veb aplikacije mogu da koriste robusne kapacitete serverskih performansi koji su najčešće vrlo fleksibilni, dok mikrokontroleri koji su u središtu modula za bežičnu komunikaciju imaju ograničene performanse i po pitanju memorijske dostupnosti i po pitanju brzine. Ipak, razvojem modula za Wi-Fi komunikaciju, pre svega modula kompanije *Espressif*, omogućena je značajna fleksibilnost i sa strane hardvera [2]. Ova fleksibilnost, pre svega u pogledu memorije, omogućila je da se firmver (*firmware* - kod za hardverski modul) može pisati istom sintaksom kao i kod veb aplikacija, a da se odgovarajućim *bootloader* alatima može bez grešaka prevesti u potreban mašinski kod.

Jedan od najzastupljenijih programskih jezika koji se koristi za pisanje i projektovanje veb aplikacija je *JavaScript*. Projekcija je da oko 98 % aplikacija koje se pokreću iz internet pretraživača, u potpunosti, ili na neki način sadrži delove koji su pisani korišćenjem *JavaScript* jezika [3]. Inicijalno je zasnovan na C++ jeziku, iako su neki principi rada slični sintaksama nekih drugih programskih jezika. Kako se za programiranje naprednijih *Espressif* modula takođe koriste objektno orijentisani jezici, povećanjem dostupne memorije javila se ideja da se, pogotovo za primenu u IoT sistemima, gde je potrebno povezivanje sa odgovarajućim veb aplikacijama, programiranje, odnosno projektovanje vrši na nekom prilagođenijem programskom jeziku. Ova inicijativa dovela je do nastajanja *Espruino* platforme [4].

Korišćenjem *Espruino* platforme i *bootloadera* koji nudi ova platforma, kao viši programski jezik za programiranje naprednijih *Espressif* modula moguće je koristiti *JavaScript*. Naravno, sintaksa i komande unete na ovaj način će ponovo morati da se prevedu u odgovarajući objektno orijentisani program, koji će se, već od ranije dostupnim metodama prevesti u odgovarajući mašinski kod. Na prvi pogled, može se dovesti u pitanje uvođenje dodatnog nivoa prevođenja u radu, jer će resursi potrebni za dodatno prevođenje zauzeti više memorije i više opteretiti modul sa stanovišta potrošnje. Zbog toga, za

mnoge, pogotovo jednostavnije primene, se ovakav pristup i ne preporučuje. Ipak, preporučuje se u onim primenama gde je potrebno napraviti vezu, odnosno uspostaviti komunikaciju između korišćenog modula i potrebne veb aplikacije. Glavna prednost se u tom scenariju pre svega ogleda u jednakoj interpretaciji tipova, dužine i količine podataka svih elemenata koji učestvuju u kreiranoj mreži za protok podataka. Baš ovaj detalj predstavlja kritičnu tačku za brzinu i pouzdanost komunikacije. To je i glavni razlog zbog koga se ovakav pristup koristi prilikom projektovanja aplikacija i sistem gde je glavni zahtev sistem brzina komunikacije, a najčešći primer tih sistema su *real time* sistemi za prikaz podataka.

III. REALIZACIJA APLIKACIJE

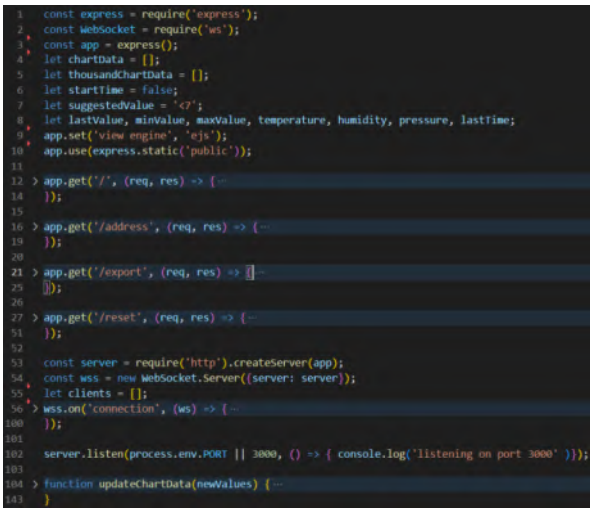
Projektovana veb aplikacija pisana je, u najvećem delu, korišćenjem *node.js* platforme koja se bazira na *JavaScript* jeziku [5]. Aplikacija se pokreće na serveru unutar klijent-server konfiguracije, gde su klijenti hardverski uređaji koji vrše merenje i šalju podatke kao i pretraživači (*browsers*) u kojima se prikazuju rezultati merenja. Aplikacija vrši prijem i obradu podataka, nakon čega su oni vidljivi svim klijentima kroz grafički prikaz unutar pretraživača. Stablo podataka projektovane aplikacije prikazano je na slici 1.



Slika 1. Stablo podataka projektovane aplikacije.

Svi podaci potrebni za projektovanje aplikacije mogu se podeliti u dve grupe. Prva su podaci templejta, odnosno fajlovi i folderi neophodni da bi projektovana aplikacija mogla da se razvija lokalno, da se kasnije sistemski menja i slično. Ovi fajlovi su nezavisni od same konkretne primene i predstavljaju deo svake veb aplikacije ovog tipa. To su fajlovi *package.json* i *package-lock.json*. Fajl *package.json* nastaje pokretanjem komande *npm init -y* iz terminala, pod pretpostavkom naravno da je instaliran alat *node.js*. Sadrži osnovne karakteristike aplikacije a mogu mu se dodavati *dependencies*, odnosno moduli koji će biti korišćeni u procesu razvoja aplikacije. Pokretanjem komande *npm install* kreira se folder *node_modules* koji sadrži potrebne fajlove za rad aplikacije i fajl *package-lock.json* koji će verzije svih upotrebivih paketa „zaključati“ na standardne kako bi se izbegla desinhronizacija aplikacije i paketa koje aplikacija koristi.

Druga grupa podataka iz stabla aplikacije su podaci koji su striktno vezani za konkretnu primenu aplikacije. To su fajlovi *server.js* i fajlovi unutar datoteka *public*, odnosno *views*. Glavni deo aplikacije nalazi se u fajlu *server.js*. Kao što se može videti po ekstenziji, ovo je fajl koji je pisan korišćenjem *JavaScript* jezika. Sadržina ovog fajla sa skupljenim zagradama data je na slici 2.

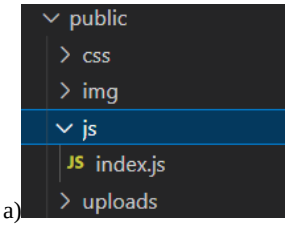


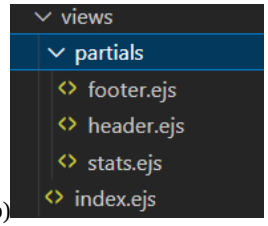
Slika 2. Sadržina fajla *server.js*.

Da bi se prikazala sadržina fajla *server.js* na jednoj slici, iskorišćena je mogućnost softverskog alata *Visual Studio Code* [6] koji je iskorišćen za razvoja aplikacije, da telo funkcija bude uvučeno (skupljene zagrade) kako bi čitav kod (143 linije) stao u dosta manje linija. Unutar ovog fajla opisan je prijem i obrada podatka od klijenta (funkcija *wss.on*), kao i ažuriranje podataka za grafički prikaz (funkcija *updateChartData*).

Aplikacija je projektovana tako da je kao protokol za prenos podatka sa hardverskog dela do aplikacije iskorišćen *WebSocket* protokol, iako su razmatrani i *polling* i *long-polling* metodi. Ovaj protokol je odabran jer omogućava veću brzinu u odnosu na druge razmatrane protokole, iako u datom trenutku ograničava broj trenutno aktivnih klijenata u klijent-server konfiguraciji koji vrše slanje na jedan. Baš ova metoda omogućava da se podaci ustupe na obradu odmah nakon prijema, samim tim, da se, ukoliko brzina prijema podataka nije prevelika, podaci prikazuju u realnom vremenu.

Fajlovi unutar datoteka *public*, odnosno *views*, zaduženi su za prikaz obrađenih podataka. Sadržina foldera public obuhvata potrebne fajlove za pokretanje stranice klijenta unutar pretraživača (*browsera*), dok sadržina foldera *views* obuhvata fajlove za prikaz stranice korišćenjem *ejs* modula [5].

a)


b)


Slika 3. Pregled podataka foldera *public* i *views*.

Kao što se može videti sa slike 3, unutar foldera *public* nalaze se drugi podfolderi koji sadrže elemente koji će biti prikazani korisniku. U podfolderu *css* nalaze se fajlovi sa stilovima prikaza, u folderu img slike koje će biti deo interfejsa, u folderu uploads ostali medijalni delovi interfejsa, a glavni deo ovog folder nalazi se u podfolderu js, čija je sadržina fajl *index.js*. Unutar ovog fajla aktiviran je modul *Chart.js* [7], koji predstavlja iskorišćeni modul za grafičku prezentaciju primljenih podataka. Sadržina fajla *index.js* prikazana je na slici 4.

```
1   const updateButton = document.querySelector('#update-button');
2   const dateValueElement = document.querySelector('#pc-date-value');
3   const timeValueElement = document.querySelector('#pc-time-value');
4   const minValueElement = document.querySelector('#pc-min-value');
5   const maxValueElement = document.querySelector('#pc-max-value');
6   const suggestedValueElement = document.querySelector('#pc-suggested-value');
7   const temperatureElement = document.querySelector('#pc-temperature-value');
8   const humidityElement = document.querySelector('#pc-humidity-value');
9   const pressureElement = document.querySelector('#pc-pressure-value');
10  const exportButton = document.querySelector('#export-button');
11  const currentDate = new Date();
12  dateValueElement.textContent = currentDate.toLocaleDateString();
13  timeValueElement.textContent = currentDate.toLocaleTimeString();
14  let chartLabels = [];
15
16  for(let i=0; i<1000; i++){···
18  }
19
20  var ctx = document.getElementById('pollution-chart').getContext('2d');
21  var pollutionChart = new Chart(ctx, {···
65  });
66
67  const socket = new WebSocket(location.origin.replace(/^http/, 'ws'));
68  socket.addEventListener('message', (event) => {···
90  });
91
92  updateButton.addEventListener('click', () => {···
95  });
96  exportButton.addEventListener('click', () => {···
114 });
115 async function updateFromCSV() {···
129 }
130 updateFromCSV();
```

Slika 4. Sadržina fajla index.js.

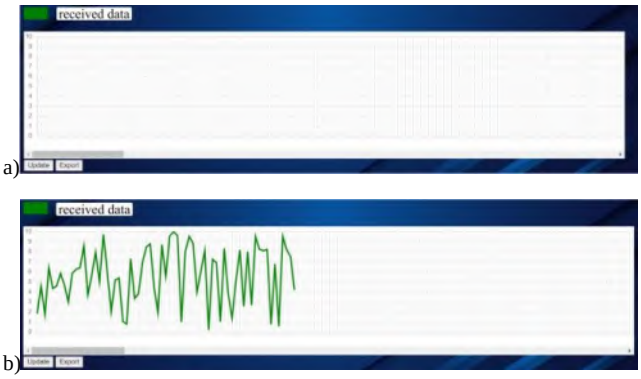
Kao što je ranije navedeno, ovaj fajl predstavlja glavni fajl potreban da se u okviru pretraživača pokrene projektovana aplikacija. Zbog toga, u njemu su opisana sva polja dostupna korisniku kroz pretraživač. Glavno polje koje na taj način dostupno korisniku dato je kroz funkciju *Chart* (linija 21 na slici 4). Telo ove funkcije prikazano je na slici 5.

Korišćenje navedenih alata kroz modul *Chart.js* omogućava kreiranje više različitih nizova podataka koji se prikazuju na grafiku (*datasets*). Moguće je podešavati boju, simbol, veličinu i druge osnvne vizuelne elemente grafičkog prikaza. Opcijama u okviru polja *options* (linija 36) moguće je podesiti *responsivness* (opcija da se veličina grafika prilagođava veličini ekrana uređaja na kome se prikazuje), animaciju prilikom prijema podatka, aktivaranje legende i slično. Poljem *scales* unutar polja *options*, moguće je detaljno podešavati ose, njihovu poziciju na ekranu, maksimalne i minimalne vrednosti broj podeoka i dužinu polja. Pored ovoga, postoje i drugi elementi modula *Chart.js* koji se mogu koristiti, ali zahtevaju određeno vreme za realizaciju. Kako je cilj da se podaci prikazuju odmah nakon prijema, iskorišćeni su samo osnovni elementi za prikaz ovog modula. Takođe, pored samog grafičkog prikaza, korisniku je omogućen i izvoz primljenh podataka, zajedno sa vremenom prijema u CSV formatu.

```
21  var pollutionChart = new Chart(ctx, {
22      type: 'line',
23      data: {
24          labels: chartLabels,
25          datasets: [···
34          ]
35      },
36      options: {
37          responsive: true,
38          maintainAspectRatio: false,
39          normalized: true,
40          animation: false,
41          plugins: {
42              legend: {
43                  display: false
44              }
45          },
46          scales: {
47              x: {
48                  minRotation: 0,
49                  maxRotation: 0,
50                  ticks: {
51                      sampleSize: 120
52                  }
53              },
54              y: {
55                  beginAtZero: true,
56                  max: 10,
57                  steps: 9,
58                  stepValue: 1,
59                  ticks: {
60                      sampleSize: 10
61                  }
62              }
```

Slika 5. Telo funkcije *Chart*.

Unutar foldera *views* nalaze se podaci zaduženi za pojedine grafičke elemente koje korisnik vidi iz pretraživača (*header*, *footer* i slično), objedinjeni *ejs* templejtom. Ovi elementi nisu neophodni, moguće je koristiti standardni *html* prikaz, ali korišćenje ovog templejta omogućava bolje struktuiranje podataka, što je od velike važnosti prilikom razvoja kompleksnijih projekata. Interfejs projektovane veb aplikacije iz pretraživača prikazan je na slici 6.



Slika 6. Izgled interfejsa projektovane veb aplikacije: a) bez prijema podataka; b) sa prijemom podataka.

Kao što se može videti, centralno mesto na interfejsu zauzima koordinatni sistem u kome se na x-osi nalazi vreme a na y-osi nalazi vrednost primljenog podatka. Bitno je napomenuti i da je u projektovanu aplikaciju jednostavno

implementirati, između ostalog, ispisivanje minimalne i maksimalne primljene vrednosti u određenom vremenskom interval, kao i funkcionalnost automatskog pomeranja usled primljenih novih vrednosti kada je prikaz grafika popunjen (*autoscroll*).

IV. Zaključak

U radu je prikazano projektovanje jedne veb aplikacije koja bi, kao deo IoT sistema, omogućila prikaz primljenih podataka u realnom vremenu. Dalji planovi istraživanja podrazumevaju razvijanja različitih baferovanja, odnosno skladištenja podataka. Ova funkcionalnost veb aplikacija bi bila korisna u slučajevima kada je učestanost primanja podataka jako velika. U tim slučajevima realna brzina prijema podataka je jako velika, pa nije moguće adekvatno očitavanje i analiza tih podataka. Baferovanje bi omogućilo korisnicima aplikacije da mogu da vide primljene podatke sporije nego što stižu, što bi možda omogućilo bolju analizu primljenih podataka, gde bi korisnik preko interfejsa mogao da bira odnos brzine prikaza u odnosu na realnu.

Literatura

[1] B. Radenković, M. Despotović-Zrakić, Z. Bogdanović, D. Barać, A. Labus i Ž. Bojović, *Internet inteligentnih uređaja*, Fakultet organizacionih nauka, 2017.

[2] Esspresif, ESP32 Series datasheets, 2022. Dostupno na: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf (pristupljeno 09.10.2022.)

[3] W3Techs, Usage statistics of JavaScript as client-side programming language on websites, 2022. Dostupno na: https://w3techs.com/technologies/details/cp-javascript/ (pristupljeno 09.10.2022.)

[4] Softverski alat *Espruino*. Dostupno na: https://www.espruino.com/Download (pristupljeno 09.10.2022.)

[5] Okruženje *node.js*. Dostupno na linku: https://nodejs.org/en, (pristupljeno 09.10.2022.)

[6] Softverski alat *Visual Studio Code*. Dostupno na: https://code.visualstudio.com/download (pristupljeno 09.10.2022.)

[7] Modul Chart.js. Dostupno na: https://www.chartjs.org/ (pristupljeno 09.10.2022).

Sistem za popunjavanje baze podataka veštački generisanim podacima

Milan Danković

Sadržaj – U ovom radu prikazan je novi sistem za popunjavanje baze podataka veštački generisanim podacima. Najpre je dat pregled i analiza postojećih rešenja, zatim je opisan razvijeni sistem i na kraju je demonstrirana njegova primena.

I. UVOD

Prilikom razvoja aplikacije, testiranje je neizostavni deo procesa. Kod implementiranog softverskog rešenja se može testirati sama ispravnost rešenja, kao i to kako se ono ponaša prilikom velikog opterećenja. Oba pomenuta slučaja testiranja imaju koristi od velikog broja podataka koji se mogu pregledati, ali se automatski generišu. Ručno umetanje nekoliko redova u svaku od tabela baze podataka jednostavno nije dovoljno dobro. Imati dovoljnu količinu ispravnih podatka može biti otežavajuća okolnost prilikom testiranja aplikacija. Ne samo da su potrebni različiti skupovi podataka kako bi bilo moguće testiranje različitih scenarija, već je potrebno obezbediti kombinacije koje mogu biti od suštinskog značaja za hvatanje grešaka pre nego što rešenje ode u proizvodno okruženje. U ovom radu prikazan je jedan pristup rešavanju ovog problema.

Povećanjem potrebe za velikom količinom veštački generisanih podataka, pojavila su se brojna rešenja koja programerima značajno olakšavaju postupak njihovog generisanja. Većina dostupnih rešenja poseduje određeni nivo automatizacije, kao i prikaz generisanih podataka u različitim formatima koji predstavljaju industrijski standard. Dostupan nivo automatizacije omogućava generisanje podataka na nivou jedne tabele, međutim poželjno je omogućiti automatsko generisanje podataka za sve tabele jedne baze podataka, ali tako da budu ispoštovana sva ograničenja definisana nad podacima te baze.

Bilo je potrebno razviti sistem koji podržava jednostavno i intuitivno generisanje modela baze podataka poput popularnih rešenja koja se koriste u industriji. Pored ovoga, sistem treba da omogući definisanje različitih tipova ograničenja podržanih u strukturiranom upitnom jeziku, kao i definisanje veza između odabranih kolona tabela generisane baze podataka. Dodatno, sistem bi trebalo da generiše ne samo formalno ispravne podatke već i semantički smislene podatke koji omogućavaju pogled na stanje baze podataka prilikom realne upotrebe. Uz sve već navedene stavke, potrebno je omogućiti i jednostavno proširenje razvijenog sistema tako da radi sa različitim tipovima baza podataka.

M. Danković, završio osnovne studije na Elektrotehničkom fakultetu u Beogradu, Bul. Kralja Aleksandra 73, Beograd, Srbija. Student prve godine master studija na Polytechnic University of Milan, Milan, Italy, E-mail: milan.dankovic@mail.polimi.it.

II. PREGLED I ANALIZA POSTOJEĆIH REŠENJA

U ovom poglavlju su ukratko predstavljena neka od postojećih rešenja opisanog problema, njihove prednosti i mane. Takođe su predstavljeni i zahtevi koje rešenje mora zadovoljavati kao i koje od tih zahteva postojeća rešenja podržavaju, a koje bi bilo potrebno dodati kako bi se napravio jedan kompletan sistem namenjen popunjavanju baze podataka veštački generisanim podacima. Mnogi komercijalni alati pružaju složene funkcije baze podataka, za razliku od alata otvorenog koda u ovoj oblasti koji često nemaju dovoljan nivo automatizacije i ne podržavaju sve funkcionalnosti koje bi jedan ovakav sistem trebalo da podržava. Opisana rešenja i njihovi slučajevi korišćenja bili su inspiracija za razvoj rešenja prikazanog u ovom radu.

A. DATAPROF

DATAPROF [1] sistem nudi više od 20 generatora sintetičkih testnih podataka koji se mogu koristiti za zamenu podataka osetljivih na privatnost kao što su imena, kompanije, IBAN, brojevi socijalnog osiguranja itd. Ovaj sistem podržava sve glavne tehnologije baze podataka i ima mogućnost generisanja podataka na više različitih jezika. Mana ovog sistema je ta što postoji podrška isključivo za Windows operativni sistem.

B. IRI RowGen

IRI RowGen generator podataka je prvi put objavljen 2004. godine [2]. Podržava mnoge baze podataka i ciljne formate datoteka u više operativnih sistema. Takođe nudi zaštitu privatnosti podataka koristeći sintezu podataka, maskiranje, mapiranje i/ili transformaciju osetljivih podataka polje po polje. Sistem pruža podršku za očuvavanje referencijalnog integriteta poštujući odnose između primarnih i stranih ključeva, kao i složene i samoreferencirajuće ključeve. Performanse ovog sistema su bez premca u generisanju ogromnih količina podataka za testiranje. Međutim, zbog svih osobina koje ovaj proizvod nudi, on nije besplatan i to predstavlja njegovu glavnu manu.

C. GENERATEDATA.COM

GENERATORDATA.COM je alat otvorenog koda, napisan u programskim jezicima PHP, Javascript i MySQL [3]. Alat je intuitivan i jednostavan za korišćenje, ali ne podržava složene odnose između entiteta u bazi podataka. Alat omogućava odabir formata izlaznih podataka.

D. Analiza postojećih rešenja

Pomenuta rešenja predstavljaju gotove softverske proizvode sa više ili manje slučajeva korišćenja koje podržavaju. Za svako od rešenja predstavljeni su njegovi najznačajniji delovi. Pregledom već razvijenih sistema i pažljivom analizom slučajeva korišćenja koje oni podržavaju, proistekla je inspiracija za implementaciju rešenja opisanog u ovom radu. Odabrani su slučajevi korišćenja koji su od najvećeg značaja za korisnike koji žele da testiraju implementacije svojih projekata. Odabrani zahtevi koje bi sistem trebalo da ispunjava su:

• Definisanje kolona uz prosleđivanje njihovog imena i tipa kolone pomoću enumeratora podržanih tipova.
• Zadavanje ograničenja strukturiranog upitnog jezika (UNIQUE, MIN_MAX, MAX_LEN) generisanim kolonama.
• Zadavanje referencijalnog integriteta između kolona.
• Definisanje entiteta uz prosleđivanje imena i dodavanje odabranih kolona.
• Definisanje objekta koji simulira bazu podataka opisanu generisanim entitetima i njima dodeljenim kolonama sa zadatim ograničenjima.
• Mogućnost generisanja podataka za svaki entitet zasebno.
• Mogućnost automatskog popunjavanja baze podataka generisanim podacima uz poštovanje svih ograničenja.
• Mogućnost biranja formata izlaznih podataka.

U tabeli 1 dat je uporedni prikaz osobina pomenutih postojećih rešenja. Glavna stavka koju nijedan od pomenutih sistema ne podržava jeste mogućnost generisanja podataka programskim putem. Takođe, zaključak je da predstavljena komercijalna rešenja pružaju veliki broj funkcionalnosti, ali zbog toga nisu besplatna. Ovakav trend prisutan je i među ostalim sistemima ovog tipa koji ovde nisu pomenuti.

TABELA I
UPOREDNI PREGLED OSOBINA POSTOJEĆIH REŠENJA: A) DATAPROF, B) IRI ROWGEN, C) GENERATEDATA.COM

	A)	B)	C)
Generisanje semantički ispravnih podataka za jedan entitet	DA	DA	DA
Modelovanje baze podataka i definisanje ograničenja nad kolonama tabela	DA	DA	NE
Popunjavanje celokupne baze podataka uz poštovanje ograničenja	DA	DA	NE
Podrška za generisanje podataka na više jezika	DA	DA	DA
Generisanje neograničenog broja podataka odjednom	DA	DA	NE
Pristup funkcionalnostima iz programskog koda	NE	NE	NE
Besplatan	NE	NE	DA

III. OPIS RAZVIJENOG SISTEMA

Alat opisan u ovom radu razvijen je kao biblioteka u Python programskom jeziku. Cela biblioteka razvijena je u PyCharm integrisanom razvojnom okruženju uz upotrebu alata komandne linije Poetry za organizaciju verzija biblioteka korišćenih prilikom razvoja i alata GIT za verzionisanje izvornog koda. Razvoj rešenja pratio je sve moderne principe i konvencije razvoja softvera.

A. Korisnički zahtevi koje razvijeni sistem ispunjava

Prilikom razvoja sistema u obzir su uzeti i pažljivo razmotreni svi korisnički zahtevi koje bi jedan ovakav alat trebalo da ispunjava. Detaljnom analizom došlo se do skupa svih zahteva. Dodatno, uviđeno je da je jako često potrebno generisati podatke koji ispunjavaju dodatne uslove, pa su najbitniji od tih slučajeva korišćenja implementirani. U nastavku su prikazani i detaljno objašnjeni dodatni korisnički zahtevi koji su implementirani u odnosu na već pomenute:

• Generisanje numeričkih tipova podataka sa zadatom raspodelom (uniformna, normalna i Poasonova).
• Generisanje podataka nabrojivog tipa sa zadatim verovatnoćama.
• Generisanje podataka koji su semantički ispravni za modeliranu biznis logiku.

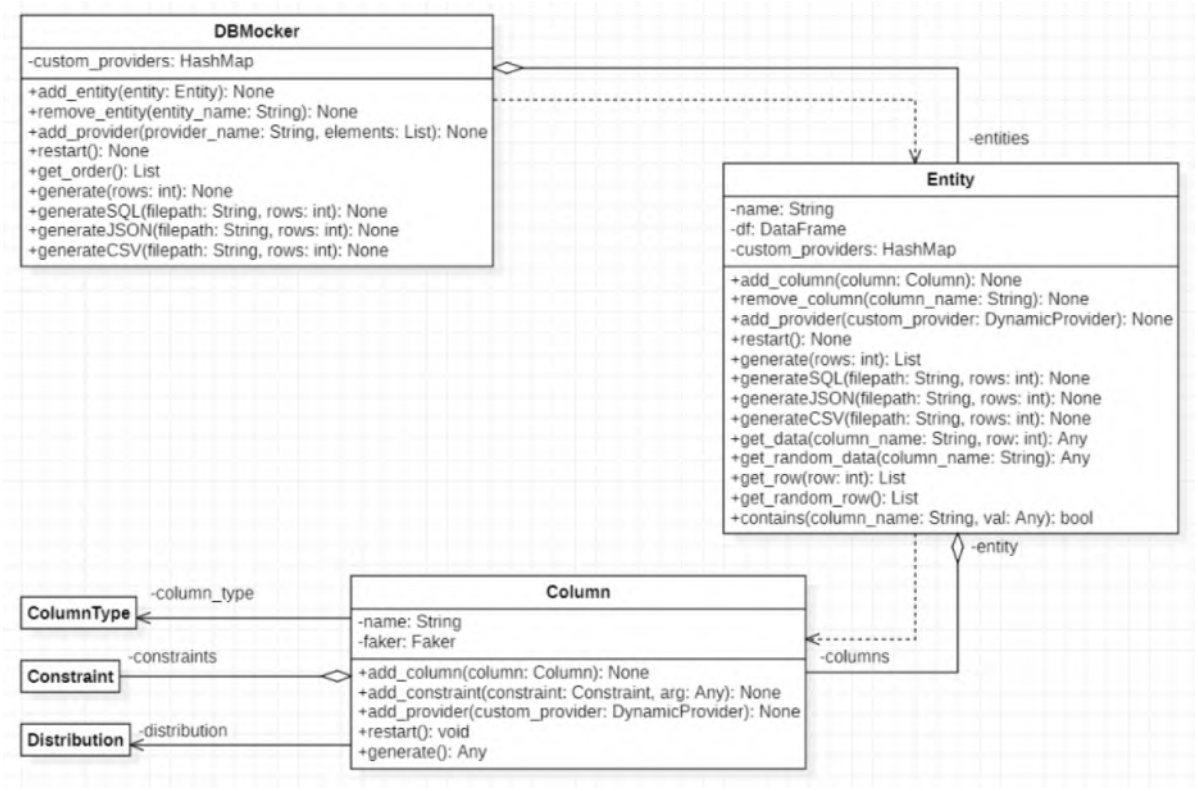
Generisanje numeričkih podataka sa zadatom raspodelom je važno, jer analitičari podataka obično žele da primene statističke testove na skup generisanih podataka, a mnogi statistički testovi dolaze sa osnovnim pretpostavkama o raspodeli podataka na koje se primenjuju. Osiguranje da će primenom odgovarajuće statističke metodologije doći do korektnih zaključaka zahteva ispravno generisane podatke sa zadatom raspodelom. Implementirane su neke od najpoznatijih raspodela, a omogućeno je i jednostavno proširenje sistema novim raspodelama.

Generisanje podataka nabrojivog tipa sa zadatim verovatnoćama omogućava korisnicima da naprave proizvoljnu diskretnu raspodelu nad proizvoljnim podacima. Ovo omogućava generisanje skupa podataka specifičnog za konkretan slučaj upotrebe i omogućava temeljno testiranje sistema generisanim podacima. Ovakav pristup je dobar ukoliko korisnik poseduje informacije o statističkim osobinama i raspodelom podataka koji će se naći u realnom sistemu.

Generisanje semantički korektnih podataka za modelovanu biznis logiku je važno kako bi korisnik imao realan pregled stanja sistema tokom razvoja ili testiranja. Takođe, ovo olakšava ispravljanje i detekciju problema u samoj logici aplikacije ukoliko to obavlja čovek jer radi sa čoveku čitljivim i smislenim podacima. Pažljivo generisani podaci koji najbliže moguće odgovaraju realnim podacima iz datog biznis domena omogućavaju približno tačno merenje performansi i ostalih parametara sistema.

B. Dizajn sistema

Sistem je projektovan u objektno-orijentisanom stilu programiranja. Dijagram klasa sistema sa najvažnijim detaljima prikazan je na slici 1. Sistem je dizajniran da omogući jednostavan i intuitivan interfejs. Važan faktor prilikom dizajniranja predstavljala je mogućnost jednostavnog proširenja sistema i prilagođavanja različitim tipovima baza podataka.



Slika 1. Dijagram klasa sistema.

B.1. Klasa DBMocker

Klasa DBMocker predstavlja glavnu klasu sistema koja simulira modelovanu bazu podataka. Klasa sadrži kolekciju objekata klase Entity iz koje tokom rada može uklanjati ili dodavati elemente. Nakon modelovanja baze podataka, korisnik može pozvati neku od metoda generate, generateSQL ili generateJSON, generateCSV uz prosleđivanje odgovarajućih parametara kako bi generisao podatke za popunjavanje modelovane baze podataka.

B.2. Klasa Entity

Klasa Entity modeluje entitet u bazi podataka i poseduje ime, kolekciju generisanih podataka, dodatno definisane generatore podataka i kolekciju objekata klase Column iz koje tokom rada može uklanjati ili dodavati elemente. Pozivom odgovarajućih metoda za generisanje podataka, uz prosleđivanje odgovarajućih parametara, korisnik može da izvrši generisanje podataka samo za odabrani entitet iz baze podataka. Klasa sadrži metode koje omogućavaju dohvatanje jednog podatka definisanog imenom kolone i rednim brojem reda, kao i dohvatanje liste podataka definisane rednim brojem reda. Uz ovo, postoji mogućnost dohvatanja nasumično odabranog podatka iz kolekcije generisanih podataka. Zbog potrebe za definisanjem referencijalnih integriteta i ostalih ograničenja upitnog jezika, postoji metoda *contains* koja proverava da li u kolekciji generisanih podataka zadata kolona sadrži zadati podatak.

B.3. Klasa Column

Klasa Column modeluje jedan atribut entiteta, kolonu u tabeli baze podataka, poseduje ime, podatak o objektu klase Entity kome je data kolona dodeljena i generator podataka koji je objekat klase Faker. Objekat klase Column poseduje i podatke o tipu kolone, zadatim ograničenjima i podatak o tipu raspodele pseudoslučajnih vrednosti, ako kolona sadrži podatke nekog numeričkog tipa.

B.4. Klasa ColumnType

Klasa ColumnType sadrži definisane konstante podržanih tipova podataka. Svaka konstanta ima njoj

dodeljeno ime koje se koristi za povezivanje sa odgovarajućim metodama iskorišćenog generatora podataka iz biblioteke *faker*. Proširenje dostupnih tipova vrši se jednostavno, dodavanjem nove vrednosti enumeratora u pomenutu klasu.

C. Generisanje podataka

Nakon modelovanja baze podataka kroz dostupan interfejs koji sistem pruža, sledeći korak predstavlja generisanje podataka. Korisnik biblioteke može generisati podatke zasebno za svaki entitet ili odjednom za sve entitete prisutne u modelovanoj bazi podataka. Ukoliko se podaci generišu za svaki entitet zasebno, potrebno je u obzir uzeti ograničenja definisana nad kolonama tabela baze podataka. Ukoliko postoji zavisnost između kolona dva ili više entiteta, neophodno je da korisnik generiše podatke odgovarajućim redosledom.

Zbog pomenutih nedostataka generisanja podataka zasebno za svaki entitet, kao i zbog povećanja same automatizacije procesa, razvijen je algoritam za automatsko popunjavanje modelovane baze podataka. Algoritam se zasniva na radu sa grafovima [4], koji se često koriste kao struktura podataka za predstavljanje odnosa između njenih elemenata. Da bi rad sa grafovskim algoritmima bio omogućen, neophodno je modelovanu bazu podataka apstrahovati i konvertovati u graf. Entiteti modelovane baze podataka predstavljaju čvorove grafa, a međusobna uslovljenost kolona između entiteta modelovana je usmerenim vezama između čvorova u dobijenom grafu. Opisani algoritam implementiran je i iskorišćen u opisanim metodama za generisanje podataka u klasi DBMocker. Pseudokod algoritma za konvertovanje baze podataka u graf uslovljenosti između njenih entiteta dat je u nastavku.

```
# inicijalizacija grafa
g = Graph(entities)
# dodavanje grana grafu
for entity in entities:
    for column in entity.columns:
       if column.depends_on is not None:
       g.addEdge(column.depends_on.entity.na
me, entity.name)
```

Čvorovi grafa identifikovani su imenom entiteta koje je jedinstveno u bazi. Neophodno je za svaki entitet iz baze podataka obići sve kolone koje taj entitet sadrži i proveriti da li je neka od tih kolona uslovljena nekom drugom kolonom. Ako takva relacija postoji, neophodno je dodati usmerenu granu od entiteta koji sadrži kolonu koja uslovljava trenutnu kolonu, ka entitetu koji se trenutno obilazi. Nakon kreiranja grafa koji predstavlja modelovanu bazu podataka, redosled entiteta za koje se generišu podaci utvrđen je topološkim poretkom dobijenog grafa.

D. Formati izlaznih podataka

Zbog velikog broja različitih primena koje razvijena biblioteka može imati, pored generisanja podataka za relacione baze podataka, jedan od zahteva prilikom analize slučajeva korišćenja bila je mogućnost generisanja izlaznih podataka u različitim formatima. Još jedna osobina koja izdvaja ovo rešenje jeste mogućnost pristupa generisanim podacima direktno iz programa, bez potrebe za nepotrebnim upisivanjem i čitanjem podataka iz fajla. Pored ovoga, omogućeno je generisanje SQL skripte za umetanje podataka u tabele, koja se može koristiti u svim komercijalnim ili alatima otvorenog koda koji podržavaju rad sa bazama podataka i strukturiranim upitnim jezikom. Dodatno omogućeno je generisanje podataka u JSON i CSV formatu. Ovi formati generisanih podataka odabrani su kao jedni od najkorišćenijih u industriji i samo su primer različitih mogućnosti razvijenog rešenja.

IV. PRIMER KORIŠĆENJA BIBLIOTEKE

U nastavku će biti prikazan primer korišćenja razvijene biblioteke. Biće prikazan biznis problem za koji će biti modelovan logički model podataka. Zatim će uslediti opis implementacije razvijenog modela korišćenjem rešenja opisanog u ovom radu, i način generisanja podataka.

A. Postavka problema

Razmatra se biznis model kompanije koja upravlja porudžbinama za veleprodaju proizvoda. Akteri u ovom biznis modelu su kupci i kompanija koja vrši prodaju. Potrebno je omogućiti osnovne očekivane funkcionalnosti koje poseduju komercijalni sistemi ove namene. Primer je pojednostavljen, kako akcenat ne bi bio na samom modelovanju baze, već na primeni razvijene biblioteke opisane u ovom radu. Postupak modelovanja baze prikazan je kako bi se prikazao uobičajeni tok rada prema kome je razvijena biblioteka dizajnirana.

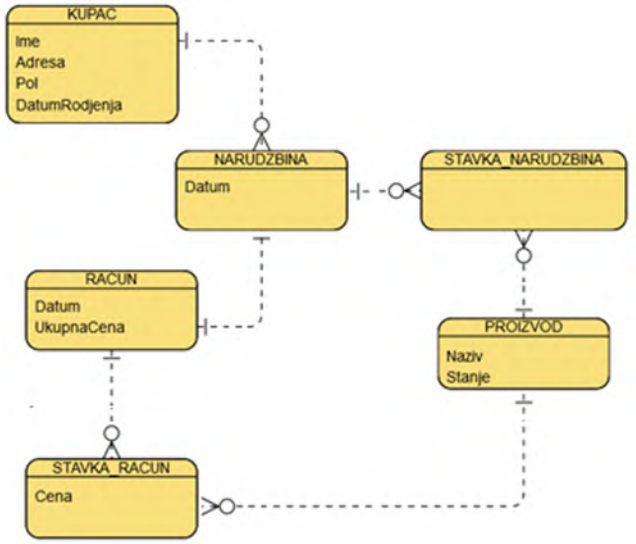
Kupci od kompanije mogu da poručuju proizvode, iako neki kupci možda još nisu poslali nijednu porudžbinu. Porudžbina može da sadrži više stavki, a svaka stavka porudžbine opisuje jedan proizvod. Mogu postojati proizvodi za koje ne postoji kreirana porudžbina. Nakon kreiranja porudžbine i prikupljanja proizvoda u porudžbinu, za tu porudžbinu će biti pripremljena faktura. Faktura ima mnogo stavki fakture, sa po jednim proizvodom u svakoj stavci fakture.

B. Model baze podataka opisanog problema

Na osnovu opisanog problema, može se dizajnirati logički model baze podataka koji je prikazan na slici 2.

Model sadrži 6 entiteta sa odgovarajućim atributima: KUPAC, NARUDZBINA, STAVKA_NARUDZBINA, PROIZVOD, RACUN i STAVKA_RACUN koji modeluju informacije iz realnog sveta. U modelu postoji 6 veza koje predstavljaju logičke odnose između entiteta i omogućavaju održavanje korektnih podataka i konzistentno stanje modelovane baze podataka. Ovako definisani model,

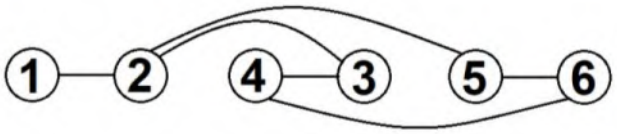
omogućava njegovu pravolinijsku implemementaciju uz korišćenje razvijenog sistema i generisanje skripti za popunjavanje baze podataka.



Slika 2. Logički model baze podataka modelovane na osnovu postavke problema.

C. Algoritam za generisanje podataka

U nastavku će biti opisan implementirani algoritam za pronalazak redosleda popunjavanja entiteta na opisanom primeru. Usled zavisnosti između tabela neophodno je utvrditi ispravan redosled po kome će podaci za odgovarajuće entitete biti generisani. Kako je prethodno opisano, model baze podataka može se predstaviti kao usmereni graf. Algoritam koji je iskorišćen u svrhu razrešavanja zavisnosti između entiteta je algoritam topološkog sortiranja [5]. Ukoliko modelovanim entitetima prikazanim u prethodnom pasusu dodelimo redom brojeve od 1 do 6 kao identifikatore, graf dobijen predstavljenim algoritmom za konverziju modela baze podataka u graf prikazan je na slici 3. Sve grane ovog grafa usmerene su u desno. Da bi bilo moguće generisati podatke za entitet sa identifikatorom *i*, neophodno je generisati podatke za sve entitete *k* koji imaju izlaznu granu do entiteta koji je označen sa *i*.



Slika 3. Apstrakcija modelovane baze podataka grafom.

Kao što se može primetiti na konkretnom primeru, topološko sortiranje grafa može imati više korektnih rešenja. Konkretan poredak zavisi od implementacionih detalja, ali je svaki korektan poredak odgovarajući za potrebe razvijenog sistema. Kada je ustanovljen redosled entiteta po kome će se vršiti generisanja podataka, po kreiranom redosledu za svaki entitet iz baze generišu se podaci redom za svaku kolonu. U zavisnosti od tipa kolone, proveravaju se moguća ograničenja definisana nad tim tipom podataka i generisanje se ponavlja sve dok podatak ne ispunjava sva zadata ograničenja. U zavisnosti od izbora formata izlaznih podataka, dobijeni podaci se smeštaju u odgovarajući fajl ili u radnu memoriju računara na kome se program izvršava. U nastavku je dat prikaz metode klase DBMocker koja koristi opisani algoritam i generiše poziva metode entiteta za generisanje podataka.

```
def generate(self, rows: int) -> None:
    self.restart()

    order = self.get_order()

    for entity_name in order:
        for entity in self._entities:
            if entity_name == entity.name:
                entity.generate(rows)
                break
```

Na samom početku vrši se resetovanje svih vrednosti kako bi bila održana konzistentnost i zadovoljena sva definisana ograničenja. Nakon toga pozivom metode get_order vrši se konverzija modelovane baze u graf i topološko sortiranje dobijenog grafa. Zatim se po tom poretku dohvata jedan po jedan entitet i vrši generisanje podataka.

V. ZAKLJUČAK

Razvojem sve većih informacionih sistema, porasla je potreba za generisanjem velike količine podataka za njihovo testiranje i merenje performansi poput odziva sistema, zauzeća resursa, itd. Kriterijumi koje podaci u ovakvom sistemu moraju da ispunjavaju su veoma strogi, pa je potrebno omogućiti generisanje sintetizovanih podataka koji zadovoljavaju sve ove uslove. Za sisteme ovog obima umetanje nekoliko redova u svaku od tabela baze podataka ne predstavlja zadovoljavajući način testiranja korektnosti same aplikacije kao i merenja njenih performansi. Pored pomenute primene, ovakvi sistemi mogu se koristiti za generisanje skupova podataka za školske primere baza podataka i ispitne zadatke. Tokom izrade ispita studenti često mogu da testiraju samo osnovne slučajeve malih dimenzija koji nisu dovoljno dobar pokazatelj korektnosti njihovog rešenja. Uz ovakav sistem, omogućeno bi bilo kompleksnije i obuhvatnije testiranje.

Od postojećih rešenja, predstavljeno rešenje problema razlikuje se po tome što omogućava programski pristup celom procesu generisanja podataka i može se koristiti uporedno sa izvornim kodom same aplikacije. Implementirane funkcionalnosti predstavljaju skup pažljivo odabranih, najkorisnijih funkcionalnosti za sisteme ovog tipa. Dizajn biblioteke omogućava njen jednostavan dalji razvoj i izradu unapređenih verzija.

Sistem je moguće unaprediti i dopuniti pisanjem omotača oko razvijene biblioteke u drugim programskim jezicima, kao i dodavanjem novih tipova podataka i dodatnih ograničenja koja se mogu definisati nad njima. Takođe, moguće je ostvariti napredak na polju generisanja podataka lokalizovanih za zadati korisnički jezik. Paralelno za razvojem biblioteka ove namene, moguće je ostvariti napredak u bibliotekama za generisanje semantički korektnih podataka upotrebom metoda mašinskog učenja.

ZAHVALNICA

Istraživanja prikazana u ovom radu sprovedena su tokom izrade mog diplomskog rada. Iz tih razloga želim da se zahvalim mentoru, prof. dr Milošu Cvetanoviću, na izuzetnoj fleksibilnosti oko odabira teme, sugestijama tokom pisanja diplomskog rada, kao i ostaloj pruženoj podršci tokom studija.

LITERATURA

[1] Test Data Simplified – DATPROF. Dostupno na: https://www.datprof.com/
[2] Innovative Routines International, „IRI RowGen – Test Data Software for Databases, Files, and Reports”. Dostupno na: https://www.iri.com/products/rowgen
[3] „Generate test data. Quickly”. Dostupno na: https://generatedata.com/
[4] Milo Tomasevic, „Algoritmi i strukture podataka”. Akademska misao, 2008.
[5] T. Cormer, C Leiserson, R. Rivest, C. Stein, „Introduction to Algorithms, Third Edition”, 1989.

Testiranje karakteristika Džonsonovog brojača na primeru kola 74HC4017

Nikola Janković, Stefan Janković

Sadržaj – U ovom radu prikazano je je testiranje Džonsonovog brojača na primeru kola 74HC4017. Kolo je fizički analizirano, a korišćenjem logičkog analizatora izvršena je provera funkcionalnosti.

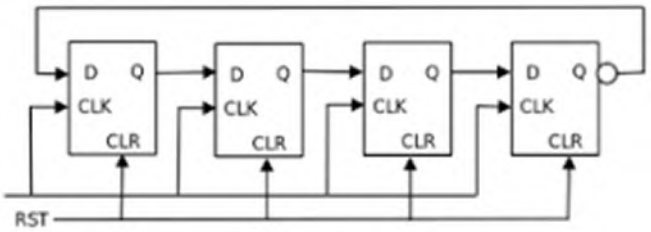
I. UVOD

Danas, gotovo svi automatizovani sistemi koriste neki računar za praćenje mnogobrojnih procesa koji se odvijaju, parametara mašina, proizvodnih podataka i tome slično. Naime, kako su nam potrebne informacije koliko trenutno ima proizvedenih komada nekog proizvoda, koliko se količine materijala utroši za proizvodnju tog istog prozivoda, brojači tu imaju značajnu primenu i zbog toga su veoma korisni i prisutni u industriji.

II. PRSTENASTI BROJAČI

Postoje dva tipa prstenastog brojača:

1. *Overbeck brojač* povezuje izlaz poslednjeg šiftovanog registra sa izlazom prvog šiftovannog registra i bit jedan ili nula cirkuliše oko prstena.(npr. četvororegistni sa inicijalnom vrednošću 1000, ima sledeću šemu ponavljanja:1000, 0100, 0010, 0001, 1000…);
2. *Upleteni prstenasti brojač* (Džonsonov brojač) povezuje komplemnt izlaza poslednjeg šiftovanog registra sa prvim šiftovanim registrom na ulazu i cirkuliše oko prstena sa nizom jedinica za kojima slede nule : 0000, 1000, 1100, 1110, 1111, 0111, 0011, 0001, 0000.



Slika 1. Šema prstenastog brojača.

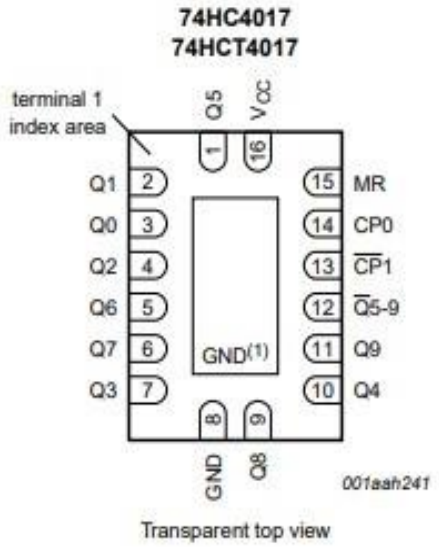
N. Janković i S. Janković su studenti Elektronskog fakulteta u Nišu, modul Elektronske komponente i mikrosistemi, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: stefancecijankovic@elfak.rs.

Registri se pokreću okidanjem signala na CLK (*Clock*) ulazu. Trenutno stanje na registru se menja, a izlaz (Q) trenutnog registra je vezan na ulaz (D) sledećeg registra. Tako se praktično informacija pomera.

Prstenasti brojač se koristi u hardverskom projektovanju (kao što su: ASIC I FPGA) za pravljenje konačnog automata. Binarnom sabiraču je potreban sabirač strujnog kola, koji je dosta komplikovaniji od već pomenutog prstenastog brojača. Najgora vrsta kašnjenja na kružnom sabiraču je proporcionalna broju bitova u kodu. Kašnjenje prstenastog brojača biće konstantno bez obzira na broj bitova u kodu. Mana prstenastih brojača je da mogu smanjiti gustinu koda. Binarni kod moze reprezentovati 2^N. stanja, gde je N broj bitova u kodu. (Overbeck brojač realizuje N stanja; Džonsonov brojač kao takav realizuje 2^N. stanja). Ovo zapravo znaci da registri mogu da budu skuplji od same kombinacione logike.

III. PRINCIP RADA DŽONSONOVOG BROJAČA

Džonsonov prstenasti brojač se sastoji od D registara. Unutar samog kućišta se nalazi 5 D registra.



Slika 2. Komponenta 7HC4017.

Pinovi Q0 do Q9 su izlazi, CP0 i /CP1 su klok ulazi, MR je pin za reset brojača, /Q5-9 izlaz se koristi za kaskadno vezivanje više brojača. Brojač menja stanje kada CP0 predje sa stanja logičke nule na jedinicu, gde je pri tome /CP1 u

stanju logičke nule, ili kada /CP1 promeni stanje sa logičke jedinice na nulu, gde je CP0 na logičkoj jedinici.

TABELA I
PRINCIP RADA PRSTENASTOG BROJAČA

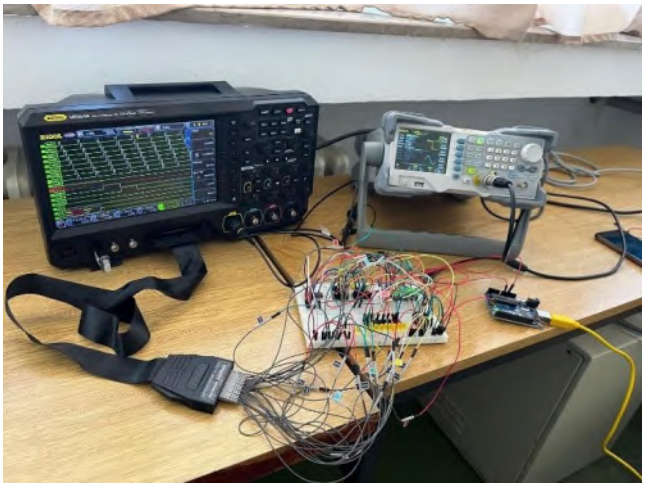
Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	1

Pre početka brojanja, svi registri se resetuju na stanje logičke nule. To se postiže dovođenjem logičke jedinice na MR (*Master Reset*) pin. Dovodimo logičku jedinicu na prvi registar koja se prosleđuje na sledeći registar, a na trenutni dolazi logička nula. Proces se ponavlja. Komponenta Džonsonovog brojača koju možemo kupiti gotovu je 74HC4017.

IV. REALIZACIJA DEKADNOG BROJAČA POMOĆU DVA PRSTENASTA BROJAČA

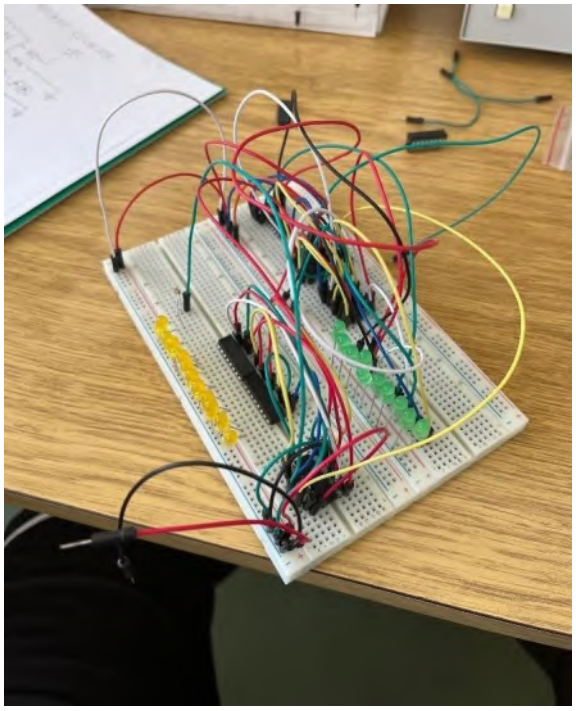
Brojač je realizovan praktično. Prilikom praktične realizacije iskorišćena su 2 Džonsonova brojača 74HC4017, 20 dioda za pinove Q0-Q9 i 2 diode za pinove /Q5-Q9, kao i odgovarajući otpornici za diode (slika 3).

Na slici 4 može se videti kako su povezani brojači na proto ploči. Vezani su kaskadno, a za to povezivanje su iskorišćeni /Q5-9 izlazi, koji su prethodno spomenuti.



Slika 3. Praktična realizacija.

Za signal pobude je iskorišćen signal generator, dok je za 5V napajanje, u nedostatku adekvatnog uređaja, iskorišćen *Arduino*. Za merenje je iskorišćen spektralni analizator.



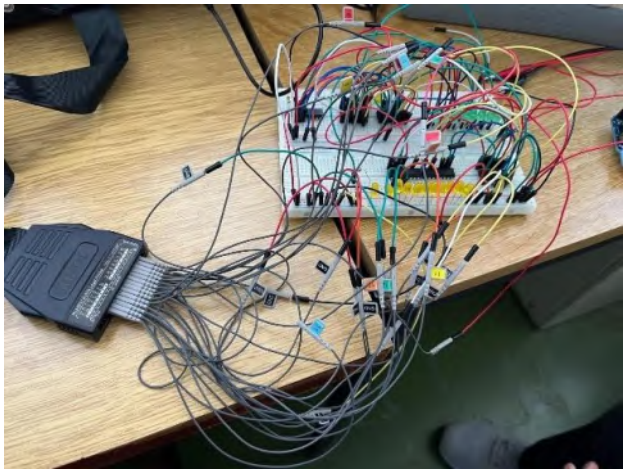
Slika 4. Realizacija prstenastog brojača.

Spektralni analizator (slika 5), je sličan osciloskopu, ali osim što može da prikazuje signale kao i osciloskop, ima još i mogućnost merenja i prikaza samo logičkih nula i jedinica, što ćemo u nastavku i videti kako to izgleda. Na slici 5 može se primetiti da analizator ima ulaze za upotrebu osciloskop moda (četiri obojena ulaza u donjem desnu uglu uređaja). Levo od ovih ulaza se nalazi ulaz za prikaz isključivo logičkih stanja. Takođe spektralni analizator ima opcije za menjanje različitih parametara, kao što su perioda, amplituda, odabir kanala itd. i te opcije se nalaze na desnoj polovini uređaja. Sam displej je osetljiv na dodir, ali ima i dodatne tastere za odabir opcija pri prikazu merenih signala.



Slika 5. Spektralni analizator RIGOL MBO5354.

Povezivanje spektralnog analizatora je vrlo jednostavno. Za prikaz logičkih stanja na ekranu analizatora postoji poseban priključak, koji ima 16 kanala, gde svaki kraj ima plus i uzemljenje. Takođe, svaki kanal ima i brojnu oznaku, da bi se kasnije na ekranu analizatora znalo koji kanal šta meri. Povezuje se tako što se plus stavi na komponentu, dok se uzemljenje vrši za svaki iskorišćeni kanal.



Slika 6. Povezivanje spektralnog analizatora i korišćenje njegove mogućnosti za detekciju logičkih stanja.

Pošto je povezivanje završeno, započinje se sa merenjem.

V. Proces merenja

Kako je na MR pin dovedena logička jedinica, svi registri su resetovani i stanje na njima je logička nula. Logička stanja zapravo detektuju prisustvo napona na diodama. Sa led01 do led91 su označeni kanali koji su dovedeni na diode prvog brojača, dok su na drugom brojaču kanali označeni sa led12 i led22.



Slika 7. Pre merenja.

Dovođenjem logičke jedinice na prvi flip-flop otpočinje brojanje. Logička jedinica pređe na sledeći flip-flop, a na trenutni se logičko stanje spusti u nulu. Kako smo realizovali dekadni brojač, da bi se logičko stanje promenilo na dekadnim diodama, neophodno je da prođu sva stanja za jedinice. Tj. tek pošto logička jedinica pređe sa 9 na 0, pin /Q5-Q9 okida desetice.

Na slici 8 vidi se proces brojanja desetica. Takođe, može se videti „glič" u prikazu označenu crvenim kvadratom (slika 9). Nastala je usled lošijeg povezivanja, kao i različitih smetnji.



Slika 8. Početak merenja.



Slika 9. Započeto je brojanje desetica kao i prisutnost „glića".

VI. Zaključak

U ovom radu prikazano je funkcionisanje dekadnog brojača uz pomoć dva Džonsonova brojača. Merenjem smo ustanovili da se posle celog ciklusa smenjivanja logičkih jedinica i nula prelazi na promenu stanja drugog brojača. Zbog toga danas ima široku primenu u mehanizmima tajmera, stoperica, semafora i svih drugih uredjaja koji se automatizuju uz pomoć elektronike.

ZAHVALNICA

Praktičnu realizaciju kao i sve nejasnoće oko realizacije projekta uspeli smo da rešimo uz pomoć asistenta Nikole Mitrovića.

LITERATURA

[1] Z. Prijić, Analogna mikroelektronika, predavanja Dostupno na: http://mikroelektronika.elfak.ni.ac.rs/
[2] Z. Prijić, D. Danković, Digitalna mikroelektronika, predavanja. Dostupno na: http://mikroelektronika.elfak.ni.ac.rs/

Projektovanje video igre za desktop računare "Remorse"

Aleksa Minčić, Mihajlo Radosavljević

Sadržaj – Ovaj rad ima za cilj da prezentuje i objasni projekat „Remorse”, koji je u stanju ranog razvoja već par meseci. Mimo vizualne prezentacije projekta, takođe je objašnjen fragment koda.

I. ISTORIJA RAZVOJA

„Remorse” je video igra za desktop računare koja je u razvoju već 7 meseci. Zbog konflikta različitih ideja, projekat je bivao odlagan, pa čak i pauziran.

„Remorse” je u početku trebala da bude *2D, Top-Down, Rogue-Like* igra, po ugledu na već postojeće igre kao što su „The Binding of Isaac”[1] i „The Legend of Bum-Bo” [2]. Projekat se tokom svog razvoja drastično menjao, ne menjajući svoj unikatan stil.

Na slici 1 se možete videti prvi dizajn igre „Remorse”.



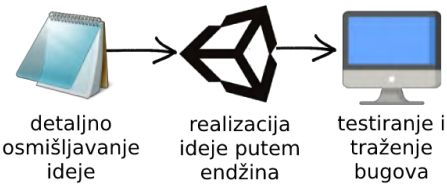
Slika 1. Prvi dizajn igre „Remorse”.

II. RAZVOJ IGRE

Video igra je igra koja se igra uz pomoć analognih ili digitalnih računara[3]. Za izradu video igre korišćeni su: program za 3D modelovanje Blender[4], program za crtanje i obradu slika Adobe Photoshop[5], program za obradu zvuka FL Studio[6] i viseplatformski program pogon igre Unity[7], koji je veoma dobro kreiran za početnike u programiranju.

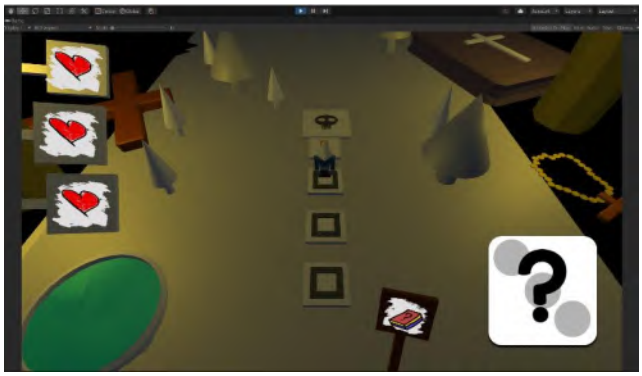
Postupak izrade celog projekta, prikazan je na slici 2.

A. Minčić i M. Radosavljević su učenici 4. godine smera Elektrotehničar Informacionih Tehnologija u Tehničkoj školi, Zaječar, Kneginje Ljubice 3-5 Zaječar, Srbija, E-mail: aleksamincic@gmail.com.



Slika 2. Šema realizacije video igre.

Konačan rezultat gotove igre u endžinu je prikazan na slici 3.



Slika 3. Prikaz glavne scene igre.

III. PRINCIP RADA

Pri pokretanju igre, igrača dočekuje početni meni na kome se nalaze tri osnovne opcije: „Započni igru”, „Opcije” i „Izadji iz igre”. Sam početni meni možete videte na slici 4.



Slika 4. Prikaz početnog ekrana.

Klikom na dugme „Izadji iz igre”, aplikacija će se isključiti i korisnik će biti odveden na desktop računara.

Isključivanje same aplikacije je veoma jednostavno, i sam kod je prikazan na slici 5.

```
0 references
public void ExitButton()
{
    //pustanje onclick animacije za exit znak
    exitOnclick.SetInteger("onclick", 1);

    //gasenje aplikacije
    Application.Quit();
}
```

Slika 5. Prikaz koda za isključivanje aplikacije.

Unity sam po sebi već ima spremnu funkciju *Application.Quit()* koja automatski isključuje sam program.

Klikom na dugme „Opcije”, kamera će se rotirati i pred korisnikom će biti opcija da isključi muziku, ili uključi, ako je prethodno bila ugašena.

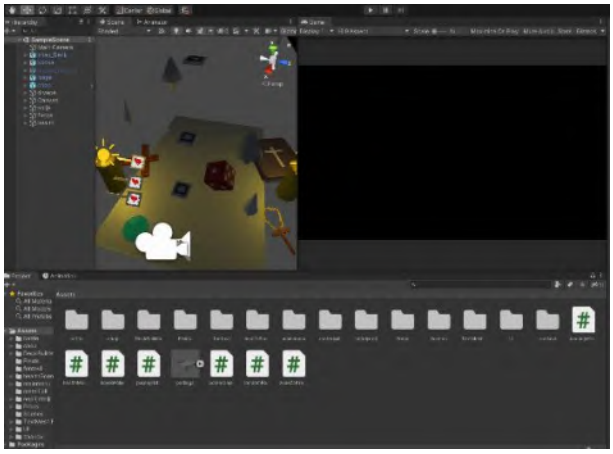
Segment opcije je prikazan na slici 6.



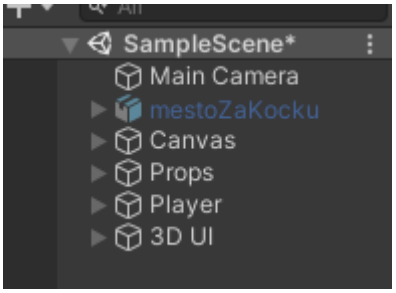
Slika 6. Prikaz segmenta „Opcije”.

Klikom na poslednju i glavnu opciju „Pokreni igru”, igra počinje učitavanjem scene 1. Na slici 7 je prikaz scene 1 sa svim svojim elementima.

Scena 1 je glavna scena igre „Remorse”, na kojoj korisnik provodi najviše vremena. Na sceni se nalaze elementi koji su podeljeni na: korisnički interfejs (*UI*), rekviziti scene (*Scene Props*) i model našeg igrača (*Player Model*). Slika same hijerarhije elemenata je prikazana na slici 8.

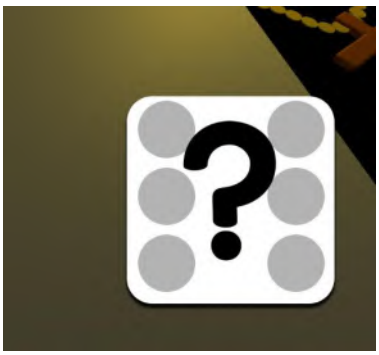


Slika 7. Prikaz scene 1 sa svojim elementima.



Slika 8. Prikaz hijerarhije elemenata.

Samim učitavanjem scene 1, pred igračom se nalazi animirani *2D Sprite* koji svojom animacijom nagoveštava da klikom na njega, kockica će biti bačena. Na slici 9 je prikazan *sprite* kockice.



Slika 9. Prikaz kockice.

Klikom na dugme za bacanje kocke, kocki, koja se nalazi van ekrana, se dodeljuje nasumična x, y i z rotacija, kao i mogućnost da gravitacija scene utiče na istu.

Na sledećoj slici je prikazan kod koji generiše tri nasumična broja i na osnovu njih rotira našu kocku.

```
Unity Script (1 asset reference) | 0 references
public class randomRotacijaKocke : MonoBehaviour
{

    public GameObject kocka;

    public Rigidbody kockaRb;

    public ConstantForce silaBacanja;
    0 references
    public void RandomRotacijaKocke()
    {
        float rngX = Random.Range(-36000, 36000);
        float rngY = Random.Range(-36000, 36000);
        float rngZ = Random.Range(-36000, 36000);

        kocka.transform.rotation = Quaternion.Euler(rngX, rngY, rngZ);

        kockaRb.useGravity = true;

        Invoke("VracanjeKocke", 4);
    }
```

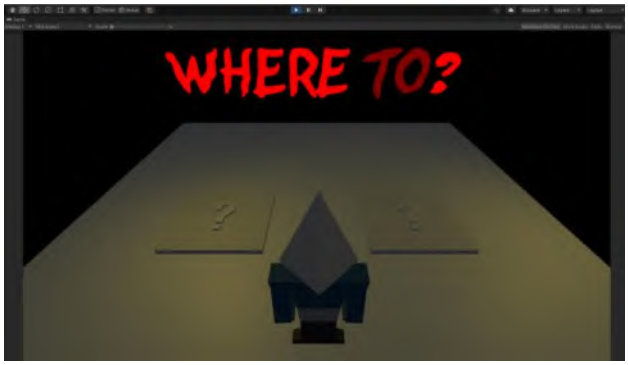
Slika 10. Prikaz koda za rotaciju kocke.

Funckija *Random.Range(n, m)* generiše nasumičan broj u opsegu n i m, dok funkcija *Quaternion.Euler(x, y, z)* primenjuje nasumične brojeve na rotaciju kocke.

Nakon bačene kocke, *loop* animacija na kocki se zaustavlja i zamenjena je istom vrednošću koju je sama kocka pokazala i igrač se pomera za jedno polje unapred.

Ako je broj kockice neparan, igrač je odved na scenu broj 3 gde se ispred igrača stavlja veoma bitno pitanje. Igrač ima mogućnost da izabere na koju stranu želi da se pomeri, a na raspolaganju su mu leva i desna strana.

Pomoću nasumične funkcije, već unapred je određeno koja strana je „loša", a koja „dobra". Na slici 11 je prikazana scena 3.



Slika 11. Prikaz scene 3.

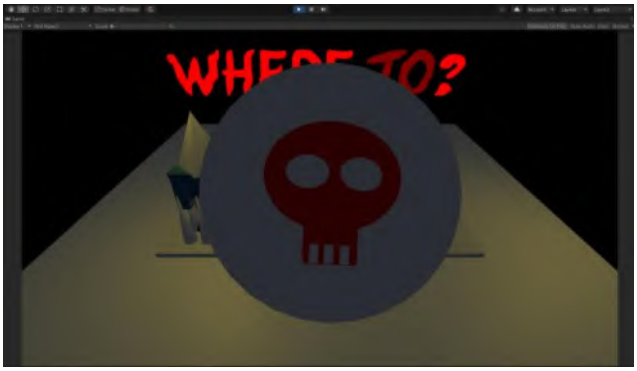
Ako igrač kojim slučajem stane na „lošu" stranu, njegov život će biti umanjen, ali će za uzvrat dobiti jedan novčić koji kasnije može iskoristiti u prodavnici.

Na slici 12 se može videte kako izgleda kada igrač stane na „lošu" stranu.

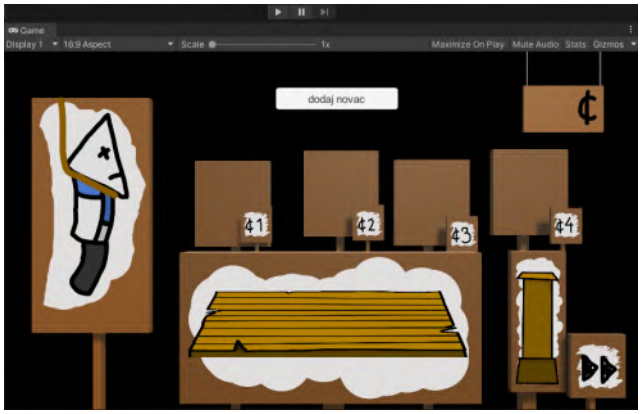
Sa prikupljenim novcem, igrač u prodavnici može kupiti specijalne „moći" koje će biti od velike pomoću u bitci sa „glavnim neprijateljem" na kraju igre.

Ako kojim slučajem igrač smatra da mu „moći" nisu potrebne, zarađeni novac može potrošiti da vrati izgubljeni život.

Prodavnica i „glavni neprijatelj" su idalje u fazi osmišljavanja i eksperimentisanja, ali prototip prodavnice se može videti na slici 13.



Slika 12. Prikaz „loše" strane.



Slika 13. Prikaz prototipa prodavnice.

IV. Zaključak

Celokupan projekat je osmišljen od strane učenika, a realizovan uz pomoć profesora-mentora.

Bilo je potrebno primeniti znanje ne samo iz programiranje već i iz Photoshop/a[5], Blendera[4] i ostalih programa. Ovaj projekat pruža i mogućnost za dalje optimizacije i unapređenja, kao što je dovršavanje prodavnice i programiranje „glavog neprijatelja". Takođe je moguća optimizacija i za mobilne uređaje i konzole.

Zahvalnica

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

LITERATURA

[1] The Binding of Isaac Wiki. Dostupno na: https://bindingofisaac.fandom.com/wiki/The_Binding_of_Isaac

[2] The Legend of Bum-Bo Wiki. Dostupno na: https://thelegendofbumbo.fandom.com/wiki/The_Legend_of_Bum-bo

[3] Video igra Wiki. Dostupno na: https://sr.m.wikipedia.org/sr/Видео-игра

[4] Blender. Dostupno na: https://www.blender.org

[5] Adobe Photoshop. Dostupno na: https://www.adobe.com/products/photoshop.html

[6] FL Studio. Dostupno na: https://www.image-line.com

[7] Unity Engine. Dostupno na: https://unity.com

Automatski punjač za LiFePO4 bateriju

Strahinja Dimitrijević, Nemanja Nedeljković

Sadržaj − Ovaj projektni zadatak se bavi izradom automatskog punjača za LiFePO4 baterijske ćelije. Najpre će biti izložena električna šema samog uređaja, zatim simulacija analognih delova u programu LTSpice, električna šema i PCB (*Printed Circuit Board*) *layout* u programskom paketu Altium Designer, i na kraju modelovanje kućišta za uređaj u programskom paketu Solid Edge 2019.

I. UVOD

Litijum-gvožđe-fosfatne (LiFePO4) ćelije predstavljaju vrstu litijum – jonskih ćelija koje imaju znatno veći kapacitet i mogu isporučiti značajno veću snagu u poređenju sa standardnim olovnim baterijama (akumulatorima). LiFePO4 baterije imaju duži životni vek zbog njihove robusne kristalne strukture usled postojanja gvožđe – fosfata i imaju veću toleranciju napona za vreme punjenja i pražnjenja [1].

LiFePO4 ćelija se može sa sigurnošću napuniti čak i do 4.2 V i uglavnom ne podleže zapaljivanju kao što je to slučaj sa litijum – jonskim ili LiPo ćelijama. Ove ćelije se često koriste u električnim automobilima upravo zbog svojih performansi i dužeg životnog veka [1].

LiFePO4 ćelija zahteva punjenje pri konstantnom naponu ili konstantnoj struji, što će upravo i biti prezentovano u ovom projektnom zadatku. Dato kolo može puniti jednu bateriju, ili dve u paraleli. Baterije se najpre pune do 3.4 V primenom konstantne struje, nakon čega prelaze u režim dopunjavanja primenom konstantnog napona [1].

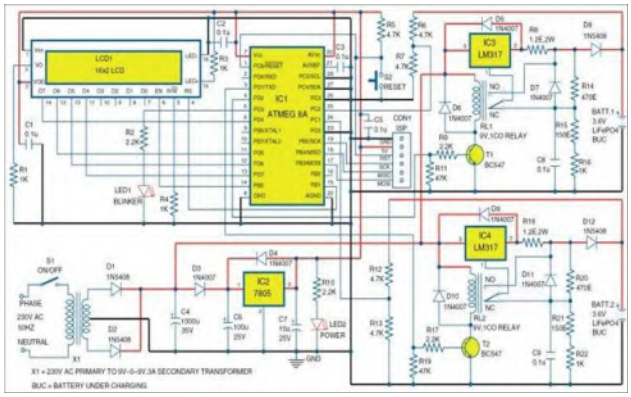
II. ELEKTRIČNA ŠEMA UREĐAJA SA OBJAŠNJENJEM

Električna šema automatskog punjača za LiFePO4 baterije prikazana je na slici 1. S obzirom da je šema uređaja već osmišljena, u radu će biti prikazana realizacija koristeći programski paket LTSpice, Altium Designer i Solid Edge [4] [5] [6].

Celokupno kolo izrađeno je oko mikrokontrolera Atmega8A (IC1), naponskog regulatora od 5 V uA7805 (IC2), dva LM317 regulatora sa promenljivim naponom (IC3 i IC4), 16x2 LCD-a (LCD1) i dva releja (RL1 i RL2). Kolo se sastoji iz tri celine:

- ulazni deo kojeg čine PCB mrežni transformator (230 V, 2x9 V, 2,8 VA, 50/60 Hz) u *center tapped* konfiguraciji, nakon čega slede dve diode 1N5408.

S. Dimitrijević i N. Nedeljković su studenti sa Katedre za mikroelektroniku, Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: straxi271@gmail.com, neca.nedeljkovic@gmail.com.



Slika 1. Električna šema uređaja [1].

(D1 i D2) koje čine punotalasni ispravljač i na kraju naponski regulator LM7805 (IC2) u *fixed-output* konfiguraciji. Ovaj regulator koristi se kako bi obezbedio stabilisani napon od 5 V za mikrokontroler. Pored toga, za indikaciju se koristi crvena LE dioda (LED2) na red sa otpornikom od 2,2 kΩ (R10). Kondenzator C4 koristi se kako bi dodatno „ispeglao“ napon na ulazu regulatora, a dioda D4 1N4007 koristi se kao *shunt* diode i služi da zaštiti regulator u slučaju inverzne polarizacije regulatora. Ovo se može desiti ukoliko napon na ulazu padne brže od napona na izlazu, i ukoliko je razlika napona između izlaza i ulaza veća od 7 V, može doći do oštećenja regulatora, pa dioda D3 sprečava potencijalno pregorevanje aktivnih komponenata unutar regulatora. Kondenzatori C6 i C7 su *bypass* kondenzatori koji se stavljaju po preporuci proizvođača kako bi uklonili šumove koji potiču od mrežnog transformatora. U primaru transformatora prisutan je *on/off* prekidač za celokupni uređaj [3].

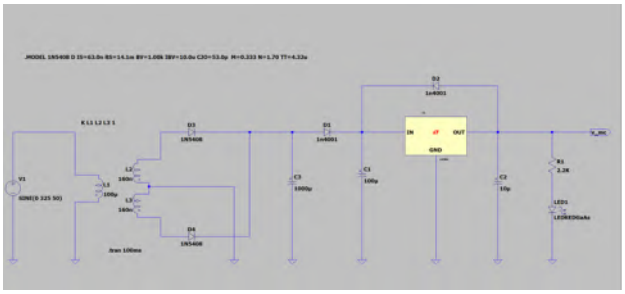
- regulatorski deo kojeg čine dva naponska regulatora LM317 u identično povezanoj konfiguraciji, dva releja sa odgovarajućim tranzistorima i zaštitnim diodama i ostale pasivne komponente koje određuju konfiguraciju regulatora. Razlog postojanja dva regulatora je mogućnost istovremenog punjenja dve baterije paralelno. Diode D5 i D9 imaju istu ulogu kao i kod LM7805 regulatora. Kao što je već rečeno, regulatorski deo može raditi u dve konfiguracije: konstantna struja (CC) i konstantni napon (CV). Prebacivanje iz jedne u drugu konfiguraciju vrši se pomoću releja RL1 i RL2, a tranzistori T1 i T2 služe kao prekidači da isključe ili uključe releje. Diode D10 i D6 su zaštitne diode koje služe da isprazne kalemove u relejima i tako spreče pojavu varničenja usled prevelikog napona. Kada je prekidač releja u NC položaju, regulator se nalazi u CV konfiguraciji. Kada je prekidač releja u NO položaju, regulator se nalazi u CC konfiguraciji. Otpornici R8 i R18

služe da ograniče struju kroz bateriju da ne bi došlo do oštećenja. Za naše potrebe, a da bismo obezbedili struju od 1 A, uzeti su otpornici od 1.2 Ω. Otpornici R14 – R16 kao i R20 – R22 imaju date vrednosti kao na Slici 1 da bi obezbedili odgovarajući napon od 3.6 V i time omogućili regulatoru da održava dati napon onda kada se baterije napune i regulatori pređu u CV konfiguraciju [2].

- deo mikrokontrolera i LCD-a kojeg čine mikrokontroler ATmega8A i 16x2 LCD. Mikrokontroler je zapravo komponenta koja kontroliše punjenje, mereći konstantno napon na bateriji i menjajući konfiguraciju regulatora između CC i CV. Stanje regulatora, kao i nivo napunjenosti baterija prikazuju se na LCD-u. Kada se dostigne odgovarajući napon na bateriji, preko namenskog pina, mikrokontroler uključuje tranzistor, on ulazi u zasićenje i uspostavlja vezu između jednog kraja releja i mase, kroz koji onda protiče struja pomerajući prekidač i vodeći regulator u CV konfiguraciju.

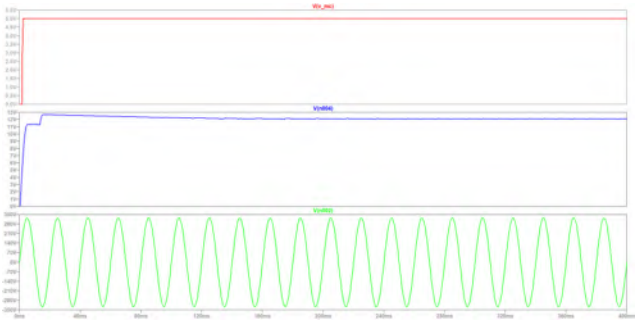
III. PRORAČUN I REZULTATI SIMULACIJE

U programu LTSpice izvršena je simulacija analognih delova datog uređaja [6]. To podrazumeva ulazni deo i regulatorski deo. Električna šema simulacije, kao i rezultati simulacija ulaznog dela kola nalaze se na slici 2 pod a) i b), respektivno.



Slika 2. a) Električna šema simulacije ulaznog dela kola.

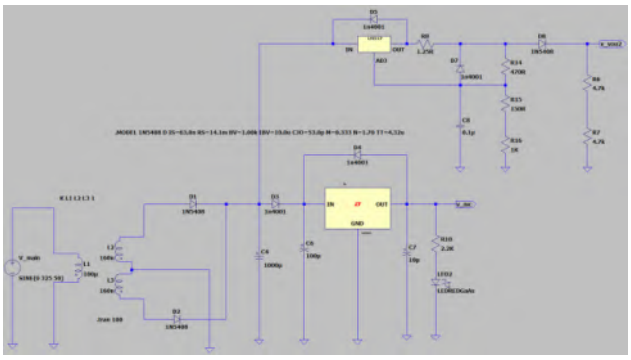
Kao što se sa slike vidi, na ulazu je postavljen idealni naponski generator koji predstavlja mrežni napon efektivne vrednosti 230 V i frekvencije 50 Hz. Nakon toga sledi transformator koji je realizovan ponoću tri kalema za koje je definisan koeficijent sprege, K. Na osnovu odnosa tranformacije kalemova obezbeđuje se da se na sekundaru pojavi naizmenični (AC) napon amplitude 12 V, a efektivne vrednosti 9 V. Pri projektovanju uređaja biće izabran transformator 230 V, 2x9 V da bi se obezbedi o upravo taj napon na sekundaru. S obzirom da u LTSpice-u ne postoji dioda 1N5408, dodata je SPICE direktiva koja opisuje tu diodu. Umesto diode 1N4007 iskorišćena je dioda 1N4001, koja ima identične specifikacije, samo je nešto manji maksimalni inverzni napon.



Slika 2. b) Električna šema simulacije ulaznog dela kola.

S obzirom da podrazumevana vrednost napona na izlazu regulatora LM7805 iznosi 5 V, ta vrednost se i očekuje pri proveri rezultata simulacije, što je i prikazano na Slici 2 pod b). Crvenom bojom označen je napon na izlazu regulatora, plavom bojom napon na ulazu u regulator, a zelenom bojom ulazni mrežni napon. Napon na izlazu regulatora je prilično stabilan tokom vremena sa skoro neprimetnim fluktuacijama, što je idealno za napajanje mikrokontrolera.

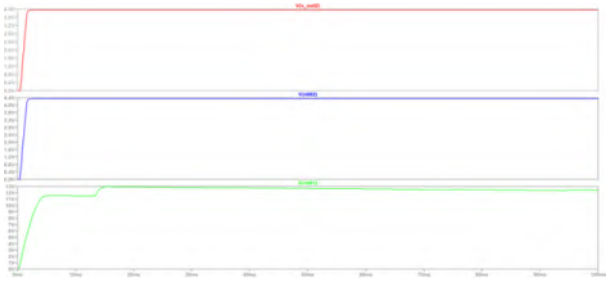
Sledeći deo za koji treba uraditi simulaciju je regulatorski deo. Električna šema simulacije, kao i odgovarajući rezultati simulacije prikazani su na slici 3 pod a) i b), respektivno.



Slika 3. a) Električna šema simulacije regulatorskog dela.

Pošto u LTSpice-u ne postoji regulator LM317, njegov simbol i model su dodati u podrazumevanu biblioteku. Računajući izlazni napon na regulatloru LM317 po formuli Vout=Vref*(1 + (R4+R5)/R3), dobija se vrednost od 4.3 V, što se i vidi na grafiku plavom bojom. Kada se taj napon umanji za vrednost pada napona na diodi 1N5408, na izlazu se dobija vrednost 3.9 V, to je napon kojim će se puniti baterija kada pređe u CV režim. proveru rada CC režima neophodno je dodati bateriju koja predstavlja opterećenje.

Simulacija same litijumske baterije u LTSpice-u nije baš jednostavna. Tako da je za simulaciju same baterije iskorišćen model sa interneta koji nije baš pouzdan, ali je bilo sasvim dovoljno da se potvrdi ispravnost kola.



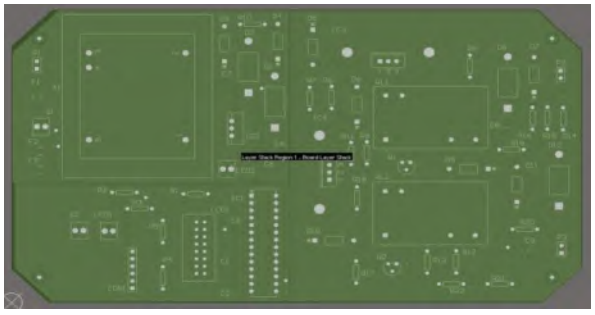
Slika 3. b) Rezultati simulacije.

Namera jeste da se kolom za punjenje pune LiFePO4 baterije, koje su jedna vrsta litijumskih baterija, prema tome verodostojnost simulacije litijumske baterije treba uzeti sa rezervom.

IV. Električna šema i PCB layout u Altium Designer-u

Kao što je i na početku navedeno, električna šema automatskog punjača za LiFePO4 baterije je podeljena u 3 celine. S tim pristupom je i krenuto u projektovanje šeme u Altium Designer-u, gde je potrebno otvoriti *New Shematic* za crtanje šeme i simbola komponenata deo po deo.

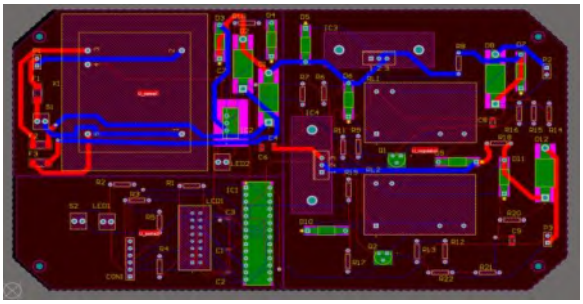
Prvo je obuhvaćen ulazni deo, koji je u projektu nazvan *sema1.SchDoc*. Sledeća celina koja je obuhvaćena jeste šema sa mikrokontrolerom i LCD displejom. koja je u projektu nazvana *sema2.SchDoc*. Poslednja celina koja je deo ovog kola jeste regulatorski blok, koja je u projektu nazvana *regulatori.SchDoc*. Preko odgovarajućih portova ovaj regulatorski blok je povezan sa blokom mikrokontrolera i sa ulaznim blokom. Svi ovi blokovi su međusobno povezani čineći tako jedinstvenu šemu *Sheet_MAIN.SchDoc* [4].



Slika 4. Izgled pločice.

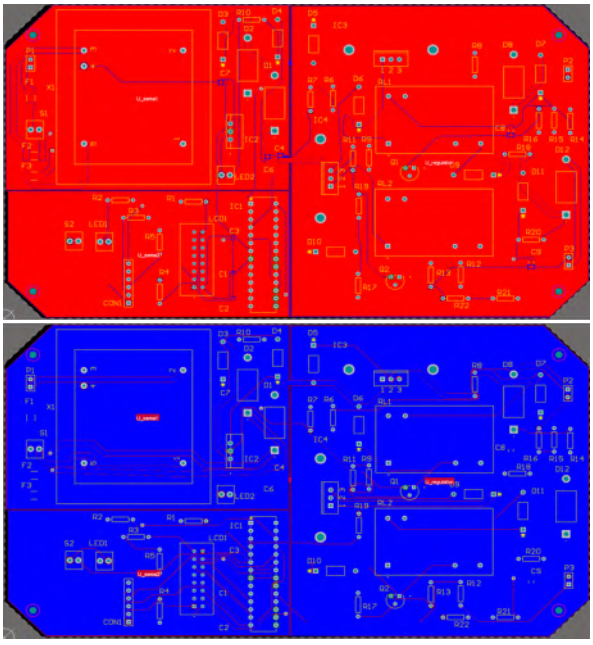
Nakon nacrtanih simbola komponenata i spajanja u jednu celinu u Schematic dokumentu, prebacuje se sada blok po blok na PCB, kako bi se adekvatno rasporedile komponente sa prethodno dodeljenim footprint-ima. Pre toga, što se može videti na slici 4 oblik PCB je malo modifikovan u skladu sa odgovarajućim kućištem. Pločica je zasečena sa sve četri strane odgovarajućih dimenzija, dok su pri samim ćoškovima postavljene rupe za šrafove kako bi zašrafili pločicu za kućište.

Na slici 5 je prikazan PCB *layout*, gde su smešteni svi blokovi iz *Shematic* dokumenta, počev od seme1 koja je smestena u levom delu pločice, zatim sema2 koja je smestena u donjem levom delu pločice i regulatorski blok, koji zauzima celu desnu stranu počice. Svaki blok se nalazi u svoom odgovarajućem prostoru (*room*), pod nazivima redom, U_sema1, U_sema2, U_regulatori.



Slika 5. PCB *layout*.

Kao što se može videti na slici 6, komponente su raspoređene u odgovarajućim prostorima i povezane linijama veze, tj. opcijom *Route* u Altiumu izvršeno je rutiranje cele šeme.



Slika 6. Prikaz razlivene mase u *Top Layer-u* i *Bottom Layer-u*.

Crvenom linijom su prikazane linije veze u *Top Layer-u*, a plavom u *Bottom Layer-u*. Takođe, još jedna primetna stvar je to da su linije veze u bloku U_sema1 i delimično u bloku U_regulatori, podebljane. To je urađeno zbog pojave visoke struje usled napajanja iz mreže i transformatora kao centralne figure, dok je u bloku regulatora to urađeno sa pristupnim linijama do konektora za bateriju zbog predostrožnosti [4].

Na kraju je ostalo proveriti greške u dizajnu i korigovati ili ispoštovati prateća pravila, *opcijom Design Rule Check*, proveravamo pravila u dizajnu. U Altiumu su sve granice i pravila postavljena inicijalno prilikom instalacije samog programa. Prilikom prvog pokretanja bilo je više od 100 kršenja pravila, tj. *Rule Violations*. Prateći prikazane greške, taj broj je smanjen na 0. Opcijom BOM (*Bill of Materials*) je određen spisak iskorišćenih komponenti [4].



	Comment	Description	Designator	Footprint	LibRef	Quantity
1	Cap	Capacitor	C1, C2, C3	CAPC1005X55N	Cap	3
2	Cap Pol3	Polarized Capacito...	C4, C6, C7	C0805	Cap Pol3	3
3	Cap Semi	Capacitor (Semico...	C8, C9	C1206	Cap Semi	2
4	Header 6	Header, 6-Pin	CON1	HDR1X6	Header 6	1
5	Diode 1N5408	3 Amp Medium P...	D1, D2	AFL-W2X2.6D	Diode 1N5408	2
6	Diode 1N4007	1 Amp General Pu...	D3, D4	DO-204AL_VIS	Diode 1N4007	2
7	1N4007	1 Amp General Pu...	D5, D6, D7, D9, D...	DO-204AL_VIS	Diode 1N4007	6
8	1N5408	1 Amp General Pu...	D8, D12	AFL-W2X2.6D	Diode 1N4007	2
9	Fuse 2	Fuse, 2410 SMT F...	F1, F2	FUSC6125X279N	Fuse 2, 0679L920...	2
10	Fuse 3	2410 SMT Fuse	F3	FUSC6125X279N	0679L9200-05	1
11	ATMEGA8A-PU	No Description Av...	IC1	28P3_ATM	ATMEGA8A-PU	1
12	7805	Positive-Voltage R...	IC2	KC3_TEX	µA7805CKC	1
13	LM317T	3-Terminal Adjust...	IC3, IC4	TO-220	LM317T	2
14	LCD-H	Header, 8-Pin, Du...	LCD1	TE_5-87589-4	Header 8X2	1
15	BLINKER	Typical RED GaAs...	LED1	MOLEX_22272021	LED1	1
16	otisak za konektor	Typical RED GaAs...	LED2	MOLEX_22272021	LED1	1
17	Header 2	Header, 2-Pin	P1, P2, P3	MOLEX_90120-0...	Header 2	3
18	NPN	NPN Bipolar Transi...	Q1, Q2	TO-92_OKS	NPN	2
19	Res1	Resistor	R1, R2, R3, R4, R5...	AXIAL-0.3	Res1	20
20	2W	Resistor	R8, R18	AXIAL-0.3	Res1	2
21	Relay-SPDT	Single-Pole Dual-T...	RL1, RL2	MODULE5B	Relay-SPDT	2
22	SW-SPST	Single-Pole, Single...	S1	MOLEX_22272021	SW-SPST	1
23	RESET	Switch	S2	MOLEX_22272021	SW-PB	1
24	X1 = 230V AC Pri...	Center-Tapped Tra...	X1	TRF_5	Trans CT	1

Slika 7. *Bill of Materials.*

V. MEHANIČKI DIZAJN U SOLID EDGE-U

Za kućište uređaja izabrana je gotova kutija od strane proizvođača *Polycase*. U skladu sa dimenzijama tog kućišta, određena je i dimenzija štampane ploče.



Slika 8. Kućište sa pločicom i korišćenim komponentama.



Slika 9. Prikaz kućišta.

Sa desne strane kućišta nalaze se prorezi za dodatno hlađenje uređaja. Na prednjem panelu kućišta nalaze se LDC displej, on/off prekidač, banana priključci za kablove i odgovarajuće diode za indikaciju. Na zadnjoj strani nalaze se ženski mrežni konektor sa 3 pina s obzirom da je kućište od metala, i dugme za resetovanje mikrokontrolera [5].

VI. ZAKLJUČAK

Što se tiče samog zaključka možemo reći da smo dosta naučili tokom projektovanja ovog punjača za LiFePO4 bateriju, s obzirom da je bilo neophodno više puta proći kroz same simulacije pre izrade dizajna u Altiumu. Takođe, kasnije smo se sve više upoznali sa samim softverskim programom za izradu štampanih ploča i uspešno realizovali projekat.

Neke dodatne stavke za unapređivanje projekta jesu: projektovani uređaj može se proširiti za punjenje većeg broja baterija uz određene izmene, pomoću mikrokontrolera i LCD displeja mogu se prikazati bilo koje informacije u vezi napunjenosti baterije, korišćenjem konektora za programiranje mikrokontrolera moguće je "spustiti" na njega bilo koji željeni kod i tako modifikovati njegov rad, postoji mogućnost montiranja socket-a za punjenje baterija, pri čemu nisu potrebni banana jack-ovi.

ZAHVALNICA

Autori rada se zahvaljuju Katedri za mikroelektroniku na konstantnoj podršci prilikom realizacije rada, pre svega profesorima Danijelu Dankoviću i Zoranu Prijiću na nesebičnoj pomoći i usmeravanju tokom izrade projekta.

LITERATURA

[1] Automatski punjač baterija. Dostupno na: https://www.electronicsforu.com/electronics-projects/automatic-lifepo4-cell-charger
[2] Tehnička dokumentacija regulatora LM317, Dostupno: https://www.ti.com/product/LM317?keyMatch=LM%20317
[3] Tehnička dokumentacija regulatora LM7805, Dostupno: www.sparkfun.com/datasheets/Components/LM7805.pdf
[4] Altium Designer uputstvo. Dostupno: https://www.altium.com/documentation/altium-designer?version=20.0
[5] Video tutorijali za SolidEdge, jedan od tutorijala dostupan na: https://www.youtube.com/watch?v=MJSHi7bcRKo
[6] LTSpice tutorijal. Dostupno na: https://www.analog.com/en/education/education-library/videos/video-series/ltspice-getting-started-tutorial.html

Ekonomski dispečing termosistema

Teodora Denić

Sadržaj − U ovom radu obrađena je problematika ekonomskog dispečinga (ED-a) termoenergetskog sistema. Kroz pregled najvažnije literature koja se bavi ovom tematikom, data su opšta razmatranja o problemu ED-a i izvršeno je matematičko modelovanje problema. U okviru praktičnog dela rada, rešavan je problem dinamičkog ekonomskog dispečinga (DED-a) termosistema sa 10 agregata, na vremenskom horizontu od 24 sata. Rešavanje konkretnog optimizacionog problema bazirano je na implementaciji matematičkog modela DED-a na agregate razmatranog termosistema, za šta je korišćen komercijalni programski paket GAMS (*General Algebraic Modeling System*). Predloženi model implementiran je u GAMS kao problem nelinearnog programiranja. Kao rezultat izvršenja GAMS koda dobijena je optimalna preraspodela opterećenja između angažovanih agregata na satnom nivou, tokom celokupnog optimizacionog perioda.

I. UVOD

Problem ED-a predstavlja važan segment kratkoročnog planiranja rada elektroenergetskog sistema (EES). Izvodi se u okviru dnevnog planiranja proizvodnje, tj. izrade voznog reda elektrana za naredni dan, kao i u realnom vremenu. Značaj ED-a ogleda se u ekonomičnosti i efikasnosti korišćenja proizvodnih kapaciteta koji su u pogonu, sa ciljem minimizacije pogonskih troškova.

ED pripada klasi nelinearnih problema iz domena optimizacije i može se definisati kao problem optimalnog raspoređivanja proizvodnje na raspoložive generatorske jedinice. Pri tome se moraju zadovoljiti uslovi sigurnosti i kvaliteta rada sistema. Optimizacioni kriterijum ekonomične raspodele opterećenja obuhvata troškove pogona, dok su ograničenja nametnuta potrebama potrošača, zahtevima obrtne rezerve i ograničenjima rada agregata i prenosne mreže. Zbog velike dimenzionalnosti i nelinearnosti problema, kao i zastupljenosti velikog broja ograničenja, ED predstavlja izrazito složen optimizacioni postupak. U literaturi se zbog toga sreću različiti prilazi ovom problemu, koji se međusobno razlikuju i po modelima i po metodama rešavanja.

Problem klasičnog ED-a termosistema detaljno je obrađen u [1]-[4]. Pored toga, u literaturi koja se bavi ovom tematikom, nailazi se na moguće brojne aspekte proširenja modela klasičnog ED-a: uvažavanje sigurnosnih ograničenja [2], [3]; inkorporiranje zahteva za ograničenom emisijom štetnih gasova u model ED-a [5], [6]; ED u prisustvu obnovljivih izvora energije [7]; DED [5], [8]; uvažavanje efekta sekvencijalnog puštanja pare u turbinu sa više regulacionih ventila i zabranjenih oblasti rada pojedinih proizvodnih jedinica [8].

Optimizacione metode koje se koriste za rešavanje navedenih problema, mogu se podeliti na determinističke (klasične, egzaktne) i (meta)heurističke metode [5]. Poslednjih nekoliko decenija, razne klasične metode optimizacije, kao što su λ-iterativni metod, gradijentni metod, Njutnov metod optimizacije, dinamičko programiranje, Lagranžova relaksacija i dr. [1]-[4], primenjivane su na različite probleme ED-a. Među klasičnim metodama, za rešavanje problema ED-a se u [1] kao najčešće korišćene navode tehnike nelinearnog i dinamičkog programiranja. Linearizacija funkcije troškova „deo po deo" (eng. *piecewise linear approximation*), kao i ograničenja koja se postavljaju, omogućavaju primenu i nekih od tehnika linernog programiranja za rešavnje problema ED-a [2]. Međutim, većina klasičnih metoda nailazi na poteškoće pri rešavanju izrazito nelinearnih modela proširenih problema ED-a, sa nediferencijabilnim i nekonveksnim funkcijama, zbog čega često ne uspevaju da obezbede globalni optimum, već konvergiraju ka lokalnom optimalnom rešenju. Metaheurističke metode optimizacije u stanju su da prevaziđu ove nedostatke i da se izbore sa pomenutim poteškoćama. U [6] i [8] detaljno je sagledan trend rešavanja problema ED-a primenom (meta)heurističkih metoda, koji se poslednjih godina sve više okreće ka hibridnoj metaheuristici.

U ovom radu je za rešavanje problema ED-a korišćen GAMS softver, baziran na egzaktnim rešavačima. U okviru praktičnog dela rada rešavan je optimizacioni problem DED-a. U cilju formiranja adekvatnog matematičkog modela za rešavanje ovog problema, prethodno je, u odeljku II, izvršeno matematičko modelovanje klasičnog ED-a i proširenog problema DED-a. Na primeru rešavanja problema optimalne raspodele opterećenja između proizvodnih jedinica konkretnog termosistema, u odeljku III prezentovana je primena formiranog modela DED-a u GAMS kao problem nelinearnog programiranja.

II. MATEMATIČKI MODELI

A. Modelovanje termoelektrana u problemu ED-a

U matematičkom modelovanju termoelektrana, pogonski troškovi bazirani na troškovima goriva (uz pretpostavku da su zanemareni troškovi pogona i održavanja, tj. da su satni troškovi goriva jednaki pogonskim troškovima), za svaku proizvodnu jedinicu, predstavljaju se

T. Denić je završila master akademske studije na studijskom programu Elektroenergetika, Elektronski fakultet u Nišu, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: teodra.denic@elfak.rs.

funkcijom troškova aproksimiranom glatkom krivom, odnosno kvadratnom funkcijom oblika:

$$C_n\left(P_{T_n}\right) = a_n {P_{T_n}}^2 + b_n P_{T_n} + c_n \left[\frac{\text{NJ}}{\text{h}}\right], \qquad (1)$$

gde su za $n = 1, \dots, N_T$: a_n [NJ/(MW)^{2}h], b_n [NJ/MWh] i c_n [NJ/h] koeficijenti pogonskih troškova za određeni termoagregat n; P_{T_n} [MW] je odata snaga agregata; dok je N_T ukupan broj termoagregata u sistemu.

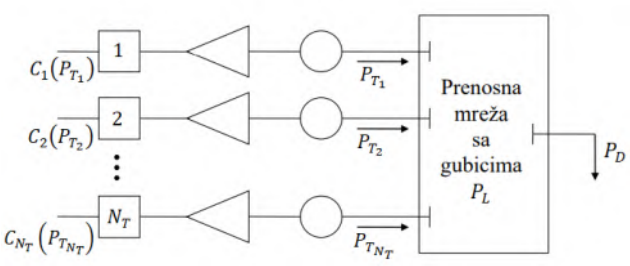
Za svaku proizvodnu jedinicu uvažava se dozvoljeni opseg rada, odnosno ograničenje maksimalne, ${P_{T_n}}^{max}$, i minimalne, ${P_{T_n}}^{min}$, snage:

$${P_{T_n}}^{min} \le P_{T_n} \le {P_{T_n}}^{max}. \qquad (2)$$

Ograničenje mogućeg opsega rada je ograničenje tipa nejednakosti i mora biti ispoštovano za svaki agregat tokom celog optimizacionog perioda.

B. Optimizacioni model ED-a termosistema

Slika 1 prikazuje konfiguraciju EES-a za koji se rešava problem ED-a. Posmatra se termosistem sa N_T termoagregata, angažovanih da snabdevaju opterećenje sistema, P_D, preko prenosne mreže sa gubicima, P_L. Svaki termoagregat je modelovan sa pripadajućom funkcijom satnih troškova goriva u zavisnosti od odate snage, P_{T_n}, (1) i dozvoljenim opsegom rada (2).



Slika 1. Šema sistema za koji se rešava problem ED-a.

Funkcija ukupnih pogonskih troškova predstavlja objektivnu funkciju, odnosno funkciju cilja u matematičkoj formulaciji ED-a. Funkcija ukupnih pogonskih troškova u sistemu jednaka je zbiru pogonskih troškova pojedinih termoagregata:

$$C_T(\boldsymbol{P_T}) = \sum_{n=1}^{N_T} C_n\left(P_{T_n}\right) \ [\text{NJ/h}], \qquad (3)$$

gde je $\boldsymbol{P_T} = \left[P_{T_1} \quad P_{T_2} \quad \dots \quad P_{T_{N_T}}\right]$.

Osnovni uslov u sistemu koji mora biti zadovoljen je bilans snaga - ukupna proizvodnja angažovanih agregata treba da zadovolji potrebe opterećenja sistema, P_D, kao i potrebe gubitaka u mreži, P_L:

$$\sum_{n=1}^{N_T} P_{T_n} = P_D + P_L \ [\text{MW}]. \qquad (4)$$

Za proračun gubitaka, P_L, najadekvatnije bi bilo koristiti jednačine tokova aktivnih i reaktivnih snaga sa egzaktnim modelom prenosne mreže. Ipak, najčešće se koristi jednostavniji pristup. U slučaju ED-a aktivnih snaga u realnom vremenu često se zanemaruje efekat tokova reaktivnih snaga, tako da se gubici izražavaju samo kao funkcija aktivnih snaga generatora. Taj metod je poznat pod nazivom *metod B-koeficijenata* [1].

Zadatak ED-a je da se pronađe raspodela opterećenja između $n = 1, \dots, N_T$ generatora, takva da se dostigne optimum objektivne funkcije - minimum pogonskih troškova u sistemu:

$$\min\{C_T(\boldsymbol{P_T})\}, \qquad (5)$$

uz uvažavanje ograničenja dozvoljenog opsega rada svakog agregata (2) i održavanja ravnoteže između proizvodnje, potrošnje i gubitaka u sistemu (4).

C. Matematička formulacija problema DED-a

DED predstavlja proširenje problema konvencionalnog ED-a. Za razliku od modela (5) klasičnog ED-a, koji uvažava samo ograničenja vezana za statičko ponašanje proizvodnih jedinica (statički ED - SED), DED uzima u obzir i dinamička ograničenja brzine promene izlazne snage generatora (eng. *ramp-rate limit*). Dozvoljena brzina promene opterećenja agregata određena je termičkim naprezanjima materijala i zavisi od konstrukcije i karakteristika turbine. Dispečing proizvodnih jedinica obavlja se na datom vremenskom horizontu, uz pretpostavku da je *on*/*off* stanje agregata poznato. Za svaki vremenski interval operativnog horizonta određuje se optimalna preraspodela opterećenja između angažovanih agregata, sa ciljem da se postignu minimalni troškovi rada, uz uvažavanje operativnih ograničenja. Preraspodela se obično vrši na satnom nivou. Za rešavanje problema u svakom vremenskom intervalu koristi se SED, koji za dati interval pretpostavlja da je opterećenje konstantno.

Posmatrani vremenski period od interesa, T, deli se na J intevala, $t_j = T/J$, u [h], gde se obično uzima da je $t_j = 1$ h. Objektivna funkcija, koju treba minimizovati, je suma pogonskih troškova svih N_T agregata za koje se vrši analiza, tokom celokupnog razmatranog vremenskog perioda T:

$$min\, C_T = \sum_{j=1}^{J} \sum_{n=1}^{N_T} t_j C_n\left(P_{T_n}(j)\right) \ [\text{NJ}], \qquad (6)$$

gde je $C_n\left(P_{T_n}(j)\right) = a_n P_{T_n}(j)^2 + b_n P_{T_n}(j) + c_n$ funkcija troškova goriva za *n*-tu termoelektranu u *j*-om vremenskom intervalu, u [NJ/h]; t_j je trajanje vremenskog intervala *j*, u [h].

U svakom vremenskom intervalu $j = 1, 2 \ldots, J$ moraju biti zadovoljena sledeća ograničenja:

1) ograničenja proizvodnje generatora (2);

2) jednačina bilansa aktivnih snaga u sistemu (4);

3) ograničenja brzine promene snage generatora - opseg rada generatorske jedinice takođe je određen brzinom promene njene izlazne snage. Pretpostavka je da ona trenutno odgovara na promenu svoje izlazne snage unutar definisanog opsega. Ograničenja brzine promene izlazne snage generatora definisana su izrazima (7) i (8), kada snaga generatora raste i kada snaga generatora opada, respektivno:

$$P_{T_n}(j+1) - P_{T_n}(j) \leq RU_n\,, \qquad (7)$$

$$P_{T_n}(j) - P_{T_n}(j+1) \leq RD_n\,, \qquad (8)$$

gde RU_n i RD_n predstavljaju maksimalni dozvoljeni porast (eng. *upper ramp rate*) i smanjenje (eng. *down ramp rate*) snage, redom, pri prelasku iz tekućeg intervala j u naredni $j+1$ interval, za agregat n.

III. REŠAVANJE PROBLEMA DED-A

Primena formiranog matematičkog modela DED-a biće prezentovana na konkretnom primeru EES-a. Konfiguracija EES-a, za koji se rešava problem DED-a, prikazana je na Slici 1. U konkretnom slučaju, razmatra se termoenergetski sistem sa $N_T = 10$ termoagregata. Za rešavanje problema korišćen je GAMS softver.

A. Kratak opis korišćenog softvera

GAMS je programski paket za algebarsko modelovanje visokog nivoa, razvijen za rešavanje složenih optimizacionih problema velikih dimenzija. Dizajniran je za modelovanje i rešavanje različitih vrsta problema – linearnih, nelinearnih, kao i problema optimizacije mešovitih celih brojeva. Formirani modeli mogu pripadati različtim klasama problema, za čije se rešavanje primenjuje niz ugrađenih rešavača (*solver*-a). Među ovim rešavačima su BARON, CONOPT, CPLEX, GUROBI, MINOS, XPRESS i dr. Sintaksa GAMS jezika inspirisana je matematičkom notacijom modela, što omogućava sažeto i čitljivo kodiranje problema. Definisani problem optimizacije, GAMS ne rešava direktno, već kompajler GAMS jezika prevodi izvorni kod u format koji ugrađeni rešavači mogu da „razumeju" i reše. Ovo omogućava fokusiranje na modelovanje, bez zalaženja u tehničke detalje povezane sa algoritmima rešavanja.

B. Definisanje zadatka i ulazni podaci

U Tabeli I [8] prikazani su karakteristični parametri angažovanih termoagregata (G_1 - G_{10}).

TABELA I
KARAKTERISTIČNI PARAMETRI TERMOAGREGATA

G_n	a_n [NJ/(MW)²h]	b_n [NJ/MWh]	c_n [NJ/h]	P_n^{min} [MW]	P_n^{max} [MW]	RU_n [MW]	RD_n [MW]
G_1	0,00043	21,6	958,2	150	470	80	80
G_2	0,00063	21,05	1313,6	135	460	80	80
G_3	0,00039	20,81	604,97	73	340	80	80
G_4	0,0007	23,9	471,6	60	300	50	50
G_5	0,00079	21,62	480,29	73	243	50	50
G_6	0,00056	17,87	601,75	57	160	50	50
G_7	0,00211	16,51	502,7	20	130	30	30
G_8	0,0048	23,23	639,4	47	120	30	30
G_9	0,10908	19,58	455,6	20	80	30	30
G_{10}	0,00951	22,54	692,4	55	55	30	30

B-koeficijenati gubitaka dati su matricom gubitaka $\bar{\boldsymbol{B}}$ [8], dimenzije (10×10), oblika:

$$\bar{\boldsymbol{B}} = 10^{-5} \times \qquad (9)$$

$$\begin{bmatrix} 8{,}7 & 0{,}43 & -4{,}61 & 0{,}36 & 0{,}32 & -0{,}66 & 0{,}96 & -1{,}6 & 0{,}8 & -0{,}1 \\ 0{,}43 & 8{,}3 & -0{,}97 & 0{,}22 & 0{,}75 & -0{,}28 & 5{,}04 & 1{,}7 & 0{,}54 & 7{,}2 \\ -4{,}61 & -0{,}97 & 9 & -2 & 0{,}63 & 3 & 1{,}7 & -4{,}3 & 3{,}1 & -2 \\ 0{,}36 & 0{,}22 & -2 & 5{,}3 & 0{,}47 & 2{,}62 & -1{,}96 & 2{,}1 & 0{,}67 & 1{,}8 \\ 0{,}32 & 0{,}75 & 0{,}63 & 0{,}47 & 8{,}6 & -0{,}8 & 0{,}37 & 0{,}72 & -0{,}9 & -0{,}69 \\ -0{,}66 & -0{,}28 & 3 & 2{,}62 & -0{,}8 & 11{,}8 & -4{,}9 & 0{,}3 & 3 & -3 \\ 0{,}96 & 5{,}04 & 1{,}7 & -1{,}96 & 0{,}37 & -4{,}9 & 8{,}24 & -0{,}9 & 5{,}9 & -0{,}6 \\ -1{,}6 & 1{,}7 & -4{,}3 & 2{,}1 & 0{,}72 & 0{,}3 & -0{,}9 & 1{,}2 & -0{,}96 & 0{,}56 \\ 0{,}8 & 0{,}54 & 3{,}1 & 0{,}67 & -0{,}9 & 3 & 5{,}9 & -0{,}96 & 0{,}93 & -0{,}3 \\ -0{,}1 & 7{,}2 & -2 & 1{,}8 & 0{,}69 & -3 & -0{,}6 & 0{,}56 & -0{,}3 & 0{,}99 \end{bmatrix}.$$

Treba naći optimalnu preraspodelu opterećenja između agregata u pogonu, sa ciljem da se postignu minimalni troškovi rada, uz uvažavanje operativnih ograničenja. Pri tome se mreža posmatra kao crna kutija, tj. ne vodi se računa o ograničenjima vodova, kao i tokovima aktivnih i reaktivnih snaga po njenim granama. Preraspodela se vrši na satnom horizontu, za vremenski period od interesa, $T = 24$ [h]. Predviđeno opterećenje, izraženo u [MW], za svaki sat optimizacionog perioda T, iznosi, redom [8]: 1036, 1110, 1258, 1406, 1480, 1628, 1702, 1776, 1924, 2072, 2146, 2220, 2072, 1924, 1776, 1554, 1480, 1628, 1776, 2072, 1924, 1628, 1332, 1184.

Rešavanje zadatog problema bazira se na implementaciji matematičkog modela DED-a na agregate razmatranog termosistema, uz uvažavanje podataka o predviđenom satnom opterećenju. Gubici u sistemu dobijaju se na osnovu *B*-koeficijenata gubitaka (9), korišćenjem Georgeove formule gubitaka [1].

C. Rezultati

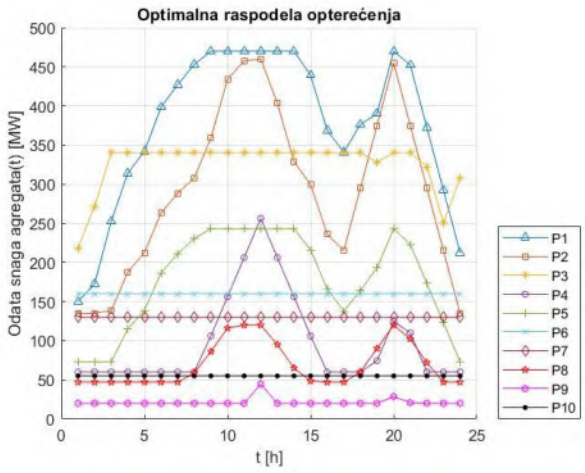
Predloženi model implementiran je u GAMS kao problem nelinearnog programiranja. Model u GAMS-u ima ukupno 530 jednačina i 266 varijabli. Primenjeni rešavač je CONOPT. Broj iteracija je 53. Vreme koje *solver* koristi za rešavanje modela, pri konkretnom pokretanju programa, iznosi 0,079 [s]. Ukupno vreme izvršenja koda, uključujući i eksportovanje podataka u kreiranu Excel datoteku, iznosi 3,695 [s]. Ova vremena su uslovljena performansama računara na kome je GAMS softver instalisan.

Kao rezultat izvršenja GAMS koda dobijena je optimalna raspodela opterećenja između angažovanih agregata, prikazana na Slici 2. Pri tome je vrednost ukupnih pogonskih troškova, tokom celokupnog razmatranog perioda T:

$$C_T{}^{min} = 1022636{,}697\ [\text{NJ}]\,, \tag{10}$$

uz ostvarene gubitke u mreži od:

$$\sum_{j=1}^{24} P_L(j) \times 1\ [\text{h}] = 839{,}473\ [\text{MWh}]\,. \tag{11}$$



Slika 2. Optimalna preraspodela opterećenja.

IV. Zaključak

ED predstavlja važan optimizacioni problem u elektroenergetici. Zbog nelinearne prirode problema i velikog broja podataka kojima se opisuje i modeluje EES, problem ED-a je veoma složen. Uvođenje određenih aproksimacija, formiranje adekvatnih matematičkih modela i korišćenje moćnih softvera mogu omogućiti pronalaženje optimalnog rešenja na brz i efikasan način.

U ovom radu sagledana je problematika ED-a termosistema, sa posebnim osvrtom na prošireni problem DED-a. Nakon što je izvršeno matematičko modelovanje problema, na konkretnom primeru EES-a prezentovana je primena formiranog modela DED-a. Predloženi model implementiran je u GAMS kao problem nelinearnog programiranja, a rezultati proračuna pokazuju da se, uz adekvatno modelovanje problema, ED termosistema može efikasno rešavati u GAMS okruženju.

Zahvalnica

Ovaj rad predstavlja deo master rada „Ekonomski dispečing“ koji je osmišljen i započet pod mentorstvom prof. dr Aleksandra Janjića, a završen pod mentorstvom prof. dr Lidije Korunović kojoj se autor zahvaljuje na korisnim sugestijama i pomoći prilikom izrade ovog rada.

Literatura

[1] M. S. Ćalović, A. T. Sarić, P. S. Stefanov, *Eksploatacija elektroenergetkih sistema u uslovima slobodnog tržišta*. Beograd: Tehnički fakultet u Čačku, 2005.

[2] J. Zhu, *Optimization of Power System Operation*. Hoboken, NJ: Wiley, 2009.

[3] A. E. Gómez, A. J. Conejo, C. Cañizares, *Electric Energy Systems: Analysis and Operation*. Boca Raton: CRC Press, 2020.

[4] A. J. Wood and B. F. Wollenberg, *Power generation, operation, and control*. Wiley, 1996.

[5] A. Soroudi, *Power system optimization modeling in GAMS*. Cham, Switzerland: Springer, 2017.

[6] J. Radosavljević, *Metaheuristic Optimization in Power Engineering*. London: The Institution of Engineering & Technology, 2018.

[7] D. Fateh, M. Eldoromi, and A. A. Moti Birjandi, „Uncertainty modeling of renewable energy sources”, *Scheduling and Operation of Virtual Power Plants*, pp. 193–208, 2022.

[8] D. Santra, A. Mukherjee, K. Sarker, S. Mondal, „Dynamic Economic Dispatch using hybrid metaheuristics - journal of electrical systems and information technology”, *SpringerOpen*, 10-Feb-2020. [Online]. https://jesit.springeropen.com/articles/10.1186/s43067-020-0011-2. [Pristupljeno: 11-Jul-2022]

Design and Implementation of a Self-Synchronizing LFSR based Descrambler

Jana Vasić, Jovana Nesić

Abstract − This paper presents design and implementation of a descrambler based on D flip-flops and XOR circuits. Descramblers are used to decode the information that is at the input of the corresponding scrambler. Firstly, we analyze the theoretical and functional properties of descramblers. Then, a simulation of a working descrambler is created using LT Spice. This process is necessary to create a schematic which will then guide us in the further product realization. We test the functionality of our prototype by connecting it with a corresponding scrambler. The functionality of our realized prototype is proven using an oscilloscope, which showed us the original bit sequence. This prototype can be used for bit rates up to 1Mb/s.

I. Introduction

In Digital Communications, a descrambler is a device that transposes or inverts signals or otherwise decodes a message on the receiver's end in order to make the message encoded by a scrambler readable [1]. It is often used for encoding various types of data, such as images, videos and audio signals [2]-[4]. In order to fully understand the process of descrambling, we must first define the process of scrambling. A scrambler is a logic circuit that performs randomization of binary data. This logic circuit is a generic linear feedback shift register (LFSR) with EX-OR gates.

In Fig. 1 we can see a theoretical schematic of the process of scrambling and descrambling.

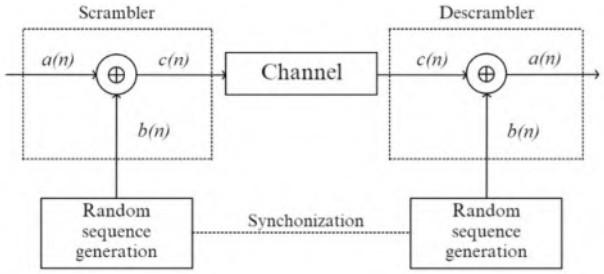


Fig. 1. Theoretical mode we made of the Scrambling-Descrambling process

To encode an original message, the sequence at the sender's end is modified in one of these two ways: 1) The original message is combined with additional components; 2) The important components of the original message are changed. Regardless of the chosen option, the original message can be read only with the corresponding descrambler. The purpose of scrambling is to remove long strings of ones and zeros from the digital binary data [5]. Analogous to that, descrambling is a process of reverse scrambling.

J. Vasić and J. Nesić are third-year students of bachelor studies at the department of Telecommunications at the faculty of Electronic Engineering, Univerity of Nis. Aleksandra Medvedeva 14, 18000 Nis, Serbia, E-mail: janavasic@elfak.rs; jovananesic@elfak.rs.

II. Functional Diagram

The shift register circuit shown in Fig. 2(a) is a linear sequential filter with feedback paths and is an example of a scrambling circuit. The linear sequential filter with feed-forward paths shown in Fig. 2(b) is an example of a descrambling circuit and is the complementary circuit to that shown in Fig. 2(a) and it regenerates the data sequence from the channel sequence. It is assumed in these two examples that data is presented as a binary sequence, the addition is taken moduo 2, and that the storage elements provide one bit of delay.

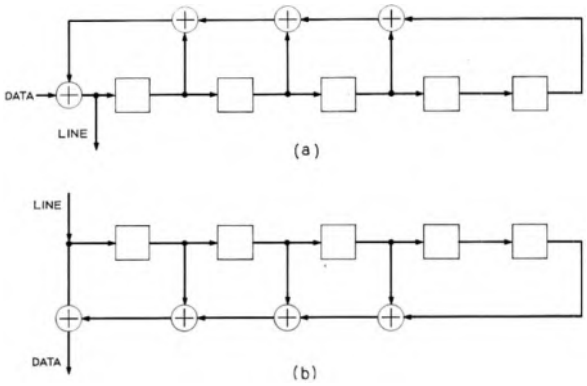


Fig. 2. (a) A basic scrambler; (b) a basic descrambler [2].

Examination of the circuit of Fig. 2 shows that they have the required synchronization property since the effect of a bit lost or added in the line sequence is felt only as long as the values stored in the descrambler disagree with those stored at the scrambler, which is five bit intervals in our example [6].

III. Simulation

The first step was to create a descrambler schematic in LT Spice. It was designed so that it consists of 16 cells and has 2 XOR circuits. The shift register consists of a series of

D flip-flops that have a common clock signal. The output was taken from the 14th and 16th D flip-flops, which can be seen in Fig. 3(a).

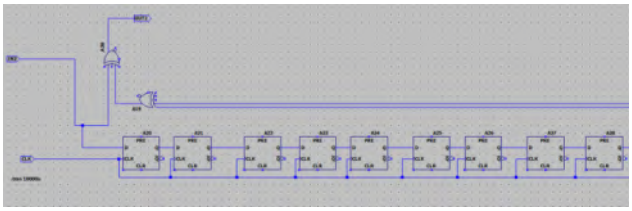


Fig. 3(a). Simulation schematic we made in LT Spice.

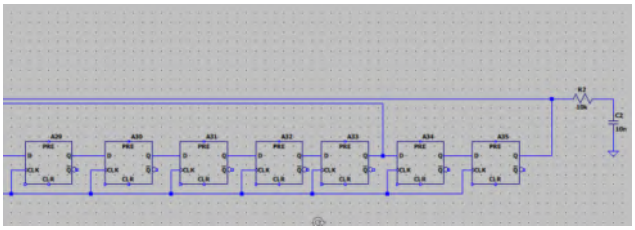


Fig. 3(b). Simulation schematic we made in LT Spice.

In Fig. 3(b) resistor R2 [7] and capacitor C2 [8] represent a low-pass filter and in this simulation serve as an indicator of the quality of the pseudo-random sequence.

IV. Component Characteristics

The following components were used for this project:

1. One-sided raster plate type 1 50x100;
2. XOR – IC TTL-H.S.CMOS 74HC86 (Fig. 4);
3. Shift register x2-IC TTL-H.S.CMOS 74HC595 (Fig. 5);
4. Light-emmiting diode (LED);
5. Resistor;
6. Capacitor;
7. Capacitor of type C;

The XOR 74HC86 provides four independent 2-input Exclusive OR gates with standard push-pull outputs. The device is designed for operation with a power supply range of 2.0 V to 6.0 V [9].

The 74HC595 devices contain an 8-bit serial-in, parallel-out shift register that feeds an 8-bit D-type storage register. The storage register has parallel 3-state outputs. Separate clocks are provided for both the shift and storage register. The shift register has a direct overriding clear (SRCLR) input, serial (SER) input, and serial outputs for cascading. When the output-enable (OE) input is high, the outputs are in the high-impedance state. Both the shift register clock (SRCLK) and storage register clock (RCLK) are positive-edge triggered. If both clocks are connected together, the shift register always is one clock pulse ahead of the storage register [10].

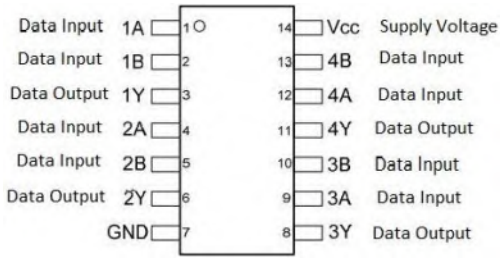


Fig. 4. XOR circuit 74HC86.

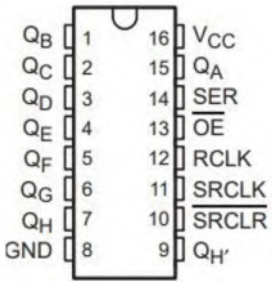


Fig. 5. Shift register 74HC595.

The gates perform the Boolean function: Y = A ⊕ B or Y = AB + AB. Fig. 6 is a schematic we created in LT Spice that shows us which elements should be connected to each other at the level of integrated circuits.

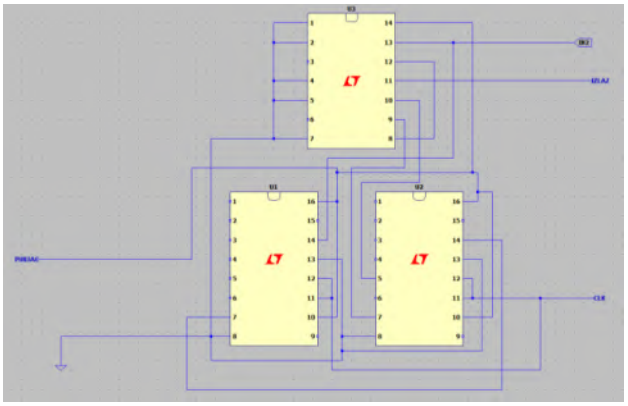


Fig. 6. The schematic of the descrambler that we created in LT Spice.

V. Implementation

After preparing all the necessary components and the schematic according to which we will work, the realization process can begin. The corresponding components are arranged on the raster board and connected to each other according to the previously mentioned schematic.

We soldered all components and connections, taking care not to short-circuit when soldering adjacent pins. The use of wires was avoided as much as possible, because it reduces the reliability of the prototype, but it was not possible to avoid them completely, because they enabled easy implementation for pins that are at a greater distance.

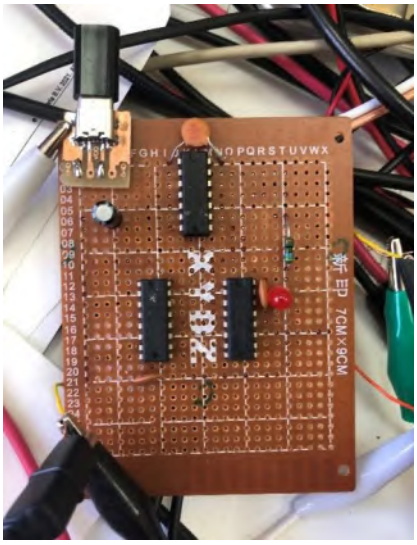


Fig. 7. Circuit board with all of the components.

VI. Results

After creating the prototype, we connected it with the previously implemented scrambler, as shown in Fig. 8, in order to determine that the entire system is functional, that is, the information at the scrambler input is successfully decoded by our descrambler.

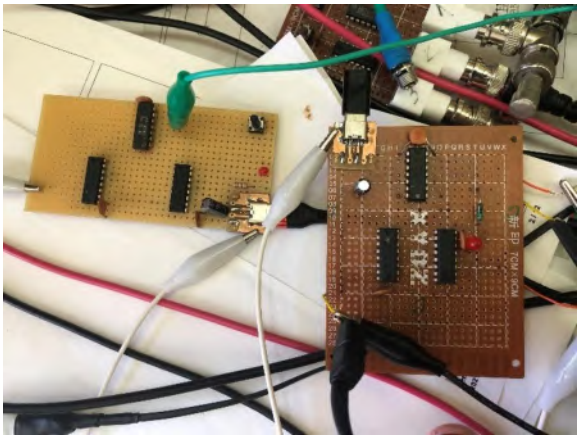


Fig. 8. Developed scrambler (left) and descrambler (right) connected.

We connected the scrambler and the descrambler to the oscilloscope, Fig. 9, and with the help of the buttons on the scrambler we set the input signal in binary which the descrambler detects and decodes. In the input signal, 1 is represented by the pressed key, and 0 is obtained without touching the key. In order to easily demonstrate the functionality of our prototype, we added an LED to the descrambler, which lights up when the button on the scrambler is pressed.

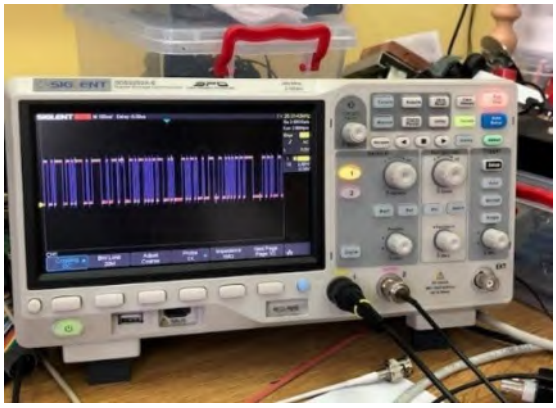


Fig. 9. The decoded bit sequence shown on the oscilloscope.

We concluded that when we have no excitation (tact) we have no signal on the oscilloscope. However, when the button on the scrambler was pressed, thus sending a 1 through the system, the diode on the descrambler lights up and the oscilloscope shows a presence of an impulse. Conversely, when a 0 is sent, the descrambler detects it correctly and the oscilloscope shows no impulse.

VII. Conclusion

The prototype we created and tested was functional and fully working. After analyzing the theoretical model of the descrambler, we performed the simulation of the scrambler-descrambler pair, since it was necessary to test various different values for some parameters. This process proved beneficial in helping us find a schematic that was both functional and easy to implement. Upon connecting our descrambler prototype to its corresponding scrambler and to the oscilloscope, we successfully managed to send, encode and finally decode a message. The device can be further upgraded so it descrambles more complex messages, such as audio signals. Despite being mostly familiar with the theoretical aspect of descramblers, this project taught us to design, implement and ultimately, create a working piece of device that is very common in the telecommunications field.



Fig. 10. Jana and Jovana during the soldering process.

ACKNOWLEDGEMENT

This paper is part of a student project from the subject of Digital Telecommunications 2 from the Department of Telecommunications at the Faculty of Electronic Engineering. The hardware implementation of the project was carried out at the laboratory for telecommunication systems, under the mentorship of prof. dr. Dejan Milić.

REFERENCES

[1] B. G. Lee and S. C. Kim, *Fundamentals of Scrambling Techniques*. In: *Scrambling Techniques for Digital Transmission. Telecommunication Networks and Computer Systems*. Springer, London, 1994. https://doi.org/10.1007/978-1-4471-3231-8_2.

[2] P. Premaratne and M. Premaratne, *Key-based scrambling for secure image communication*. In *International conference on intelligent computing*, pp. 259-263, Springer, Berlin, Heidelberg, July 2012.

[3] N. J. Corron, B. R. Reed, J. N. Blakely, K. Myneni and S. D. Pethel, "Chaotic scrambling for wireless analog video". *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 9, pp.2504-2513, 2010.

[4] W. Q. Yan, W. G. Fu and M. S. Kankanhalli, "Progressive audio scrambling in compressed domain", *IEEE Transactions on Multimedia*, vol. 10, no. 6, pp.960-968, 2008.

[5] EN 301 192, Specifications for Data Broadcasting, European Telecommunications Standards Institute (ETSI), 2004.

[6] J. Savage, "Some Simple Self-Synchronizing Digital Data Scramblers", *Bell System Technical Journal, The. 46*., pp. 449-487, 1967, DOI: 10.1002/j.1538-7305.1967.tb01066.x

[7] University of Waterloo, "Resistors", Available online at: Resistors | Water Analogy to Circuits | Douglas Wilhelm Harder | University of Waterloo (uwaterloo.ca)

[8] W. Duff, *A Text-Book of Physics (4th ed.)*, Philadelphia: P. Blakiston's Son & Co. p. 361.

[9] Diodes incomporated, "Quadruple 2-input exculsive or gates", Available online at: https://www.diodes.com/assets/Datasheets/74HC86.pdf

[10] Diodes incomporated, "8-bit shift register with 8-bit output register", Available online at: https://www.diodes.com/assets/Datasheets/74HC595.pdf

Redni i paralelni drajver za LE diodu za optičke komunikacije

Toplica Jakimov, Aleksandar Nikolić

Sadržaj − U ovom radu tema je projektovanje rednog i paralelnog drajvera. Na osnovu simulacije u LTSpice-u, nabavljene su sve potrebne komponente za realizaciju oba drajvera. I za redni i paralelni drajver su bile potrebne iste komponente, a razlika u realizaciji je u rasporedu samih komponenti. Nakon realizacije drajvera, na svakom od njih je dodata LED dioda, pri čemu je tranzistor svakog od njih korišćen kao prekidač koji uključuje i isključuje diodu. Na osciloskopu su posmatrani parametri *rise time* i *fall time*, kako bi se zaključilo koji drajver je brži.

I. UVOD

Drajver laserske diode je ključna komponenta laserske tehnologije jer obezbeđuje snagu koja osvetljava lasersku diodu. Ovaj važan deo laserskog sistema diktira tačnost, kontrolu i performanse lasera, a pravi za laser koji koristite često zavisi od tipa lasera koji imate. Pored toga, laserski drajver je zaštitnik laserske diode, ograničavajući struju i napon na bezbedne nivoe. Ova ograničenja mogu biti razlika između radnog lasera i mrtvog lasera. Laseri koriste struju da funkcionišu. Previše struje može oštetiti ili čak uništiti laser. Prekomerni rad na bilo čemu, bilo da je to deo tehnologije ili osoba, obično dovodi do nižeg kvaliteta rada. Ograničenje struje hardvera reaguje brzo kako bi se osiguralo da laser ne bude preopterećen.

U mnogim slučajevima, diodni drajver jednostavno treba da obezbedi konstantnu radnu struju, što rezultira kontinuiranim talasnim radom lasera sa približno konstantnom optičkom izlaznom snagom.

II. SIMULACIJA

Što se tiče simulacije nju je moguće izvesti u LTspice-u i to tako sto bi se napravili drajveri kao na Slici 1 i 2. Tokom realizacije nije korišćen nijedan program za obavljanje simulacije već je na osnovu prethodnih šema pređeno na direktnu praktičnu realizaciju.

III. KOMPONENTE PROTOTIPA

Prototip sadrži sledeće komponente:

- LE dioda;
- Otpornike;
- Kondenzatore;
- Tranzistor;
- Raster ploča;

Otpornik je pasivna elektronska komponenta koja pruža otpor električnoj struji, stvarajući pad napona između priključaka. U otporniku uložena električna energija većim delom nepovratno prelazi u drugi vid energije, najčešće u toplotnu, ali može i u svetlosnu, mehaničku itd.

Otpornici su uobičajeni elementi električnih i elektronskih kola, mogu biti sastavljeni od različitih materijala i imati različite oblike. Mogu biti stalni ili promenljivi.

LED je posebna vrsta poluprovodničke diode koja emituje svetlost kada kroz nju teče struja.

Kondenzator je elektronska komponenta koja služi za skladištenje elektriciteta. Sposobnost kondenzatora da primi naelektrisanje naziva se električni kapacitet. Što je vrednost veća, to u jednom trenutku kondenzator može da uskladišti više elektrona. Uskladišteni elektricitet se može iskoristiti po potrebi.

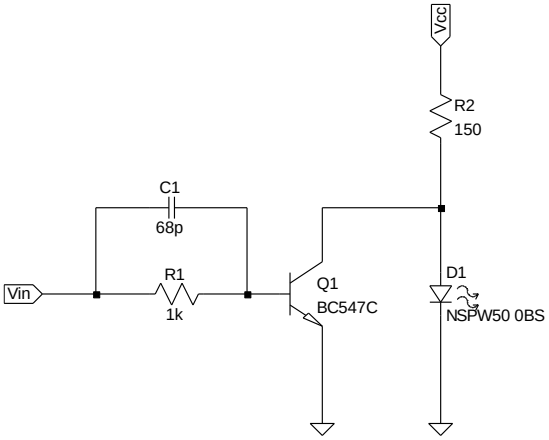
Tranzistor je aktivna poluprovodička komponenta s trima elektrodama, koja upravlja protokom struje tako što otvara i zatvara svoje „ventile“. Tranzistori se najčešće koriste kao pojačavači i prekidači signala, a takođe i za stabilizaciju napona, modulaciju signala i mnoge druge primjene.

Mogu bipolarni i unipolarni (tranzistori sa efektom polja). Pod tranzistorom se podrazumeva bipolarni, a kada se govori o unipolarnom to se posebno naglašava. Postoje i tranzistori osetljivi na svetlost, koji se uključuju kada se izlože dejstvu svetlosti.

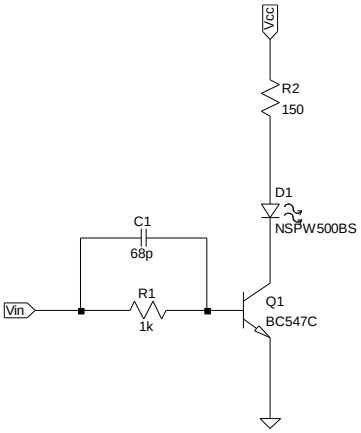
Ovo su osnovne komponente koje su nam potrebne kako bismo realizovali oba drajvera. Ovo znači da i redni i paralelni drajver imaju iste komponente i da je bitan samo raspored komponenti. Na sledećoj slici možemo videti šeme za redni i paralelni drajver i šemu kompletnog sistema koji smo koristili kako bismo zaključili koji drajver je brži.

Na slici 3 možemo videti da svetlost koju će drajver da emituje pasti na fotodiodu gde ćemo dalje koristiti terminator zatim imamo DC blok nakon toga imamo pojacavač koji ima ulaznu i izlaznu otpornost 50 Ω, a zatim imamo jos jedan terminator koji se naziva feedthrough terminator koji takođe ima otpornost 50 Ω. Na kraju imamo osciliskop koji nam služi da očitamo vrednosti i da donesemo zaključak koji je drajver brži.

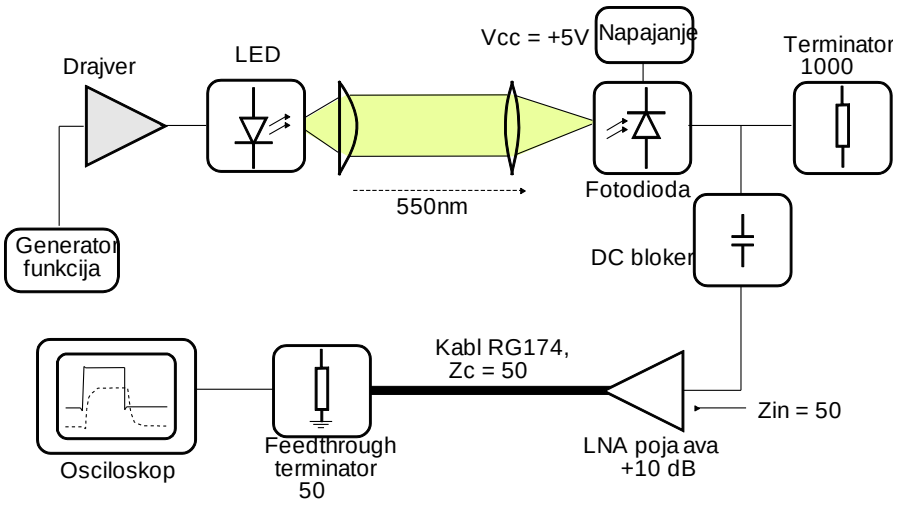
T. Jakimov i A. Nikolić su studenti Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: toplica@elfak.rs.



Slika 1. Šema paralelnog drajvera.



Slika 2. Šema serijskog drajvera.



Slika 3. Šema celog sistema za ispitivanje drajvera.

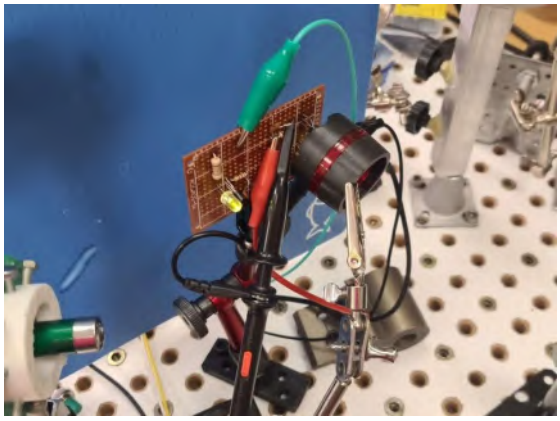
IV. PRAKTIČNA REALIZACIJA

Realizacija je obavljena na raster ploči tako što su na osnovu šema za redni i paralelni drajver pravilno spojene sve komponente. Takođe je korišćena jedna raster ploča dovoljno velikih dimenzija za realizaciju i rednog i paralelnog drajvera. Bez obzira što su na istoj ploči za vreme očitavanja rezultata korišćen je samo jedan od njih. Glavna razlika između njih je što tranzistor (koji nam služi kao prekidač da uključuje i isključuje diodu) ide redno ili paralelno sa LE diodom.

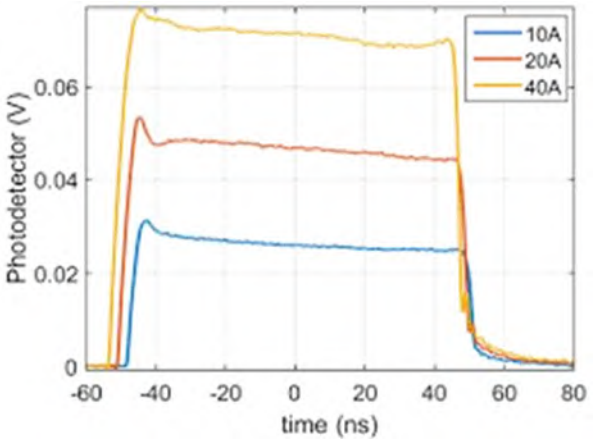
Na slici 4 prikazan je drajver koji je priključen i spreman da uz pomoć LED preda svetlost fotodiodi.

V. REZULTATI MERENJA

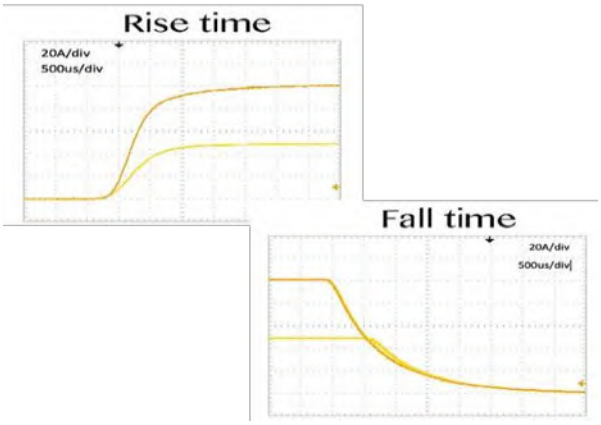
Rezultati merenja su dobijeni na osciloskopu. Vrednosti koje su očitavane jesu RISE i FALL vreme, odnosno vreme koje je potrebno da struja od nule dostigne neku maksimalnu vrednost, a nakon toga od te maksimalne vrednosti opada. Rise i fall time primeri prikazani su na slici 5.



Slika 4. Drajver koji je priključen i spreman za ispitivanje.



a)



b)

Slika 5. Primer rise i fall vremena.

Rezultat merenja sa osciloskopa jednog od drajvera prikazan je na slici 6.



Slika 6. Prikaz rezultata na osciloskopu.

Pored ovoga, obavljena su još nekoliko očitavanja kako bismo bili uvereni da nije napraviljena neka grešku. Izvršili su očitavanja za oba drajvera u 2 slučaja, kada imamo pojačavač i terminator od 50 Ω, a zatim i očitavanje bez njih. Rezultati su prikazani u sledećim tabelama:

TABELA I
REZULTATI MERENJA ZA PARALELNI DRAJVER

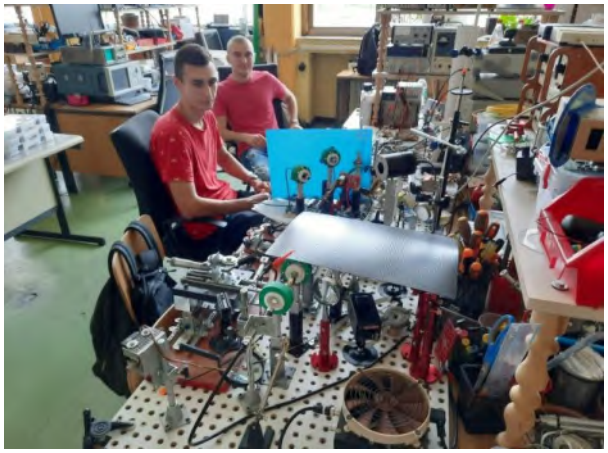
PARALELNI DRAJVER	Sa pojačavačem i terminatorom	Bez pojačavača i terminatora
RISE time	430 ns	1.50 ns
FALL time	598 ns	1.40 ns

TABELA II
REZULTATI MERENJA ZA REDNI DRAJVER

REDNI DRAJVER	Sa pojačavačem i terminatorom	Bez pojačavača i terminatora
RISE time	521 ns	1.88 ns
FALL time	807 ns	1.61 ns

VI. ZAKLJUČAK

Zadatak ovog projekta je bio da ustanovimo koji od ova dva drajvera je brži u slučaju kada je priključen pojačavač i terminator od 50 Ω i u slučaju kada nemamo pojačavac i terminator. Na osnovu dobijenih vrednosti sa osciloskopa nakon nekoliko očitavanja za oba drajvera u oba slučaja možemo doneti zaključak da je paralelni drajver brži od rednog drajvera u oba slučaja.



Slika 7. Proces merenja.

LITERATURA

[1] S. B. Alexander, "*Optical Communication Receiver Design*",, SPIE – The International Society for Optical Engineering, Bellingham, 1997.
[2] A. Marinčić, „*Osnovi optoelektronskih komunikacija*“, Građevinska knjiga, Beograd, 1986.

[3] D. Milić, „*Optičke telekomunikacije*“, Univerzitet u Nišu, Elektronski fakultet, Niš, 2016.
[4] G. P. Agrawal, „*Fiber-Optic Communication Systems*“, John Wiley & Sons, Inc., Hoboken, 2010.
[5] Vishay Semiconductors, „BPW34, BPW34S, Silicon PIN Photodiode“, kataloški podaci, Dostupno na: http://www.vishay.com/docs/81521/bpw34.pdf.
[6] Wavelength electronics. Dostupno na: https://www.teamwavelength.com/laser-diode-driver-basics/.
[7] „*Optoelektronske komponente*“ Dostupno na: 3-Optoelektronske-komponente.pdf (ni.ac.rs).

Merenje karakteristika fototranzistora i fotodiode

Jovana Nešić, Milica Mijajlović

Sadržaj – U ovom radu prikazane su osnovne karakteristike fotodiode i fototranzistora. U laboratoriji su bile dostupne obe komponente, a cilj ovog rada je da se njihove karakteristike praktično uporede kako bi se zaključilo za koje namene su podesne ove komponente. Kataloški podaci su pronađeni na internetu, a zatim verifikovani putem merenja u laboratoriji. Od karakteristika, kao najvažnija je posmatrana zavisnost odziva u funkciji frekvencije, kao i primenjenog napona. Postupak merenja uključuje odgovarajuću polarizaciju ovih komponenata, a praktična realizacija potrebnih elektronskih kola je izvedena na raster ploči. Optička pobuda komponenata je izvedena putem modulisanog poluprovodničkog lasera koji je bio dostupan u laboratoriji, dok su rezultati dobijeni posmatranjem i merenjem parametara talasnih oblika putem osciloskopa.

I. UVOD

Fotodioda reaguje na pojavu svetlosti generišući električnu struju. Fotoni izbijaju elektrone iz orbitala u oblasti spoja što je uzrok pojave električne struje. Fotodiode se mogu koristiti kao solarne ili fotonaponske ćelije i u fotometriji [1]. Ako foton nema dovoljno energije, neće pobuditi elektron i samo će proći kroz spoj. Čak se i svetleća dioda može upotrebiti kao foto dioda u nekim primenama, ali će tada njena efikasnost biti veoma niska. Nekada se svetleća dioda i fotodioda pakuju u isto kućište [2],[3]. Ovaj uređaj se tada zove "opto izolator", "opto kapler" ili "opto razdvajač". Slično kao što transformator omogućava galvansko razdvajanje naizmeničnog napona, optokapler omogućava galvansku izolaciju uključujući i jednosmernu komponentu. Ovo je izuzetno korisno, recimo kod zaštite pacijenata koji su priključeni na medicinske uređaje ili kada se osetljiva niskostrujna kola razdvajaju od problematičnih napojnih sklopova ili jakih elektromotora. Solarne ćelije su takođe jedna vrsta fotodiode. Unutrašnja građa fotodiode je slična građi obične diode. Razlika se sastoji u tome što je kod fotodiode PN spoj otkriven i na njega mogu da padaju svetlosni zraci. Kada svetlost padne na (osvetli) PN spoj, stvaraju se parovi šupljina-elektron [4].

Ako se dioda inverzno polariše kroz nju teče inverzno polarisana struja. Povećanje inverzne struje kroz diodu je približno srazmerno povećanju osvetljenosti. Fotodioda normalno radi u oblasti vidljive svetlosti i u infracrvenoj oblasti [5].

J. Nešić i M. Mijajlović su studenti treće godine osnovnih akademskih studija na Elektronskom fakultetu Univerziteta u Nišu, A. Medvedeva 14, 18000 Niš, Srbija, E-mail: jovananesic@elfak.rs, milica.mijajlovic@elfak.rs.



(a) (b)

Slika 1. a) Šematski simbol fotodiode, b) Izgled fotodiode.

Fototranzistor je bipolarni tranzistor koji pripada grupi optoelektronskih komponenata. Realizuje se tako da mu je oblast baze izložena dejstvu upadne svetlosti. Fototranzistor se polariše tako da mu je kolektor na pozitivnom potencijalu u odnosu na emitor. Pod dejstvom incidentne svetlosti, unutar osiromašene oblasti p–n spoja baza–kolektor dolazi do generacije parova elektron–šupljina. Pošto je spoj baza–kolektor inverzno polarisan, šupljine iz osiromašene oblasti prelaze u bazu, a elektroni u kolektor, čineći na taj način fotostruju. Usled toga se povećava pozitivni potencijal baze u odnosu na emitor.

Ako su fotoni svetlosti koji padaju dovoljno visoke frekvencije, generisaće struju baze koja nakon pojačanja postaje struja kolektora. Fototranzistori stoga imaju samo dva izvoda: kolektor i emitor.

Fototranzistor se može upotrebiti i u aktivnom režimu rada, tipično kao senzor intenziteta svetlosti. Međutim, generisana fotostruja nije idealno linearno proporcionalna intenzitetu upadne svetlosti. U tom slučaju je na izlazu tranzistora potrebno dodatno elektronsko kolo koje će izvršiti linearizaciju odziva tranzistora na svetlosnu pobudu. Za proizvodnju fototranzistora se, pored silicijuma, koriste i poluprovodnička jedinjenja (GaAlAs, GaAs, InGaP, itd.). Fototranzistori su u širokoj upotrebi u elektronskim uređajima koji služe za detekciju svetlosti, kao što su detektori dima [6].

Koriste se i u uređajima kao što su CD plejeri i laseri.



(a) (b)

Slika 2. a) Šematski simbol fototranzistora b) Izgled fototranzistora.

Ključne razlike između fotodioda i fototranzistora [5]:
1. Fotodioda je poluprovodnički uređaj koji pretvara energiju svetlosti u električnu struju. Dok fototranzistor koristi tranzistor za konverziju svetlosne energije u električnu struju.
2. Fototranzistor stvara struju dok fotodioda proizvodi i napon i struju.
3. Odziv fotodiode je mnogo brži od fototranzistora.
4. Fotodioda je manje osetljiva u odnosu na fototranzistor jer fototranzistor proizvodi veliku izlaznu struju.
5. Fotodioda se koristi u solarnoj elektrani, u meraču svetlosti, itd. Dok se fototranzistor koristi za detekciju svetlosti.

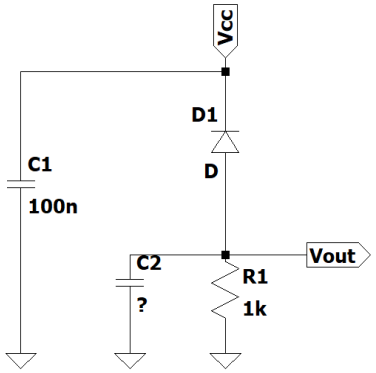
II. POSTUPAK MERENJA

Prvo smo krenuli sa dostupnim komponentama. Fotodiodu smo identifikovali pomoću datasheet-a kao BPW34 [7] a fototanzistor kao PT204-6C [8]. Kod fotodiode struja mraka je 2 nA, max inverzni napon je 60 V, a takodje je inverzni napon 60 V i talasne dužina je 900 nm. Kod fototranzistora pronašli smo sledeće karakteristike:

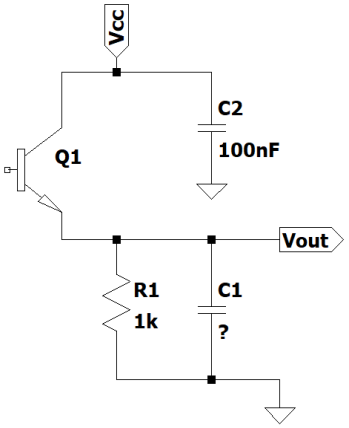
Kada imamo struju svetlosti od 100 uA tada je Vce=5V, a Ev=100 lx. S druge strane kada je struja svetlosti 1000 uA, napon izmedju kolektora i emitora je 5V a Ev= 1000 lx. Opseg talasne dužine je od 370 do 670 nm, dok je optimalna talasna dužina 630 nm.

Zatim smo pronašli sve potrebne elemente, kao što su raster ploča za lemljenje, kalaj, lemilica, otpornik, fotodioda, fototranzistor i kondenzator i krenuli u postupak lemljenja. Cilj nam je bio da električno kolo polarizacije bude identično u oba dva slučaja i da jedina razlika u ova dva kola bude fotoosetljivi element. Na slici 3. je prikazana šema koju smo koristili za tesitranje fotodiode. Napajanje u kolu fotodiode je predvidjeno zbog toga što inverzni napon polarizacije smanjuje prostornu kapacitivnost na diodi. Iz kataloskih podataka diode smo pročitali da za inverzni napon od 5V, prostorna kapacitivnost diode iznosi oko 20pF. Kondenzator C1 u kolu sprečava brze promene inverznog napona Vcc koje mogu nastati kao posledica različitih smetnji ili indukovanja struje usled spoljašnjih smetnji. Otpornik R1 predstavlja opterećenje fotodiode. Ukoliko je ovaj otpornik veliki onda će mala fotostruja da na njemu proizvede veći napon. Ukoliko je otpornik mali onda fotostruja proizvodi mali napon. Na taj način opterećenje direktno odredjuje osetljivost u kolu fotodiode. Vrednost od 1kΩ je kompromis. Kondenzator C2 predstavlja parazitnu kapacitivnost, kao i prostornu kapacitivnost fotodiode. Potrebno je da ova vrednost bude što manja kako bi propusni opseg bio što veći.

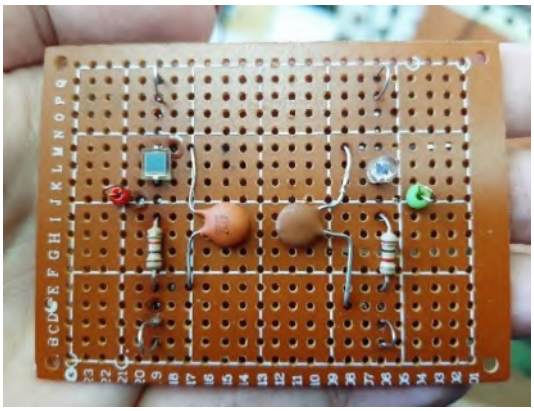
Na slici 4. je prikazana šema koju smo koristili za testiranje fototranzistora. Napajanje je neophodno da bi tranzistor NPN tipa funkcionisao. Ostale komponente imaju istu ulogu kao u prethodnom kolu. Možemo zaključiti da sa povećanjem intenziteta svetlosti raste i kolektorska struja. U ovom slučaju nismo koristili pojačavač zato što smo želeli da minimizujemo potencijalna izobličenja ili ograničenja propusnog opsega koje pojačavač može da unese.



Slika 3. Šema kola za testiranje fotodiode.



Slika 4. Šema kola za testiranje fototranzistora.

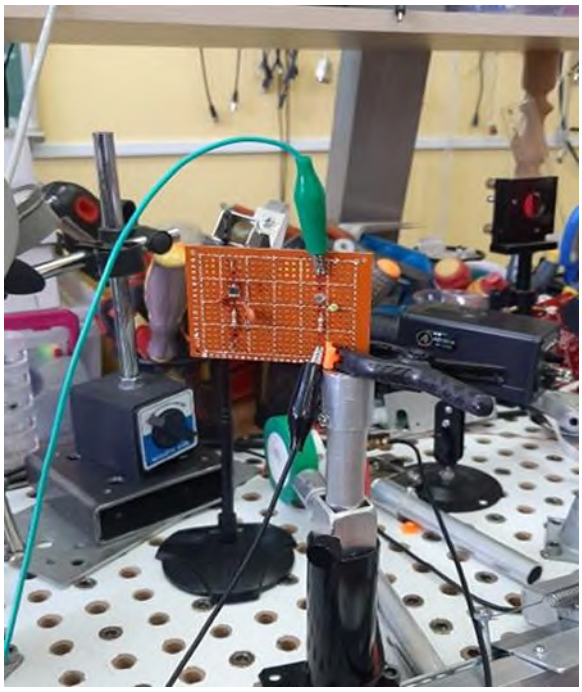


Slika 5. Raster ploča sa komponentama.

Prilikom realizacije potrudili smo se da raspored elemenata i njihove vrednosti budu identične u obadva slučaja, da bismo mogli da uradimo fer poredjenje pod istim uslovima. Prema tome jedina razlika u realizaciji ovih kola se sastoji u tome da li je aktivni element fotodioda ili fototranzistor.

Nakon lemljenja, zadatak je bio da se izmere karakteristike fotodiode i fototranzistora. Postavili smo

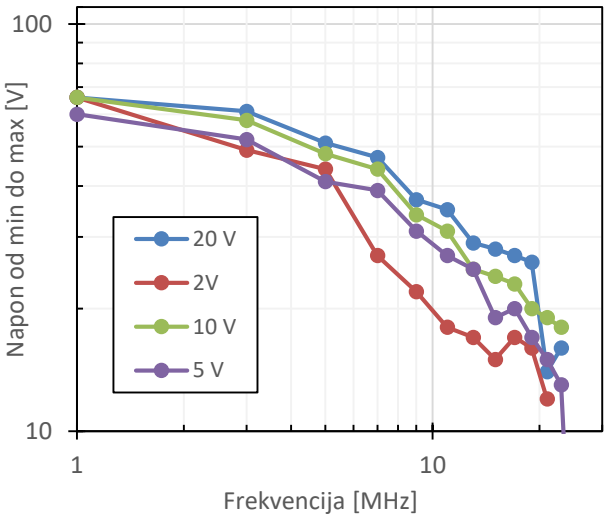
ploču i uperili laser u fotodiodu odnosno fototranzistor. Pobudili smo laser pravougaonim impulsom, menjali frekvenciju i na osnovu toga smo posmatrali zavisnost odziva u funkciji frekvencije. Za realizaciju smo izabrali laser crvene boje čija je talasna dužina 650 nm. Uz pomoć grafika očitali smo da je relativna spektralna osetljivost fotodiode 60 % od maksimalne koja se dostiže na talasnoj dužini od 850 nm. Dok je relativna spektralna osetljivost fototranzistora maksimalna za talasnu dužinu od 900 nm, a osetljivost fototranziszora na talasnoj dužini 650 nm iznosi 70 % od maksimalne.



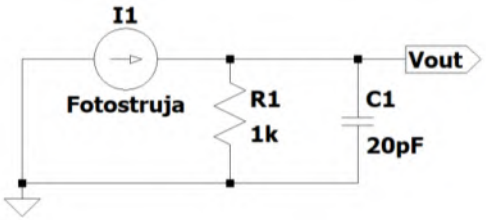
Slika 6. Postupak merenja.

Na kraju povezali smo napon napajanja Vcc i masu sa naponskim generatorom, a na izlazu Vout smo zakačili sondu osciloskopa. Kao generator napona Gwinstek GPS-4303, generator signala za pobudu lasera koristili smo Siglent SDG830 ko koji podržava izlazne frekvencije do 30 MHz, a osciloskop je bio Siglent SDS2202X-E koji ima propusni opseg 200 MHz tako da zadovoljava zahteve u pogledu posmatranja pomenutih signala. Na taj način smo mogli da pratimo dinamiku signala kada se menja frekvencija optičke pobude.

Na slici 5. je prikazana zavisnost izlaznog napona (od vrha do vrha) kada se menja frekvencija optičke pobude. Prikazane su slike za različite vrednosti inverznog napona. Inverzni napon ima veoma mali uticaj kada su frekvencije pobude male. Medjutim za velike frekvencije ovaj uticaj je mnogo veći i može se primetiti da propusni opseg raste kada se povećava inverzni napon. Za inverzni napon od 20 V procenili smo da je propusni opseg oko 13 MHz, dok je za 5 V propusni opseg 9 MHz. Kada se inverzni napon smanji na 2 V propusni opseg opada do vrednosti 7 MHz.



Slika 7. Zavisnost amplitude primljenog signala sa fotodiode u funkciji frekvencije i za različite vrednosti inverznog napona na fotodiodi.



Slika 8. Teorijski model za analizu propusnog opsega.

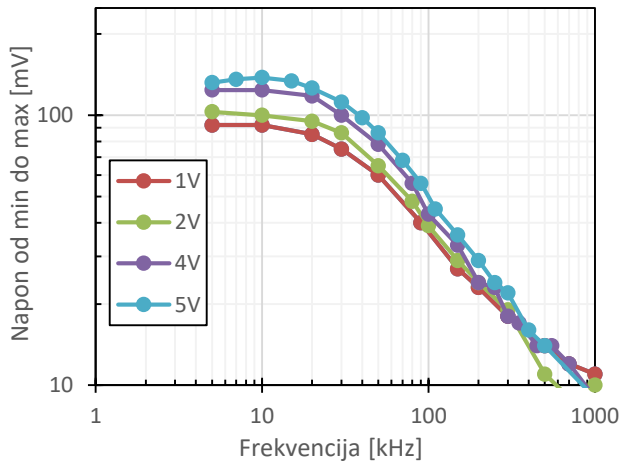
Da bismo poredili teorijske vrednosti koristili smo model prokazan na Slici 7. Opterećenje fotodiode vrednosti R1 = 1kΩ, ukupna kapacitivnost fotodiode i parazitnih elemenata iznosi 20pF kao sto smo već pomenuli za inverzni napon polarizacije od 5V. Teorijski se propusni opseg kola na Slici 7, se može izračunati

$$f_0 = \frac{1}{2\pi RC} \tag{1}$$

Korišćenjem prethodne formule dobija se da je teorijska vrednost propusnog opsega 8MHz, što je u saglasnosti sa rezultatima koje smo i mi dobili.

Kada je u pitanju fototranzistor uočavamo da odziv opada sa frekvencijom i da je dobro definisana gornja granična frekvencija. Za razliku od fotodiode ovde smo uočili da odziv zavisi od napona polarizacije, čak i na niskim frekvencijama. Što je veći napon napajanja veća je i amplituda odziva. Uočava se da propusni opseg ne zavisi značajno od napona polarizacije, što je suštinska razlika u odnosu na fotodiodu. Pored toga dok su propusni opsezi u slučaju fotodiode bili reda veličine nekoliko MHz, kod fototranzistora je ova vrednost izražena u kilohercima. U kataloškim podacima videli smo da je maksimalni napon napajanja manji nego kod fotodiode, pa smo zbog toga

koristili manje vrednosti. Za napon napajanja od 5V smo sa Slike 6. procenili da propusni opseg iznosi oko 60kHz. Sa druge strane kada je napon napajanja 1V, ova vrednost iznosi oko 80kHz. Zaključujemo da sa povećanjem napona polarizacije opada propusni opseg, ali ovo smanjenje je relativno malo i nema neki suštinski značaj. Sa druge strane odziv fototranzistora je veći nego u slučaju fotodiode.



Slika 9. Zavisnost amplitude primljenog signala sa fototranzistora u funkciji frekvencije i za različite vrednosti napona između kolektora i emotora.

Koristeći ovaj model možemo da procenimo vrednost ekvivalentne kapacitivnosti fototranzistora koja iznosi 2.65nF.

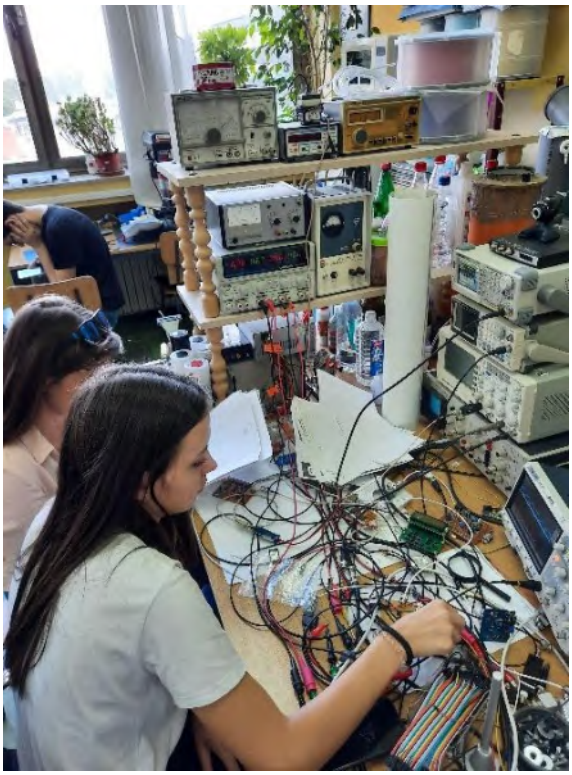
III. Zaključak

Na osnovu relizacije ovog projekta utvrdili smo da je amplituda signala veća kod fototranzistora u odnosu na fotodiodu. Sa druge strane, propusni opseg fotodiode je znatno veći u odnosu na propusni opseg koji ima fototranzistor. Promenom napona između kolektora i emitora na fototranzistoru, menja se i amplituda izlaznog signala, dok frekvencijski odziv ostaje gotovo isti. Kod fotodiode smo primetili da sa povećanjem inverznog napona raste i granična frekvencija, dok amplituda izlaznog signala ostaje približno ista.

Na osnovu ovoga, zaključujemo da je merenjem karakteristika nepoznatnog elementa moguće utvrditi da li se radi o fototranzistoru ili fotodiodi, čak i kada unapred nije poznata ova informacija.

Zahvalnica

Ovaj rad napisan je kao deo studentskog projekta na kursu Optičke komunikacije, na smeru Telekomunikacije, Elektronskog fakulteta, Univerziteta u Nišu. Kurs je na 6. semestru osnovnih akademskih studija. Hardverska realizacija projekta izvedena je pri Laboratoriji za telekomunikacione sisteme, pod mentorstvom prof. dr Dejana Milića.



Slika 10. Izvodjenje eksperimenta u laboratoriji 304.

Literatura

[1] D. Milić, „Optičke telekomunikacije“, Univerzitet u Nišu, Elektronski fakultet, Niš, 2016.
[2] A. Marinčić, „Osnovi optoelektronskih kumunikacija“, Građevinska knjiga, Beograd, 1986.
[3] Wikipedia,”Fototranzistor”, dostupno putem interneta na: https://sh.wikipedia.org/wiki/Fototranzistor, datum pristupa 09.09.2022.
[4] G. P. Agrawal, "Fiber-Optic Communication Systems", John Wiley & Sons, Inc., Hoboken, 2010.
[5] Illustrationprize, ”Difference between photodiode and phototransistor”, dostupno putem interneta na adresi: https://illustrationprize.com/sr/59-difference-between-hotodiode-amp-phototransistor.html, datum pristupa 09.09.2022.
[6] “Optoelektronske komponente” Dostupno putem interneta na sajtu: 3-Optoelektronske-komponente.pdf (ni.ac.rs) , datum pristupa 11.09.2022
[7] Vishay Semiconductors, "BPW34, BPW34S, Silicon PIN Photodiode“, kataloški podaci, dostupno putem interneta na: http://www.vishay.com/docs/81521/bpw34.pdf
[8] Everlight, „3mm Phototransistor PT204-6C“, kataloški podaci, dostupno putem interneta na adresi: https://everlighteurope.com/index.php?controller=attachment&id_attachment=5339

Projektovanje i realizacija samosinhronišućeg skremblera

Toplica Jakimov, Aleksandar Nikolić

Sadržaj − U ovom projektu smo realizovali skrembler kao deo sistema sa samosinhronišućim deskremblerom. Cilj nam je bio da realizujemo prototip koji će nam pokazati da je princip dobar i da realizacija može da se izvede relativno jednostavno. Za realizaciju smo odabrali standardna integrisana kola digitalne logike koja su dostupna po veoma niskim cenama i ispunjavaju zahteve koje smo postavili. Realizacija je pokazala da je koncept bio ispravan i da se ovakva vrsta skremblera može jednostavno realizovati. Ustanovili smo da ovaj prototip može da se koristi za bitske brzine do 1Mb/s.

I. UVOD

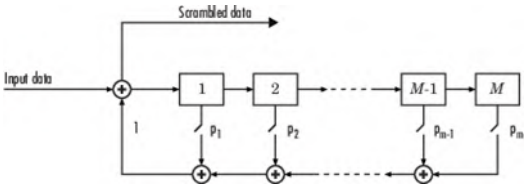
Zadatak ovog projekta je projektovanje skremblera. U telekomunikacijama, skrembler je uređaj koji transponuje ili invertuje signale ili na drugi način kodira poruku na strani pošiljaoca kako bi poruku učinio nerazumljivom kod primaoca koji nije opremljen odgovarajućim uređajem za dekodovanje. Dok se šifrovanje obično odnosi na operacije koje se izvode u digitalnom domenu, kodiranje se obično odnosi na operacije koje se izvode u analognom domenu. Šifrovanje se postiže dodavanjem komponenti originalnom signalu ili promenom neke važne komponente originalnog signala kako bi se otežalo izdvajanje originalnog signala.

U digitalnim telekomunikacijama nam je mnogo značajnija jedna druga osobina skremblera, a to je da u skremblovanom signalu imamo podjednak broj nula i jedinica. Pored toga, ne postoje dugački nizovi identičnih bitova, što je veoma važno prilikom ekstrakcije digitskog takta u digitalnim komunikacijama. Prema tome, signal koji skrembler generiše ima osobine pseudoslučajnog niza bitova, a što je najbitnije, on ovu osobinu prenosi i na ulazni digitalni signal. Usled toga, ulazni signal se pretvara u naizgled slučajni niz bitova. Sa druge strane ovaj niz bitova je moguće vratiti u originalni oblik putem operacije deskremblovanja. Postoje različite vrste skremblera i deskremblera.

Zbog jednostavnosti deskremblovanja, odlučili smo se za princip samosinhronišućeg skremblovanja. Cilj nam je da realizujemo skrembler najednostavnije moguće uz što manji broj komponenti.

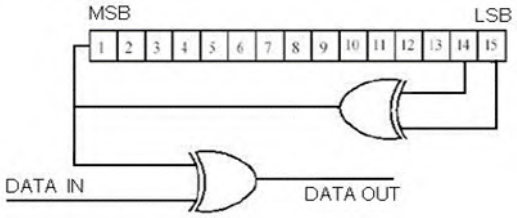
II. FUNKCIONALNA ŠEMA

S obzirom da je literatura na srpskom dosta skromna, šemu skremblera smo našli u stranoj literaturi na internetu [4][5]. Ova šema u opštem slučaju predstavlja transverzalni filtar gde se podaci iz istorije signala kombinuju sa trenutnim ulazom. Ono što je neophodno jeste odrediti koji su koeficijenti potrebni da bi ovaj transverzalni filtar funkcionisao kao skrembler.



Slika 1. Opšta šema skremblera kao transverzalnog filtra.

Ispostavlja se da je najveći broj koeficijenata jednak nuli i da je za najednostavniju realizaciju dovoljno da postoje bar dva koeficijenta koji su razliciti od nule. Na slici 2 prikazana je ovakva situacija u kojoj postoje samo dve povratne sprege.



Slika 2. Izgled jednog tipičnog skremblera.

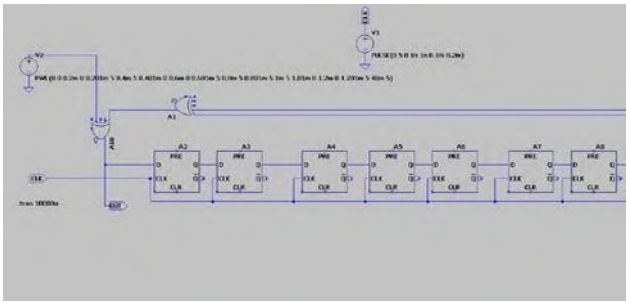
Gledajući ovu šemu možemo da vidimo da je realizacija ovog skremblera jednostavna i da su za realizaciju korišćeni pomerački registri i XOR logička kola. Pošto nam se ova šema učinila jednostavnom odlučili smo da nju iskoristimo za realizaciju.

III. SIMULACIJA

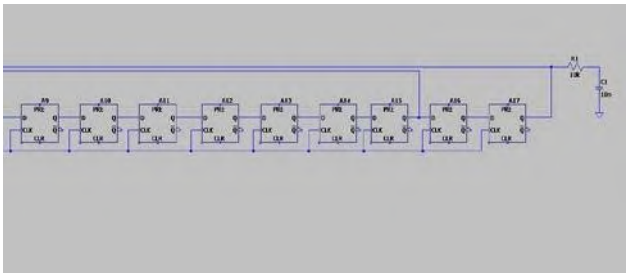
Najpre smo u LTspice-u napravili šemu skremblera i zamislili smo ga da bude sa 16 ćelija, takođe će da koristi i 2 XOR kola. Na sledećim slikama je prikazana kompletna šema skremblerau LTspice-u. Ulazni deo skremblera je identičan kao u prethodnoj šemi koju smo našli u literaturi. Pomerački registar smo simulirali kao niz D flip – flopova koji imaju zajednički signal takta. Na slici b) vidi se da smo uzeli izlaz sa 14-og i 16-og D flip-flopa. Otpornik R1 i kondenzator C1 predstavljaju filtar propusnik niskih

T. Jakimov i A. Nikolić su studenti treće godine osnovnih akademskih studija na Elektronskom fakultetu Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: toplica@elfak.rs.

frekvencija i u ovoj simulaciji nam služe kao indikator kvaliteta pseudslucajne sekvence. Ukoliko je signal na izlazu ovog filtra jednak polovini napona napajanja onda znamo da u sekvenci imamo isti broj nula i jedinica što nam potvrđuje da je skrembler ispunio svoju funkciju.



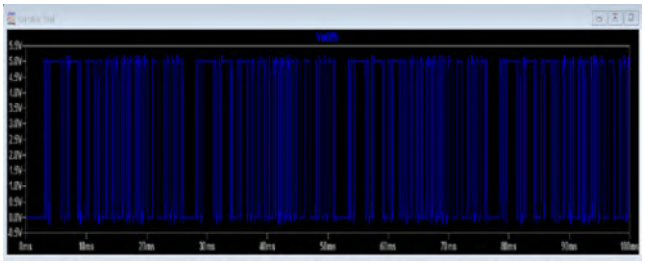
a)



b)

Slika 3. Simulaciona šema u Ltspice-u a) prvi deo šeme, b) drugi deo šeme.

Naravno, ovakvu šemu ne možemo realizovati na ploči već nam ona samo pokazuje kakve bi rezultate trebali da dobijemo kada završimo realizaciju skremblera. I takođe na ovoj šemi možemo izvršiti sva potrebna testiranja koja se odnose na pravilan izbor ćelija koje treba povezati sa XOR kolom kako bismo dobili sto dužu sekvencu koja se ne ponavlja. Realizacija ovakve šeme nije moguća iz prostog razloga što D flip-flopovi ne postoje u ovakvom obliku već u obliku integrisanog kola kao što je prikzano na slici 5.



Slika 4. Talasni oblik signala(napona) nakon pokretanja simulacije u LTspice-u.

Na slici 4 vidimo da je grafički realizovan talasni oblik napona. Razmera grafika je u voltima na y-osi, a u milisekundama na x-osi. Na x-osi je predstavljeno vreme trajanja sekvence dok na y-osi je predstavljen napon.

IV. KARAKTERISTIKE KOMPONENTI

Za realizaciju navedenog protipa korišćene su sledeće komponente:

1. Shift register x2 - IC TTL-H.S.CMOS 74HC595
2. Xor – IC TTL-H.S.CMOS 74HC86
3. Taster
4. Raster ploca tip 1 50x100
5. Konektor tipa C
6. LE Dioda 7.Otpornik 8.Kondenzatori

74HC595 se sastoji od 8-bitnog registra pomeranja i 8-bitnog D-tipskog latcha sa paralelnim izlazima u tri stanja. Registar pomeranja prihvata serijske podatke i obezbeđuje serijski izlaz. Registar pomeranja takođe obezbeđuje paralelne podatke 8- bitnoj vezi. Ovaj uređaj takođe ima asinhronovano resetovanje registra pomeranja. HC595 se direktno povezuje sa SPI serijskim portom podataka sa CMOS MPU-ima i MCU-ovima.

Features

- Output Drive Capability: 15 LSTTL Loads
- Outputs Directly Interface to CMOS, NMOS, and TTL
- Operating Voltage Range: 2.0 to 6.0 V
- Low Input Current: 1.0 μA
- High Noise Immunity Characteristic of CMOS Devices
- In Compliance with the Requirements Defined by JEDEC Standard No. 7A
- ESD Performance: HBM > 2000 V; Machine Model > 200 V
- Chip Complexity: 328 FETs or 82 Equivalent Gates
- Improvements over HC595
  - Improved Propagation Delays
  - 50% Lower Quiescent Power
  - Improved Input Noise and Latchup Immunity
- These are Pb−Free Devices

74HC86 obezbeđuje četiri nezavisna dvoulazna ILI kola sa standardnim push-pull izlazima. Uređaj je dizajniran za rad sa opsegom napajanja od 2.0 V do 6.0 V.

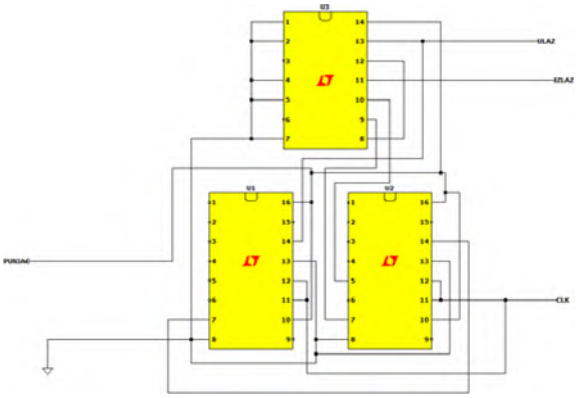
Features

- Wide Supply Voltage Range from 2.0V to 6.0V
- Sinks or sources 4mA at V_{CC} = 4.5V
- CMOS low power consumption
- Schmitt Trigger Action at All Inputs
- ESD Protection Exceeds JESD 22
  - 200-V Machine Model (A115-A)
  - 2000-V Human Body Model (A114-A)
  - Exceeds 1000-V Charged Device Model (C101C)
- Range of Package Options SO-14 and TSSOP-14
- **Totally Lead-Free & Fully RoHS Compliant (Notes 1 & 2)**
- **Halogen and Antimony Free. "Green" Device (Note 3)**

LED je posebna vrsta poluprovodničke diode koja emituje svetlost kada kroz nju teče struja.

Taster u našem prototipu služi za kreiranje sekvence nula ili jedinica koja se u XOR kolu sabira sa sekvencom koja dolazi iz pomeračkih registra.

Da bismo mogli da realizujemo skrembler na ploči moramo napraviti novu šemu u LTspice-u koja pokazuje šta treba da bude spojeno sa čim na nivou integrisanih kola.

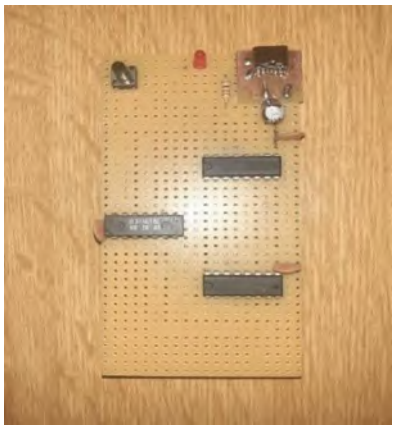


Slika 5. Šema koju ćemo realizovati.

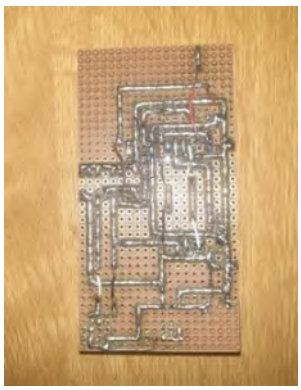
Na slici 5 je prikazana šema koju smo predvideli za realizaciju. Komponente U1 i U2 predstavljaju pomeračke registre dok komponenta U3 predstavlja XOR logičko kolo. Naziv komponenata koje smo koristili za ćelije je 74HC595 a XOR-a je 74HC86.

V. REALIZACIJA

Nakon što imamo sve potrebne komponente i šemu za realizaciju ovog prototipa mozemo da krenemo sa lemljenjem odnosno realizacijom na raster ploči. Izabrali smo layout koji je prikazan na sledećoj slici. Nakon toga smo krenuli sa lemljenjem komponenti na osnovu šeme koju smo predhodno napravili (slika 5). Na slici 6 nam je predstavljeno kako bi sve trebalo da izgleda kada se konačno realizuje:



a)



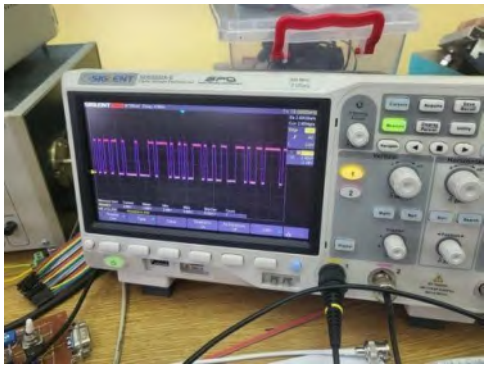
b)

Slika 6. Predstavljen je izgled realizovanog prototipa.

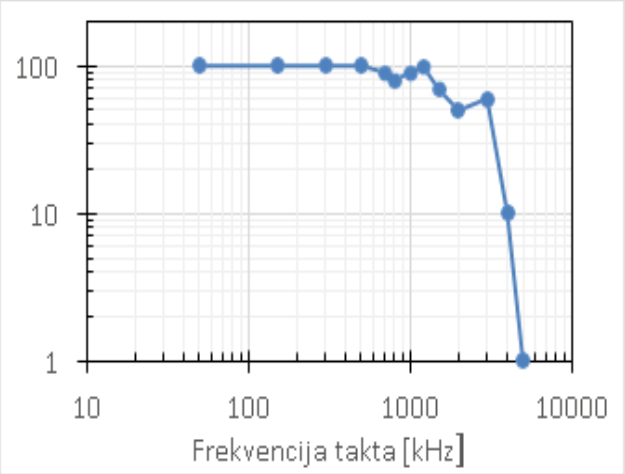
Pokušali smo da maksimalno izbegnemo korišćenje žica jer one mogu da smanje pouzdanost našeg prototipa, naravno zbog prevelikog povezivanja i takvog layout-a nismo mogli u potpunosti da izbegnemo njihovo korišćenje. Takodje smo prilikom lemljenja vodili racuna da ne napravimo kratke spojeve odnosno da ne spojima 2 lema koja ne trebaju da budu spojena.

VI. REZULTATI MERENJA

Rezultate naše realizacije smo očitali na osciloskopu na koji je bio priključen naš skrembler. Izvrsili smo nekoliko provera i očitavanja kako bismo došli do što pouzdanijih rezultata. Skrembler smo testirali na nekoliko različitih frekvencija kako bismo ustanovili na kojim najvisim frekvencijama može pouzdano da radi. Na svakoj frekvenciji na kojoj smo testirali skrembler, taster smo kliknuli 10 puta i onda videli koliko puta je skrembler radio. Za nas je zadovoljavajući rezultat bio 8 ili vise klika na koje je skrembler radio. Najpre smo krenuli od 50 kHz sve do 10 MHz na osnovu čega smo zaključili da ovaj skrembler koji smo realizovali ima dobru pouzdanost rada na frekvencijama manjim od 5 MHz, a na frekvencijama većim od 5 MHz ne ispunjava naše uslove rada. Nakon izvršene provere i merenja dobili smo rezultate koji su prikazani na grafiku.



Slika 7. Talasni oblici signala snimljeni pomoću osciloskopa.



Slika 8. Grafički prikaz pouzdanosti realizovanog skremblera u zavisnosti od frekvencije na kojoj radi.

VII. Zaključak

Realizacija je bila uspešna. Uspeli smo da pokažemo da je moguće realizovati šeme koje smo zamislili i napravili u LT spice-u. Zaključili smo da naš prototip može pouzdano raditi na frekvencijama od 100 kHz do 1.5 MHz. Na frekvencijama višim od 1.5 MHz naš prototip ne zadovoljava uslove koje smo predhodno postavili. Moguće je ostvariti efikasniji rezultat boljim layout-om (tako što ćemo eliminisati provodnike odnosno smanjenjem korišćenja žica kako bismo povezali elemente). Ostvareni rezultat ispunjava početne planove projektovanja.

Zahvalnica

Ovaj rad napisan je kao deo studentskog projekta na kursu Digitalne telekomunikacije 2, na smeru Telekomunikacije, Elektronskog fakulteta Univerziteta u Nišu. Kurs je na 5. semestru osnovnih akademskih studija. Hardverska realizacija projekta izvedena je pri Laboratoriji za telekomunikacione sisteme, pod mentorstvom prof. dr Dejana Milića.

Literatura

[1] DIODES INCOMPORATED, “Pomerački registri”, dostupno putem interneta na adresi: https://www.diodes.com/assets/Datasheets/74HC595.pdf

[2] DIODES INCOMPORATED, ”XOR logička kola”, dostupno putem internet na adresi: https://www.diodes.com/assets/Datasheets/74HC86.pdf

[3] Knjiga uvod u digitalne telekomunikacije Autori(Predrag Ivanis I Vesna Blagojevic)

[4] MathWorks, ”Scrambler”, dostupno putem internet na: https://www.mathworks.com/help/comm/ref/scrambler.html, datum pristupa 23.09.2022.

[5] RF Wireless World, “Scrambler MatLab source code”, dostupno puteminternet na adresi: https://www.rfwireless- world.com/source-code/MATLAB/scrambler-descrambler- matlab-code.html, datum pristupa 23.09.2022

[6] Wikipedia, “Dioda”, dostupno putem internet na adresi: https://sr.wikipedia.org/wiki/%D0%94%D0%B8%D0%BE%D0%B4%D0%B0, datum pristupa 23.09.2022

[7] Wikipedia, “Otpornik”, dostupno putem internet na adresi: https://sh.wikipedia.org/wiki/Otpornik, datum pristupa 23.09.2022

[8] Wikipedia, “Kondenzator”, dostupno putem internet na adresi: https://sr.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D0%B4%D0%B5%D0%BD%D0%B7%D0%B0%D1%82%D0%BE%D1%80, datum pristupa 23.09.2022

Sistem osvetljenja baziran na fotootporniku i LED diodi

Teodora Bogdanović, Mina Dikić

Sadržaj – U ovom radu je prikazan primer projektovanja uređaja kojim će se pomoću fotootpornika meriti osvetljenje i u zavisnosti od njegove jačine, LED dioda će se uključivati mikrokontrolerom.

I. UVOD

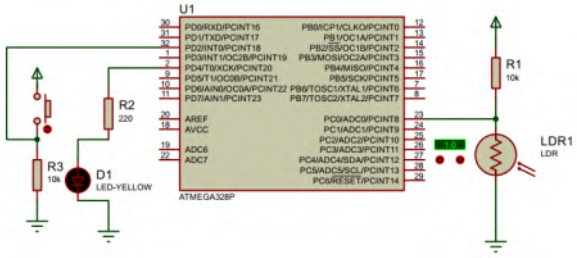
Trenutna globalna dešavanja poput pandemije, rata, toplotnih talasa i suša su glavni uzrok za energetsku krizu svuda u svetu. To zahteva preduzimanje određenih mera od strane društva, ne bi li došlo do nepoželjnih posledica. Usled nedostatka energenata i poskupljenja struje, došlo je do smanjenja korišćenja iste.

Jedan od načina smanjenja potrošnje jeste preko javne rasvete, koja se pokazala kao jedan od najvećih potrošača. Ovaj sistem osvetljenja je planiran da se koristi upravo u tu svrhu. Svojim funkcionisanjem mogao bi da smanji ukupnu količinu potrošene energije za čak 10%. Njegov rad se bazira na svega par komponenata čija bi uloga bila detektovanje svetlosti – pri čemu bi se sijalica ugasila, a ako svetlosti nema, sijalica bi se automatski uključila ili isključila. Taj proces može prekinuti ugrađeni taster koji po volji korisnika upravlja sijalicom.

II. PRINCIP RADA SISTEMA

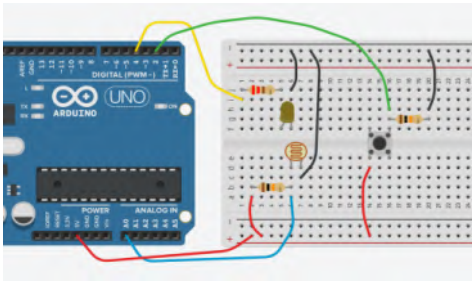
Ovaj sistem se zasniva na Arduino Uno razvojnom okruženju [1]. Glavna komponenta ovog sistema jeste fotootpotnik. To je otpornik čiji se otpor menja pod uticajem svetlosti koja pada na njega tako što fotoni iz svetlosti daju vezanim elektronima u provodniku dovolno energije da postanu slobodni što prouzrokuje njegovu manju otpornost [2]. On je zadužen za detektovanje svetlosti i slanje signala sijalici (u ovom slučaju LED diodi). Ukoliko detektuje određenu jačinu svetlosti, LED dioda će se isključiti. U suprotnom, pri manjku iste, ona će se uključiti. Na slici 1 je prikazana električna šema ovog sistema. Potrebna su tri pina sa Arduino okruženja, gde je jedan pin zadužen da simulira rad sijalice, u ovom slučaju LED diode. Drugi pin koji je iskorišćen je povezan u konfiguraciji naponskog razdelnika i preko njega se očitava očitava izlazni napon između otpornika i fotootpornika. U slučaju da je u prostoriji jaka svetlost, fotootpornik će imati malu otpornost, pa će i izlazni napon sa naponskog razdelnika biti manje vrednosti. Za slučaj da se fotootpornik nalazi u mraku, njegova otpornost će biti velika, pa će se povećati i vrednost izlaznog napona.

U ovaj sistem osvetljenja je uključen i taster u konfiguraciji sa *pull down* otpornikom koji je postavljen između pina 2 Arduino okruženja i GND-a tj. mase. Taster ima ulogu da prekine merenje izlaznog napona na fotootporniku i direktno uključi LED diodu, te ostvaruje ulogu prekidača i svetla kao i kod klasičnog osvetljenja u prostorijama. Ovo znači da je moguće kontrolisati i uključivati LED diodu čak i ako nije detektovan nedostatak svetlosti u prostoriji, a isto tako je moguće i isključiti LED diodu ukoliko za tim ima potrebe. To bi doprinelo dodatnoj uštedi električne energije i daje korisniku potpunu kontrolu nad uređajem.



Slika 1. Električna šema uređaja.

Na slici 2 je prikazana šema povezivanja uređaja na Arduino razvojnom okruženju urađena u softverskom paketu *TinkerCAD* [3]. Nakon povezivanja komponenata na protoploči, potrebno je napisati kod programa.



Slika 2. Šema povezivanja na uređaju [2].

III. KOD PROGRAMA I REZULTATI

Na slici 3 prikazan je softverski kod za Arduino razvojno okruženje. Za LED diodu je iskorišćen pin 4, koji

T. Bogdanović je učenica gimnazije "Bora Stanković" Vožda Karađorđa 27, 18000 Niš, Srbija, E-mail: bogdanovicteodora4@gmail.com.

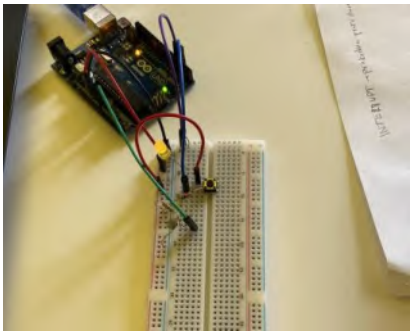
M. Dikić je učenica gimnazije "Bora Stanković" Vožda Karađorđa 27, 18000 Niš, Srbija, E-mail: mina.dikic5@gmail.com.

je izlazni. Taster je povezan na pin 2 i postavljen je da bude ulazni, a senzor, tj. fotootpornik je postavljen da takođe bude ulazni zato što se preko njega mere analogne vrednosti napona sa naponskog razdelnika. U beskonačnoj petlji se vrši očitavanje sa senzora i vrši se poređenje sa određenim vrednostima koje su određene eksperimentalnim putem. To predstavlja kalibraciju senzora (fotootpornika) i svetlosti veću od 150, LED dioda će se isključiti što znači da prostorija ima dovoljno svetlosti. U suprotnom, pri manjku iste, ona će se uključiti. Ukoliko postoji potreba da

Kod programa

```
#define ledPin 4
#define tasterPin 2
#define senzorPin A0
volatile bool fleg=false;
void setup() {
  // put your setup code here, to run once:
  pinMode(ledPin, OUTPUT);
  pinMode(tasterPin, INPUT);
attachInterrupt(digitalPinToInterrupt(tasterPin),tasterISR,
RISING);
  Serial.begin(9600);
}
void loop() {
  // put your main code here, to run repeatedly:
  int senzor=analogRead(senzorPin);
  Serial.println(senzor);
  if(senzor>150 || fleg == true){
    digitalWrite(ledPin, HIGH);
  }
  else{
    digitalWrite(ledPin, LOW);
  }
}
  void tasterISR(){
    fleg =!fleg;
  }
```

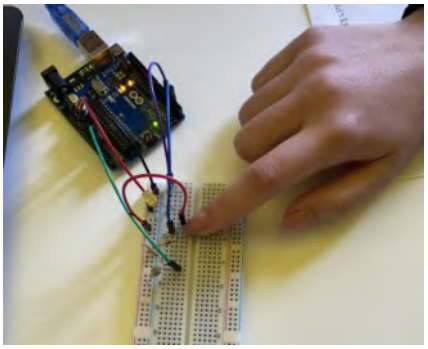
Na slikama 3 i 4 je prikazan izgled finalnog projekta. Prikazane slike pokazuju rad LED diode u mračnijim uslovima, kao i njeno isključivanje pomoću tastera. Sistem je moguće implementirati i za korišćenje u realnim uslovima, uz dodatne komponente za proširenje ovakvog projekta.



Slika 3. Izgled gotovog projekta i testiranje u mračnim uslovima.

se LED dioda uključi, a nije detektovala jačinu svetlosti manju od 150, onda možemo ručno pritiskom na taster uključiti istu, što se odigrava u prekidnoj proceduri (*interrupt*).

Vrednosti koje su postavljene u kodu su dobijene tako što su vršenja očitavanja senzora u osvetljenoj prostoriji, a zatim je određena granica gde se softverski vrši uključivanje i isključivanje LED diode. Dobijena vrednost je 150.



Slika 4. Izgled gotovog projekta i isključivanje LED diode pomoću tastera.

IV. ZAKLJUČAK

Osmišljeni uređaj je kompaktan, mobilan i može se dovesti do stadijuma velike preciznosti pomoću analize podataka dobijenih iz njegovog koda. Potrošnja energije je svedena na minimum korišćenjem Arduino okruženja koje ne iziskuje veliku količinu iste.

Sistem bi pomogao u borbi protiv zagađenja životne sredine i doprineo bi rešenju problema brojnih država i velikih korporacija.

ZAHVALNICA

Ovaj rad nastao je kao rezultat realizacije projekta „Let STEM visit again IEEESTEC“, finansiranog u okviru 2022 IEEE Pre-University STEM Portal Grant Program-a. Zahvaljujemo se mentoru Jani Vračar na usmeravanju i sugestijama pri izradi ovog rada.

LITERATURA

[1] Arduino Uno Development Kit, tehnička dokumentacija dostupna na: https://docs.arduino.cc/hardware/uno-rev3;
[2] Fotootpornik Wikipedia, https://sh.wikipedia.org/wiki/Fotootpornik, pristupljeno: 20.10.2022. godine;
[3] TinkerCad, dostupno na https://www.tinkercad.com/circuits, pristupljeno 12.10.2022. godine;

Comparison of Capacitive and Resistive Methods for Water Salinity Measurements

Milan Stojanović, Aleksandra Stojković, Neda Stanojević, Jana Vračar

Abstract – The water salinity is one of the crucial physical parameters in many processes where water usage is necessary. The fundamentals of salinity measurement using capacitive and resistive methods are presented in this study. The output sensor voltage, at different salt concentrations in the temperature range from 298 K to 338 K, was measured by two sensors simultaneously. The results show significant changes in the resistive sensor's output voltage than the capacitive at the same salt concentration because the salinity has more impact on changing resistivity than on the dielectric permittivity of water.

I. Introduction

Water or soil salinity is a crucial parameter in many fields such as environment, agronomy, etc. Water level can be determined by fiber-optical sensors that measure the intensity of light reflected from the water surface, and it depends on temperature and salinity [1]. The salt concentration is an essential parameter in the estimation of power production in an osmosis power plant [2]. There are different electrical methods used for salinity measurement.

The resistive methods, based on resistive sensors, are used to measure electrical conductivity, that can be regarded as an indicator of water or soil salinity. The paper [3] describes that the resistivity of soil decreases with salt concentration. The other one, the capacitive method is based on sensors which can detect changes in capacity influenced by lower or higher salt concentration.

The influence of salt concentration on changes in water resistance or capacitance significantly depends on temperature [4]. The number of free ions or their mobility may increase with a rise in temperature, increasing the conductivity of water [5].

The goal of this paper is present and compare the possibility of measurement of water salinity using the capacitive soil moisture sensor SKU:SEN0193, and the simple resistive rain sensor.

II. Sensors

A. Capacitive soil moisture sensor SKU:SEN0193

As already mentioned, the first sensor used in the paper is a capacitive soil moisture sensor SKU:SEN0193, presented in Figure 1, which is commercially available on the market, in small-thin sizes. Due to the materials, it is made of, it is resistant to corrosion which is why it is quite durable. This capacitive sensor's operating voltage ranges from 3.3 V to 5.5 V, and its operating current is 5 mA [6].



Fig. 1. The capacitive soil moisture sensor SKU:SEN0193 [6].

The first part of the sensor is the electronic circuit that serves to generate the excitation of the input signal, and the second part is the coplanar capacitor whose capacitance depends on the dielectric permittivity (ε) of the surrounding material. Figure 2a) is shown a simplified schematic of sensor SKU: SEN0193. When using the generated impulse voltage, the capacitor C_{sense} is charged to a certain level, and that voltage V_{OUT} is taken to the output. The value of the voltage V_{OUT} depends on the value of the capacitance, whereby a higher value of the capacitance causes a higher value of the voltage and vice versa. Depending on the water content, the dielectric permittivity changes, which directly affects the change in the capacitance of the capacitor. The dielectric permittivity of water depends on salinity, so the capacitance and output voltage of the sensor are changing for different salinity. It's important to emphasize that the output signal of the sensor SKU:SEN0193 is a direct DC voltage, and that due to the influence of temperature and water content, it's necessary to perform calibration before the start of the experiment [7].

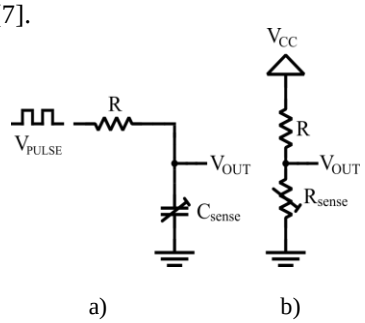


Fig. 2. Simplified electric circuit of a) capacitive SKU:SEN0193 sensor and b) resistive rain sensor.

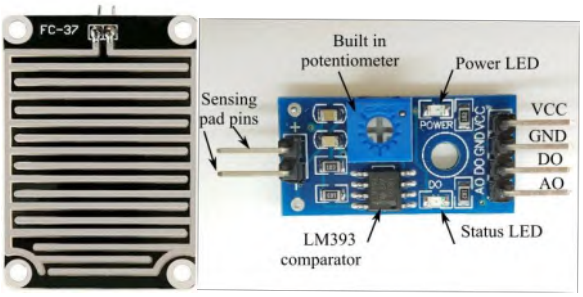
M. Stojanović, A. Stojković, N. Stanojević, and J. Vračar are with the Department of Microelectronics, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: milan.stojanovic@elfak.ni.ac.rs.

B. The resistive rain sensor

The resistive rain sensor [8] includes two PCBs: a sensing pad and an electronics board, which are separate for convenience. The sensing pad should be placed out in the open where it can be affected by rain/water, while the board with electronics must be isolated. The role of the sensing pad is to collect raindrops, while the electronics board is in charge of the output signals. Dimensions of the sensing pad are 5 cm x 4 cm, and of the electronics board 3.2 cm x 1.4 cm.

The sensing pad (Figure 3a) acts as a variable resistor (potentiometer) whose resistance changes according to the amount of water on its surface. This pad is a resistive dipole that shows less resistance when wet and more resistance when dry, and that is because resistance is inversely proportional to the amount of water. The sensing pad is a PCB that has a series of exposed copper traces that are not connected but can be bridged by water. The total resistance is decreased when drops are detected on the circuit board because parallel resistance paths form between the traces. None or a little bit of water on the surface means poor conductivity which will result in higher resistance. More water on the surface means better conductivity and results in a lower resistance. Depending on the resistance of the sensing pad, the corresponding output signal is formed.

The electronics board (Figure 3b) has six pins, two of which are used to connect to the sensing pad and the other four to the microcontroller (AO, DO, GND and VCC). GND represents ground while the power supply is connected to the VCC pin. Power supply values between 3.3 V and 5 V are advised. AO (Analog Output) pin gives an analog signal between the ground and the supply value, while DO (Digital Output) pin provides a digital output of the internal comparator circuit.



a) b)

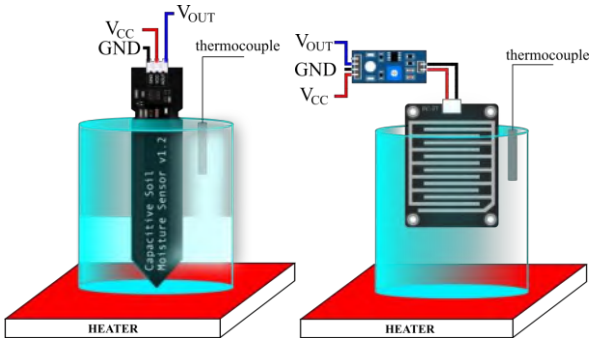
Fig. 3. The resistive rain sensor consists of a) a sensing pad and b) an electronics board.

The electronics board consists of a voltage comparator LM393, a built-in potentiometer for sensitivity adjustment, and two LEDs (Power and Status LED). At the AO pin, an output voltage is generated according to the resistance of the sensing pad. That same signal is fed to the LM393 comparator as the input signal. The comparator output will be HIGH (1) when the input signal is higher than the threshold value and will change to LOW (0) when the input signal falls beneath the threshold value.

The built-in potentiometer is used for sensitivity adjustment of the digital output (DO) in the example comparator threshold value. Apart from this, the board has two LEDs. The Power LED will light up when the board has a power supply, and the Status LED will light up when the digital output goes LOW.

III. The Experiment Setup and Results

The experiment was set up by testing the water with different concentrations of table salt (NaCl) but at different temperature ranges. The first measurement was performed with a salt concentration of 10 ‰; subsequent measurements were done with concentrations that ranged from 10 ‰ to 50 ‰, respectively. The vessel containing the mixture was heated, starting at a temperature range of 298 K to 338 K. The measurement was executed for the particular salt concentration in water, and the step for temperature range is 10 K. Both the resistive rain sensor, and the capacitive sensor SKU:SEN0193 were submerged in water at the same depths, and readings were made simultaneously. A thermocouple (K type) was used in parallel with the sensors to precisely measure the water temperature. The experimental setup is shown in Figure 4. Even though the measurements were carried out simultaneously in one vessel, the sensors are depicted individually in containers to better illustrate how the experiment was set up. Figure 4a) illustrates the capacitive sensor SKU: SEN0193 in the experimental setting, whereas Figure 4b) depicts the resistive rain sensor.



a) b)

Fig. 4 The experimental setup of a) the capacitive sensor SKU: SEN0193, and b) the resistive rain sensor.

Capacitance soil moisture sensors respond to the dielectric permittivity ($\varepsilon = \varepsilon_0 \cdot \varepsilon_r$) of a soil mixture content, and in this paper of a water-salt mixture. Figure 5 illustrates capacitive sensor readings at different water salinities and different temperature levels, as stated before. The intended outcomes of this experiment are that when

the capacitance value in the RC circuit, which is a component of the capacitive sensor, declines, the output voltage from the V_{OUT} sensor should also decrease in value. ε_r is a crucial element for the measurement findings because of its impact on the capacitance value and the fact that the capacitance itself is influenced by the environment in which the sensor is positioned.

Therefore, when the amount of salt in the water increases, ε_r's value lowers, which also decreases the capacitance's value. As stated earlier in the text, this results in a decrease in the output voltage's value, as shown by the outcomes in Figure 5. For a variety of soil types, the impact of soil temperature on capacitance systems has been documented. A soil temperature fluctuation experiment [9] revealed a relationship between temperature and capacitance probe values. Thus, the results show an increase in the output voltage V_{OUT} from the sensor with increasing water temperature. For a water temperature of 338 K, the output voltage reaches its maximum values, while the voltage decreases with an increasing salt concentration in the water, as stated before.

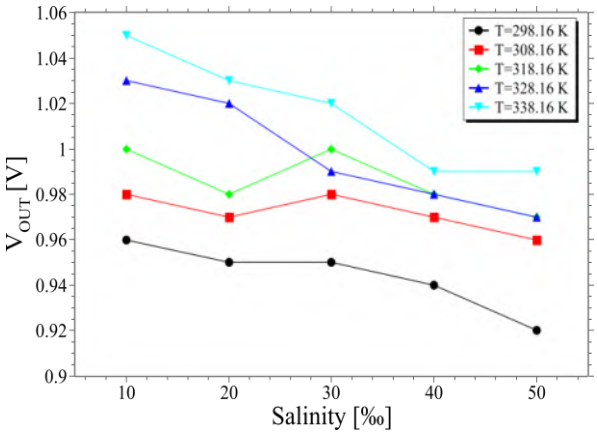


Fig. 5. The measured results of the capacitive sensor SKU:SEN0193 for the salt-water mixture with the water salinity range from 10 ‰ to 50 ‰, with power supply V_{CC} of 5 V.

More water on the sensor pad results in improved conductivity and reduced resistance for the resistive rain sensor. Likewise to this, when there is less water on the surface pad, there is low conductivity and high resistance. Thus, the output of this sensor is primarily determined by its resistance. Figure 6 shows measuring results for the resistive rain sensor. Since the main working principle of this sensor is a voltage divider, changes in the voltage output values are anticipated. These output voltage changes occur because the part of the sensor submerged in the water represents R_{sense} from the electrical diagram (Figure 2b).

The output voltage value should decrease as the R_{sense} resistance value does. Additionally, since dissolved salt in water acts as a good conductor, it influences the reduction of electrical resistance, which, in turn, impacts the decrease of the resistance value R_{sense}. The electrical resistance R_{sense} reduces as the concentration of salt in the water rises, which causes the decrease of output voltage value as seen by the results. Furthermore, as the water temperature rises, conductivity rises as well. Because water becomes less viscous at higher temperatures, ions can move more freely.

As a result, the resistance at the same salt concentration in the water decreases its value progressively more with increasing temperature values, and therefore the value of the output voltage V_{OUT} decreases.

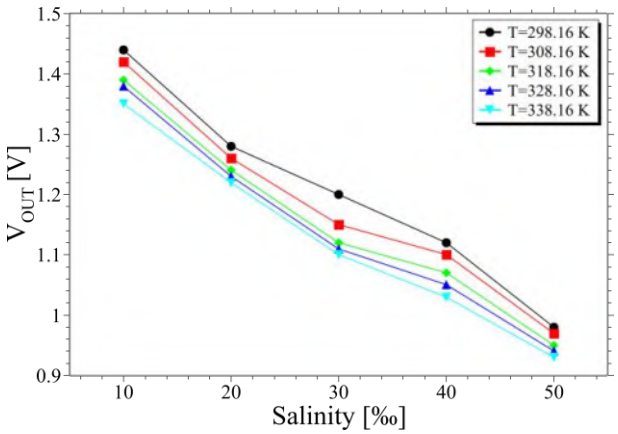


Fig. 6. The measured results of the resistive rain sensor for the salt-water mixture with the water salinity range from 10 ‰ to 50 ‰, with power supply V_{CC} of 5 V.

Figure 6 shows a sharp voltage drop compared to Figure 5, at lower salt concentrations in the water. Therefore, it can be concluded that the dielectric permittivity fluctuates less than the electrical resistance when the salinity changes.

IV. Conclusion

The presented results show that capacitive sensor SKU:SEN0193, and resistive rain sensor can be used to measure the salinity of the water. At the same level of the power supply, the capacitive sensor is less sensitive than the resistive rain sensor. It is because the changes in the dielectric permittivity are less pronounced than the changes in the resistivity of water in the salinity range from 10 ‰ to 50 ‰. A higher salinity lever should cause larger changes in the dielectric permittivity. When measuring salinity with a rain sensor module (or another resistivity sensor), water electrolysis may be a problem. When measuring, pulse excitation instead of constant DC current can solve the problem. Future research will involve designing an electric circuit to excite pulses and a sensor with multiple electrodes for sensing. Also, future research will deal with testing higher concentrations of salt, where it will be necessary to develop a sensor model that is resilient to salt corrosion. Furthermore, measurements will be made at a broader temperature range while testing at higher salt concentrations in water.

ACKNOWLEDGEMENT

This work was supported in part by the Serbian Ministry of Education, Science and Technological Development under Grant 451-03-68/2022-14/200102.

REFERENCES

[1] Yoo, W., Sim, H., Shin, S., Jang, K., Cho, S., Moon, J., & Lee, B. (2014). A Fiber-Optic Sensor Using an Aqueous Solution of Sodium Chloride to Measure Temperature and Water Level Simultaneously. Sensors, 14(10), 18823–18836. doi:10.3390/s141018823.

[2] Sivertsen, E., Holt, T., & Thelin, W. (2018). Concentration and Temperature Effects on Water and Salt Permeabilities in Osmosis and Implications in Pressure-Retarded Osmosis. Membranes, 8(3), 39. doi:10.3390/membranes8030039

[3] Zohra-Hadjadj, F., Laredj, N., Maliki, M., Missoum, H., & Bendani, K. (2019). Laboratory evaluation of soil geotechnical properties via electrical conductivity. Revista Facultad de Ingeniería Universidad de Antioquia, (90), 101–112. doi:10.17533/udea.redin.n90a11.

[4] Zhang, W., Chen, X., Wang, Y., Wu, L., & Hu, Y. Experimental and Modeling of Conductivity for Electrolyte Solution Systems. ACS Omega, 2020, 5(35), pp. 22465–22474. doi:10.1021/acsomega.0c03013.

[5] Marshal Soni Variation of conductivity of the different sources of water with temperature and concentration of electrolyte solution NaCl. Pharma Innovation 2017, 6(3): pp. 119-120.

[6] Available on: https://nettigo.eu/products/capacitive-analog-soil-moisture-sensor.

[7] Milan Stojanović, Neda Stanojević, "Use, testing and calibration of capacitive sensor SKU:SEN0193", Proceedings of the 13th Student projects conference, IEEESTEC, Niš (Serbia), 2020, pp. 157-160, ISBN: 978-86-6125-230-3.

[8] Available on: https://components101.com/sensors/rain-drop-sensor-module?fbclid=IwAR0PmR4S68xVpHE0oZr2FZePuTs89NxKn6Yfhrkk1ErDKETK08tChDqDTr4.

[9] Baumhardt, R. L., Lascano, R. J. and Evett, S. R. 2000. Soil material, temperature, and salinity effects on calibration of multisensor capacitance probes. Soil Sci. Soc. Am. J. 64: pp. 1940–1946.

Primer korišćenja Arduino kontrolera za pravilno sedenje

Luka Vuksanović, Ana Milosavljević, Anastasija Savković

Sadržaj − Osnovu ovog sistema čini: Arduino UNO ploča, zujalica modul za Arduino, ultrazvučni senzor i nekoliko žica. Cilj sistema jeste da pomogne deci prilikom sedenja na stolici, tako da njihovo sedenje bude pravilno. Ovo se realizuje preko "pametne pčelice" namenjene za decu.

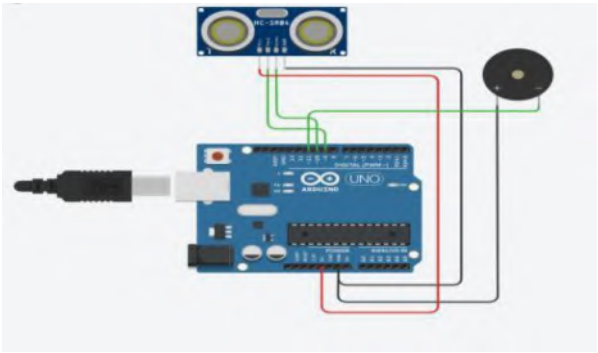
I. UVOD

Projekat ima za cilj da osobu, koja sedi na stolici zvučnim signalom podseti na pravilno sedenje [1].



Slika 1. Crtež pravilnog i nepravilnog sedenja.

Projekat se zasniva na Arduinu [4], ultrazvučnom senzoru [6], kao i zujalici [5]. Svi delovi projekta upakovani su u igračku pčelice. "Pametna pčelica" postavlja se na zadnjem delu stolice. Senzor će neprekidno meriti rastojanje do leđa osobe koja sedi. Svako povećanje udaljenosti ukazuje na savijanje leđa i uključivanje zvučnog signala. Pčelica je namenjena primarno za decu, kako bi oni razvili naviku da sede uspravno, a kada je ne koriste za pravilno sedenje može da posluži kao njihova omiljena igračka.



Slika 2. Slika šeme.

L. Vuksanović, A. Milosavljević i A. Savković su učenici Šeste beogradske gimnazije, Bulevar despota Stefana 90, Beograd, Srbija, E-mail: luka.vuksanovic2004@gmail.com.

II. PRINCIP RADA

Ultrazvučni senzor [6] emituje ultrazvučne talase, koji se prilikom sedenja na stolici gde je pčelica postavljena, odbijaju od korisnika i šalju nazad ka datom senzoru. Ultrazvučni senzor [6] prima informacije distance korisnika u odnosu na njegovu poziciju i šalje signal Arduino UNO [4], koji dobija informacije od ultrazvučnog senzora i obrađuje ih preko koda koji je napisan u C programskom jeziku.

```
const int trigPin = 9;  // definišemo TRIG pin senzora na pin 9 od Arduina
const int echoPin = 10; // definišemo ECHO pin senzora na pin 10 od Arduina
const int buzzer = 11;  // definišemo zujalicu na pin 11 od Arduina
const int ledPin = 13;  // definišemo lampicu na pin 13 od Arduina

// definišemo promenjive
long duration; // promenjiva za dužinu zvučnog talasa
int distance;  // promenjiva za merenje udaljenosti
int safetyDistance;


void setup() {
pinMode(trigPin, OUTPUT); // postavljamo trigPin kao izlaz
pinMode(echoPin, INPUT);  // postavljamo echoPin kao ulaz
pinMode(buzzer, OUTPUT);  // postavljamo zujalicu kao izlaz
pinMode(ledPin, OUTPUT);  // postavljamo lampicu kao izlaz
Serial.begin(9600); // pokrećemo serijalnu komunikaciju
}


void loop() {
// brišemo trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(2);

// postavljamo trigPin na HIGH stanje na 10 mikrosekundi
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

// čitamo echoPin i vraćamo vreme putovanja zvučnog talasa u mikrosekundama
duration = pulseIn(echoPin, HIGH);

// računamo distancu
distance= duration*0.034/2;

safetyDistance = distance;
if (safetyDistance >= 30
)
{ // unosimo distancu
  digitalWrite(buzzer, HIGH);
  digitalWrite(ledPin, HIGH);
}
else{
  digitalWrite(buzzer, LOW);
  digitalWrite(ledPin, LOW);
}

// printujemo distancu na Serijalnom Monitoru
Serial.print("Distance: ");
Serial.println(distance);
}
```

Slika 3. Slika koda.

Zatim, Arduino UNO [4] obrađene informacije šalje zujalici, koji emituje glasan zvuk i daje do znanja korisniku da sedne uspravno. Momenat kada korisnik sedne uspravno, distanca između ultrazvučnog senzora [6] i korisnika se smanjuje i ultrazvučni senzor šalje signal Arduino UNO [4], koji obrađuje novu informaciju, šalje signal zujalici [5] da prestane sa emitovanjem zvuka. Osnovu koda predstavlja if naredba pomoću koje se proverava udaljenost.



Slika 4. Arduino UNO.



Slika 5. Zvučnik.



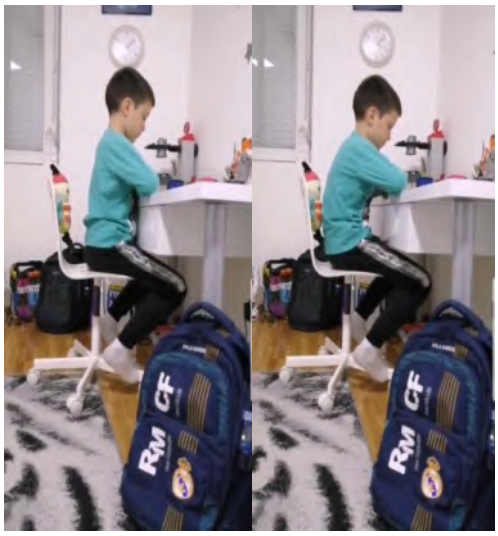
Slika 6. Ultrazvučni senzor.



Slika 7. “Pametna” pčelica.

III. Prenos koda

Kod se prenosi koristeći USB 2.0 kabl A/B, koji se povezuje za računar i Arduino UNO ploču [4] i koristi Arduino 2.0 IDE softver za lako pravljenje i otpremanje koda na Arduino UNO ploču[4].



Slika 8. Pravilno i nepravilno sedenje deteta.

IV. Zaključak

Ovaj projekat je osmišljen od strane učenika, a realizovan uz pomoć profesora-mentora. Bilo je potrebno primeniti znanje iz programiranja i elektronike.

Ovaj projekat pruža i mogućnost za dalja poboljšanja kao što je na primer da se umesto zvučnog signala čuje glas omiljenog crtanog junaka, kako bi dete sa radošću prihvatalo pametnu pčelicu.

Zahvalnica

Autori rada zahvaljuju se školi, a posebno mentoru profesorki Gordani Ćeranić.

Literatura

[1] https://saznajlako.com/2013/11/17/vaznost-pravilnog-drzanja-tela-kod-skolske-dece/
[2] https://www.tinkercad.com/
[3] https://create.arduino.cc/editor
[4] https://elektroleum.rs/product/arduino-kod-programiranje-uno-r3-raspberry-kodiranje-hardver-softver-senzor-senzori-kit-ard-modul-modeliranje-
[5] daljinski-robot-mega-atmega-data-detektor-daljinski-upravljanje-beograd-2/
[6] https://robocraze.com/blogs/post/how-to-use-buzzer-with-arduino
[7] https://create.arduino.cc/projecthub/csw1/ultrasonic -sensor-with-arduino-uno-f33ca1
[8] http://vigimnazija.edu.rs/?page_id=128
[9] http://vigimnazija.edu.rs/?page_id=12

Detekcija pola pomoću konvolucionih neuronskih mreža

Ivana Petković

Sadržaj – Ovaj rad predstavlja principe i način rada konvolucionih neuronskih mreža i kako se uz pomoć konvolucionih neuronskih mreža može postići detekcija pola čoveka na osnovu njegovog lica.

I. UVOD

Živimo u doba kada se neprestano stvara i skladišti ogromna količina podataka. Ovi podaci se mogu pretvoriti u znanje pomoću naučenih algoritama iz oblasti mašinskog učenja. Veštačke neuronske mreže su napravljene po uzoru na biološke neuronske mreže. Osim neuronskih mreža, postoji još mnogo modela koji mogu da se primene iz oblasti mašinskog učenja, ali u ovom radu će biti obrađivane konvolucione neuronske mreže za potrebe detekcije pola čoveka [1] [2].

II. KONVOLUCIONE NEURONSKE MREŽE

Konvolucione neuronske mreže nisu ništa drugo nego proširene višeslojne neuronske mreže. One se proširuju dodavanjem nove vrste sloja koji se naziva konvolucioni sloj. Konvolucione neuronske mreže su privukle veliku pažnju i dovele do ogromnog poboljšanja u mašinskom učenju za računarski vid zbog svojih izvanrednih performansi za klasifikaciju slika. Konvolucione neuronske mreže mogu automatski da uče atribute iz neobrađenih podataka koji su najkorisniji za određeni problem. Uobičajeno je razmotriti slojeve ove mreže kao ekstraktore atributa [1].

Konvolucione neuronske mreže konstruišu hijerarhiju atributa kombinovanjem atributa niskog nivoa na slojevit način za formiranje atributa visokog nivoa. Ako je u pitanju slika, onda su atributi niskog nivoa ivice i mrlje, oni se kombinuju za formiranje atributa višeg nivoa koji mogu da formiraju kompleksnije oblike kao što su konture nekog objekta. Svaki sloj u hijerarhiji ekstraktuje atribute iz izlaza prethodnog sloja. Poenta ekstraktovanja atributa iz ulaznih slika je da se formira mapa atributa [3].

Neuronska mreža u svakom sloju uči određeni broj filtra. Skup filtra koji deluju nad istim ulazom se naziva konvolucioni sloj. Konvolucioni sloj na svom izlazu daje tenzor čiji svaki kanal odgovara izlazu jednog filtra. Svaki filtar posebno može da nauči neke korisne informacije. Na primer, jedan filtar može da nauči da detektuje horizontalne ivice, drugi vertikalne, treći kose ivice. Svaki kanal se transformiše nelinearnom aktvacionom funkcijom. Nad izlazom jednog konvolucionog sloja je moguće izgraditi novi konvolucioni sloj pa zato filtri viših konvolucionih slojeva konstruišu nova svojstva nad svojstvima koji su konstruisali niži slojevi. Konvolucioni slojevi višeg nivoa traže složenije obrasce u slici, a koji su obrasci od značaja se uči minimizacijom greške koju mreža pravi nad podacima [4].

Pored konvolucionih slojeva, često se koriste i slojevi za smanjivanje uzoraka koji služe agregaciji informacije u cilju smanjenja količine izračunavanja i broja parametara u mreži. Smanjivanje parametara se, obično, primenjuje u dve forme operacija udruživanja u konvolucionim neuronskim mrežama i to: udruživanje maksimumom (eng. *max pooling*) i udruživanje srednjom vrednošću (eng. *mean pooling*), poznato i kao udruživanje prosekom (eng. *average pooling*) [3][4].

III. AUGMENTACIJA PODATAKA

Augmentacija podatka se u dubokom učenju koristi u cilju povećanja broja podataka u trening skupu podataka, tj. povećanja varijabilnosti mogućih položaja i izgleda objekata od interesa kako bi mreža imala više primera za učenje. Određene tehnike proširenja podataka omogućavaju da modifikujemo ili čak veštački sintetizujemo više podataka i tako poboljšamo performansu modela mašinskog učenja smanjivanjem preprilagođavanja. Iako proširenje podataka nije samo za slike, postoji skup transformacija koje su jedino primenljive na slike. Za potrebe treniranja konvolucione mreže primenjuje se niz nasumičnih ili izabranih transformacija na svaku sliku u trening skupu. Neke od transformacija su: rotacija, zumiranje, afina transformacije, menjanje osvetljenja, zamagljenje, opsecanje delova slike, izokretanje i menjanje kontrasta, zasićenosti. Cilj augmentacije podataka je povećanje generalizacije problema, odnosno, sprečavanje pojave preprilagođavanja modela. Ovim postupkom je mreža u mogućnosti da nauči više robusnih karakteristika [5] [6].

IV. IMPLEMENTACIJA

Zadatak ovog rada je detekcija pola na osnovu lica čoveka. Ovo je realizovano pomoću konvolucione neuronske mreže. Trening set koji je korišćen za treniranje

I. Petković je master inženjer elektrotehnike i računarstva na katedri za automatiku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: ivana.petkovic@elfak.rs.

mreža sa sastoji iz dve particije, slika muškaraca i slika žena. U trening setu postoji 1173 slika muškaraca i 1134 slika žena [7]. Ovo nije veliki skup podataka, ali je pokušano da se pokaže da se i sa nešto manjim skupom podataka može rešiti ovakav problem ukoliko se upotrebi funkcija *ImageDataGenerator* i time se veštački proširi skup podataka. Slike su u boji (što znači da je slika predstavljena pomoću tri kanala za boje) i u pitanju su slike različitih veličina. Iskorišćeno je 80 % slika za treniranje modela, a ostalih 20 % za testiranje. Ova podela podataka je moguća uz pomoć funkcije *train_test_split.* Implementacija je rađena u programskom jeziku *Python,* a realizacija ovog zadatka i same konvolucione neuronske mreže je olakšana uz pomoć biblioteka *Tensorflow* i *Keras*.

Za potrebe ovog rada su trenirane 4 različite arhitekture mreže, u cilju poboljšanja tačnosti na test skupu podataka. Cilj ovog rada je da kada je kamera PC-a uključena, ona detektuje lice i proceni na osnovu slika koje su date mreži u toku treninga, sa određenom sigurnošću da li je detektovano lice zapravo lice muškaraca ili žene.

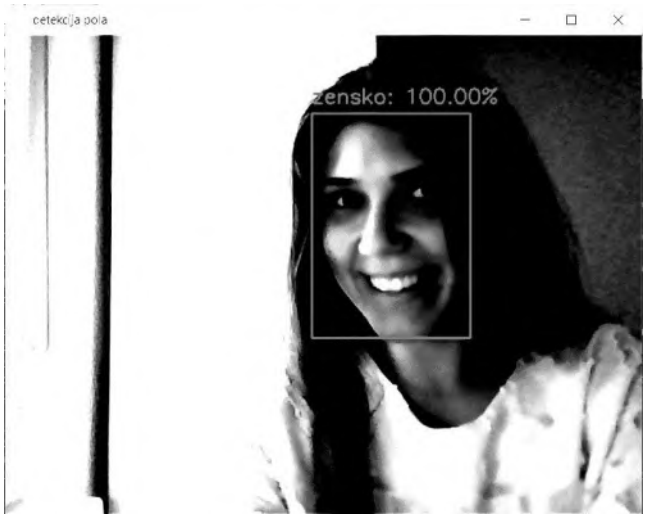
U sve 4 athitekture je korišćena aktivaciona funkcija *ReLU* u svim slojevima mreže, sem u poslednjem, gde je korišćena *sigmoidalna* aktivaciona funkcija. Što se tiče dopunjavanja, uvek je korišćeno jednako dopunjavanje (*padding = "same*"). Veličina filtra je takođe uvek bila ista 3x3 jer se poslednjih godina ta veličina pokazala kao dobra praksa. Ono sa čim je eksperimentisano je broj slojeva mreže, broj filtra koji se uče, vrednost za verovatnoću izostavljanja (eng. *dropout*) i u koliko se epoha trenira mreža.

Sledi tabelarni prikaz 4 arhitekture mreže. Prikazane su sledeće metrike za svaku arhitekturu: funkcija greške na trening podacima (eng. *loss*), funkcija greške na validacionim podacima (eng. *val_loss*), tačnost na trening podacima (eng. *accuracy*), tačnost na validacionim podacima (eng. *val_accuracy*). Bitnije metrike su *val_loss* i *val_accuracy*. Potrebno je da *val_loss* bude što manja, a *val_accuracy* što veća, za bolje performanse modela.

TABELA I
PREGLED PERFORMANSI MODELA

CNN	CNN1	CNN2	CNN3	CNN4
Br.slojeva	5	7	9	9
Dropout	25 %	50 %	50 %	50 %
Br.epoha	20	100	150	200
Loss	0.1466	0.0604	0.0504	0.0337
Val_loss	0.2309	0.1524	0.0814	0.0669
Accuracy	0.9410	0.9770	0.9809	0.9865
Val_accuracy	0.9242	0.9589	0.9762	0.9827

Na osnovu tabele I je očigledno da je poslednja trenirana arhitektura postigla najveću tačnost na validacionim podacima od 98,27 %. Ona uspešno prepoznaje pol čoveka. Sledi i vizuelni prikaz detekcije pola na slici 1.



Slika 1. Prikaz rada modela CNN4.

V. ZAKLJUČAK

Na osnovu treniranja nekoliko arhitektura primećuje se da model daje bolje rezultate ukoliko je mreža dublja. Dodavanje konvolucionih slojeva i povećanje broja epoha može pomoći u povećanju tačnosti i smanjenju greške na validacionim podacima. Takođe, augmentacija podataka je vrlo bitan aspekt u sprečavanju preprilagođavanja modela i poboljšanja performansi. Ovaj model bi mogao da se unapredi tako što bi mreža pored detekcije pola učila da prepoznaje i broj godina čoveka. Ovo bi moglo da pomogne u problemu segmentacije kupaca i poboljšanju prodaje u online i fizičkim prodavnicama kako bi se poboljšale usluge u prodajnim objektima.

LITERATURA

[1] Sebastian Raschka, Vahid Mirjalili, „Python mašinsko učenje“, *Packt Publishing*, 2020.
[2] Zsolt Nagy, „Osnove veštačke inteligencije i mašinskog učenja“, *Packt Publishing*, 2019.
[3] Predrag Janičić, Mladen Nikolić, „Veštačka inteligencija“, Matematički fakultet univerziteta u Beogradu, 2021.
[4] Aleksandar Milosavljević, „Računarski vid“, Elektronski fakultet univerziteta u Nišu, 2020
[5] Dostupno na: https://www.megatrend.com/augmentacija-slika-koristeci-tensorflow
[6] Dostupno na: https://ivanamedojevic.medium.com/dan28-augmentacija-podataka-100daychallenge-b5079be40fc2
[7] Dostupno na: https://github.com/topics/gender-detection

Ispitivanje karakteristika digitalnih kola 74LS68 i 74LS69

Milica Đorđević, Emilija Kotlo, Anđela Ružić, Sara Savić, Marta Stamenković

Sadržaj − U ovom radu izvršeno je ispitivanje karakteristika digitalnih kola 74LS68 i 74LS69. Glavni zadatak ovog istraživanja je pokazati i analizirati digitalne signale koji se dobijaju na izlazima ovih digitalnih kola.

I. Uvod

Brojači su sekvencijalne mreže koje se sastoje od memorijskih elemenata, odnosno flip-flopova, što omogućava memorisanje rezultata brojanja. Brojač po nailasku odgovarajuće ivice taktnog signala prelazi iz prethodnog (ili sadašnjeg) u odgovarajuće naredno stanje.

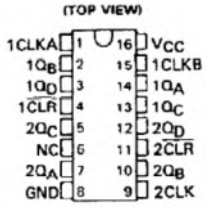
Oba kola su sastavljena od dva četvorobitna brojača. 74LS68 je dvostruki dekadni brojač, a 74LS69 je dvostruki binarni brojač. Kada govorimo o binarnom brojaču (74LS69) brojanje kreće od 0 do 15, pri čemu na izlazu dobijamo četvorobitni broj u binarnom obliku, dok kod dekadnog brojača (74LS68) brojanje se vrši u opsegu od 0 do 9, što znači da ima 10 stanja.



Slika 1. Komponente 74LS68 i 74LS69.

Kola 74LS68 i 74LS69 su *through-hole* komponente sa 16 izvoda (slika 1.). Prvi brojač u sastavu oba kola sadrži dva pina za taktni signal (CLK), od kojih jedan okida prvi od četiri flip-flopa, a drugi okida ostala tri flip-flopa. Takođe, svi flip-flopovi se okidaju zadnjom ivicom taktnog signala. Postoji i pin rezervisan za resetovanje brojača, koji resetuje izlaze kada se na njega dovede napon logičke nule, bez obzira na trenutno stanje brojača. Raspored pinova dat je na slici 2.

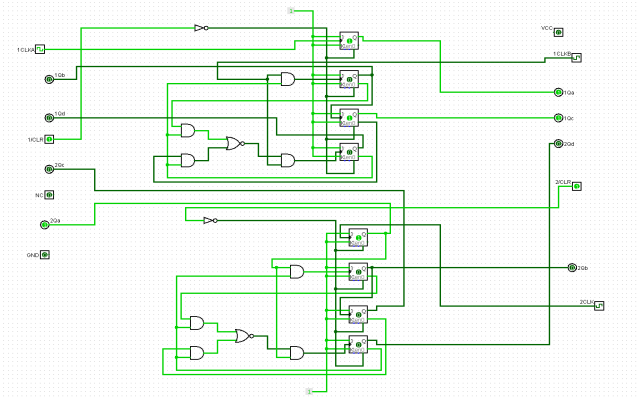
M. Đorđević, E. Kotlo, A. Ružić, S. Savić, M. Stamenković, studenti četvrte godine na modulu Elektronske komponente i mikrosistemi, Elektronskog fakulteta Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: milicadjordjevic@elfak.rs,kotloemilija@elfak.rs,andjela.r@elfak.rs, sarassavic@elfak.rs, marta.stamenkovic@elfak.rs.



Slika 2. Raspored pinova iz tehničke dokumentacije.

II. Simulacija

Na početku je potrebno odraditi simulaciju kola, koja će za potrebe ovog rada biti odrađena u programu *Logisim*. Na osnovu tehničke dokumentacije, napravljena je ekvivalentna šema oba brojača i ispitano je poklapanje dobijenih rezultata simulacije sa tablicom istinitosti datom u specifikacijama (slike 3, 4, 5, 6).

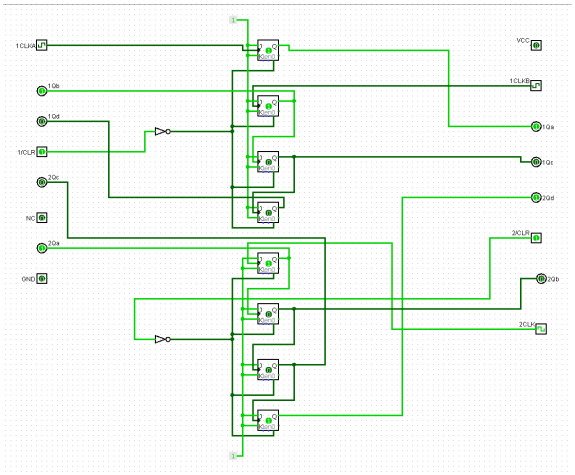


Slika 3. Šema kola 74LS68 realizovana u *Logisim*-u.

'LS68 DECADE COUNTER
BCD COUNT SEQUENCE
(See Note 1)
Applies to Counters 1 & 2

COUNT	OUTPUT			
	Q_D	Q_C	Q_B	Q_A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H

Slika 4. Tablica istinitosti kola 74LS68.



Slika 5. Šema kola 74LS69 realizovana u *Logisim*-u.

'LS69 BINARY COUNTER
BCD COUNT SEQUENCE
(See Note 3)
Applies to Counters 1 & 2

COUNT	OUTPUT			
	Q_D	Q_C	Q_B	Q_A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H
10	H	L	H	L
11	H	L	H	H
12	H	H	L	L
13	H	H	L	H
14	H	H	H	L
15	H	H	H	H

Slika 6. Tablica istinitosti kola 74LS69.

III. REALIZACIJA KOLA

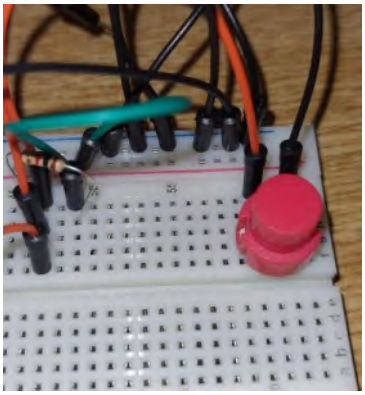
A. Oprema

Za realizaciju kola korišćeno je: proto ploča, žičani provodnici, taster i otpornik u *pull-up* konfiguraciji, kao i kola 74LS68 i 74LS69; instrumenti: dva signal generatora i izvor napajanja (slika 7).

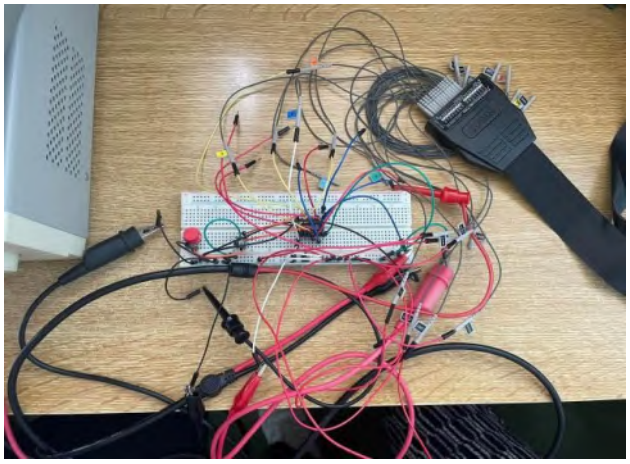


Slika 7. Dva signal generatora i izvor napajanja.

Signal generatori su korišćeni za generisanje taktnih signala 1CLKA, 1CLKB i 2CLK različitih frekvencija, amplitude 5 V. Prakčna realizacija prikazana je na slikama 8 i 9.



Slika 8. Taster i otpornik u *pull-up* konfiguraciji.



Slika 9. Izgled realizovanog kola.

B. Merni instrumenti

Digitalni osciloskop MSO5104 (slika 10.) je dizajniran model osciloskopa visokih performansi koji je baziran na RIGOL *UltraVision* II tehnologiji. Sa 9-inčnim kapacitivnim multi-tač ekranom, integriše 7 nezavisnih instrumenata u jedan, isporučujući odnos propusnog opsega uzorka, izuzetno veliku memoriju i druge odlične specifikacije. Ovaj digitalni osciloskop korišćen je u funkciji logičkog analizatora.

Logički analizator je elektronski instrument koji detektuje i prikazuje više signala iz digitalnog sistema ili digitalnog kola. Logički analizator može pretvoriti detektovane podatke u vremenske dijagrame, vrši dekodiranje protokola i može povezati asembler sa softverom na nivou izvora. Logički analizatori imaju napredne mogućnosti pokretanja i korisni su kada korisnik treba da vidi vremenske odnose između mnogih signala u digitalnom sistemu.



Slika 10. Digitalni osciloskop.

Za rad kao logički analizator, digitalni osciloskop je potrebno postaviti na kanal L (LA) koji prikazuje 16 digitalnih kanala (D0-D15). Preko takozvane "viljuške" ovih 16 kanala povezuju se sa proto pločom korišćenjem specijalnog seta žica (slika 11).



Slika 11. Set žica korišćenih pri povezivanju osciloskopa sa proto pločom.

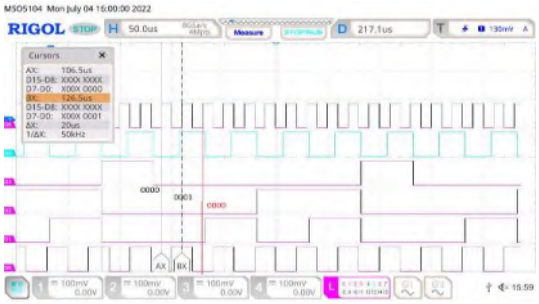
IV. ANALIZA REZULTATA IZLAZNIH SIGNALA

A. Integrisano kolo 74LS68

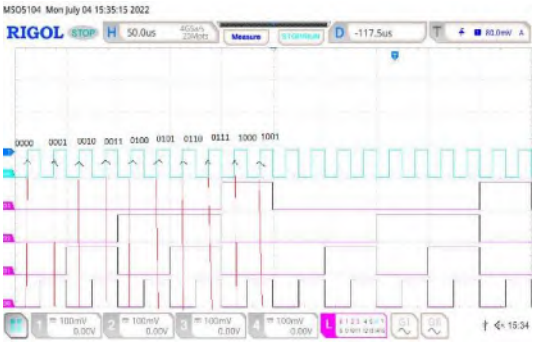
Prilikom snimanja izlaznih signala dobijenih sa pinova koji su deklarisani kao izlazi brojača kontrolisanog sa dva taktna signala (1QA, 1QB, 1QC i 1QD), može se videti da brojač ne funkcioniše onako kako je u teoriji predviđeno. Daljom analizom dobijenih rezultata ustanovljeno je da se greška dešava zbog neusaglašenosti okidanja. Ta neusaglašenost je rezultat različitih frekvencija taktnih signala.

U idealnom slučaju taktni signali moraju biti usaglašeni, što se nije moglo realizovati zbog upotrebe dva signal generatora. To je prouzrokovalo da se stanja na flip-flopovima neće promeniti u trenutku u kome je to predviđeno i na izlazu nećemo imati očekivana stanja na izlazima brojača.

Usnimili smo i izlaze sa pinova 2QA, 2QB, 2QC i 2QD (slika 13.) gde se jednim taktnim signalom kontrolišu četiri flip-flopa. Sa dolaskom odgovarajuće ivice taktnog signala izvršiće se promena stanja na izlazima, tako da će rezultat prikazan u binarnom obliku predstavljati brojeve od 0 do 9. Takođe, može se zaključiti da se odgovarajuća ivica taktnog signala poklapa sa promenom stanja na izlazu i da nema nikakvog kašnjenja, što je rezultat kontrolisanja četiri flip-flopa istim taktnim signalom (slike 12 i 13).



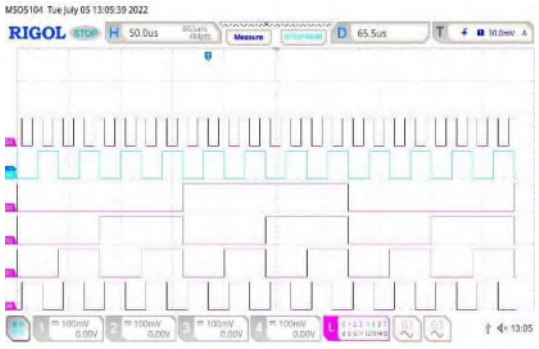
Slika 12. Digitalni signali na izlazima 1QA, 1QB, 1QC i 1QD kola 74LS68.



Slika 13. Digitalni signali na izlazima 2QA, 2QB, 2QC i 2QD kola 74LS68.

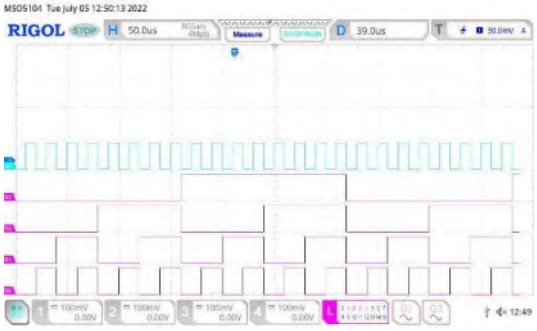
B. Integrisano kolo 74LS69

Izlazi 1QA, 1QB, 1QC i 1QD usnimljeni su iz prvog četvorobitnog brojača u okviru kola 74LS69. Prvi flip-flop okida se zasebnim taktnim signalom označenim kao 1CLKA, dok se ostala tri flip-flopa okidaju taktnim signalom 1CLKB i to asinhrono. Na slici 14. prikazano je kako ovaj binarni brojač broji od 0 (0000) do 15 (1111). Pri čemu je taktni signal 1CLKA frekvencije 50 kHz, dok je 1CLKB frekvencije 25 kHz.



Slika 14. Digitalni signali na izlazima 1QA, 1QB, 1QC i 1QD kola 74LS69.

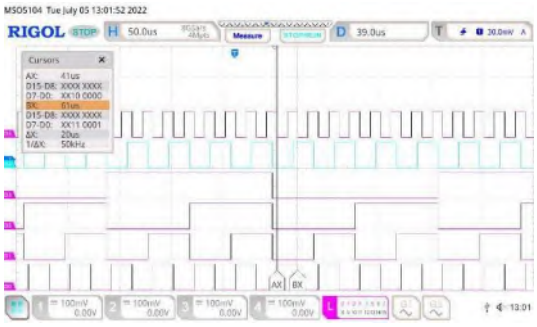
Izlazi 2QA, 2QB, 2QC i 2QD usnimljeni su iz drugog četvorobitnog brojača u okviru kola 74LS69. Ova četiri flip-flopa čine jedan asinhroni brojač (izlaz prethodnog flip-flopa okida naredni flip-flop) koji ima jedan taktni signal (2CLK). Na slici 15. može se videti kako ovaj binarni brojač broji od 0 (0000) do 15 (1111). Pri čemu je taktni signal frekvencije 50 kHz.



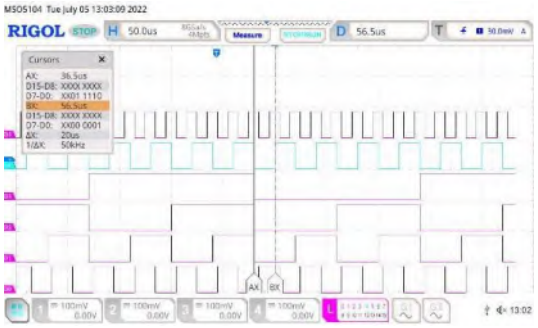
Slika 15. Digitalni signali na izlazima 2QA, 2QB, 2QC i 2QD kola 74LS69.

V. ZAKLJUČAK

Zbog različitih frekvencija koje okidaju jedan brojač dolazi do nepoklapanja. Na slici 16. prikazano je, pomoću kursora, kako se signal sa nultog kanala, koji je zapravo izlaz 1QA prvog flip-flopa koji se okida frekvencijom od 50 kHz, u potpunosti poklapa sa promenama taktnog signala (na svaku njegovu zadnju ivicu izlaz prelazi u suprotno stanje). Na slici 17. prikazano je, pomoću kursora, kako je jedna poluperioda taktnog signala 1CLKB, frekvencije 25 kHz, ekvivalentna jednoj periodi taktnog signala 1CLKA frekvencije 50 kHz, kao i to da se promene na kanalima 1-3 dešavaju upravo prateći drugi taktni signal.

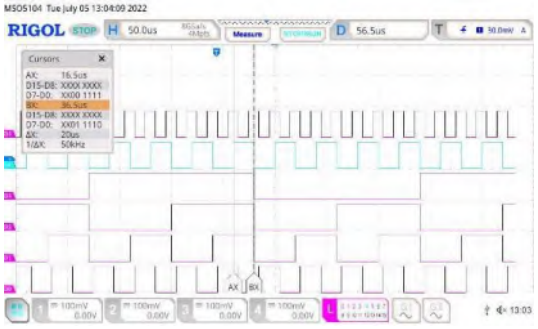


Slika 16. Izlazni signali na osciloskopu.



Slika 17. Izlazni signali na osciloskopu.

Na kraju, na slici 18. prikazan je prelaz stanja 1111 u 0000, gde se jasno vidi razilaženje signala sa nultog kanala i ostalih signala, zbog nemogućnosti istovremenog pokretanja oba taktna signala i propagacionog kašnjenja asinhronih brojača.



Slika 18. Izlazni signali na osciloskopu.

ZAHVALNICA

Autori se zahvaljuju prof. dr Danijelu Dankoviću i master inž. Nikoli Mitroviću na izdvojenom vremenu, sugestijama i pomoći pri realizaciji ovog rada.

LITERATURA

[1] Tehnička dokumentacija, dostupna na linku: https://www.circuits-diy.com/74ls68-dual-4-bit-decade-binary-counter-ic-datasheet/
[2] Uputstvo za digitalni osciloskop, dostupno na: https://rigolshop.eu/product-oscilloscope-mso5000-mso5104.html
[3] Način rada logičkog analizatora: https://bahasa.wiki/bs/Logic_analyzer
[4] *Logisim*, dostupan na: https://sourceforge.net/projects/circuit/

Projektovanje višeslojne štampane ploče za komunikaciju sa HC-05 bluetooth modulom

Nikoleta Džunić

Sadržaj – U ovom radu prikazan je način projektovanje višeslojne štampane ploče koja komunicira sa *bluetooth* modulom. Dat je detaljan opis svih delova šeme kao i osnovnih komponenata koje se nalaze na njoj. Za pisanje koda prilikom testiranja korišćen je *Aduion IDE* kompajler zbog kompatibilnosti sa mikrokontrolerom koji je iskorišćen u projektu.

I. UVOD

Razvojem tehnologije u današnje vreme vrlo je jednostavno i ekonomski prihvatljivo koristiti neku vrstu uređaja za komunikaciju u okolini, bilo za prenos velikih količina podataka, poput uživo prenosa videa ili slanja velikih datoteka, ili prenosa jako male količine informacija, kratkog dometa, uz vrlo malu i ekonomičnu potrošnju energije. Tako da *bluetooth* uređaji u današnjem vremenu su jako zastupljeni pre svega zbog potrošnje, ali i zbog značaja koji pružaju prenosom potrebnih podataka do krajnjeg uređaja kome oni mogu služiti ili za pružanje obaveštenja korisniku ili za obavljanje neke dužnosti.

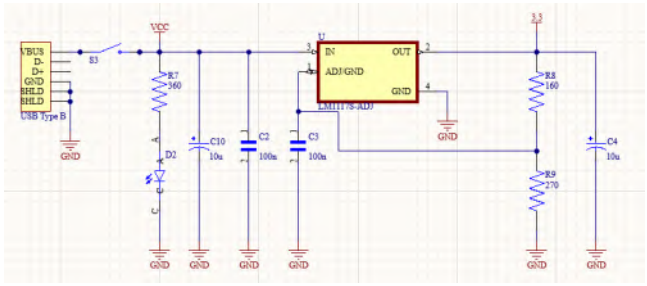
Uprkos povećanoj ceni izrade ploče, zbog složenosti usled težnje za minijaturizacijom i korišćenja višeslojne štampane ploče, projektovanje i fizička realizacija štampane ploče koja omogućava komunikaciju korišćenjem *bluetooth*-a nudi razne mogućnosti iskorišćenja. Proces projektovanja ovakve ploče se odvija u nekoliko koraka, od projektovanja šeme, preko *Layout*-a i rutiranja do konačnog izgleda ploče i eksportovanja izlaznih fajlova. Međutim, kako bi se proverila ispravnost ploče, mogućnost komunikacije s njom, neophodno je izvršiti testiranje.

II. ELEKTRIČNA ŠEMA

Električna šema projekta se sastoji iz tri dela i to iz kola za napajanje, kola mikrokontorlera i kola *bluetooth* modula. Kolo za napajanje obezbeđuje određeni naponski nivo kako mikrokontroleru tako i *bluetooth* modulu, dok se komunikacija između mikrokontrolera i *bluetooth* modula odvija preko UART serijske komunikacije. Dakle, oni komuniciraju direktno pomoću dve linije za prenosa podataka, između TX pina na predajniku i RX pina na prijemniku.

Prvi deo šeme, prikazan na Slici 1, predstavlja kolo za napajanje. Napon od 5 V koji se dovodi preko USB porta, se LM117S [1] linearnim regulatorom smanjuje na vrednost od 3.3 V koji je neophodan za rad *bluetooth* modula. Izlazni napon od 3.3 V se definiše na osnovu vrednosti otpornika R8 i R9 pomoću jednačine (1):

$$Vout=1.25(1+R9/R8) \quad (1)$$



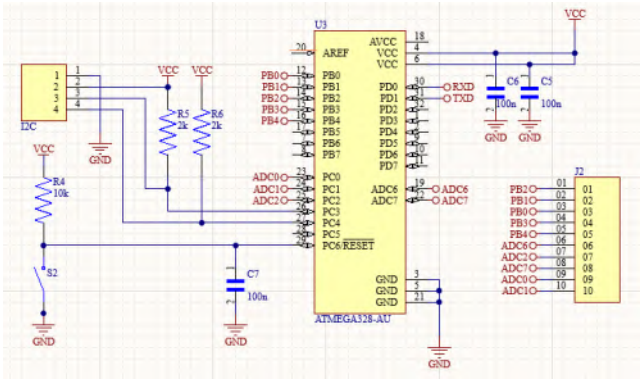
Slika 1. Šema kola napajanja.

Kondenzator C10 služi da smanji poskakivanje signala, elektrolitski kondenzator C4 na izlazu smanjuje AC komponente signala sa strane opterećenja, dok C2 i C3 služe da smanje šumove ulaznog napona. Za indikator rada se koristi LED dioda čiji se intenzitet svetlosti određuje otpornikom R7.

Drugi deo šeme, prikazan na Slici 2, predstavlja kolo mikrokontrolera. Mikrokontroler korišćen za izradu ovog projekta jeste ATMega 328 [2]. Ovaj mikrokontroler je odabran zbog kompatibilnosti sa *Arduino UNO* razvojnim okruženjem i zato što omogućava da se softverski razvoj vrši korišćenjem *Arduino IDE* kompajlera. ATMega 328 predstavlja CMOS 8-bitni mikrokontroler male snage zasnovan na AVR poboljšanoj RISC arhitekturi, omogućava optimizaciju potrošnje energije u odnosu na brzinu rada jer izvršava instrukcije u jednom ciklusu takta. Sa taktom od 16 MHz, ATMega 328 može dostići brzinu prenosa podataka pomoću serijske UART komunikacije od 1 Mbps (125000 karaktera svake sekunde). Za potrebe UART komunikacije iskorišćeni su pinovi PD0 i PD1 koji predstavljaju RX i TX pinove mikrokontrolera respektivno. Kako bi se izbegli visokofrekventni šumovi na liniji napajanja su povezani *bypass* kondenzatori C6 i C5. Za resetovanje mikrokontrolera se koristi taster koji je *pull up* otpornikom R4 povezan sa napajanjem. *Pull up* otpornicu R5 i R6 podižu pinove SCL i SDA koji služe za prenos podataka korišćenjem I^2C porta na visok naponski nivo. I^2C port je iskorišćen za povezivanje OLED (*Organic Light Emitting Diode*) *display* koji služi za praćenje kontrole rada, debagiranje i slično. *Socket* koji se nalazi na šemi,

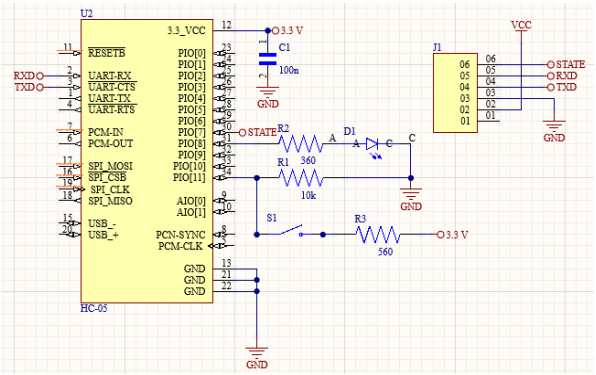
N. Džunić je master inženjer elektrotehnike i računarstva za Elektoniku i mikrosisteme, Elektronski fakultet Niš, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: nikoleta.dzunic@elfak.rs.

nakon fizičke izrade ploče korisniku će omogućiti pristup analognim i digitalnim pinovima mikrokontrolera.



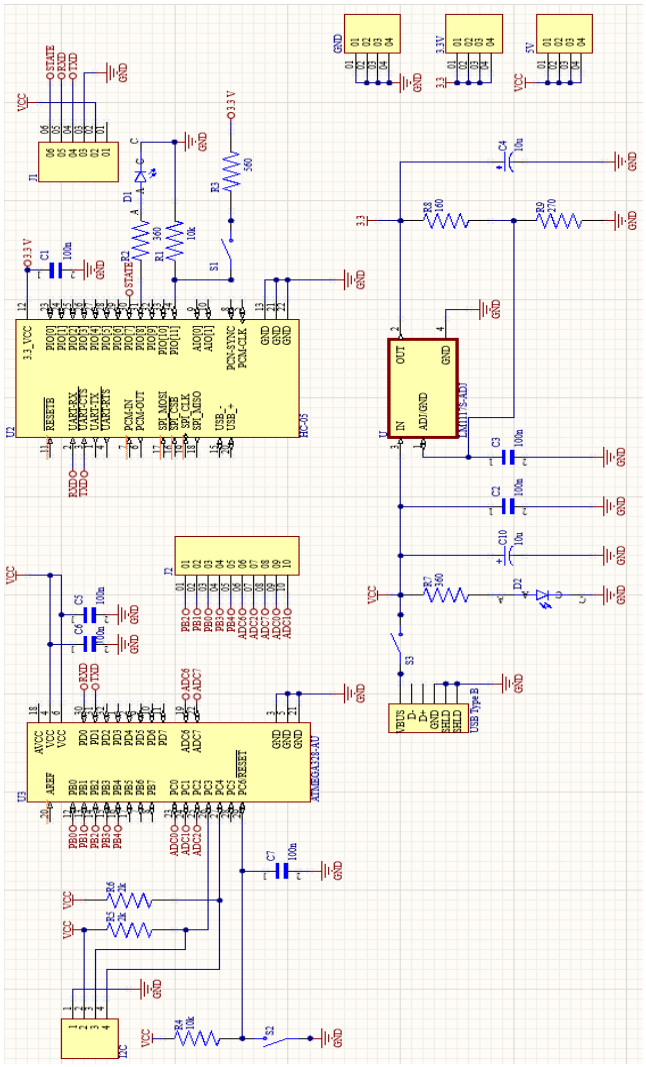
Slika 2. Šema kola mikrokontrolera.

Poslednji deo šeme, prikazan na Slici 3, jeste kolo HC-05 *bluetooth* modula [3]. Jezgro ovog modula je *bluetooth* kolo koje se zove CSR BC417 [4]. Ovo kolo je dizajnirano da smanji broj potrebnih spoljašnjih komponenti čime smanjuje troškove proizvodnje. CSR BC417 se povezuje sa eksternom fleš (*flash*) memorijom kapaciteta 8 Mbit i potpuno je usaglašen sa *bluetooth* v2.0+ EDR (*Enhanced Data Rate*) kod kojih brzina prenosa dostiže i 3 Mbps. HC-05 sadrži ukopanu antenu koja može da podnese širok opseg propusnog opsega, a struktura nalik tankom filmu čini antenu kompaktnijom i bestežinskom. LED u ovom delu kola služi za prikaz režima rada HC-05 *bluetooth* modula. Ukoliko ona treperi nekoliko puta u 1 sekundi to znači da modul čeka uspostavljanje veze, ukoliko treperi dvaput u 1 sekundi onda je HC-05 Bluetooth modul povezan, tj. uspostavljena je veza, a ukoliko treperi jednom u 2 sekunde modul se nalazi u AT komandnom režimu, koji služi za njegovo programiranje. Pristup AT komandnom režimu se vrši pomoću tastera, dok *jumper* služi za ostvarivanje UART komunikacije sa mikrokontrolerom.



Slika 3. Električna šema *bluetooth* modula.

Kompletna šema projekta je prikazana na Slici 4. Dakle, ona se sastoji iz tri dela koja su prethodno detaljnije objašnjena, međutim na slici se mogu videti dodatna tri *socket*-a za pristup napajanjima od 3.3 V, od 5 V i masi (GND). Ovim *socket*-ma se omogućava da korisnik pločice u svakom trenutku kratkospojnikom može iskoristiti potrebno napajanje ukoliko mu je ono neophodno.



Slika 4. Kompletna šema projekta.

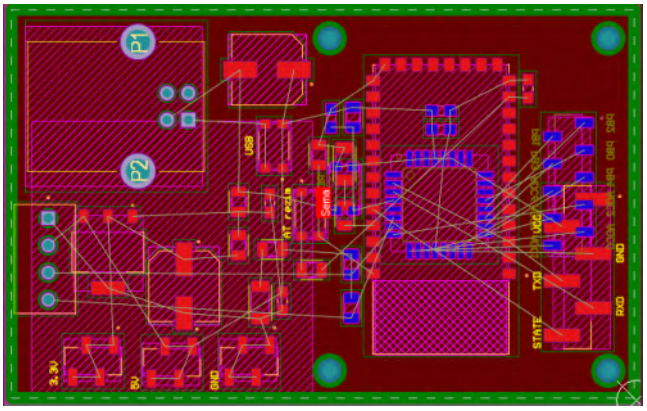
Na šemi se mogu primetiti dva načina povezivanja koje *Altium Designer* podržava. Jedan od njih je povezivanje komponenti pomoću labela, čime se omogućava veća preglednost same šeme, tzv. logičko povezivanje, dok drugi način podrazumeva direktno povezivanje komponenata linijama veza koje se iscrtavaju između njih.

III. *LAYOUT* PROJEKTOVAN U *ALTIUM DESIGNER*-U

Svakom simbolu komponenti prilikom crtanja šeme pridružen je odgovarajući *footprint*, otisak gde će ona biti zalemljena prilikom fizičke izrade ploče. *Update PCB Document* kome se pristupa u okviru kartice *Design* prebacuje sve *footprint*-ove u .PcbDoc dokument. Prilikom

prebacivanja sve komponente se uglavnom nalaze sa desne strane ploče, tako da ih je potrebno prerasporediti na štampanoj ploči tamo gde će se fizički nalaziti nakon izrade [5].

Prilikom ređanja *footprint*-ova na ploči potrebno je voditi računa o tome da se na što efikasniji način oni rasporede, kako u električnom tako i u vizuelnom smislu. Usled potrebe za minijaturizacijom u ovom projektu kolo mikrokontrolera zajedno sa OLED *display*-em je smešteno na *Bottom Layer*-u, dok su komponente kola za napajanje i *bluetooth* modula na *Top Layer*-u. *Layout*-a sa raspoređenim komponentama je prikazan na Slici 5.



Slika 5. *Layout* projekta.

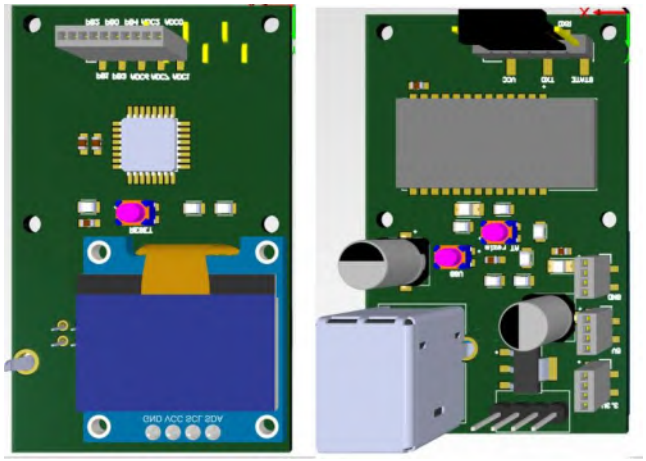
Sastavni deo ređanja komponenti na ploču jeste i uvoženje njihovih 3D modela i dodeljivanje istih *footprint*-ovima. 3D modeli služe kako bi se dobio vizuelno bolji pregled onoga što se projektuje, jednostavnije rečeno dodavanjem 3D modela moguće je videti kako će ploča fizički izgledati nakon realizacije. Na *Layout*-u roze šrafirane površine predstavljaju 3D modele.

Pored uvoženja postojećih 3D modela, pomoću naredbe *3D Body* u okviru kartice *Place*, *Altium Designer* nudi mogućnost crtanja različitih oblika poput cilindra, četvorougaonika ili sfere koji mogu vizuelno predstaviti određenu komponentu. 3D model HC-05 modula je nacrtan na taj način. Važno je samo da se nakon postavljanja 3D modela on zaključa zajedno sa *footprint*-om, koristeći dugme *Primitives* u okviru *Properties* komponente koju on predstavlja.

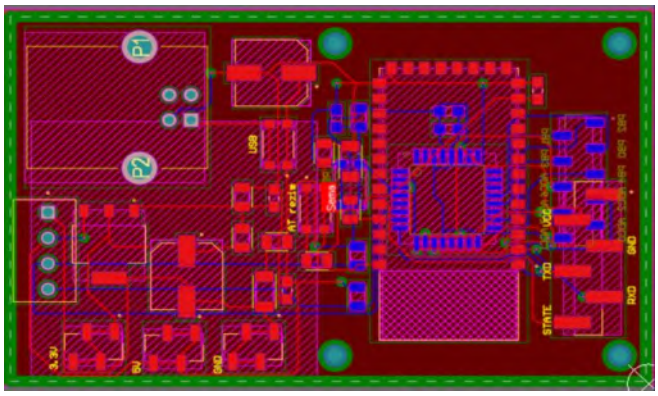
Izgled *Layout*-a sa 3D modelima sa gornje i donje strane prikazn je na Slici 6. Sa desne strane je prikazan izgled *Top Layer*-a ploče sa 3D modelima komponenata, dok je na levoj strani prikazan izgled *Bottom Layer*-a.

Kada je šematski realizovano kolo i okvirno raspoređene komponente, potrebno ih je povezati u električnom smislu, odnosno izvršiti rutiranje. Rutiranje se vrši po pravilima projektovanja štampanih ploča.

Pravila projektovanja koja su predefinisana u ovom projektu su: *Minimum Solder Mask Sliver* (minimalno rastojanje između metalizacije), *Width* (minimalno dozvoljenu širinu linije), *Clearence* (rastojanje između dva objekta) , *Hole size* (minimalno i maksimalno dozvoljenu širinu rupe) i *SilkToSilkClearance* (minimalni dozvoljeni razmak između teksta i drugih objekata na ploči). Nakon njihovog podešavanja izvršeno je rutiranje pomoću opcije *AutoRoute* koju nudi A*ltium Designer*, tako da konačni *Layout* projektovane ploče izgleda kao na Slici 7.



Slika 6. *Layout*-a sa 3D modelima sa gornje i donje strane.



Slika 7. *Layout* projekta nakon rutiranja.

Kako bi ploča bila spremna za fizičku izradu, neophodno je razliti masu i sa njene gornje i donje strane. To se radi zbog smanjenje električnog šuma, minimiziranja neželjenih petlji i preslušavanja između pojedinih veza na štampanoj ploči. Razlivanje mase je moguće realizovati pomoću opcije *Polygon Pour* koja se nalazi u okviru kartice *Place*.

IV. TESTIRANJE

Testiranje pločice podrazumeva proveru rada HC-05 *bluetooth* modula kako u AT komandom režimu tako i u režimu podataka.

AT režimu se pristupa pomoću tastera na ploči koji je označen sa „AT rezim“. Testiranje ploče u ovom rada zahteva slanje određenih AT komandi pomoću serijskog terminala *Arduino IDE* kompajlera, koji je pre toga programiran za tu svrhu, i primanje odgovora od strane

modula. U Tabeli I su date neke od osnovnih AT komandi [6].

TABELA I
OSNOVNE AT KOMANDE

AT komande	Opis	Odgovor
AT	Test naredba	OK
AT+VERSION?	Provera verzije	+VERSION:***
AT+PSWD?	Provera pina	+ PSWD:***
AT+ROLE?	Provera načina rada	+ROLE: (0/1/2)
AT+ADDR?	Provera adrese	+ADDR:***
AT+NAME?	Provera imena	+NAME:***
AT+NAME= *ime*	Promena imena	OK/FAIL

Testiranje ploče u režimu podataka se obično vrši uparivanjem modula sa telefonom (koji je podešen da bude *maser*) preko kog se šalju naredbe za uključivanje/isključivanje LED diode pri čemu je modul podešen kao *slave*.

Korišćenjem određenih AT komandi može se vršiti konfigurisanje HC-05 modula na način koji je pogodan za njegovu primenu. Kako provera rada modula u režimu podataka zahteva da on prima podatke, odnosno da radi kao *slave*, korišćenjem AT komandi iz Tabele II on se konfiguriše za tu svrhu.

TABELA II
AT KOMANDE ZA PODEŠAVANJE SLAVE UREĐAJA

AT komande	Odgovor
AT	OK
AT+NAME?	+NAME: HC-05
AT+PSWD?	+PSWD:1234
AT+VERSION?	+VERSION:2.0-20100601
AT+ADDR?	+ADDR:96d3:31:f41822
AT+UART?	+UART:9600,0,0
AT+INIT?	OK
AT+STATE?	+STATE:INIALIZED
AT+RMAAD?	OK
AT+CMODE=1	OK
AT+INQ?	OK
AT+STATE?	+STATE:PAIRABLE
AT+STATE?	+STATE:PAIRED

Na samom početku potrebno je proveriti osnovne podatke HC-05 modula (naziv, pin, verzija i adresa), zatim omogućiti serijsku komunikaciju, dodeliti inicijalizaciju, obrisati lista trenutnih ovlašćenih uređaja i podesiti način povezivanje modula da bude isti kao i kod modula koji je konfigurisan da bude *master*. Nakon čega se započinje pretraživanje dostupnih uređaja za taj modul i na kraju se vrši uparivanje

Prema tome, uređaj koji šalje podatke modulu treba da bude uparen s njim. To je moguće jedino ako je *bluetooth* modul tog uređaja podešen kao *master*. Konfigurisanje modula da radi kao *master* se vrši koristeći AT komande iz Tabele III.

TABELA III
AT KOMANDE ZA PODEŠAVANJE MASTER UREĐAJA

AT komande	Odgovor
AT	OK
AT+NAME?	+NAME: MODUL
AT+PSWD?	+PSWD:1234
AT+VERSION?	+VERSION:2.0-20100601
AT+ADDR?	+ADDR:96d3:34:9056fa
AT+UART?	+UART:9600,0,0
AT+INIT?	OK
AT+STATE?	+STATE:INIALIZED
AT+ROLE?	+ROLE:1, OK
AT+CMODE?	+CMODE:1, OK
AT+INQM?	+INQM: 0,5,10, OK
AT+INQM=0,5,5	OK
AT+INQM?	+INQM: 0,5,5, OK
AT+INQ?	+INQ: <available address will displa>
AT+INQ?	+INQ: <slave address will display>
AT+ RNAM? <slave address>	HC-05
AT+STATE?	+STATE:INQUIRING
AT+PAIR=[slave address]	OK
AT+STATE?	+STATE:PAIRED
AT+LINK=[slave address]	OK

Za potrebe i ove kofiguracije se na samom početku proveravaju osnovni podaci o HC-05 modulu (naziv, pin, verzija, adresa i parametar), nakon čega mu je potrebno omogućiti serijsku komunikaciju, dodeliti inicijalizaciju, omogućiti preuzimanje i povezivanje, proveriti trenutni status modula, dodeliti mu adresu uređenja sa kojim ga je potrebno upariti i proverava da li je uparivanje uspelo i na samom kraju dati mu link uređuje s kojim se uparuje [7].

Dakle, za uparivanje dva *bluetooth* modula, potrebno je da jedan od njih bude konfigurisan kao *master*, a drugi kao *slave*. Osim toga, brzina prenosa podataka oba modula treba da bude ista, isto tako i SPP (*Serial Port Profile*) i INQ (*Inquiring*) moraju biti pokrenuti kod oba modula, što se definiše korišćenjem odredjenih AT komandi. SPP omogućava ostvarivanje komunikacije, dok INQ prikazuje listu uređaja sa kojim može modul da se upari.

V. ZAKLJUČAK

U ovom radu je opisan način projektovanja i testiranja ploče za komunikaciju sa HC-05 *bluetooth* modulom. Projektovanje je urađeno u *Altium Designer* jer on pruža puno pogodnosti kako u pogledu projektovanja tako i kompatibilnosti sa *footprint*-ovima i 3D modelima koji se mogu naći na internetu. Dok je testiranje urađeno koristeći *Arduino IDE* kompajler koji je pogodan za programiranje ATMega 328 mikrokontrolera.

ZAHVALNICA

Zahvalnost za ovaj rad uputila bih asistentu Nikoli Mitroviću, koji mi je od samog početka davao ideje i savete kako bih na što bolji način realizovala projekat i profesoru dr Danijelu Dankoviću, na nesebičnoj saradnji, podršci i pomoći koju mi je pružio od samog početka realizacije projekta.

LITERATURA

[1] Tehnička dokumentacija za LM117S. Dostupno na: https://www.ti.com/lit/ds/symlink/lm1117.pdf?ts=1665901512624.

[2] Tehnička dokumentacija za ATMega 328. Dostupno na: https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061A.pdf.

[3] Tehnička dokumentacija za HC-05. Dostupno na: https://components101.com/sites/default/files/component_datasheet/HC-05%20Datasheet.pdf

[4] Tehnička dokumentacija za CSR-BC417. Dostupno na: https://cdn.sparkfun.com/datasheets/Wireless/Bluetooth/CSR-BC417-datasheet.pdf

[5] Prateći materijal za predavanja i vezbe iz predmeta projektovanje štampanih ploča, Danijel Danković. Dostupno na: http://mikro.elfak.ni.ac.rs/wp-content/uploads/PROJEKTOVANJE-STAMPANIH-PLOCA-2017.pdf

[6] Povezivanje bluetooth HC-05 modula na arduino. Arduino lekcije: upravljanje pametnim telefonom za lutke. Povezivanje sa pametnog telefona pomoću bluetooth terminala. Dostupno na: https://whatsappss.ru/bs/tools/podklyuchenie-bluetooth-hc-05-k-arduino-uroki-arduino-upravlenie-ustroistvami-so.html4

[7] Primer uparivanja dva *bluetooth* modula. Dostupno na: https://www.engineersgarage.com/pairing-two-hc-05-bluetooth-modules-and-share-data-between-them/*Analysis of Integrated Circuits IPFA 2001*, Singapore, 2001, pp. 243-248.

GPU Implementation of SVM Algorithm

Nikola Vukotić

Abstract - The development of machine learning algorithms is essential for the future of computing. Of old software systems were designed to work as defined, today they have become "smart" and can predict results, showing users suggestions on various topics and of great help in everyday life. One of the machine learning algorithms that has gained great popularity is certainly the SVM algorithm for classification. He provided to be very accurate, but one of his problems is that model training takes quite a long time. The aim of this paper is to speed up this part and thus reduce the execution time of model training, but also the entire program. An interesting way to achieve this is seen in the use of increasingly popular graphics processors. Speedup has been archived by using CUDA software model which allows incredibly fast graphics processors to be used even for general purpose computing. The highest speedup was archived by using 32 blocks and 1024 threads, and the one for the part of the program that is parallelized is 6.11. The acceleration of the part of the algorithm for training the model is 4.02 and the overall execution of the program is 3.94.

I. INTRODUCTION

Over the past ten years, supervised learning models have experienced a rapid expansion. Applications of such models can be seen in the most diverse spheres of everyday life, and they have found special application in business, science, and industry.

The SVM algorithm is one of the most widely used classification algorithms. This algorithm is a binary classifier by nature, but like all binary classifiers, it can also be applied for multi-class classification. The mathematical apparatus behind this algorithm is quite complex, but the prediction is very fast and accurate. The challenge with this algorithm is the performance of model training as it can take quite a long time.

Graphics processors are developing exponentially with the progress of computing and with the need to constantly provide users with better experiences in using the device. The 3D industry is also developing rapidly. Over time, it has been established that graphics processors are ideal if they are used to execute parts of the program on them, considering that a high degree of parallelism can be achieved on them, and thus significant improvements in system performance are achieved.

Training of machine learning algorithms is one of the obvious examples of very computationally intensive algorithms, and for this reason, in recent years, a lot of work has been done on the use of graphics processors for the execution of certain parts of it. In this paper, the graphics processor was used to train the SVM model.

N. Vukotić is a master's student at the Department of Computer Science, Faculty of Electronics, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: nikolaraca@elfak.rs.

The exposition below is organized as follows: The second area is devoted to a detailed explanation of the SVM algorithm and the mathematical apparatus according to which it works. The third area discusses graphics processors, their architecture and the benefits they bring to improve system performance. In the fourth chapter, the CUDA programming model is explained as one of the most current methods for using graphics processors for general purpose calculations. The next chapter contains a description of the sequential and parallel implementation of the SVM algorithm. Finally, the sixth area is reserved for the evaluation of the results achieved using the parallel SVM algorithm, and the performance comparisons of sequential and parallel implementations are also given.

II. SVM ALGORITHM

Support vector machines or abbreviated SVM (SVM support vector machines) belong to the group of non-probabilistic supervised training models. SVM is a binary linear classifier, which means that it separates, i.e., classifies objects into two categories. The objects with which this model works are represented by multidimensional vectors of features or features. Since every object is associated with a vector, which means that each object is mapped to a point in some multidimensional space, the goal of the SVM method is to find a hyperplane that separates the points associated with objects of opposite classes. If it is a two-dimensional space, the required hyperplane is a line (Figure 1).

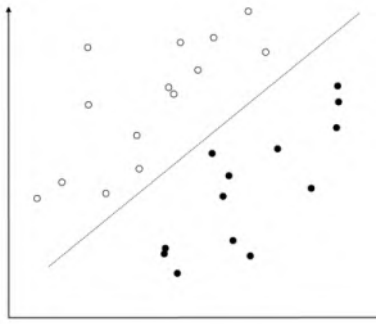


Fig. 1. Black and white dots as objects of opposite classes.

Objects are represented by vectors $x^{(i)}$ where each vector is associated with a corresponding class $y^{(i)}$. The hyperplane is represented by the equation:

$$\omega^T x + b = 0, \qquad x, \omega \in \mathbb{R}^n, \qquad b \in \mathbb{R}, \qquad (1)$$

In the mentioned equation, ω is a vector normal to the plane, while b is a constant that determines the distance of the plane from the coordinate origin. Therefore, the goal is to find values for ω and b so that all points $x^{(i)}$ on one side of the plane are those marked with 1, while on the other side those marked with -1 should be found [1]-[4]. The problem is, however, that the number of planes that make such a division is mostly infinite. SVM solves this problem by setting a plane so that the margin, i.e. the distance from the nearest points should be as large as possible. This is why SVM is said to be a high-margin classifier. This can be understood in the way that instead of a line (plane), a strip of certain thickness is drawn between the points. In that case, the number of possible positions is no longer infinite.

The points with the smallest margin are the points closest to the given hyperplane. In the example in Figure 2, there are three points, framed by a red circle, located on a broken line parallel to the line representing the hyperplane. These three points are called support vectors. Generally, the number of such vectors is much smaller than the total number of points in the training set. In other words, the hyperplane that separates the training set is uniquely determined by the support vectors.

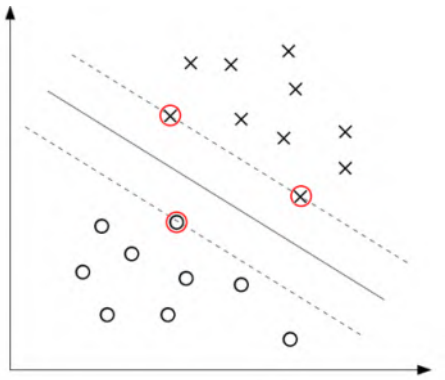


Fig. 2. Position of support vectors in the training set.

The SMO algorithm is an iterative algorithm for training support vector machines, which is reduced to solving the optimization problem [2]. The algorithm appeared in 1998, as a result of the work of Microsoft researcher John Platt and is considered one of the most effective algorithms in this field.

The algorithm works by dividing the initial problem into a series of subproblems of the smallest possible size, each of which is solved analytically. The algorithm ends when every element of the training set $(x^{(i)}, y^{(i)})$ satisfies the Karush-Kuhn-Tucker (KKT) conditions, whose form is as follows [3]:

$$\begin{aligned} \alpha_i = 0 &\Rightarrow y^{(i)}(\omega^T x + b) \geq 1 \\ \alpha_i = C &\Rightarrow y^{(i)}(\omega^T x + b) \leq 1 \\ 0 < \alpha_i < C &\Rightarrow y^{(i)}(\omega^T x + b) = 1 \end{aligned} \qquad (2)$$

In the roughest outline, the algorithm consists of the following steps:

1. Find the Lagrange multiplier α_j that violates the KKT conditions
2. Choose the second Lagrange multiplier α_i and optimize the pair (α_i, α_j)
3. Repeat 1. and 2. until all elements meet the KKT conditions

The algorithm consists of three main components: an analytical method for optimizing a pair of multipliers (α_i, α_j), a heuristic for choosing a multiplier and calculating the parameter b with updating the error cache [2].

III. GRAPHICS PROCESSOR ARCHITECTURE

The basic idea behind graphics processors is the use of many simpler and weaker processing units that simultaneously execute instructions on different data. The main difference between a central processor and a graphics processor is in the design. The GPU is oriented towards data processing, while the CPU is oriented towards the traditional execution of tasks, so in the case of graphics, many more transistors are intended for data processing, than for caching and flow control. Figure 3 shows the difference in CPU and GPU design [5].



Fig. 3. Comparison of CPU and GPU designs.

A graphics processor is also called a device in a different way. It consists of several multiprocessors consisting of a set of processors and shared memory. A set of arithmetic logic units, registers and common logic for obtaining and decoding instructions make up one streaming multiprocessor, while the GPU consists of a larger number of them. Individual streaming multiprocessors (SMs) are not overly powerful, but when they work in parallel, they have a lot of computing power, so they are very convenient for computationally intensive calculations.
A graphics processor is also called a device (English device). As said, it consists of several multiprocessors consisting of a set of processors and shared memory.

A. A software view of the graphics processor

For the sake of easier monitoring of the further flow of the presentation, a software view of the graphics processor should be shown. Namely, the program that is executed on the graphics processor is called the kernel in a different way. The kernel consists of a grid consisting of a set of blocks of threads that execute the kernel. Blocks

therefore consist of many threads that execute the same instructions on different data and that can communicate via shared memory. So, the kernel executes as a grid of thread blocks. Figure 4 shows the mentioned hierarchy.

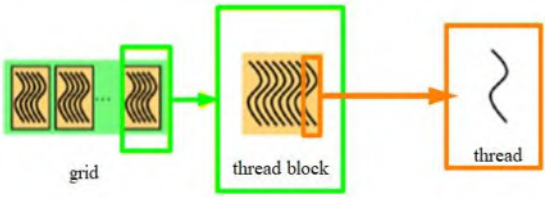


Fig. 4. Graphics processor in software view.

An obvious mapping between the hardware and software view of the graphics processor can be observed, as the entire grid is served by the device or graphics chip, each streaming multiprocessor is responsible for one or more blocks, while the individual scalar processor is responsible for an individual thread. It is important to note that there are no guarantees for the order in which blocks, and threads are executed.

Due to the structure of graphics processors, they are specialized for computationally intensive, highly parallel calculations. Although they were initially intended for graphics processing, today they are also used for general-purpose calculations, where they greatly contribute to the additional improvement of system performance if they are used together with the central processor in co-processing mode. The central processor is then used for the sequential part of the application where the delay is visible (input, output, data preparation, …), while the graphics processor is used for parts of the program that represent the processing of large amounts of data.

B. Programming on a graphics processor

Programming on the GPU is possible using the graphics API, however this approach has proven to be disadvantageous given that it is quite difficult and unintuitive to program using the graphics API, and there are also many restrictions in memory access.

One such way is provided by Microsoft's DirectX set of APIs, where for a developer to access computing resources, he would have to adapt his problem to graphics operations so that the computation could be run through OpenGL or the DirectX API, which, clearly, is not an easy job for the developer.

Unlike the previous method, the NVIDIA CUDA programming model provides a much better experience for programming on the GPU. CUDA enabled general purpose computations on GPUs without the need to use the graphics API and is available on NVIDIA 8000 series graphics cards and above.

IV. NVIDIA CUDA PROGRAMMING MODEL

Parallel computing refers to the process of dividing a problem into smaller, independent, often identical parts that can be executed simultaneously by multiple processors communicating via shared memory, and whose results are combined upon completion as part of an overall algorithm. The primary goal of parallel computing is to increase the available computation power for faster application processing and problem solving [6].

CUDA (Compute Undefined Device Architecture) is a parallel processing platform created by NVIDIA and implemented on their graphics processors. It is used to manage general purpose computations on graphics processing units. It is available to programmers through libraries, compiler directives, and extensions to the C, C++, and Fortran programming languages. A CUDA program consists of parts of the code that are executed on the central and parts that are executed on the graphics processor, which is the goal because the graphics processor can be best used in co-processing mode with the central processor.

To translate such programs, you need a compiler that can generate code that is executed on the central processor and code that is executed on the graphics processor. Considering that the program does not only contain the code of the central processor, otherwise known as the host, but also part of the code that is executed on the device, in addition to the traditional compiler, the NVIDIA compiler is also necessary, and for the programming language C and C++ it is the *nvcc* compiler [7], [8].

As mentioned, a CUDA program is executed on a central processor, otherwise called a host, and on a graphics processor, called a device. The execution of the program is started by the host, then after calling certain functions, the execution continues the graphics processor and on the so-called kernel, which executes a set of thread blocks. When kernel execution ends, the program continues the host until the kernel is called again, and so on.

V. IMPLEMENTATION OF SVM ALGORITHM

The algorithm is implemented sequentially and in parallel, and a comparison of these two implementations can be seen in the next chapter. Implementations written in C++ programming language, and CUDA version 11.4 was used for parallel implementation.

The idea of parallel implementation is to speed up the part of the algorithm that involves training the model. Before starting the training of the model, it is necessary to load from the file the data representing the set of vectors $x^{(i)}$ from the training set $\mathbb{D}$, as well as the data representing the series of labels $y^{(i)}$ with which the vectors $x^{(i)}$ are marked. Below is a description of the sequential algorithm for training the model by steps, and a detailed explanation of the algorithm as well as the mathematical apparatus is given in [2].

Step 1. Filling the kernel matrix
Step 2. Finding the Lagrange multiplier α_j that violates KKT conditions from the equation 2
Step 3. Selection of the second Lagrange multiplier α_i and then optimization of the pair (α_i, α_j)
Step 4. Calculating the parameter *b* and updating the error cache
Step 5. Repeat steps 2, 3 and 4 until all elements meet the KKT conditions

Step 1 is the most computationally intensive, since the kernel matrix is filled, which is of huge dimensions, and it is needed in the following steps of model training. Because filling the kernel matrix is the most computationally intensive part of the algorithm, it will be implemented later on the graphic processor.

A. Realization of the parallel part of the program

As it was said, the goal in achieving better performance compared to the sequential SVM algorithm lies in the fact that the kernel matrix that is filled during the training of the data set is of huge dimensions, and it requires a lot of time in execution. Namely, the dimensions of this matrix represented as an array are *(m * (m + 1)) / 2*, where *m* is actually the number of instances of the training data set. Due to the large dimensions, the parallel filling of the kernel matrix K can greatly improve the performance.

At the very beginning of the calculation of the kernel matrix, it is necessary to send the data loaded from the data set to the device. After that, the kernel matrix is filled on the device, and after it is filled, it is returned to the host. The kernel matrix filled in this way is later used on the host in the further work of the SVM algorithm.

The idea is that practically every thread in the grid is responsible for calculating one or more elements of the kernel matrix, so this filling takes place in parallel.

Figure 5 shows the pseudo-code of the sequential filler of the kernel matrix.

```
ComputeKernelMatrix() : void

    dimK <- (m * (m + 1))/2
    K[dimK]

    for(i = 0, m)
        for(j = i, m)
            offset <- (i * (2 * m - i - 1)) / 2 + j
            K[offset] <- Kernel(X[i], X[j])

end
```

Fig. 5. Sequential filling of the kernel matrix.

As said, a parallel algorithm for populating the kernel matrix would imply that each thread would be responsible for computing one or more elements of the kernel matrix. The realization of such an idea is carried out using n blocks and m threads, where n and m may or may not be different. Namely, given the fact that there is a limit on the maximum number of blocks and threads, as well as the fact that it is not known in advance how many instances the training data set has, the idea will be that each block is responsible for one or more types of kernel matrix, and each thread is responsible for one or more elements in a type.

Example 1. Let a training set with 5 instances be given.

The size of the kernel matrix represented as an array will be 15, the number of species will be 5, where the species have 5, 4, 3, 2 and 1 elements respectively. If a kernel with 2 blocks and 2 threads is created, the block with index 0 will oversee types 0, 2 and 4, while the block with index 1 will be in charge of types 1 and 3. Thus, the blocks are cyclically assigned types. As for the threads, each thread will have a part of the elements in the type, so in type 0, the thread with index 0 will process elements 0 and 1 and the thread with index 1 elements 2 and 3 and additionally element 4 because it is a thread with index 1 the last thread in the block. The requirement that the last thread in the block process some number of additional elements is necessary because it is not known in advance whether the number of elements in the row is divisible by the number of threads in the block.

Special attention should be paid to another specific case, which is the case when the number of elements in a kind of kernel matrix is less than or equal to the number of threads in the block. In that case, exactly as many threads as there are elements in the array will participate, and each thread will be responsible for exactly one element in the array. Figure 6. shows an illustration of example 1. The block with index 0 is marked in green, the block with index 1 in blue. The thread with index 0 is marked in red, and the thread with index 1 in purple.

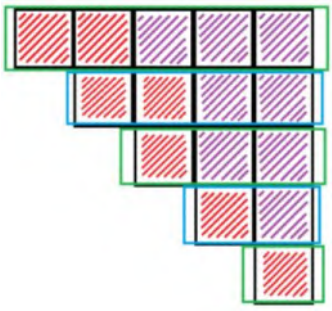


Fig. 6. Illustration of example 1.

VI. EVALUATION OF RESULTS

The goal of parallelizing the SVM algorithm is to improve its performance, i.e., to shorten the execution time. Unlike sequential programs, with parallel programs, the execution time depends on several factors, among which are the number of processing units used, their processing power, as well as the speed of interprocess communication. That is why, in addition to the execution time, as a metric for evaluating the performance of parallel systems, among other performance measures, speedup and efficiency are used.

Speedup (S) is defined as the ratio of the execution time of a program on a single-processor system to the execution time of a program on a system with *n* identical processors.

$$S(n) = \frac{T(1)}{T(n)} \quad (3)$$

Efficiency (E) is a measure of processor utilization. It is calculated as the ratio of the acceleration achieved and the number of processing units used. As the number of processing units increases, acceleration increases and efficiency decreases. In an ideal parallel system with acceleration *n*, the efficiency would be 1 or 100%.

$$E(n) = \frac{S(n)}{n} \quad (4)$$

A comparison of the results was performed on a banknote dataset [9] which involves predicting whether a banknote is authentic given a series of measures taken from a photograph. It is a binary (two-class) classification problem.

The evaluation of the results was performed on the NVIDIA RTX 2060 SUPER card with Turing architecture and 8GB GDDR6 memory [10].

As for the validity of the classifier's prediction, first in sequential and then in parallel, it was established by the fact that the classifier proved to be reliable considering that it predicts with an accuracy of 94.9% over the test data set, and 94.1% over the entire data.

The sequential execution times of the compute kernel matrix function, the entire model training, and the entire program are shown in Table I. Times are expressed in milliseconds.

TABLE I
SEQUENTIAL PROGRAM EXECUTION TIMES

Compute kernel matrix function	Model training	Program execution
1291	14587	15125

The execution times of the parallel implementation of the SVM algorithm were performed over different numbers of blocks and threads, and by comparing the performance of the sequential and parallel implementation, one can really see the shorter execution time of the parallel program.

Tables II, III and IV respectively show the program execution times for 1, 16 and 32 blocks.

By comparing the execution time of the sequential and parallel implementation when using one block and 16, 32, 64, 1024 threads, a shorter execution time can be observed already using 16 threads. However, as expected, the execution time with 32 threads is even shorter, while when using 64 threads it is longer compared to 32 but shorter compared to 16. Longer execution time with 64 threads compared to 32 although at first glance illogical, it is completely expected considering the hardware structure of the used processor described in [3]. Since only 1 block and 64 threads are used, actually 16 threads will actually work in parallel, and the other 48 will wait for the previous ones to finish. However, the greatest improvement when using one block is visible when using the maximum number of threads in one block, i.e. 1024, which also corresponds to expectations because the processor used is designed so that for the maximum number of threads in the block, it works with the greatest amount of optimization and time saving.

TABLE II
EXECUTION TIMES OF A PARALLEL PROGRAM WITH ONE BLOCK

Threads number	Compute kernel matrix function	Model training	Program execution
16	856	4227	4432
32	709	4069	4271
64	786	4175	4379
1024	277	3659	3864

TABLE III
EXECUTION TIMES OF A PARALLEL PROGRAM WITH 16 BLOCKS

Threads number	Compute kernel matrix function	Model training	Program execution
16	284	3683	3885
32	264	3618	3814
64	261	3676	3878
1024	214	3584	3787

TABLE IV
EXECUTION TIMES OF A PARALLEL PROGRAM WITH 32 BLOCKS

Threads number	Compute kernel matrix function	Model training	Program execution
16	239	3646	3847
32	238	3620	3822
64	241	3615	3818
1024	211	3629	3831

Speedups achieved by measuring the execution of the computational core matrix method are shown in Figure 7. The maximum speedup is achieved using 32 blocks and 1024 threads. On this occasion, speedup of 4.02 was achieved for model training and 3.94 for the entire execution of the program.

VII. CONCLUSION

Support vector machines, although less popular than neural networks in recent times, are still a common choice when deciding whether to implement a supervised learning algorithm. Training the model takes a long time, but after the model is trained and support vectors are found for a given problem, the prediction of new data is quite accurate and fast since there are not many support vectors. However, it is

possible to additionally speed up part of the training of the model, and one of the possibilities is certainly the use of graphics processors. Within this paper, an acceleration of 4.02 was achieved during model training. When testing the parallel algorithm, it was compared with the sequential one that was also implemented in the work.

Considering that the testing was performed on one data set, which was not suitable for parallelizing additional parts of the algorithm, one of the further steps to achieve even better performance is certainly to parallelize some other parts of the algorithm. Data sets with an even larger number of instances, a larger number of input variables, and a larger number of labels with which the data can be classified will be suitable for testing such an improved algorithm.

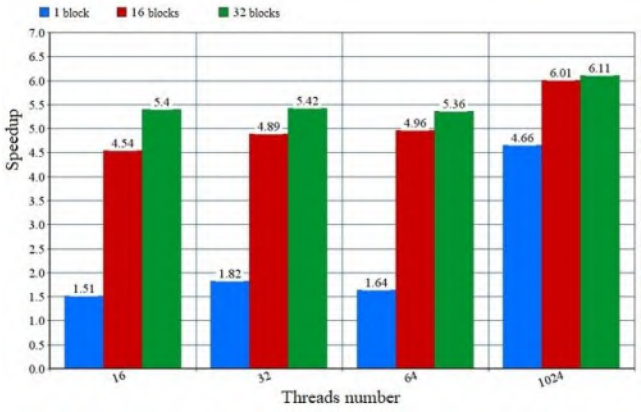


Fig. 7. Achieved speedups.

REFERENCES

[1] Bryan Catanzaro, Narayanan Sundaram, Kurt Keutzer, *Fast Support Vector Machine Training and Classification on Graphics Processors*, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, USA, 2008

[2] *The Simplified SMO Algorithm*, Available at: http://cs229.stanford.edu/materials/smo.pdf (accessed October 29, 2021)

[3] John Platt, *Fast Training of Support Vector Machines using Sequential Minimal Optimization, in Advances in Kernel Methods – Support Vector Learning*, B. Scholkopf, C. Burges, A. Smola, eds., MIT Press, Cambridge, USA, 1998

[4] Jiashuai Shi, Zeyi Wen, Bingsheng He, Jian Chen, *Efficient Support Vector Machine Training Algorithm on GPUs*, South China University of Technology, Guangzhou China, National University of Singapore, 2018

[5] *What is GPU and CPU (differences and similarities)*, Available at: https://www.quora.com/What-is-GPU-and-CPU-differences-and-similarities (accessed 1 October 2021)

[6] *What is parallel computing? Definitions and FAQs*, Available at: https://www.omnisci.com/technical-glossary/parallel-computing (Accessed: 3 November 2021)

[7] *CUDA by Example, An Introduction to general-purpose GPU programming*, Available at: http://www.mat.unimi.it/users/sansotte/cuda/CUDA_by_Example.pdf (accessed 23 October 2021)

[8] *A performance study of general-purpose applications on graphics processors using CUDA*, Available at: https://www.sciencedirect.com/science/article/abs/pii/S0743731508000932 (accessed 1 November 2021.)

[9] *10 Standard Datasets for Practicing Applied Machine Learning*, Available at: https://machinelearningmastery.com/standard-machine-learning-datasets/ (accessed October 18, 2021)

[10] NVIDIA Corporation, "*CUDA Turing GPU Architecture v1.1*", 2017*10*

Uređaj za merenje nivoa vode

Katarina Marković

Sadržaj − U ovom radu opisan je princip rada uređaja za merenje nivoa vode. Koristi se u svrhu očitavanja nivoa vode u sudu. Za realizaciju ovog projekta potreban je senzor za merenje nivoa vode, dva otpornika, dva tastera (pomoću kojih korisnik zadaje graničnu vrednost nivoa vode u sudu), potenciometar, LCD display (pomoću kojeg se prikazuje nivo vode i granična vrednost nivoa vode zadata od strane kosrisnika) i zujalica (pomoću koje se obaveštava kada nivo vode postane jednak ili pređe graničnu vrednost nivoa vode koju je zadao korisnik).

I. UVOD

Stalne klimatske promene, zagađenja i trenutna globalna zagrevanja i suše, glavni su uzrok nedostatka pijaće vode u mnogim delovima sveta.

Prema podacima UN-Water-a (*United Nations Water*), 2,3 milijarde ljudi živi u zemljama sa problemom vode. Prema UNICEF-u (*United Nations International Children's Emergency Fund*), 1,42 milijarde ljudi – uključujući 450 miliona dece – živi u područjima visoke ili izuzetno visoke ugroženosti vode. Svetska zdravstvena organizacija-SZO (na engleskom *WHO*, od izraza *World Health Organization*) [1] izveštava da 884 miliona ljudi nema pristup bezbednoj vodi za piće. Dve trećine svetske populacije doživljava ozbiljnu nestašicu vode tokom najmanje jednog meseca u godini. Posebno treba istaći da otprilike 73% ljudi pogođenih nestašicom vode živi u Aziji [2].

Da bismo pomogli smanjenju gubitka pijaće vode širom sveta, potrebno je da stalno imamo uvid u stanje iste u vodovodima i vodotokovima. U tu svrhu možemo upotrebiti uređaj za merenje nivoa vode koji će biti opisan u ovom radu. Rad ovog uređaja bazira se na očitavanju nivoa vode u sudu.

II. PRINCIP RADA SISTEMA

Ovaj sistem se zasniva na Arduino Uno razvojnom okruženju [3]. Glavna komponenta ovog sistema jeste senzor za merenje nivoa vode [4], prikazan na slici 1.

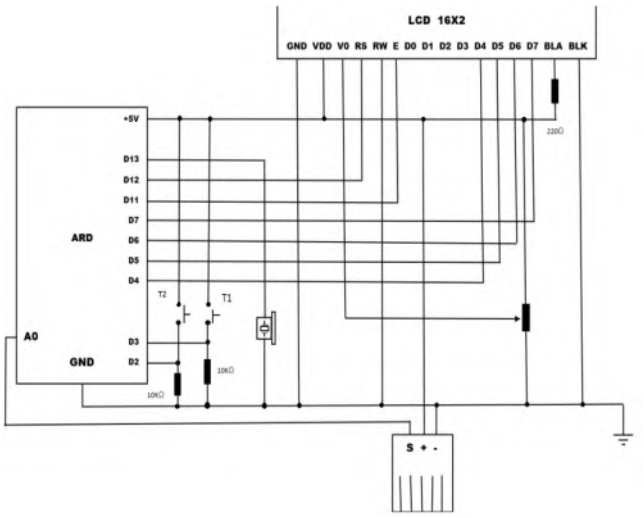


Slika 1. Senzor za merenje nivoa vode.

K. Marković, učenik četvrtog razreda Tehničke škole “Rade Metalac”, Dimitrija Tucovića bb, 16000 Leskovac, Srbija, E-mail: delekaca@gmail.com.

To je senzor kod kojeg se pokazivanje izmerenog nivoa vode menja u zavisnosti količine vode u sudu. Korisnik pomoću jednog tastera može povećati graničnu vrednost nivoa vode za 50, a pomoću drugog tastera smanjiti. Inicijalno, granična vrednost nivoa vode je 0, a zatim je korisnik može menjati. Radi se o otpornom senzoru, tj. promenom nivoa vode menja se otpornost provodnih linija na senzorskom modulu. Da bi sve ovo bilo vidljivo, na LCD-u će se prikazivati senzorom izmeren nivo vode i granična vrednost nivoa vode koju korisnik može podešavati. Ukoliko je nivo vode jednak ili prelazi trenutno definisanu graničnu vrednost od strane korisnika aktiviraće se neki vid alarmanog sistema, a u ovom slučaju to je zujalica. Međutim, ukoliko je nivo vode manji od trenutne granične vrednosti zadate od strane korisnika alarmni sistem, odnosno zujalica, neće biti aktivna.

Na slici 2 je prikazana električna šema ovog sistema. Za realizaciju ovog projekta potrebno je trinaest pinova sa Arduino okruženja, od kojih su devet digitalni pinovi, a jedan analogni. Sa pinova GND i 5 V dovodimo uzemljenje i napajanje na proto ploču.



Slika 2. Električna šema uređaja.

Senzor za merenje nivoa vode povezan je tako što je pin S sa senzora povezan na analogni pin A0, plus sa senzora je povezan na 5 V na Arduinu, a minus sa senzora je povezan na masu na proto ploči.

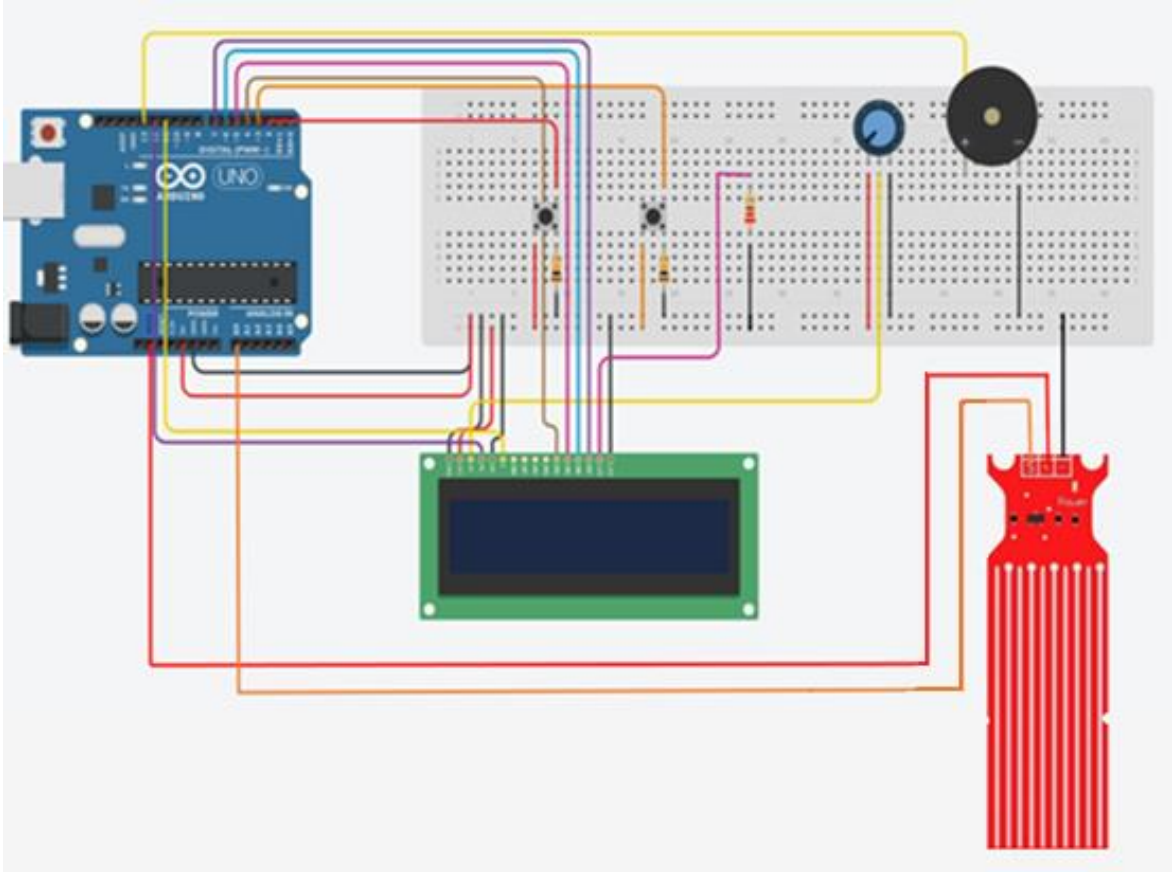
LCD je povezan na sledeći način. GND sa displeja je povezan na masu na proto ploči, VDD sa displeja je povezan na 5 V na proto ploči, V0 sa displeja povezan je na srednju nožicu potenciometra, RS sa displeja povezan je na digitalni pin 12 na Arduinu, RW sa displeja povezan je na masu na

proto ploči, E sa displeja povezan je na digitalni pin 11 na Arduinu, D4 sa displeja povezan je na digitalni pin 4 na Arduinu, D5 sa displeja povezan je na digitalni pin 5 na Arduinu, D6 sa displeja povezan je na digitalni pin 6 na Arduinu, D7 sa displeja povezan je na digitalni pin 7 na Arduinu, BLA sa displeja povezan je na otpornik R=220 Ω, koji je dalje povezan na masu na proto ploči, BLK sa displeja povezan je na masu na proto ploči. Jedan kraj zujalice povezan je na digitalni pin 13 na Arduinu, dok je drugi kraj povezan na masu na proto ploči. Pinovi RS, RW i E su kontrolni pinovi, dok D4-D7 služe za prenos podataka koji se prikazuju, Takođe, otpornik R=220 Ω služi za ograničenje struje kroz LED koji predstavlja pozadinsko osvetljenje LCD display-a.

Što se tiče prvog tastera, jedan kraj tastera povezan je na digitalni pin 2 na Arduinu, dok je drugi kraj tastera povezan na otpornik R=10 kΩ koji je dalje povezan na masu na proto ploči. Na sličan način je povezan i drugi taster. Jedan kraj tastera povezan je na digitalni pin 3 na Arduinu, dok je drugi kraj tastera povezan na otpornik R=1 kΩ koji je dalje povezan na masu na proto ploči. Oba povezana tastera su u *pull-down* konfiguraciji. Korisnik pomoću jednog tastera može povećati graničnu vrednost nivoa vode za 50, a pomoću drugog tastera smanjiti.

Potenciometar je povezan na sledeći način. Prvi kraj potenciometra povezan je na 5 V na proto ploči, sredina je povezana na V0 na displeju, a krajnja nožica povezana je na masu na proto ploči. Potenciometar se koristi za podešavanje kontrasta displeja.

Na slici 3 prikazana je električna šema uredjaja realizovanog korišćenjem Arduino razvojnog okruženja urađena u softverskom paketu *TinkerCAD* [5]. Nakon povezivanja komponenata na protoploči, potrebno je napisati kod programa.



Slika 3. Šema povezivanja uređaja.

III. KOD PROGRAMA

Na slici 4 prikazan je softverski kod za Arduino razvojno okruženje. Za senzor za merenje nivoa vode je iskorišćen analogni pin A0. Tasteri su povezani na pin 2 i 3, koji su postavljeni da budu ulazni digitalni pinovi, dok je zujalica povezana na digitalni pin 13, koji je definisan kao izlazni digitalni pin. Ukoliko senzor očita jednak ili veći nivo vode od trenutno postavljenje granične vrednosti, aktiviraće se zujalica. U suprotnom, zujalica neće biti aktivna. Pritiskom na tastere korisnik može manipulisati graničnom vrednošću (što se odigrava u prekidnoj proceduri *interrupt*). Uz sve to, naravno, na LCD-u očitaće se trenutna vrednost nivoa vode i granična vrednost zadata od strane korisnika. Za LCD definisani su digitalni pinovi 12, 11, 4, 5, 6, i 7.

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 4, 5, 6, 7);
#define SensorPin A0
#define downTaster 2
#define upTaster 3
#define zujalicaPin 13

volatile int granica = 0;
void setup(){
pinMode(downTaster,INPUT);
pinMode(upTaster,INPUT);
pinMode(zujalicaPin, OUTPUT);
attachInterrupt(digitalPinToInterrupt(downTaster), downISR, RISING);
attachInterrupt(digitalPinToInterrupt(upTaster), upISR, RISING);
Serial.begin(9600);
lcd.begin(16, 2);
}
void downISR(){
 if(granica>0){
  granica=granica-50;
 }}
void upISR(){
 if(granica>=0){
  granica=granica+50;
 }}
void loop(){
 int lvl = analogRead (SensorPin);
 Serial.print("Level vode: ");
 Serial.println(lvl);
 delay(1000);

 lcd.setCursor(0, 0);
 lcd.print("Level= ");
 lcd.print(lvl);
 lcd.setCursor(0, 1);
 lcd.print("Granica= ");
 lcd.print(granica);

 if(granica<=lvl){
  tone(zujalicaPin, 440);
  }
 else{
  noTone(zujalicaPin);
  }
}
```

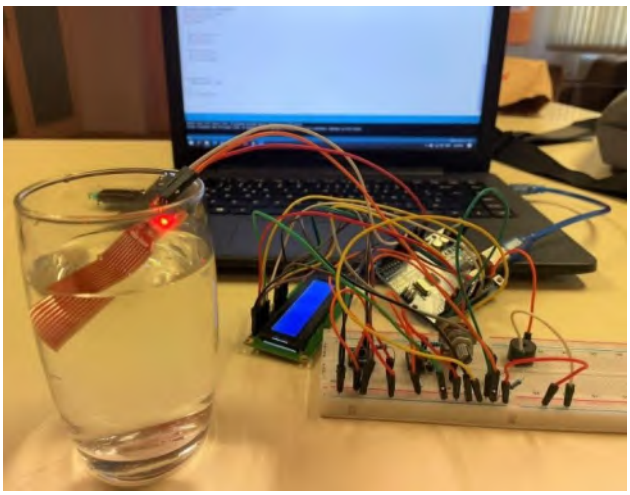
Slika 4. Kod programa.

IV. REZULTATI I ANALIZA

Na slici 5 je prikazan izgled finalnog projekta koji omogućava testiranje u laboratorijskim uslovim. Prikazana slika pokazuje rad LCD-a i senzora za merenje nivoa vode, tastera i zujalice.

Ovaj sistem je moguće implementirati i za korišćenje u realnim uslovima. Međutim, u edukativne svrhe testiranje sistema izvršeno je samo u laboratoriji.

Kao što je prikazano na slici 6, u čaši se nalazi nešto malo vode. Granična vrednost podešena je na 450. Trenutno izmeren nivo vode je 244 i on je manji u odnosu na zadatu graničnu vrednost. Samim tim, zujalica se neće oglasiti.



Slika 5. Izgled gotovog projekta i testiranje.



Slika 6. Testiranje gotovog projekta.

Sada se u čaši, kao što je prikazano na slici 7, nalazi više vode u odnosu na prethodni primer (ilustrovan na Slici

6). Graničnu vrednost koju smo podesili ostaje ista, a to je 450. Trenutno izmeren nivo vode je 636 i on je veći u odnosu na zadatu graničnu vrednost, pa će se zbog toga zujalica oglasiti.



Slika 7. Testiranje gotovog projekta.

Testiranje sistema sprovedeno je za više različitih kombinacija, pokazano je njegovo uspešno funkcionisanje, a slike 6 i 7 izabrane su samo kao primer demonstracije uspešnog rada.

V. ZAKLJUČAK

Osmišljeni uređaj je veoma pouzdan i može se doveti do stadijuma visokog nivoa preciznosti. Takođe, može se upotrebljavati i u drugačije svrhe koje imaju dodirnih tačaka sa merenjem nivoa vode.

Ovaj uređaj bi mogao da pomogne u tome da u svakom trenutku imamo uvid u količinu pijaće vode u okviru celog sveta što bi moglo i da umanji intenzivnu neštasicu iste.

ZAHVALNICA

Zahvaljujem se Milošu Marjanoviću na usmeravanju i sugestijama pri izradi ovog rada. Takođe, zahvaljujem se koleginici Milici Danković, studentu treće godine Elektronskog fakulteta u Nišu, za savete tokom fizičke realizacije ovog projekta. Konačno, zahvaljujem se rukovodstvu škole što mi je omogućilo realizaciju ovog projekta.

LITERATURA

[1] World Health Organization dostupno na https://www.who.int
[2] Statistika pijaće vode u svetu. dostupno na https://www.concernusa.org/story/countries-with-water-stress and-scarcity/
[3] Arduino Uno https://en.wikipedia.org/wiki/Arduino_Uno
[4] Water Sensor,opisan na https://lastminuteengineers.com/water-level-sensor-arduino-tutorial/
[5] TinkerCad, Dostupno na https://www.tinkercad.com/circuits.

Primena veštačke inteligencije za podešavanje kontrolera laboratorijske stanice za flaširanje

Emilija Ćojbašić

Sadržaj – U ovom radu je primenjena veštačka inteligencija za podešavanje parametara kontrolera laboratorijske stanice za simulaciju procesa flaširanja u procesnoj industriji. Opisane su komponente i način rada FESTO MPS PA laboratorijske stanice za flaširanje, koja predstavlja standard za vežbe iz domena procesnog upravljanja u evropskom visokoškolskom obrazovanju. Prikazana je eksperimentalna identifikacija prenosne funkcije stanice kao objekta upravljanja, a kao upravljački sistem je primenjen PID kontroler realizovan Siemens Simatic S7-300 PLC kontrolerom. Za podešavanje upravljanja, preciznije za određivanje pojačanja PID kontrolera, primenjena je veštačka inteligencija, odnosno metaheuristička optimizacija. Primenjeno je šest različitih metoda, i to genetski algoritmi (GA), simulirano žarenje (SA), optimizacija roja čestica (PSO), metod čopora sivih vukova (GWO), metod kolonije mrava (ACO) i algoritam slepih miševa (BA). Podešavanje parametara kontrolera sprovedeno je softverskim paketom Matlab. Na osnovu dobijenih optimalnih parametara kontrolera urađena je verifikacija rada stanice, a na kraju su izvedeni i odgovarajući zaključci.

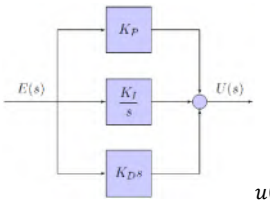
I. UVOD

Veštačka inteligencija (engl. Artificial intelligence, skraćeno AI) [1][2] je podoblast računarstva i predstavlja široko područje računarske nauke koje ima za globalni cilj da mašine funkcionišu i zaključuju podražavajući ljudski mozak. Drugim rečima cilj istraživanja veštačke inteligencije je razvijanje programa (softvera), koji će omogućiti računarima da se ponašaju na način koji bi se mogao okarakterisati inteligentnim.

Matematička optimizacija je izbor najboljeg elementa (s obzirom na neki kriterijum) iz neke grupe dostupnih alternativa. Metaheuristička optimizacija [1] je postupak višeg nivoa osmišljen da nađe, generiše ili odabere parcijalni algoritam pretraživanja koji može pružiti dovoljno dobro rešenje problema optimizacije, posebno sa nepotpunim ili nesavršenim informacijama ili ograničenim kapacitetom računanja. Svi moderni algoritmi optimizacije inspirisani prirodom i optimizacioni algoritmi sa stohastičkim komponentama danas se smatraju tehnikama veštačke inteligencije i nazivaju metaheuristikom.

Upravljanje procesa je jedna od najznačajnijih grana upravljanja kao discipline [3][4][5]. Od mnoštva različitih algoritama upravljanja, PID upravljački algoritam [3][4] se najčešće koristi kod ovakvih sistema. PID algoritam upravljanja ima nekoliko važnih osobina: obezbeđuje povratnu spregu; eliminiše statičku grešku integralnim dejstvom; anticipira buduće ponašanje diferencijalnim dejstvom. PID algoritmi upravljanja su dovoljan izbor u većini aplikacija, naročito kada dinamika objekta ili procesa nije previše zahtevna, a zahtevi za performansama su umereni. Prošli su razvoj kroz različite tehnologije, od pneumatskih i hidrauličkih do mikroprocesorskih, preko elektronskih cevi, tranzistora i integrisanih kola i danas u upravljanju procesima učestvuju sa preko 95 %.

Iako savremeni PID kontroleri sadrže napredne funkcije, podešavanje tri parametra kontrolera, proporcionalnog K_P, integralnog K_I i diferencijalnog pojačanja K_D, je netrivijalan problem za koji je razvijeno više teorijskih i eksperimentalnih metoda. Na slici 1 prikazani su paralelna realizacija PID kontrolera i njegov upravljački zakon.



$$u(t) = K_P e(t) + K_I \int_0^t e(\tau)d\tau + K_D \frac{de(t)}{dt}$$

Slika 1. Paralelna realizacija i upravljački zakon PID kontrolera.

Kada je reč o primeni veštačke inteligencije u razvoju sistema upravljanja, brz tempo razvoja veštačke inteligencije pruža neviđenu širinu mogućnosti za poboljšanje performansi, uključujući i potpunu novu oblast inteligentnog upravljanja koje uključuju veoma napredne metode. Važan segment primene veštačke inteligencije u oblasti upravljanja jeste mogućnost primene metaheurističke optimizacije za podešavanje kontrolera, odnosno optimizaciju upravljanja, bez obzira da li je reč o konvencionalnom ili inteligentnom upravljačkom sistemu.

Istraživanje u radu je sprovedeno sa ciljem da se primeni veštačka inteligencija za podešavanje kontrolera FESTO stanice za flaširanje, koja je deo FESTO MPS PA linije laboratorijskih stanica koje predstavljaju de fakto standard za obuku iz domena procesnog upravljanja i industrijske automatike. Primenjeno je nekoliko različitih algoritama metaheurističke optimizacije.

II. FESTO STANICA ZA FLAŠIRANJE

FESTO laboratorijska stanica za flaširanje predstavlja realno industrijsko okruženje [6][7]. Na ovoj stanici svi

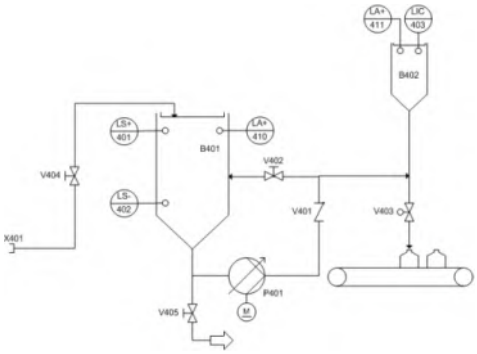
E. Ćojbašić je student četvrte godine studija Računarstva i informatike na Elektronskom fakultetu Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija i studentski član IEEE, E-mail: ecojbasic@elfak.rs.

aspekti potrebni za optimizaciju procesa punjenja mogu biti proučeni i praktično isprobani.

Stanica za flaširanje vrši punjenje bočica tečnošću. Uključivanjem pumpe se puni rezervoar, a zatim se tečnost iz rezervoara pretače u rezervoar za doziranje odakle se pune bočice. Bočice preko transportnih traka dolaze do pozicije za punjenje. Pneumatski separator razdvaja bočice, a bočice mogu biti punjene različitim količinama tečnosti. Kontroler reguliše nivo ounjenja dozirajućeg rezervoara na zadatu vrednost pomoću dinamički kontrolisane pumpe.

A. Blok šema FESTO stanice za flaširanje

Nivo tečnosti se u dozirajućem rezervoaru održava konstantnim u toku punjenja. Razni upravljački algoritmi kao što su P, PI ili PID regulatori mogu se primenjivati a kontrola procesa se takođe može vršiti i pomoću ručno podesivih ventila. Na slici 2 dat je šematski prikaz komponenti stanice sa oznakama i načinom povezivanja.



Slika 2. Šematski prikaz komponenti stanice.

Izvor tečnosti je X401, a otvaranjem ručnog ventila V404 tečnost se pušta u rezervoar B401 za skladištenje. Senzori LS+401 i LS-402 mere nivo tečnosti u rezervoaru. Senzor LA+410 vrši merenje tečnosti, ali i šalje poruku (alarm) upozorenja usled nedostatka ili prevelike količine tečnosti u rezervoaru. Ventil V405 otvara se ručno i služi za istakanje tečnosti iz rezervoara. Tečnost se iz rezervoara B401 u dozirajući rezervoar B402 prebacuje se pomoću pumpe P401 koja se napaja preko motora 4M1. Ventil V401 je nepovratni ventil koji dozvoljava protok tečnosti samo u smeru ka dozirajućem rezervoaru. Pomoću ventila V402 koji je ručni ventil može se prema potrebi vraćati tečnost iz dozirajućeg u rezervoar za skladištenje. Količina tečnosti u dozirajućem rezervoaru se meri akustičnim senzorom LIC403, dok senzor LA+411 služi i za slanje alarma upozorenja. Ventil V403 je merni ventil koji propušta zadatu količinu tečnosti u bočice [6][7].

B. Komponente FESTO laboratorijske stanice

Komponente stanice na slici 3 su: (1) Pokretna traka za transport bočica od magacina do mesta punjenja; (2) Rezervoar za skladištenje tečnosti kojom se pune bočice, a sa senzorima sa obe strane rezervoara koji mere nivo tečnosti; (3) Dozirajući rezervoar kapaciteta 3 litra, koji se napaja iz rezervoara za skladištenje tečnosti preko pumpe, a nivo tečnosti u rezervoaru se meri akustičnim senzorom; (4) Merni ventil, koji reguliše količinu tečnosti kod punjenja tako što na osnovu zadate vrednosti PLC otvara ventil i reguliše količinu tečnosti kojom će biti napunjena bočica. (5) Pneumatski razdvajač, koji osigurava da je na mestu punjenja uvek samo jedna bočica; (6) Reduktorski motor za pokretanje trake; (7) Pritisni ventil, meri pritisak tečnosti u rezervoaru a sistem se isključuje ako pritisak poraste iznad dozvoljene vrednosti; (8) Napajanje, pretvara napon sa 220 V na 24 V; (9) Akustični senzor, šalje signal kroz tečnost i meri vreme refleksije signala i konvertuje ga u naponski signal vrednosti od 0 do 10 V, mereći nivo tečnosti od 50-300 mm. (10) Panel osetljiv na dodir, preko koga se bira način rada stanice i upravlja stanicom.



Slika 3. FESTO laboratorijska stanica za flaširanje.

Upravljanje procesom punjenja se vrši preko PLC kontrolera za industrijsku primenu tipa Siemens S7-300.

C. Identifikacija prenosne funkcije stanice

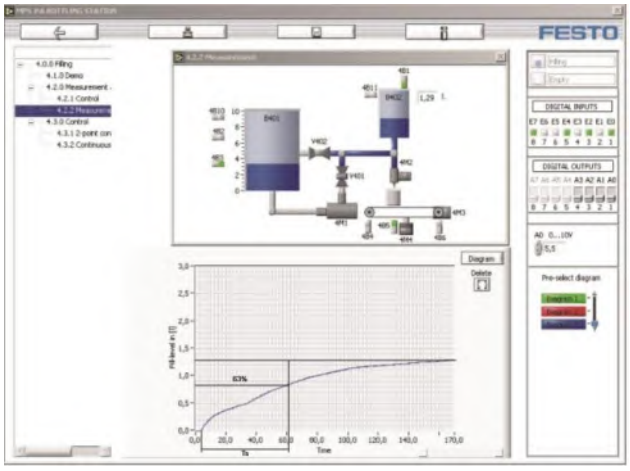
Prenosna funkcija stanice dobija se snimanjem eksperimentalnog odskočnog odziva sistema, iz koga se određuju red sistema i parametri prenosne funkcije. Eksperimentalna kriva punjenja rezervoara za doziranje se određuje u programu FluidLab-PA, a eksperiment je prikazan na slici 4.

Pomoću eksperimentalno snimljenog odziva određuje se prenosna funkcija stanice kao upravljanog objekta:

$$G(s) = \frac{0.867}{59.5s+1} \quad (1)$$

III. PODEŠAVANJE SISTEMA METAHEURISTIKOM

Klasična teorijska i eksperimentalna sinteza PID kontrolera razmatrane stanice daje veoma dobre performanse sistema. Kao alternativa, koja predstavlja univerzalnu proceduru nezavisnu od tipa kontrolera, sprovedena je optimizacija parametara kontrolera tehnikama metaheurističke optimizacije.



Slika 4. Eksperimentalno snimanje odziva stanice.

Funkcija cilja za minimiziranje greške praćenja definisana kao:

$$J = \sum_{i=1}^{n} |e_i| = \sum_{i=1}^{n} |r_i - y_i| \quad (2)$$

gde je *r* referentna vrednost, *y* je upravljani izlaz, *e* je greška u regulaciji i *n* je broj uzoraka. Cilj optimizacije je da se podese parametri kontrolera minimiziranjem funkcije cilja i na taj način ostvari poboljšanje performansi kontrolera smanjenjem greške praćenja.

Za optimizaciju kontrolera korišćene su najpre tri dobro poznate metaheurističke tehnike: genetski algoritmi (GA), simulirano žarenje (SA) i optimizacija roja čestica (PSO). Genetski algoritmi [8] su najpopularnija i najšire korišćena tehnika metaheurističke optimizacije, koja predstavlja model apstrakcije biološke prirodne selekcije, zasnovan na Darvinovoj teoriji evolucije. Optimizacija simuliranog žarenja je nastala korišćenjem analogije sa termičkom obradom metala [9][10], i poseduje sposobnost da izbegne zarobljavanje u lokalnim optimumima, za razliku od determinističkih tehnika optimizacije. Konačno, optimizacija roja čestica je tehnika razvijena na osnovu ponašanja rojeva, kao što je rojenje riba i ptica [11].

Pored tri navedene i često primenjivane optimizacione metode, za podešavanje kontrolera korišćene su i tri manje primenjivane tehnike, detaljnije prikazane u nastavku: metoda čopora sivih vukova (GWO), metoda kolonije mrava (ACO) i algoritam slepih miševa (BA).

A. Metod čopora sivih vukova (GWO)

Metod čopora sivih vukova [12] pripada tehnikama zasnovani na inteligenciji čopora. Sivi vukovi (Canis lupus) su predatori sa vrha lanca ishrane. Žive i love u čoporima i imaju veoma strogu društvenu hijerarhiju. Vođe čopora se zovu alfe. Alfa vuk nije uvek najjači već najbolji u vođenju i upravljanju grupom. Vukovi koji su podređeni alfama, pomažući im u donošenju odluka i aktivnostima, nazivaju se beta. Delta prati alfa i beta, dominira nad omegom i oni su izviđači, lovci, stražari, starešine i čuvari. Omega je vuk najnižeg ranga u čoporu.

Glavne faze grupnog lova sivog vuka su: praćenje, gonjenje i približavanje plenu; gonjenje, opkoljavanje i uznemiravanje plena; napad na plen. Jednačine (3) predstavljaju model zaokruživanja plena:

$$\vec{D} = \left|\vec{C} \cdot \overrightarrow{X_P}(t) - \vec{X}(t)\right|, \ \vec{X}(t+1) = \overrightarrow{X_P}(t) - \vec{A} \cdot \vec{D} \quad (3)$$

gde t označava trenutnu iteraciju, $\vec{A}$ i $\vec{C}$ su vektori koeficijenata, $\overrightarrow{X_p}$ je vektor pozicije plena, a $\vec{X}$ označava poziciju sivog vuka. Vektori $\vec{A}$ i $\vec{C}$ se računaju kao

$$\vec{A} = 2\,\vec{a} \cdot \vec{r_1} - \vec{a}, \ \vec{C} = 2 \cdot \vec{r_2}, \quad (4)$$

gde se komponente $\vec{a}$ smanjuju od 2 do 0 tokom iteracija, a $\vec{r_1}$, $\vec{r_2}$ su slučajni vektori između 0 i 1.

Lov (optimizacija) se završava napadom na plen, a da bi se modeliralo zaokruživanje plena vrednost $\vec{A}$ je slučajna vrednost u [*-2a,2a*], gde se *a* smanjuje od 2 do 0 u toku iteracija. Slika 5 pokazuje slučajeve $|A| < 1$ gde se agent pretraživanja približava plenu i $|A| > 1$ kada se agent pretraživanja udaljava od plena u potrazi za boljim plenom.



Slika 5. Napad na plen i traženje plena.

B. Metod kolonije mrava (ACO)

Mravi su jedna od najorganizovanijih živih bića a njihovi fascinantni mehanizmi života pokrenuli su stvaranje metod optimizacije kolonije mrava (ACO) [13]. Mravi talože feromone dok se kreću, a drugi mravi prate te feromone. Kada stignu do raskrsnice, prate stazu sa više feromona. ACO je matematička interpretacija ovog prirodnog fenomena. Nakon inicijalizacije postoje tri faze: za svaku iteraciju mravi konstruišu rešenja, koja se poboljšavaju kroz lokalnu pretragu, a feromon se ažurira:

$$\tau_{ij} \leftarrow (1-\rho) \cdot \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \quad (5)$$

gde τ_{ij} predstavlja feromone koji spajaju ciljeve *i* i *j*, ρ je brzina isparavanja, *m* je broj mrava, a $\Delta\tau_{ij}^{k}$ je količina feromona položena na putanju (*i*, *j*) od strane mrava *k:*

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{L_k} & \text{ako je mrav k koristio rutu } (i,j) \text{ na svom putu,} \\ 0 & \text{u suprotnom,} \end{cases} \quad (6)$$

gde je *Q* konstanta a L_k opisuju dužinu rute mrava *k*. Mravi biraju sledeći cilj stohastičkim mehanizmom.

C. Algoritam slepih miševa (BA)

Algoritam slepih miševa [14] je inspirisan eholokacijom slepih miševa. Slepi miševi koriste ultrazvučne talase kako bi pronašli svoj put i lovili plen. Pri je brzini zvuka u vazduhu 340 m/s, talasna dužina λ ultrazvučnog praska je u opsegu od 2 mm do 14 mm.

Algoritam pretpostavlja neke aproksimacije: slepi miševi koriste eholokaciju da osete udaljenost i znaju razliku između hrane i prepreka; Tragajući za plenom lete nasumično brzinom υ_i na poziciji sa fiksnom frekvencijom f_{min}, promenljivom talasnom dužinom i glasnoćom A_0; U zavisnosti od blizine mete, oni mogu da podese frekvenciju emitovanih impulsa i brzinu emitovanja impulsa $r \in [0, 1]$.

Pravila kako se brzine υ_i i pozicije x_i virtuelnih slepih miševa menjaju u d-dimenzionom prostoru su:

$$f_i = f_{min} + (f_{max} - f_{min}), \quad v_i^t = v_i^{t-1} + (x_i^t - x_*)f_i$$
$$x_i^t = x_i^{t-1} + v_i^t, \qquad (7)$$

gde je $\beta \in [0, 1]$ slučajan vektor. Lokacija x_x je globalno najbolje rešenje.

Kada je rešenje selektovano između trenutnih najboljih rešenja, kreira se novo rešenje za svakog slepog miša.

IV. Izbor parametara kontrolera

U svakom od razmatranih metaheurističkih algoritama optimizacije, performanse zavise od vrednosti njegovih parametara, pa su parametri pažljivo odabrani izvođenjem brojnih eksperimenata. U odnosu na parametre PID kontrolera određene eksperimentalno Zigler-Nikolsovom procedurom [7], koji su smatrani referentnim za poređenje, dozvoljena je umerena varijacija pojačanja kontrolera uz približno dvostruko uvećanje/umanjenje.

Vrednosti pojačanja kontrolera i vrednosti odgovarajuće funkcije cilja za minimizaciju greške praćenja dobijene različitim metodama sumirane su u tabeli I. Podešavanje PID kontrolera veštačkom inteligencijom sprovedeno je u softverskom paketu Matlab verzija R2022a uz korišćenje njegovih dodataka Global Optimization Toolbox i Control System Toolbox.

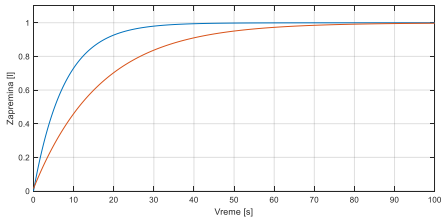
Tabela I
Parametri PID kontrolera dobijeni optimizacijom

Metod	K_P	K_I	K_D	**Funkcija cilja (manje je bolje)**
Eksperimentalno [6]	4.2	0,07	1	165,21
GA	9	0,15	0,1	76,64
SA	9	0,15	0,79	76,73
PSO	9	3	3	86,71
BA	8.7	0,14	0,56	79,84
GWO	7.2	0,12	0,1	95,39
ACO	8.4	0,14	0,1	82,03

Na slici 6 prikazan je odskočni odziv sistema za referentni eksperimentalno podešeni PID kontroler i najbolji odziv dobijen optimizacijom.

V. Zaključak

Kao referenca za poređenje rezultata optimizacije podešavanja parametara PID kontrolera, uzeti su parametri PID kontolera koji su za stanicu za flaširanje eksperimentalno podešeni i navedeni u dokumentaciji kompanije FESTO za laboratorijsku stanicu [7].



Slika 6. Odskočni odziv sistema za referentni eksperimentalno podešeni PID kontroler i najbolji odziv dobijen optimizacijom.

Pored poboljšanih performansi u poređenju sa konvencionalno podešenim PID kontrolerom, najznačajniji zaključak istraživanja je da je metaheuristička optimizacija smislen i univerzalan koncept za optimizaciju kontrolera, koji omogućava implementaciju komplikovanijih (nelinearnih) regulatora i njihovu optimizaciju.

Zahvalnica

Autorka iskazuje svoju iskrenu zahvalnost mentoru prof. dr Saši S. Nikoliću sa Elektronskog fakulteta u Nišu za nadzor nad radom, dragocene smernice i podršku.

Literatura

[1] P. Janičić, M. Nikolić, *Veštačka inteligencija*, Matematički fakultet Univerziteta u Beogradu, 2021.
[2] M. Nikolić, A. Zečević, *Mašinsko učenje*, Matematički fakultet Univerziteta u Beogradu, 2019.
[3] R. Dorf, R. Bishop, *Modern Control Systems*, Pearson, 13th edition, 2017.
[4] Milosavljević Č., *Osnovi automatike I deo*, Elektronski fakultet u Nišu, 2002.
[5] Nikolić S., *Uvod u automatiku – skripta sa predavanja*, Elektronski fakultet u Nišu, 2021.
[6] FESTO, *FESTO Didactic Catalogue – Technology for education and science, the current range of Festo didactic products 2014*, 2014.
[7] FESTO, *Festo MPS PA Workbook and Solutions*, 2007.
[8] Goldberg, D. E., *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley, USA, 1989.
[9] P.J. van Laarhoven, E.H. Aarts, *Simulated Annealing: Theory and Applications*, Springer, 1987.
[10] Soni Y. K., Bhatt R., "Simulated Annealing optimized PID Controller design using ISE, IAE, IATE and MSE error criteria", *IJARCET*, Vol. 2, No. 7, pp. 2337-2340, 2013.
[11] García-Gonzalo, E., Fernández-Martínez, J. L., "A Brief Historical Review of Particle Swarm Optimization (PSO)", *Journal of Bioinformatics and Intelligent Control*, Vol. 1, No. 1 (2012), pp. 3-16.
[12] Mirjalili S., Mirjalili S.M., Andrew L., "Grey Wolf Optimizer", *Advances in Eng. Software* (2014), pp. 46-61.
[13] Dorigo M., Gambardella, L. M., "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem", *IEEE Transactions on Evolutionary Computation*, 1997, Vol. 1, No. 1, pp. 53-66.
[14] Xin-She Yang, "A New Metaheuristic Bat-Inspired Algorithm, Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)", *SCI 284*, pp. 65-74.

Projektovanje i dizajn štampane ploče na primeru LED bedža

Jovana Janjatović, Kosta Jovanović, Mihajlo Ninkov

Sadržaj − U ovom radu se opisuje postupak izrade autentične štampane ploče (PCB) koja bi vršila funkciju svetlećeg LED bedža. Uzimajući u obzir da je ovaj rad rađen u Istraživačkoj Stanici Petnica kao seminarski projekat troje polaznika seminara elektronike, motiv ovog LED bedža je logo IS Petnica. U cilju interakcije korisnika sa štampanom pločom (bedžom), uvedena je mogućnost kontrolisanja LE dioda pomoću Arduino mikrokontrolera. U nastavku rada, detaljno je opisana metodologija izrade ovog projekta, kao i korišćene komponente, tehnike i izrada završnih dokumenata.

I. Uvod

PCB (Printed Circuit Board), odnosno štampana ploča, predstavlja pasivnu elektronsku komponentu koja omogućava ostvarivanje veze među ostalim elektronskim komponentama. Najbitnija podela štampanih ploča je na jednoslojne i višeslojne.

Kod jednoslojnih štampanih ploča, komponente su najčešće naslagane sa jedne, a trake (provodnici) sa druge, dok se između ta dva sloja nalazi sloj izolacije. Međutim, nije nužno da jednoslojna štampana ploča sadrži provodnički element sa obe strane. Kod nekih specijalnih primena, štampana ploča sadrži provodnik samo sa jedne strane, dok se na drugoj nalazi izolatorska ploča.

Višeslojne štampane ploče se koriste kod složenih elektronskih kola. Sastoje se od naizmenično raspoređenih slojeva provodnog i izolacionog materijala spojenih zajedno. Provodni slojevi su povezani preko metaliziranih otvora koji se koriste kako za montiranje, tako i za električno povezivanje komponenata.

U ovom radu opisan je postupak dizajniranja štampane ploče koristeći Proteuss 8 Professional program za dizajn štampane ploče i LCSC bazu komponenti [1].

Ovaj projekat rađen je u okviru seminara elektronike u Istraživačkoj Stanici Petnica, pa je samim tim projektovana štampana ploča sa motivima logoa IS Petnica (slika 1).

Iscrtavanje šeme i dizajn štampane ploče je rađen u programu Proteus 8 Professional.

J. Janjatović je učenica treće godine Šabačke gimnazije, Masarikova 13, 15000 Šabac, Srbija, E-mail: jovana.janjatovic.05@gmail.com.

K. Jovanović je učenik četvrte godine V beogradske gimnazije, Ilije Garašanina 24, 11000 Beograd, Srbija, E-mail: kosta.jovanovic@gmail.com.

M. Ninkov je učenik treće godine gimnazije „Jovan Jovanović Zmaj“, Zlatne grede 4, 21000 Novi Sad, Srbija, E-mail: mih.nin05@gmail.com.

Slika 1. Logo IS Petnica.

II. Šema električnog kola

Prvi korak u izradi projekta je bila električna šema. (Slika 2) [2,3].

Za ovu šemu je korišćeno:

- 22 LE SMD diode u 5 različitih boja [5]
- SMD diode [6]
- SMD otpornici različitih vrednosti otpornosti [4]
- NPN bipolarnih SMD tranzistora [7]
- Držač za dugmastu bateriju (3V)
- Konektori

Radi postizanja najveće moguće kompaktnosti celog sistema, korišćene su tzv. SMD komponente (Surface-Mount Device), koje se leme na površinu štampane ploče i znatno su manjih dimenzija u odnosu na komponente sa izvodima TH (Through Hole).

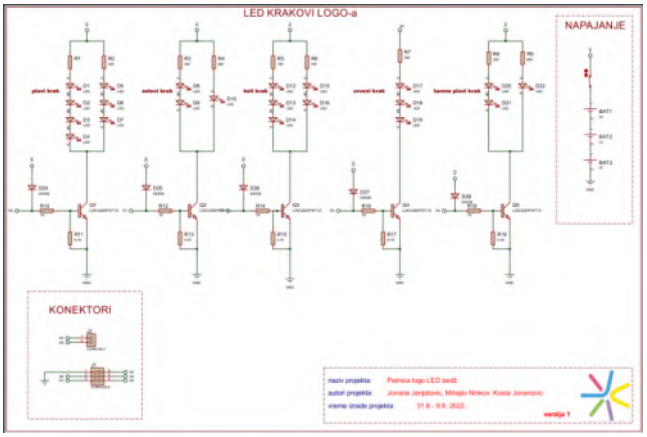
Prvi problem koji se pojavljuje jeste to što u bazi komponenata kompanije JLCPCB, koja bi trebalo da bude zadužena za izradu ove štampane ploče, ne postoji LE dioda koja emituje ljubičastu svetlost. Rešenje ovog problema jeste da se umesto ljubičaste koristi plava dioda koje emituju svetlost veće talasne dužine od plave koja se već koristi za dva kraka.

Kako bi se izbeglo nošenje glomaznih baterija od 9 V i više, sa zadnje strane ploče je planirano postavljanje 3 držača dugmaste baterije od 3 V, zato što predstavljaju dosta manju i laganiju alternativu ideji početnog napajanja.

Uzimajući u obzir da LE diode različitih boja zahtevaju različit radni napon, a poštujući i I Kirhofov zakon, bilo je potrebno razdeliti LE diode i ostale elektronske komponente u više grana kako bi se ideja sa postavljanjem baterije od 9 V koja napaja ceo sistem pokazala izvodljivom.

Raspodelom neophodnih komponenti u odgovarajući redosled na šematskom prikazu, pojavila se potreba za unapređenjem sistema, u vidu mogućnosti za kontrolu svake boje logoa nezavisno koristeći Arduino mikrokontroler. Dodavanjem NPN bipolarnih tranzistora na svaku granu električne šeme i prespojnika koji bi vršili funkciju provodnika u slučajevima kada mikrokontroler

nije priključen, korisnik je dobio mogućnost da po sopstvenoj želji isprojektuje šablon po kom će LE diode funkcionisati.



Slika 2. Šema električnog kola.

III. DIZAJN ŠTAMPANE PLOČE

Nakon kompletirane šeme sa svim željenim komponentama, neophodno je sve te komponente rasporediti u određeni raspored na štampanoj ploči. Kako bi LE diode raspodelili najpreciznije moguće, Petnički logo je prebačen u monohromatsku sliku, a zatim je sačuvan kao .bitmap dokument, iz razlog zašto je to jedini format fotografije koji Proteus dozvoljava kao uvoz. Kada su sve LE diode bile na svojim mestima, usledila je raspodela ostalih komponenti na štampanu ploču, tako da ploča zadrži kompaktan i funkcionalan izgled, odnosno da svaka komponenta ima mogućnost da se poveže u kolo, tako da ne remeti rad kako LE dioda tako i ostalih komponenti.

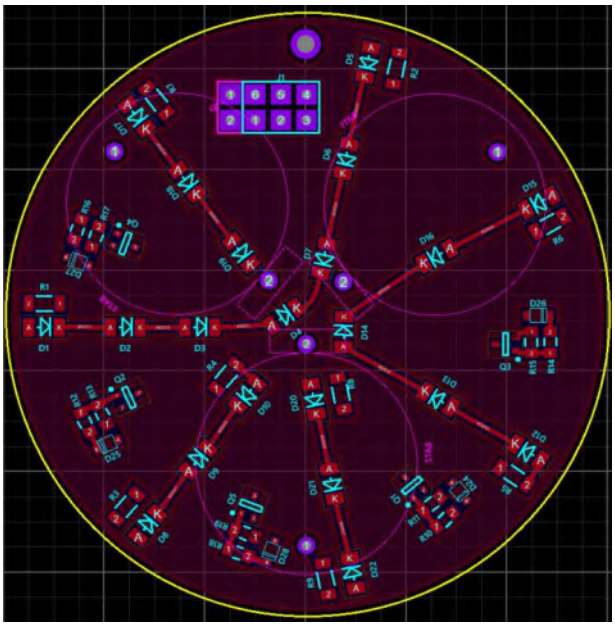
Ideja je da se na jednoj strani nalaze samo LE diode, dok ostale treba da budu smeštene sa zadnje strane i da budu povezane viama sa prednjom stranom, da ne bi remetile rad LE dioda.

Daljim istraživanjem, uviđeno je da je znatno jeftinije i brže ukoliko sve komponente stoje na jednoj strani ploče. Iz tog razloga, ploča je dizajnirana tako da ispunjava prethodno navedene potrebe, ali i da sadrži komponente isključivo na jednoj strani.

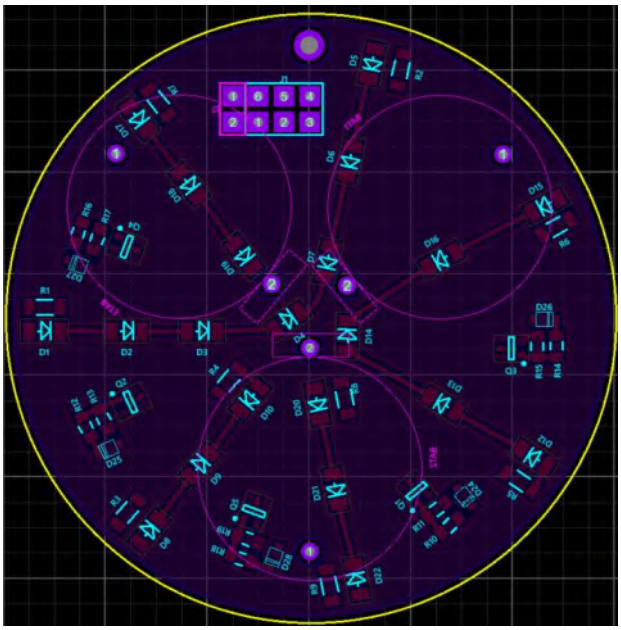
Povezivajući komponente putevima (trace), pojavio se naredni problem, tačnije, nije moguće povezati dve komponente tako da put između njih na bilo kom svom delu skreće pod pravim uglom zato što u tom slučaju ima ulogu odašiljača signala.

Budući da će sve TH (ili) komponente biti lemljene u Srbiji, nije postojao 3D model i otisak (eng. footprint) na ploči u bazi komponenti programa u kom je projekat rađen, osim konektora, pa je bilo neophodno projektovati 3D model (u programu za 3D modelovanje – Fusion 360) i otisak na ploči (u Proteus 8 Professional) [1,8].

Jedna od stvari koje su znatno olakšale postpuak izrade jeste postavljanje površine obe strane ploče tako da vrši ulogu uzemljenja. Finalni izgled štampane ploče prikazan je na slikama 3 i 4.



Slika 3. Prednja strana štampane ploče.



Slika 4. Zadnja strana štampane ploče.

IV. DOKUMENTACIJA

Po završetku procesa dizajna štampane ploče, neophodne je precizirati postupak izrade kompaniji kojoj se šalje ploča na izradu. U ovom slučaju, štampana ploča će biti izrađena od strane kompanije JLCPCB. Kako bi se precizno definisao postupak izrade neophodno je sastaviti tri ključna dokumenta - BOM, P&P i Gerber fajl.

Izrada proizvodne dokumentacije podrazumeva generisanje i proveru Gerber fajlova. Ovi fajlovi sadrže podatke, kao što su koordinate za bušenje rupa sa definisanim prečnicima, referentne tačke... GERBER je standardni format za uređaje u proizvodnji štampanih ploča.

BOM fajl (Bill of Materials) predstavlja spisak svih komponenti potrebnih za proizvodnju štampane ploče. Postoje različiti tipovi BOM fajlova, ali svaki od njih mora sadržati:

- šifru proizvoda - karakteristična je za firmu koja je dizajnirala štampanu ploču
- karakteristike komponenti: na primer: otpornost, kapacitivnost...
- oznaku: na samoj štampanoj ploči, svakoj komponenti dato je ime koje je sastavljeno od slova (koje je uglavnom karakteristično za vrstu komponente, na primer: otpornik - R, kondenzator - C...) i broja (koji označava koja je po redu to komponenta određene vrste)

Izuzetno je važno redovno ažurirati BOM file tokom dizajniranja jer je on ključni aspekt i dizajna štampane ploče i njene montaže. Kad god se doda nova komponenta tokom procesa dizajniranja, treba se uveriti da je BOM ažuriran sa ispravnom šifrom delova, opisom i količine komponenata.

P&P fajl (Pick & Place) je lista komponenti i njihovih koordinata. Ova vrsta fajla je neophodna za automatsko montiranje, jer daje instrukcije mašini, kao što su pozicija i orijentacija željene komponente. Svaki PnP file mora sadržati:

- oznaku: o kojoj komponenti je reč (npr: C21, L15...).
- preporučene jedinice: uglavnom je po metarskom sistemu to milimetar (mm).
- Layer - Top / Bottom, odnosno na kojoj strani ploče se nalazi komponenta: gornjoj/donjoj.
- Rotaciju: rotaciju date komponente izražene u stepenima. Vrednost je pozitivna za rotaciju u smeru kretanja kazaljke na satu.

V. ZAKLJUČAK

Nakon prolaska kroz kompletan proces izrade štampane ploče, zaključak je da je uz malo predznanje itekako moguće dizajnirati štampanu ploču sa proizvoljnom namenom poput ukrasa, funkcionalnog upravljača za mašine itd...

Ovaj projekat ima dosta mogućnosti za dalji napredak. Planovi za buduće modifikacije ploče su:

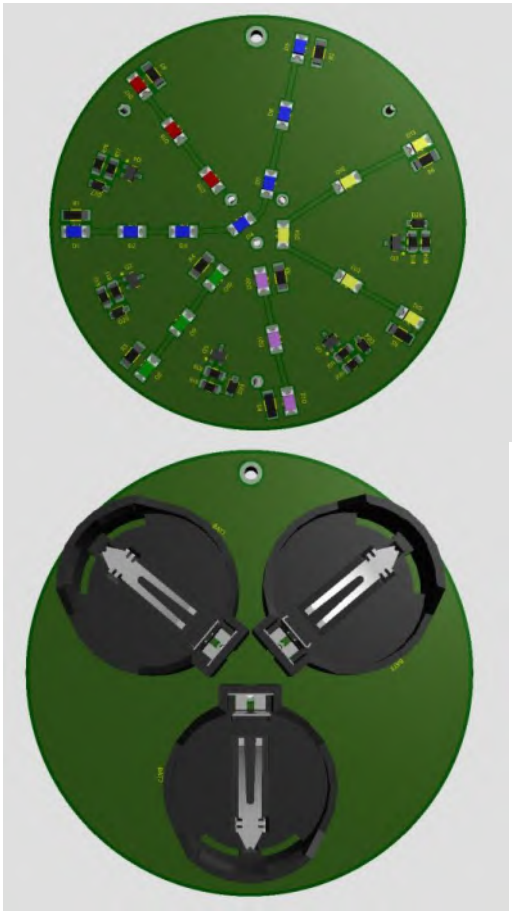
-Postavljanje svih komonenti na jednu stranu kako bi smanjili cenu izrade štampane ploče

-Dodavanje novih LE dioda radi "gušće" slike logoa

-Postavljanje štampane ploče u plastično kućište, koje će pored zaštitne svrhe, sadržati mlečnu plastiku ili klirit koji će da raširi svetlost radi jasnije slike

-Zamena 3 baterije od 3 V za jednu bateriju od 3 V koja se može pojačati DC-DC konverterom.

Sve detaljnije informacije se nalaze na GitHubu: https://github.com/jovanajanjatovic/LEDBadgePetnica.git.



Slika 5. Finalni izgled štampane ploče.

LITERATURA

[1] Proteus 8 Professional. Dostupno na: https://www.labcenter.com/.
[2] B. Antić “Uvod u projektovanje štampanih ploča” Mart 2009.
[3] D. Danković “Projektovanje štampanih ploča” Katedra za mikroelektroniku, Elektronski fakultet, Univerzitet u Nišu, 2017.
[4] Tehničke informacije korišćenih otpornika: https://datasheet.lcsc.com/lcsc/2206010130_UNI-ROYAL-Uniroyal-Elec-0603WAF1001T5E_C21190.pdf
[5] Tehničke informacije koričćenih LE dioda: https://datasheet.lcsc.com/lcsc/2009091206_HONGLITRONIC-Hongli-Zhihui--HONGLITRONIC--HL-PC-3216U97GC-L_C499460.pdf
[6] Tehničke informacije korišćenih dioda: https://datasheet.lcsc.com/lcsc/1810101110_Jiangsu-Changjing-Electronics-Technology-Co---Ltd--1N4148WS_C2128.pdf
[7] Tehniče informacije korišćenih kondenzatora: https://datasheet.lcsc.com/lcsc/1810051107_LRC-L2SC4083PWT1G_C12064.pdf
[8] Autodesk Fusion 360. Dostupno na: https://www.autodesk.com/products/fusion-360/personal

Razvoj decentralizovanih veb-aplikacija

Andrej Madanović

Sadržaj – U ovom radu je opisana realizacija decentralizovane komunikacije na Vebu i potencijal u daljem razvitku i novim pristupima rešavanja postojećih problema u uspostavljanju *peer-to-peer* konekcija u okviru veb-pregledača.

I. UVOD

Internet je osmišljen kao globalna mreža svih računara i tako i danas pričamo, ali se ne čini tako. Komunikacija se najčešće obavlja po klijent–server arhitekturi. Zbog toga, ukoliko je potrebno publikovati veb-aplikacija, neophodno je posedovati veb-server. Sa porastom broja korisnika aplikacije potreban je veb-server većih performansi, ili pak više takvih servera. Evidentno, skaliranje u klijent–server arhitekturi može predstavljati problem i danas na akcentu sve više dobijaju *serverless* i *edge computing*.

Ideja *serverless*-a je da se ne mora brinuti o sopstvenoj infrastrukturi i skaliranju aplikacije, već bi *Cloud* platforma automatski upravljala resursima dostupni aplikaciji, i to bez dodatnih podešavanja servera i slično. Ideja *edge*-a ide korak dalje, a to je da se resursi aplikacije nalaže što bliže korisnicima. Umesto da se resursi aplikacije nalaze u jednom ili par data centrima, nalazili bi se svuda u svetu i na taj način dodatno poboljšale performanse i efikasnost pristupa aplikaciji za sve korisnike širom sveta. Upravo isti cilj ima i decentralizovana komunikacija, samo što umesto servera, svaki klijent bi istovremeno imao ulogu i klijenta i servera.

Decentralizacija nije ništa novo, već ima visok stepen iskorišćenosti zahvaljujući *Torrent* kulturi. U okviru veb-pregledača je takođe moguće uspostaviti decentralizovanu, *peer-to-peer*, komunikaciju među klijentima. Međutim, češće se koristi kao pogodan način transporta audio i video podataka i centralizuje se upotrebom servera koji zatim umrežava klijente.

II. KOMUNIKACIJA U REALNOM VREMENU

Kako bi korisnik dobio neku informaciju u okviru veb-aplikacije, potrebno je da pošalje zahtev serveru, a zauzvrat će dobiti neki odgovor. Za komunikaciju u realnom vremenu to nije efikasno, jer bi se potpuno nepotrebno slali zahtevi serveru. Postoje nekoliko tehnika za zaobilaženje tog problema.

Umesto periodičnog slawa zahteva serveru radi probere da li postoji nešto novo za korisnika, može se poslati zahtev, a zatim čekati na otvorenoj konekciji sve dok server ne odgovori o novoj poruci, a zatim ponavljati proces kada se primi odgovor. Ova tehnika se zove *long polling* [1], lako se implementira i podržana je u gotovo svim veb-pregledačima i veb-serverima putem *HTTP* protokola. Ukoliko je potrebno slati informacije nazad serveru ponovo se nailazi na problem efikasnosti u komunikaciji, jer bi se puno vremena utrošilo pri *HTTP handshake*-y svakog zahteva.

Kako bi se rešio iznad napomenut problem može se upotrebiti naredni metod – *WebSocket* [2] protokol omogućava da se uspostavi bi-direkciona komunikaciji između klijenta i servera. Dok je *WebSocket* veza otvorena ka serveru, može se u bilo kom trenutku ista veza koristiti više puta za slanje podataka serveru, a takođe se i istom konekcijom mogu primati podaci. Podaci koji se transportuju su isključivo podaci aplikacije, i nije potrebno specifikovati dodatne podatke koji opisuju taj transport. Veoma je brz i pozudan, danas se koristi u gotovo svakoj aplikaciji za razmenu poruka.

Nekada je ipak potrebno razmenjivati ogromnme količine podataka između korisnika, kao npr. pri izvođenju audio-video razgovora. U tom slučaju, potrebna je što bolja konekcija među korisnicima za što brži transport podataka i što manjim odzivom, odnosno kašnjenjem. Tada je najpovoljnije realizovati *peer-to-peer* komunikaciju. To su upravo i činile ranije desktop aplikacije, kao što je, na primer, *Skype*. Zbog promene u upotrebi Veba, sada i kao platforme aplikacija, potrebno je imati ovakve vidove komunikacije i u okviru veb-pregledača.

III. *WEBRTC*

WebRTC [3] (*Web Real-Time Communications*) je protokol koji ima mnogo veću propusnost te je pogodan i često se koristi za prenos audio i video strimova u realnom vremenu. Dostupan je na svim platformama – u okviru veb-pregledača kao *Web API* i kao biblioteka za mobilne i desktop aplikacije. Tu ogromnu moć *WebRTC* poseduje zbog svoje arhitekture, i napravljen je da bude decentralizovan. Umesto potrebe za serverom kako bi prenosio podatke između klijenata, oni se međusobno umrežuju, *peer-to-peer*. Ovakav vid je poznat kao umrežen (*meshed*) ili decentralizovan (*dWebRTC*).

Nažalost, decentralizovan *WebRTC* kakav je originalno zamišljen se uglavnom centralizuje. Brzine interneta u velikom broju domaćinstava se nisu promenile u zadnjoj deceniji zbog tehničkih (ne)mogućnosti, kao i (ne)isplativosti paketa većih brzina. Takođe, brzina preuzimanja se uzima kao najvažnija karakteristika, je se

A. Madanović je student Elektrotehnike i računarstva na Elektronskom fakultetu, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18106 Niš, Srbija, E-mail: akymaky@elfak.rs.

većina operacija koje svakodnevni internet korisnik zahteva svode na preuzimanje, dok se brzina otpremanja često zanemaruje kao nebitna karakteristika, jer kada korisnik šalje podatke serveru, oni obično nisu velikih veličina.

Kada korisnici u okviru *WebRTC* komunikacije razmenjuju podatke, pošiljlac podataka (uglavnom) mora poslati taj isti podatak svih ostalim korisnicima. Ukoliko je taj podatak video strim, koji zahteva veliku brzinu otpremanja, može doći do blokade. Otpremanje uživo video strima sam po sebi je veoma spor proces zbog brzine otpremanja. Sa rastom broja korisnika koji međusobno komuniciraju, veći je problem skaliranja *WebRTC* komunikacije. Poznato je da *WebRTC* pri većem broju korisnika nije moguće lako skalirati, tako da se mora nešto učiniti za takve veće grupe korisnika. Nekako je potrebno ograničiti da svakih od korisnika samo jednom otpremaju svoje video strimove, a zatim nekako ostali korisnici da prime kopije tih strimova.

A. Strategije boljeg skaliranja WebRTC-a za potrebe video konferencijskih aplikacija

SFU (*Selective Forwarding Unit*) je najčešće korišćen pristup smanjenja broja kopija strimova koje je svakih od korisnika potrebno da otpremi mrežusobno drugim korisnicima. „Glup" je softver, te ne zahteva puno resursa za svoju operativnost. Komunikacija između korisnika se odvija preko *SFU*-a, svaki učesnik otpremljuje svoje strimove samo jedanput, a zatim nastavlja da preuzima pojedinačne strimove ostalih korisnika. Međutim i ovaj pristup ima ograničenja. Kako je potrebno pojedinačno preuzimanje svakog audio i video strima, dolazi do otvaranja ogromnog broja konekcija koje mogu dovesti do preopterećenja mrežne kartice uređaja korisnika ili do preopterećenja rutera. Dodatno je potrebno obraditi i reprodukovati svakih od preuzetih strimova, što u zavisnosti od grafičke sposobnosti korisničkog uređaja može dovesti do degradiranih performansi. *SFU* se obično koristi do oko najviše 50 korisnika.

MCU (*Multipoint Conferencing Unit*) ili *Mixer* se koristi za masovan broj ljudi, kada je potrebno da svi učesnici istovremeno šalju video strimove. Korisnici šalju svoje strimove mikseru, a on će ih objediniti u jedan video strim koji se prosleđuje svim učesnicima poziva. Veoma je zahtevan za operaciju, jer je na ophodio da oba serverski mašina bude visokih performansi, kao i da je dostupan grafički procesor (*GPU*). Još jedno veliko ograničenje je da je sam prikaz video strimova fiksan za sve učesnike, kako je to objedinjeno u jedan video strim.

B. Druge upotrebe i izazovi

Pored audio i video podataka, *WebRTC* poseduje *Data Channel* koji je moguće koristiti za prijenos bilo kojih podataka. Ukoliko bi prenosili tekstualne podatke, npr., ne bi došlo do problema ograničenja brzine interneta pa bi se mogao koristiti potpuno decentralizovan *WebRTC*.

Ipak, *WebRTC* tehnički nije moguće potpuno decentralizovati. Naime, proces uspostavljanja konekcije između korisnika je centralizovan zbog mrežnih barijera iza kojih se nalaze internet korisnici.

Zbog nedostataka IPv4 adresa nije moguće dodeliti svakom povezanom uređaju svoju IPv4 adresu, već se mora, na neki način, istovremeno deliti jedna adresa sa velikim brojem uređaja. Za tu svrhu se koristi *NAT* (*Network Address Translation*). Osim *NAT*-a na ruteru samih korisnika interneta postoji i *Carrier-Grade NAT* (*CGNAT*). Kako broj dostupnih IPv4 adresa nije dovoljan čak ni da pokrije sam broj pretplatnika internet usluga, internet provajderi taj problem prevazilaze tako što više korisnika mogu koristiti istu adresu primenom *CGNAT*-a. Zato se upotrebljavaju tehnike *hole punching*-a kako bi uopšte bilo moguće uspostaviti direktnu konekciju sa drugim korisnikom koji se takođe nalazi iza *NAT*-a.

WebRTC utilizuje *hole punching* putem *STUN* (*Session Traversal Utilities for NAT*) servera. On pomaže da samim klijentima nagovesti na koje sve načine i preko kojih *IP* adresa, im se mogu drugi korisnici umrežiti. Istovremeno iskorišćava nedostatke pojedinih implementacija tipova *NAT*-a za realizaciju *hole punching*-a, npr. kada bi veb-pregledač kontaktirao *STUN* server ruter bi privremeno otvorio port preko kojeg će omogućiti komunikaciju ka tom serveru i sačuvati tu informaciju u *NAT* tabeli, zatim će neki drugi korisnik moći da upotrebi isti taj port da komunicira nazad, pri čemu jedinu validaciju u tome kome tu konekciju treba proslediti u okviru lokalne mreže bi bio port putem kojeg je ruter primio tu konekciju.

Još jedan problem je kada korisnici saznaju koje su njihove *IP* adrese, tu informaciju na neki način je potrebno podeliti sa drugim korisnicima i to se naziva signalizacija. Taj proces se obično realizuje preko centralnog servera upotrebom *WebSocket*-a za što bržu konekciju i kontinualno signaliziranje u slučaju da klijent pridobije novu *IP* adresu ako dođe do promene mreže, npr. prelazak za *Wi-Fi* na mobilni internet.

C. Potencijalna rešenja

U *Torrent* svetu postoji koncept *Tracker*-a, čiji je cilj upravo način razmene informacija umrežavanja korisnika. Isto to se može primeniti i u okviru veb-pregledača – umesto da se samostalno razvije i održava mehanizam za signalizaciju u okviru aplikacije, moglo bi se koristiti više javnih distribuiranih servera za tu potrebu. Pošto *WebRTC* uspostavlja *peer-to-peer* konekcije, kao i *BitTorrent*, razvijen je *WebTorrent* – implementacija *BitTorrent*-a za veb-pregledače. Postoje već mnogi javni *Tracker*-i za *WebTorrent* kojima je moguće pristupiti *WebSocket*-om i koristiti ih za proces signalizacije. Može se upotrebiti biblioteku za lakšu realizaciju toga, poput *Trystero* [4] koja apstrahuje signalizaciju putem *WebTorrent Tracker*, *IPFS* ili *Firebase* strategija i pojednostavljuje rad sa *WebRTC Data Channel*-ima.

IV. *OFFLINE FIRST*

Potrebno je na neki način uskladištiti podatke korisnika, što je danas uglavnom i očekivano na Vebu kada korisnik pristupi nekoj veb-aplikaciji. Ukoliko se ne koristi centralni server za skladištenje podataka, onda je potrebno te podatke skladištiti lokalno na uređaju korisnika. Ovaj pristup se zove *offline first* [5], i ima svojih prednosti i mana [6]. Prednost je da su podaci odmah lokalno dostupni, brzo im se može pristupati i momentalno modifikovati bez potrebe da se čeka da server obradi zahtev i prosledi nazad odgovor.

Velike baze podataka mogu biti problematične. Ne može se očekivati da korisnički uređaji imaju veliko skladište. Osim toga, veb-pregledač daje određen deo prostora na raspolaganju. *Chrome* dozvoljava da veb-aplikacije iskoriste do čak 60% ukupnog prostora na disku, *Firefox* do 50%, ali u slučaju *Safari*-a samo 1GB i korisnik mora odobriti povećanje dostupnog prostora za svakih narednih 200MB. Nikako se ne može znati koliko prostora je zaista dostupno u svakom slučaju za svakog od korisnika. Može se napraviti pretpostavka koja je možda tačna za sve korisnike te veb-aplikacije. Obično se može pretpostaviti da je nekoliko stotina MB prostora dostupno na raspolaganju.

Iako je moguće što možemo skladištiti podatke lokalno, to ne znači da će oni zauvek ostati uskladišteni. Veb-pregledači imaju mehanizme za brisanje podataka veb-sajtova koje korisnik nije skoro posetio. *Safari* npr. briše sve podatke veb-sajta koji korisnik nije posetio više od nedelju dana. Sam korisnik takođe može biti inicijator brisanja svih podataka uskladištenih u okviru veb-pregledača. Kako je baza podataka distribuirana na više mesta, pri koji svaka može sadržati različite verzije i skupove podataka, može doći do pojave konflikta.

A. Strategize rešavanja konflikta

1. Najčešća strategija je čuvanje svih konfliktnih verzija podataka, a zatim pri upitu izabrati pobeđujući podatak deterministički upoređivanjem njihovih heševa. Ova metoda može izazvati gubitak nekih podataka, ali za pojedine podatke koje nisu od velike važnosti može funkcionisati, i to ako su konflikti retki i zanemarljivi..

2. Bolje rešenje je da prilikom sinhronizacije i promene podataka koji je označen kao konfliktujući da se preuzmu sve njegove verzije, a zatim napravi novi dokument koji će sadržati pobeđujuće stanje podataka.

3. Takođe, konflikti se mogu rešiti i posredovanjem servera. Server će odlučiti kako da razreši konflikt i tu promenu zatim da sinhronizuje između ostalih klijenata.

4. Nekada je potrebno da sam korisnik odluči koju verziju podataka želi da zadrži, ali bi to zahtevalo kompleksnu implementaciju korisničkog interfejsa.

5. Nije potrebno razrešavanje konflikta ukoliko se do njih ne može doći. Može se dizajnirati baza podataka u kojoj se samo mogu upisivati novi podaci, tj. postojeći podaci se ne mogu modifikovati, a ni obrisati. Umesto da se uskladišti trenutno stanje podataka, moglo bi se uskladište promene stanja koje su dovele do trenutnog stanja. Računovođe ne koriste gumice [7] – i ova strategija je poznata kao „sve je delta" ili *Event Sourcing* [8].

6. Postoje i tipovi podataka specifično napravljeni za automatsko razrešavanje konflikata, *conflict-free replicated data type* (*CRDT*). Korišćenje *CRDT* biblioteke, kao što je *automerge* [9] bi trebalo da reši sve konflikte, mada implementacija *CRDT*-a u produkciji nosi istu kompleksnost kao i ostale strategije razrešavanje konflikta.

B. GUN baza podataka

GUN [10] je decentralizovana graf baza podataka koja je distribuirana u lokalnom skladištu veb-pregledača svih korisnika veb-aplikacije. Sinhronizuje se putem *WebRTC* i *WebSocket* protokola. Ne zahteva dodatnu implementaciju servera, već slično *WebTorrent* Tracker konceptu, *GUN* se služi javnim instancama za signaliranje i sinhronizaciju baze podataka kada nema drugih korisnika na mreži. Listu instanci je moguće izmeniti, kao i pokrenuti sopstvenu instancu i izabrati da li čuvati sve podatke svih korisnika u arhivske svrhe ili ne. Ovo nije samo baza podataka, već i ekosistem alatki za kriptografiju, autentikaciju, razrešavanje konflikata, umrežavanje, … Podatke je moguće čuvati u tri različita prostora: javni, korisnički i zamrznuti. Na osnovu prostora u koje je podatak zapisan i da li je enkriptovan ili ne, različita su prava pristupa drugim korisnicima tom podatku.

V. NOVI *HYPERMEDIA* PROTOKOLI

Do sada je diskutovano o decentralizovanoj komunikaciji korisnika i skladištenju korisničkih podataka aplikacija, ali ne i o samim aplikacijima i Veb stranicama. Uz pomoć tehnologija kao što su *IPFS* [11] i *Hypercore* [12] može se graditi novi, distribuirani Veb, dizajniran sa ciljem da je zauvek trajan. Na žalost, ove tehnologije nisu tako popularne i vodeći veb-pregledači ih (još uvek) ne implementuju.

Ovi protokoli bi omogućili mnogo lakše publikovanje veb-stranica, odnosno veb-aplikacija, koje bi odmah bile dostupne svima – mogu ponovo podsticati sve da stvaraju putem veb-pregledača, kao u ranije doba kada je *GeoCities* [13] servis bio aktivan. Prototipi takvih veb-pregledača postoje, kao što je *Beaker Browser*, koji pored standardnih funkcionalnosti veb-pregledača omogućava korisnicima da putem *Hypercore* protokola publikuju svoje veb-stranice.

Možda bi, takođe, implemenacija ovih protokola u okviru veb-pregledača pomogla i rešenju problema signalizacije *WebRTC*-a bez upotrebe servera. Korišćenjem *DHT* [14] (*Decentralized Hash Table*), poput *Kademlia* koju koriste *IPFS* i *Hypercore*, mogu se na decentralizovan način vršiti pronalaženja korisnika na mreži i njihovo međusobno umrežavanje.

Iako trenutno nisu direktno dostupni u veb-pregledačima, mogu se koristiti njihovim tehnologijama i

pristupiti im alternativnim načinima, mada ne uvek tako pouzdano i brzo, ponekad ne tako idealno sa strane korisnika. Moguće je dodati podršku putem *gateway*-evima [15] koji mogu biti instalirani na korisničkom uređaju ili da se upotrebi jedna od javnih instanci koje mogu biti ponekad preopterećene i spore. Ovo jeste funkcionalno, ali ima i dosta mesta za dalja poboljšanja.

VI. Zaključak

Već danas je moguće ostvariti decentralizovanu komunikaciju na Vebu, distribuirati podatke širom korisnika, ali i dalje ima mesta za dalja unapređenja. Projekti poput *Beaker Browser*-a pokazuju obećavajuće rezultate u upotrebi novih decentralizovanih protokola, a slične funkcionalnosti je već moguće upotrebiti u Electron [16] aplikacijama za eksperimentaciju novih mogućnosti i građenje desktop aplikacija upotrebom veb tehnologija budućnosti. Mnogi još uvek nisu upoznati sa decentralizacijom ili pak to usko povezuju sa kriptovalutama i *web3* pokretom.

Literatura

[1] Long polling, https://javascript.info/long-polling
[2] The WebSocket API (WebSockets), https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API
[3] WebRTC API, https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API
[4] Trystero Serverless WebRTC matchmaking for painless P2P, https://github.com/lmangani/trystero
[5] About Offline First, https://rxdb.info/offline-first.html
[6] Downsides of Offline First, https://rxdb.info/downsides-of-offline-first.html
[7] Conflicts Accountants don't use erasers, https://pouchdb.com/guides/conflicts.html#accountants-dont-use-erasers
[8] Event Sourcing, https://martinfowler.com/eaaDev/EventSourcing.html
[9] automerge, a CRDT that can be modified concurrently by different users, https://github.com/automerge/automerge
[10] GUN – the database for freedom fighters, https://gun.eco/docs/
[11] What is IPFS?, https://docs.ipfs.tech/concepts/what-is-ipfs/
[12] Hypercore Protocol, https://hypercore-protocol.org/
[13] GeoCities, https://wiki.archiveteam.org/index.php/GeoCities
[14] Distributed Hash Tables (DHTs), https://docs.ipfs.tech/concepts/dht/
[15] IPFS Gateway, https://docs.ipfs.tech/concepts/ipfs-gateway/
[16] Electron Build cross-platform desktop apps, https://www.electronjs.org/

Elektrohemijski procesi odgovorni za nestabilnosti VDMOS tranzistora snage usled NBT naprezanja

Sandra Veljković, Nikola Mitrović

Sadržaj − U okviru ovog rada razmatrani su mehanizmi formiranja zahvaćenog pozitivnog naelektrisanja u oksidu gejta i površinskih stanja koji dovode do promene napona praga kod IRF9520 VDMOS komercijalnih tranzistora snage pri NBT naprezanju. Prikazan je proces eksperimenta, koja je aparatura korišćena i eksperimentalno dobijene strujno-naponske karakteristike. Na osnovu ovih karakteristika je praćena promena napona praga (ΔV_T) i data je njena analiza. Ova promena V_T je posledica formiranja pozitivnog naelektrisanja u oksidu gejta i površinskih stanja do kojih dolazi u toku elektrohemijskih reakcija koje su detaljno analizirane, kao i odgovarajući mehanizmi. Mehanizmi koji se odvijaju u oksidu gejta i na međupovršini su definisani ukupnim brojem defekata u oksidu i na međuporšini, kao i prisustvom čestica vodonika. Ovi mehanizmi, predstavljeni i elektrohemijskim reakcijama, koje se odvijaju u toku NBT naprezanja, su i pojedinačno šematski predstavljeni u radu.

I. UVOD

MOSFET (*Metal-Oxide Semiconductor Field Effect*) tranzistori snage, odnosno MOS tranzistor sa efektom polja, su unipolarne komponente koje su dizajnirane tako da rade pri visokim vrednostima struja i napona. U ranim fazama proizvodnje korišćeni su različiti dizajni, ali su VDMOS (*Vertical Double Diffused MOS*) komponente postale dominantan tip [1]. Pored toga što imaju visok probojni napon, oni su sposobni da izdrže velike struje drejna. Jedinstvene karakteristike VDMOS tranzistora snage su omogućile njihovu široku upotrebu u komercijalnim i specijalnim aplikacijama. Ove komponente se u velikoj meri koriste za regulaciju snage u električnim uređajima za medicinske potrebe, kao i u industrijskoj i vojnoj elektronici [1]. Takođe, njihova upotreba je uobičajena u prekidačkim izvorima napajanja i audio pojačavačima, kao i u složenim sistemima koji predstavljaju primarnu i sekundarnu opremu u automobilskoj industriji [1]. U ovim i brojnim sličnim primenama tranzistori snage mogu biti izloženi zahtevnijim radnim uslovima i/ili stresu. Ovo je razlog velikog interesovanja za utvrđivanje funkcionisanja u specifičnim uslovima i okruženjima, kao i za njihov pouzdan rad [1-14]. Veliki broj studija je istakao uticaj jakih električnih polja [3-5], zračenja [2, 6] i naprezanja pri visokim temperaturama [8, 11]. Međutim, parametri uređaja određeni dizajnom moraju ostati stabilni, tj. da se kreću unutar predviđenih granica, tokom određenog perioda upotrebe u navedenom okruženju. Ako neki od parametara odstupa od definisanog opsega, može doći do kvara uređaja. Klasifikacija se može izvršiti korišćenjem mnogih kriterijuma [1], a odabir visokopouzdanih komponenti za upotrebu u određenom okruženju uključuje tehniku provere pouzdanosti. Osim toga, standardno ispitivanje pouzdanosti za ove komponente uključuje „*burn-in*“ testove, koji obuhvataju primenu polarizacije i povišenih temperatura [11- 14].

Delovi zasnovani na dielektricima su ključni delovi komponente. Silicijum dioksid (oksid) SiO_2 i međupovršina oksid – poluprovodnik SiO_2–Si su najosetljiviji delovi MOS komponenata koje su podvrgnute ekstremnim, teškim uslovima okoline - radnim uslovima ili naprezanjima. Naime, zračenje, naprezanje jakim električnim poljem (*high electric field* - HEF) ili naprezanje negativnom polarizacijom pre visokoj temperaturi dovode do stvaranja defekata u oksidu i defekata na međupovršini koji mogu izazvati degradaciju električnih parametara MOS komponenata. Ovi nedostaci su predmet istraživanja više od 40 godina, ali su i dalje u fokusu mnogih istraživača.

Štaviše, efekti NBTI (*Negative Bias Temperature Instability*) mogu se pojaviti tokom testiranja i u nekim situacijama, tokom normalnog rada VDMOS tranzistora snage. Ovi efekti su izraženiji u p-kanalnim komponentama i javljaju se na temperaturama u rasponu od 100 do 250°C i pri električnim poljima oksida gejta od 2 do 6 MV/cm [14-16]. Rad pri maksimalnoj snazi ili na povišenim temperaturama može dovesti do aktiviranja mehanizama koji izazivaju NBTI. Ovi mehanizmi se pokreću elektrohemijskim reakcijama koje stvaraju i/ili aktiviraju defekte u oksidu gejta i na međupovršini oksid-silicijum. Pod dejstvom primenjene negativne polarizacije gejta na povišenim temperaturama - NBT stresa, elektrohemijski procesi formiraju naelektrisanja u oksidu (N_{ot}) i površinska stanja (N_{it}). Prema tome, povećanje, u apsolutnoj vrednosti, napona praga je posledica formiranja i pozitivnog naelektrisanja oksida gejta i površinskih stanja.

NBTI je postao sve važniji problem pouzdanosti, jer se debljina oksida smanjuje sa svakom generacijom tehnologije. Uprkos činjenici da se dimenzije uređaja smanjuju, ispitivanje nestabilnosti u ultra debelim oksidima gejta i dalje je od interesa zbog široke upotrebe MOS tehnologije u elektronskim komponentama.

S. Veljković i N. Mitrović su studenti doktorskih studija na modulu Nanotehnologije i mikrosistemi na Katedri za mikroelektroniku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: sandra.veljkovic@elfak.rs.

Inače, brojne metode koje se koriste u proučavanju prirode defekata SiO_2 i defekata na međupovršini SiO_2–Si i njihove karakterizacije mogu se grubo podeliti u dve kategorije: metode primenjene na MOS kondenzatorskoj strukturi i metode primenjene na složenim MOS elektronskim komponentama. Ove metode su generalno zasnovane na električnim karakterizacijama i uglavnom daju rezultate koji se odnose na distribuciju energije defekata u oksidu i na međupovršini, omogućavajući na taj način predviđanje prirode defekta i modelovanje njihovog ponašanja.

U okviru ovog rada je na osnovu električnih karakteristika analizirana promena napona praga p-kanalnog VDMOS tranzistora snage. Kako je ova promena posledica formiranja pozitivnog naelektrisanja u oksidu gejta i površinskih stanja do kojih dolazi u toku elektrohemijskih reakcija, razmatrani su mehanizmi formiranja zahvaćenog pozitivnog naelektrisanja u oksidu gejta i površinskih stanja koji dovode do promene napona praga kod IRF9520 VDMOS komercijalnih tranzistora snage pri NBT naprezanju. Ovi mehanizmi su predstavljeni i elektrohemijskim reakcijama i pojedinačno su šematski prikazani u radu.

II. Eksperimentalni uslovi i oprema

Svi eksperimenti sprovedeni su prethodno razvijenom tehnikom zasnovanom na prekidačkom kolu koje je uspostavljeno da obezbedi odgovarajući kompromis između NBT naprezanja i zahteva merenja VDMOS tranzistora snage [17]. Tehnika je potvrđena korišćenjem NBTI testiranja u različitim eksperimentima [9, 18- 21]. Kao što se može videti na slici 1, cela postavka je podeljena na tri dela.

(1)  (2) 

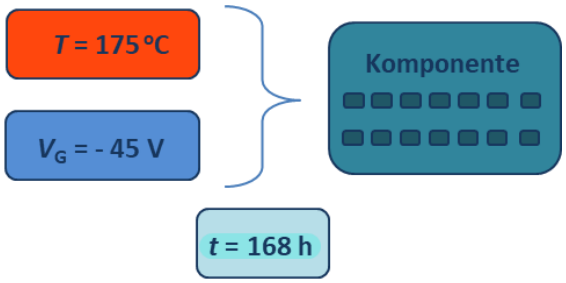
(3) 

Slika 1. Eksperimentalna postavka u laboratoriji: 1-napajanje, 2-komora za grejanje i 3-merna postavka.

Prvi deo (označen brojem 1) je odgovoran za obezbeđivanje konstantnog napona na gejtu. Za praćenje signala koji se šalje testiranoj komponenti, osciloskop se postavlja na ulaz kola. Drugi deo eksperimentalne opreme (označen brojem 2) predstavlja komoru za žarenje. Koristeći komoru, uspostavljena je temperatura od $T = 175\ ^{\circ}C$. Treći deo eksperimentalne opreme (označen brojem 3) uključuje SMU (*Source Measurement Unit*) *Keisight* B2901A kome se pristupa preko računara. Da bi se izvršila *I-V* merenja ovom tehnikom, napon koji se primenjuje radi naprezanja mora biti uklonjen sa komponente koji se testira. Napon praga se određuje na osnovu izmerene prenosne karakteristike.

Komponente koje su korišćene u eksperimentu su VDMOS tranzistori snage sa kanalom p tipa i označene su oznakom IRF9520. Imaju 1650 ćelija heksagonalnog oblika i proizvedene su u polisilicijumskoj tehnologiji. Debljina oksida ovih komponenata je oko 100 nm. Pre početka eksperimenta je proveren napon praga svih testiranih uzoraka i bio je oko -3.6 V. Ove komponente se standardno pakuju u plastična kućišta TO-220.

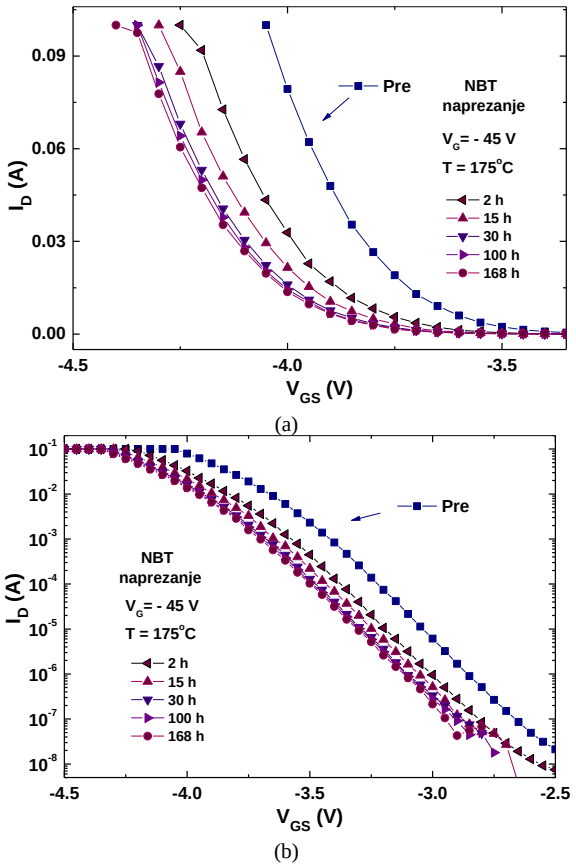
U toku eksperimenta grupa komponenata je bila NBT naprezana u trajanju od 168 h, kao što je prikazano na slici 2. Ovo naponsko temperaturno naprezanje negativnom polarizacijom je sprovedeno na Elektronskom fakultetu u Nišu, u laboratoriji Katedre za mikroelektroniku. Pri naprezanju, sors se kratkospaja sa drejnom i povezuje se sa masom, dok se na gejt primenjuje polarizacija od - 45 V. Pomoću ove aparature obezbeđena je konstantna temperatura u vrednosti od 175 °C, i ta vrednost je održavana tokom 168 h, odnosno jednu nedelju.



Slika 2. Parametri NBT naprezanja primenjeni na grupu testiranih komponenata.

III. Rezultati i njihova analiza

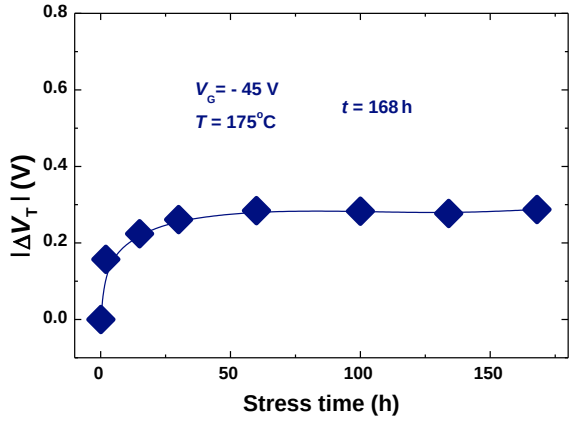
Da bi se izvršila električna karakterizacija tokom eksperimenta, naprezanja su prekidana u tačno određenim vremenskim intervalima i na sobnoj temperaturi, snimane su prenosne karakteristike u oblasti zasićenja. Na slici 3 su prikazane ove karakteristike u nadpragovskoj (a) i podpragovskoj (b) oblasti. Date su karakteristike za tipičnu komponentu, pri čemu su i za ostale komponente karakteristike vrlo bliske.



Slika 3. Snimljene prenosne karakteristike u nadpragovskoj (a) i podpragovskoj (b) oblasti.

Na osnovu slike 3. se može uočiti da tokom primene NBT naprezanja dolazi do značajnog pomeranja karakteristika duž naponske ose, pri čemu je to pomeranje najizraženije u početnoj fazi, a zatim je sve manje. Takođe se vidi da ne dolazi do značajne promene nagiba karakteristika.

Na osnovu snimljenih prenosnih karakteristika tranzistora je određivana promena napona praga (ΔV_T) a dobijeni rezultati su prikazani na slici 4. Napon praga je

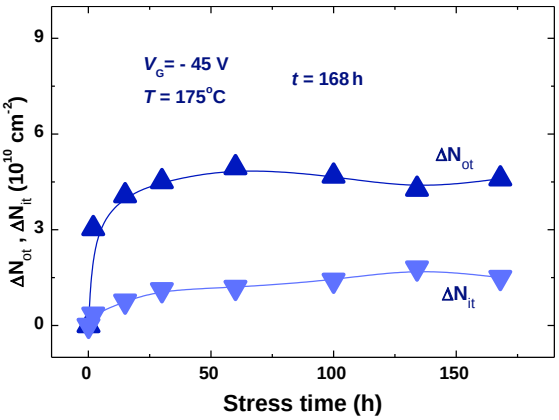


Slika 4. Promena napona praga (apsolutna vrednosti) tokom NBT naprezanja tipične komponente.

određivan ekstrapolacijom linearnog dela $\sqrt{I_D}-V_G$ krivih do preseka sa V_G-osom. Vidi se da tokom NBT naprezanja dolazi do značajnog porasta napona praga (po apsolutnoj vrednosti) u toku prvih dvadesetak sati, a zatim se taj porast značajno usporava.

Za uočeno ponašanje napona praga tokom NBT naprezanja p-kanalnog VDMOS tranzistora snage su odgovorne promene gustine pozitivnog naelektrisanja u oksidu gejta i površinskih stanja. Zato su ove promene određivane i to pomoću SMG (*Subtreshold Midgap*) tehnike [22], koja je zasnovana na merenju podpragovskih prenosnih karakteristika u oblasti zasićenja i rezultati su prikazani na slici 5.

Sa slike 5. se vidi da i promene gustine pozitivnog naelektrisanja u oksidu gejta i površinskih stanja pokazuju trend porasta tokom NBT naprezanja i da je ΔN_{ot} veće od ΔN_{it} tokom celokupnog naprezanja. Takođe, može se uočiti da ΔN_{it} ranije počinje sa usporavanjem porasta i pre ulazi u zasićenje od ΔN_{ot}.



Slika 5. Promena gustine pozitivnog naelektrisanja u oksidu gejta i površinskih stanja tokom NBT naprezanja tipične komponente.

IV. ODGOVORNI MEHANIZMI

Uočene promene gustine pozitivnog naelektrisanja u oksidu gejta i površinskih stanja su posledica elektrohemijskih reakcija do kojih dolazi u toku NBT naprezanja. Pri tome se na početku naprezanja formira pozitivno naelektrisanje u oksidu usled zahvatanja šupljina na defektima u oksidu, kao što su vakancije kiseonika [14]:

$$O_3 \equiv Si^{\bullet\bullet}Si \equiv O_3 + h^+ \rightarrow O_3 \equiv Si^{+\bullet}Si \equiv O_3 . \qquad (1)$$

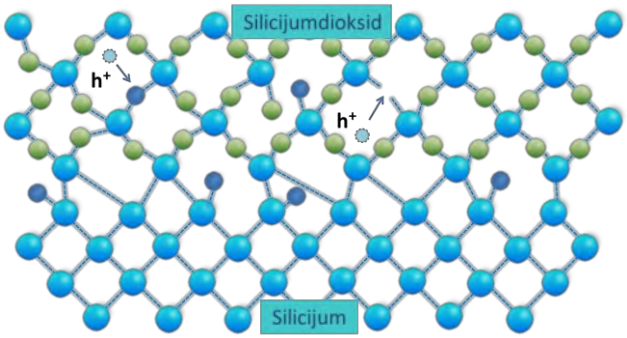
Takođe, formirane šupljine mogu da dovedu do raskidanja oslabljenih Si-H veza u oksidu, što može da se predstavi sledećom reakcijom [23]:

$$O_3 \equiv Si - H + h^+ \leftrightarrow O_3 \equiv Si^+ + H^\bullet . \qquad (2)$$

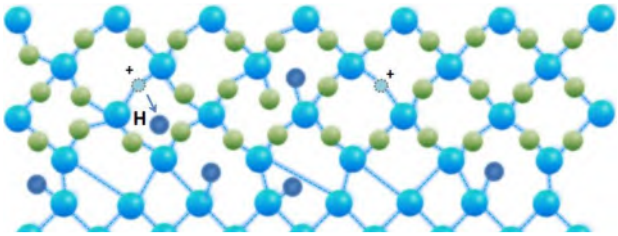
Ove elektrohemijske reakcije su i šematski predstavljene na slici 6. Pri tome je na slici 6a. prikazano zahvatanje šupljina, dok je na slici 6b. prikazano

formirano, odnosno zahvaćeno pozitivno naelektrisanje u oksidu i oslobađanje atoma vodonika.





(a)



(b)

Slika 6. Šematski prikaz zahvatanja šupljina (a) i formiranja pozitivnog naelektrisanja (b) u oksidu.

Pored ovih reakcija, u početku NBT naprezanja, na međupovršini dominira proces disocijacije oslabljenih Si-H veza (slika 7). Tako, pod uticajem jakog električnog polja dolazi do raskidanja veza, odnosno reakcije [24]:

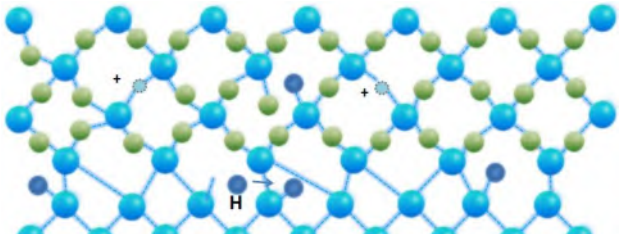
$$Si_3 \equiv Si - H \leftrightarrow Si_3 \equiv Si^{\bullet} + H^{\bullet} \qquad (3)$$

pri čemu se formiraju površinska stanja i oslobađa atom vodonika, kao što je prikazano na slici 7a. Oslobođeni atom vodonika je jako reaktivan i on takođe može da učestvuje u disocijaciji oslabljenih Si-H veza:

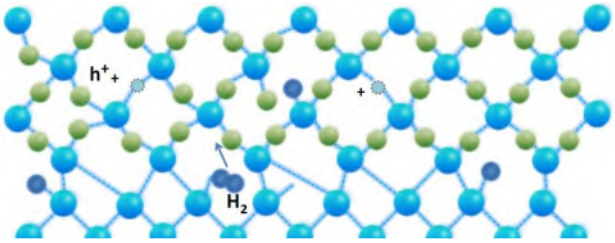
$$Si_3 \equiv Si - H + H^{\bullet} \leftrightarrow Si_3 \equiv Si^{\bullet} + H_2 \qquad (4)$$

pri čemu se formiraju površinska stanja i formira molekul vodonika, kao što je prikazano na slici 7b.

Sa povećanjem vremena naprezanja raste i broj nastalih površinskih stanja, a time i verovatnoća da i atom i molekul vodonika učestvuju u pasivizaciji površinskih stanja. Ovaj proces pasivizacije je u stvari opisan povratnim reakcijama (3) i (4) tako da se proces formiranja površinskih stanja usporava i zato vrlo brzo dolazi do zasićenja porasta gustine površinskih stanja (slika 5.).



(a)



(b)

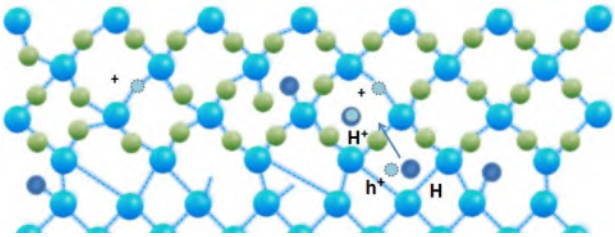
Slika 7. Šematski prikaz disocijacije oslabljenih Si-H veza i formiranja površinskih stanja, kao i oslobađanje atoma vodonika (a) i molekula vodonika (b).

Pored navedenih reakcija, veoma reaktivni atom vodonika može da reaguje sa šupljinom (slika 8a.) formirajući jon vodonika [25]:

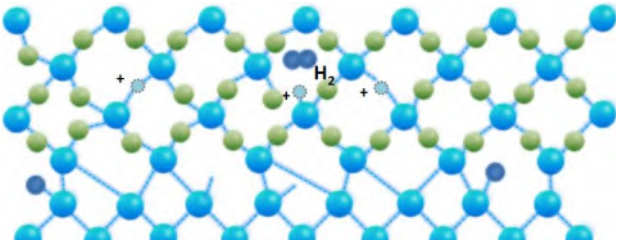
$$H^{\bullet} + h^{+} \rightarrow H^{+}. \qquad (5)$$

Nastali jon vodonika driftuje od međupovršine pod dejstvom električnog polja. U oksidu jon može da dovede do disocijacije oslabljenih Si-H veza u blizini međupovršine (slika 8b.), pri čemu se formira dodatno naelektrisanje u oksidu i molekul vodonika [24]:

$$O_3 \equiv Si - H + H^{+} \leftrightarrow O_3 \equiv Si^{+} + H_2. \qquad (6)$$



(a)



(b)

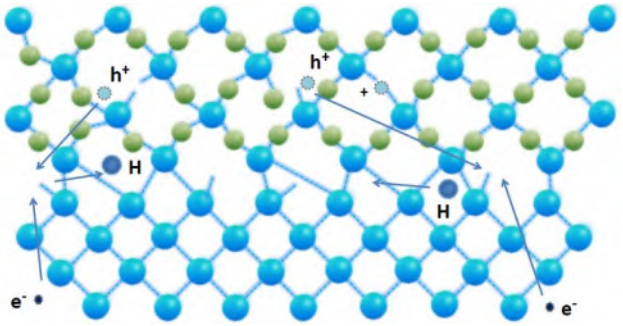
Slika 8. Šematski prikaz formiranja jona vodonika (a) i dodatnog formiranja pozitivnog naelektrisanja (b) u oksidu.

Kako se sve više formira naelektrisanje u oksidu ono postepeno smanjuje lokalno električno polje u blizini međupovršine, što dovodi do sve sporijeg porasta gustine naelektrisanja u oksidu tokom NBT naprezanja. Pored toga, zahvaćeno pozitivno naelektrisanje u oksidu može biti transformisano u površinska stanja u kasnijoj fazi naprezanja [14]:

$$O_3 \equiv Si^{+\bullet} Si \equiv O_3 + Si_3 \equiv Si - H + e^- \rightarrow Si_3 \equiv Si^{\bullet} + O_3 \equiv Si^{\bullet\bullet} Si \equiv O_3 + H^{\bullet} \quad (7)$$

$$O_3 \equiv Si^{+} + Si_3 \equiv Si - H + e^- \rightarrow Si_3 \equiv Si^{\bullet} + O_3 \equiv Si^{\bullet} + H^{\bullet} \quad (8)$$

Pri elektrohemijskim reakcijama koje opisuju ovu transformaciju dolazi do oslobađanja vrlo reaktivnog atoma vodonika koji dalje može da učestvuje u novim reakcijama, kao što je prikazano na slici 9.



Slika 9. Šematski prikaz transformisanja pozitivnog naelektrisanje u oksidu u površinska stanja.

Jasno je da na elektrohemijske reakcije i procese do kojih dolazi značajno utiču reakcije u kojima učestvuju čestice vodonika. Pri tome dolazi do interakcije vrlo reaktivnih atoma vodonika, difuzije neutralnih molekula H_2 i drifta pozitivnih jona H^+ i njihove interakcije sa defektima i oslabljenim vezama. Zato su praktično sve elektrohemijske reakcije koje dovode do formiranja naelektrisanja u oksidu i površinskih stanja i definisane ukupnom količinom čestica vodonika koje se nalaze na i u blizini međupovršine silicijumdioksid-silicijum (SiO_2-Si). Pored toga i ukupan broj defekata u oksidu i na međupovršini definiše procese formiranja zahvaćenog pozitivnog naelektrisanja u oksidu gejta i površinskih stanja. Posledica toga je promena napona praga, kao jednog od najvažnijih parametara komercijalnih VDMOS tranzistora snage. Zbog toga je i analiza elektrohemijskih reakcija i procesa koji se dešavaju u komponenti (a na osnovu uočenih promena napona praga i razdvajanja uticaja naelektrisanja u oksidu gejta i površinskih stanja) od posebnog značaja za dalju analizu rada, ali i predviđanje pouzdanog rada komercijalnih VDMOS tranzistora snage u realnim uslovima.

V. Zaključak

U radu su analizirani mehanizmi formiranja zahvaćenog pozitivnog naelektrisanja u oksidu gejta i površinskih stanja koji dovode do promene napona praga kod IRF9520 VDMOS komercijalnih tranzistora snage pri NBT naprezanju. Promena napona praga, kao jednog od najvažnijih parametara komercijalnih VDMOS tranzistora snage, uslovljena je elektrohemijskim reakcijama koje se odvijaju u oksidu i na međupovršini SiO_2-Si. U toku ovih reakcija dolazi do formiranja zahvaćenog pozitivnog naelektrisanja u oksidu gejta i površinskih stanja, pri čemu ukupan broj defekata u oksidu i na međuporšini definiše ove procese formiranja. Zbog toga u početku NBT naprezanja dominiraju procesi raskidanja oslabljenih veza, a u kasnijoj fazi postaju izraženi i procesi pasivizacije, neutralizacije, ali i transformacije zahvaćenog pozitivnog naelektrisanja u oksidu gejta u površinska stanja. U ovim reakcijama je vrlo značajno učešće čestica vodonika, pri čemu dolazi do interakcije vrlo reaktivnih atoma vodonika, difuzije neutralnih molekula H_2 i drifta pozitivnih jona H^+ i njihove interakcije sa defektima i oslabljenim vezama. Ove elektrohemijske reakcije, koje se odvijaju u toku NBT naprezanja su i pojedinačno šematski predstavljene u radu.

Zahvalnica

Prikazani rezultati dobijeni su u okviru programa za finansiranje naučnoistraživačkog rada (pod brojem 451-03-68/2022-14/ 200102) koji finansira Ministarstvo prosvete, nauke i tehnološkog razvoja Republike Srbije i u okviru projekta Evropske Komisije Horizon 2020 research and innovation program (Grant No. 857558 — ELICSIR). Autori se zahvaljuju prof. dr Danijelu Dankoviću na motivaciji, korisnim savetima i sugestijama prilikom izrade rada.

Literatura

[1] B. Jayant Baliga, *Fundamentals of Semiconductor Power Devices*, Springer, New York, 2008.
[2] S. Veljković, et al, "Effects of Bias Temperature Stress and Irradiation in Commercial p-Channel Power VDMOS Transistors", in *Proc. 32nd International Conference on Microelectronics (MIEL 21)*, Niš, October 2021, pp. 345-350.
[3] X.R. Ye, et al, "VDMOSFET HEF degradation modelling considering turn-around phenomenon", *Microelectronics Reliability*, 2018, vol. 50, pp. 37-41.
[4] G. S. Ristić, "Defect Behaviors During High Electric Field Stress of p-Channel Power MOSFETs", *IEEE Transactions on Device and Materials Reliability*, 2012, vol. 12 (1), pp. 94–100.
[5] N. Stojadinović, et al, "Effects of electrical stressing in power VDMOSFETs", *Microelectronics Reliability*, 2005, vol. 45, pp. 115-122.
[6] M. S. Park, and C. R. Wie, "Study of radiation effects in /spl gamma/-ray irradiated power VDMOSFET by DCIV technique", *IEEE Transactions on Nuclear Science*, 2001, vol. 48(6), pp. 2285-2293.

[7] K. L. Ryder, et al, "Systems engineering and assurance modeling (SEAM): A web-based solution for integrated mission assurance", *Facta Universitatis, Series: Electronics and Energetics*, 2021, vol. 34(1), pp. 1-20.
[8] A. Bhattacharjee and S. N. Pradhan, "NBTI-Aware Power Gating Design with Dynamically Varying Stress Probability Control on Sleep Transistor", *Journal of Circuits, Systems, and Computers*, 2021, vol. 30(11), pp. 2120004.
[9] D. Danković, et al, "Negative bias temperature instability in p-channel power VDMOSFETs: recoverable versus permanent degradation", *Semiconductor Science and Technology*, 2015, vol. 30(10), pp. 105009.
[10] A. N. Tallarico, et al, "Negative Bias Temperature Stress Reliability in Trench-Gated P-Channel Power MOSFETs", *IEEE Transactions on Device and Materials Reliability*, 2014, vol. 14(2), pp. 657-663.
[11] S. Djoric-Veljkovic, et al, "Effects of burn-in stressing on post-irradiation annealing response of power VDMOSFETs", *Microelectronics Reliability*, 2003, vol. 43(9-11), pp. 1455-1460.
[12] P. Magnone, et al, "Investigation of degradation mechanisms in low-voltage p-channel power MOSFETs under High Temperature Gate Bias stress", *Microelectronics Reliability*, 2018, vol. 88-90, pp. 438-442.
[13] N. Stojadinovic, et al,"Effects of high electric field and elevated-temperature bias stressing on radiation response in power VDMOSFETs", *Microelectronics Reliability*, 2002, vol. 42(4-5), pp. 669-677.
[14] N. Stojadinović, et al, "Negative bias temperature instability mechanisms in p-channel power VDMOSFETs", *Microelectronics Reliability*, 2005, vol. 45(9-11), pp. 1343-1348.
[15] J. H. Stathis and S. Zafar, "The negative bias temperature instability in MOS devices: A review", *Microelectronics Reliability*, 2006, vol. 46(2-4), pp. 270-286.
[16] T. Grasser, et al, "Recent Advances in Understanding the Bias Temperature Instability", in *Proc. Inter. Elect. Dev. Meet. IEDM 2010*, vol. 82-85, San Francisco, (CA, USA), 2010, pp. 4.4.1-4.4.4.
[17] A. Prijic, et al, "A method for negative bias temperature instability (NBTI) measurements on power VDMOS transistors", *Measurement Science and Technology*, 2012, vol. 23(8), pp. 085003-1-8.
[18] D. Dankovic, et al, "Analysis of recoverable and permanent components of threshold voltage shift in NBT stressed p-channel power VDMOSFET", *Chinese Physics. B*, 2015, vol. 24(10), pp. 106601-1-9.
[19] D. Danković, et al, "On the recoverable and permanent components of NBTI in p-channel power VDMOSFETs", *IEEE Transactions on Device and Materials Reliability*, 2016, vol. 16(4), pp. 522–531.
[20] I. Manić, et al, "Effects of pulsed negative bias temperature stressing in p-channel power VDMOSFETs", *Facta Universitatis, Series: Electronics and Energetics*, 2016, vol. 29, pp. 49-60.
[21] S. Veljković, et al, "Response of Commercial P-Channel Power VDMOS Transistors to Ionizing Irradiation and Bias Temperature Stress", *Journal of Circuits, Systems, and Computers*, 2022, vol. 31, pp. 2240003-1-25.
[22] P.J. McWhorter and P.S. Winokur. "Simple technique for separating the effects of interface traps and trapped-oxide charge in metal-oxide-semiconductor transistors", *Applied. Physics. Letters*, 1986, vol. 48, pp. 133-135.
[23] V. Davidović et al., "Effects of consecutive irradiation and bias temperature stress in p-channel power vertical double-difused metal oxide semiconductor transistors", *Japanese Journal of Applied Physics*, 2018, vol. 57, pp. 044101-1-10.
[24] S. Ogawa, et al., "Interface-trap generation at ultrathin SiO_2 (4-6 nm)-Si interfaces during negative-bias temperature aging", *Journal of Applied Physics*, 1995, vol. 77, pp. 1137-1148.
[25] D. M. Fleetwood, "Effects of hydrogen transport and reactions on microelectronics radiation response and reliability", *Microelectronics Reliability*, 2002, vol. 42(4–5), pp. 523–541.

Određivanje vlažnosti vazduha i temperature korišćenjem senzora DHT11

Petra Milićević, Marina Smiljanić, Natalija Živanović

Sadržaj – U ovom radu predstavljeno je projektovanje uređaja za merenje i kontrolu parametara kvaliteta vazduha. Uz pomoć senzora DHT11 i *Arduino Uno* okruženja, uspešno je realizovan uređaj za određivanje temperature i vlažnosti vazduha u prostoriji.

I. Uvod

Senzori za merenje vlažnosti vazduha i temperature koriste se za proveru i određivanje klimatskih parametara i uslova. Takvi uređaji se koriste u situacijama u kojima kvalitet vazduha mora biti kontrolisan.

Vlažnost vazduha predstavlja količinu vodene pare u vazduhu. Za različite procese u industriji uticaj vlažnosti vazduha mora biti pažljivo kontrolisan. Takođe, kontrola vlažnosti često je neophodna za inkubatore, respiratornu opremu, sterilizatore i biološke proizvode. Vlažnosti vazduha može imati veliki uticaj na zdravlje ljudi i bezbednost osoblja, a takođe utiče i na cenu proizvoda.

Merenje temperature je važan proces u mnogim oblastima. Kao što je napomenuto o važnosti uticaja vlažnosti vazduha, isti je slučaj i sa merenjem temperature. Veoma je bitno kontrolisati vrednost temperature u prostoriji kako bi se održali optimalni uslovi rada kako za ljude, tako i za uređaje. Savremeni računari koriste signale dobijene sa senzora temperature da bi mogli ispravno da upravljaju procesima.

Iz tog razloga, kao rezultat, senzori za merenje temperature i vlažnosti vazduha su veoma važni [1].

II. Komponente korišćene za realizaciju projekta

Za realizaciju ovog projekta korišćene su sledeće komponente: *Arduino Uno* okruženje, senzor DHT11, potenciometar, LCD displej i žičani kratkospojnici.

A. Arduiono Uno razvojno okruženje

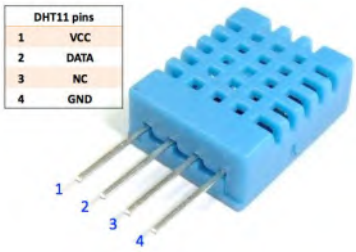
Arduino Uno (slika 1.) je razvojno okruženje sa ATmega328P mikrokontrolerom. Poseduje 14 digitalnih ulazno/izlaznih pinova, 6 analognih ulaza, keramički rezonator od 16 MHz, USB konektor, priključak za napajanje, ICSP zaglavlje i reset dugme. Raspolaže sa 2 kB SRAM memorije, 32 kB flash memorije, 1 kB EEPROM memorije, a radni napon ovog okruženja je 5 V [2].



Slika 1. Prikaz *Arduino Uno* okruženja.

B. Senzor DHT11

DHT11 (slika 2.) je senzor za merenje temperature i vlažnosti vazduha. Ovaj senzor se sastoji od termistora za merenje temperature, kapacitivnog senzora za merenje vlažnosti vazduha i elektronike koja omogućava komunikaciju sa mikrokontrolerom. Obezbeđuje visoku pouzdanost i odličnu dugoročnu stabilnost, jednostavan je za upotrebu i niske je cene. Napon napajanja ovog senzora kreće se u opsegu od 3 do 5.5 V, njegova potrošnja struje za vreme konverzije je 2.5 mA i detektuje temperaturu do 50 ˚C [3]. Poseduje četiri pina, Vcc (pin za napajanje), *data* (pin za prenos podataka), NC (*not connected* pin) i GND (pin za masu).



Slika 2. Prikaz senzora DHT11 i njegovih pinova.

C. Potenciometar

Potenciometar (slika 3.) je otpornik promenljive otpornosti, koji je korišćen za manuelno podešavanje pozadinskog osvetljenja LCD displeja.

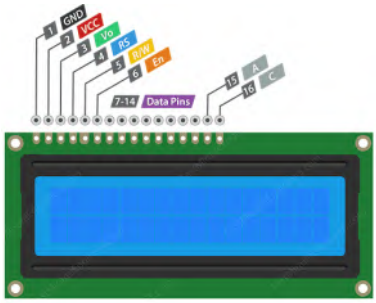
P. Milićević, M. Smiljanić, N. Živanović su učenice treće godine gimnazije "Bora Stanković" u Nišu, Vožda Karađorđa 27, 18000 Niš, E-mail: marina.m.smiljanic@gmail.com.



Slika 3. Prikaz potenciometra.

D. LCD displej

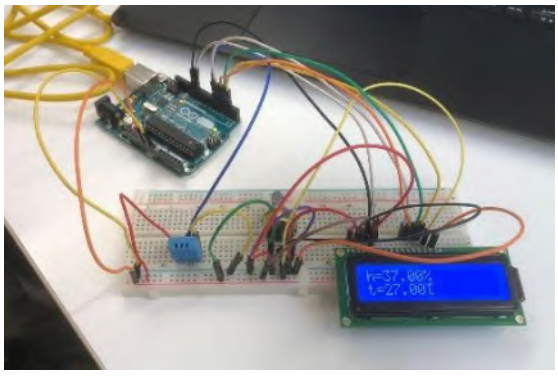
Za ovaj projekat korišćen je 16x2 LCD displej (slika 4.). Pomoću ovog displeja se prikazuju rezultati merenja i karaktera. Displej sadrži 1280 piksela podeljenih u 2 reda i 16 kolona. Ima 32 polja od 40 piksela (5x8) [4].



Slika 4. LCD displej (16x2).

III. PRAKTIČNA REALIZACIJA

Praktična realizacija ovog projekta je prikazana na slici 5.



Slika 5. Praktična realizacija sistema.

Za pokretanje projekta bilo je neophodno napisati kod (slika 6.) korišćenjem *Arduino IDE* simulatora. Program sadrži definisanje pinova, kreiranje simbola stepen Celzijusov (ºC), koji je napisan pomoću uključenih, odnosno isključenih piksela u jednom polju LCD displeja. Postavljaju se startni parametri: 0 ºC i 0 %.

Senzor detektuje temperaturu i vlažnost koji se ispisuju na LCD displeju zajedno sa oznakama za temperaturu i za vlažnost vazduha.

U nastavku rada prikazan je kompletan kod napisan u programu.

```
#include "DHT.h"
#include "LiquidCrystal.h"
#define DHTPIN 6
#define DTHTYPE DHT11

Const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
 byte stepen [8]={
B10111,
B00100,
B00100,
B00100,
B00100,
B00100,
B00111,
};

void setup() {
Serial.begin(9600);
lcd.begin(16,2);
dht.begin();
lcd.createChar(0,stepen);
}

void loop() {
delay(500);
float h = dht.readHumidity();
float t = dht.readTemperature();
lcd.setCursor(0,0);
lcd.print("h=");
lcd.print(h);
lcd.print("%");
lcd.setCursor(0,1);
lcd.print("t=");
lcd.print(t);
lcd.write(byte());
}
```

Slika 6. Kod napisan korišćenjem *Arduino IDE* simulatora.

IV. ZAKLJUČAK

U ovom radu je prikazan praktičan sistem za merenje promene temperature i vlažnosti vazduha. Objašnjene su sve korišćene komponente i kod koji se koristi za pokretanje sistema.

ZAHVALNICA

Ovaj rad nastao je kao rezultat realizacije projekta „Let STEM visit again IEEESTEC“, finansiranog u okviru 2022 IEEE Pre-University STEM Portal Grant Program-a. Takođe, autori se zahvaljuju Nedi Stanojević i Sandri Veljković na pomoći pri izradi rada.

LITERATURA

[1] DHT11 senzor, objašnjenje dostupno na linku: https://www.comptus.com/the-importance-of-temperature-and-humidity-sensors/
[2] Arduino Uno okruženje, objašnjenje dostupno na linku https://store-usa.arduino.cc/products/arduino-uno-rev3
[3] Datasheet DHT11 dostupno na linku: https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf
[4] LCD displej, objašnjenje dostupno na linku: https://www.winstar.com.tw/products/character-lcd-display-module/16x2-datasheet.html

Modeling a Fractal Antenna using Artificial Neural Networks

Nemanja Perić, Ksenija Pešić, Zoran Stanković, Nebojša Dončov

Abstract −This paper describes EM model and neural model of the Sierpinski gasket fractal antenna which is based on the MLP (Multi-Layer Perceptron) network. First, the characteristics of the fractal antenna were investigated and the training and test sets were generated using the MATLAB software environment, and then the neural model was developed. As input parameters of the neural model, i.e. the physical parameters of the fractal antenna to be modeled are *s* – side length of fractal antenna and *w* – neck width of fractal antenna while the output parameters are resonant frequencies belonging to the range [500-3000] MHz and minimum values of S_{11} parameters.

I. Introduction

With the development of microwave techniques and new technologies in the field of EM waves, printed antennas began to be studied. These antennas are widely used in wireless communications. These antennas' benefits include their compact size, low cost, ability to operate over a range of frequencies and simplicity of antenna array integration. Low gain and limited bandwidth are the disadvantages of printed antennas. [1,2] In this paper, the printed antenna that was used, described and modeled is the fractal antenna.

Maxwell's equations can be used to determine the characteristics of an antenna with a simple geometry. However, with the increasing complexity of the antenna, these equations become more and more complex and it is impossible to solve them analytically, and therefore numerical methods are used. Some of those methods are FEM [3] (Finite Element Method), MoM [3] (Method of Moments), SSPE [4] (split-step parabolic equation), FDTD[3] (Finite Difference Time-Domain) and TLM [3,5] (Transmission Line Matrix) method. Numerical EM approach in antenna modeling gives high accuracy results, however, using artificial neural networks can get a neural model that has similar accuracy, but once successfully developed, its speed exceeds the speed of other classical numerical EM models. As an alternative to numerical EM simulators, antenna models based on Artificial Neural Networks (ANN) can be utilised.[6,7]

Radio frequency identification (RFID) [8] is the name for technologies that from a use radio waves to identify objects or people at distances few centimeters to several tens of meters. It is about automatic identification,which means that any object can be automatically identifed. A reader and a tag are used for identification. The code is saved in the RFID tag that is attached to the physical object. Since the object is now recognizable,the object transmits the code.In this way,the reader receives information about the object. A frequency of 2.45GHz is used for active RFID. A fractal antenna can be used as an antenna in an RFID system because of the performance it has that corresponds to the way this system works.

N. Perić, K. Pešić, Z. Stanković and N. Dončov are with the Department of Telecomunications, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: nemanjaa.elfak.rs.

II. Architecture of Fractal Antenna

The architecture of fractal antenna is shown in Fig. 1 provides the basis for developing the EM and neural model that will be described in Sections III and IV, respectively. In this paper Sierpinski's Gasket fractal antenna is considered. Such antennas consist of several reduced and rotated copies of themselves. It starts with a triangle in the plane and then applies a repeating pattern of operations to it. The midpoints of its three sides are joined to form four congruent triangles from which the central triangle is dropped. In this way, the first phase of construction has been completed. In other words, after the first step, we have three congruent triangles whose sides are exactly half the size of the original triangle. Since such antennas are constructed using an iterative process, there are different combinations of antennas. In order to denote the number of iterations in this work, the symbol N was used. In other words, N represents the number of different sizes of triangles that are dropped and N = 2 is taken as the default value in this paper. Parameters that describe this type of antenna are *s* – sidelength and *w* – neckwidth are they are marked in Fig. 1. If we look at Fig. 1, the parts filled with gray color represent the metal construction, while the other parts are cut. The antenna is fed in the middle and this is shown by the red dot in the Fig. 1.

For an antenna designed in this way, it is possible to determine the resonant frequency using the formula from the paper [9]. However, this formula approximately determines only the resonant frequency, using an iterative procedure. The further goal of the research is the optimization of the antenna that is, obtaining the physical parameters of the antenna for a predetermined resonant frequency, so that the antenna radiates well at the given frequency. For the optimization of the antenna, it is very important to adapt the antenna so that the losses are as small as possible,

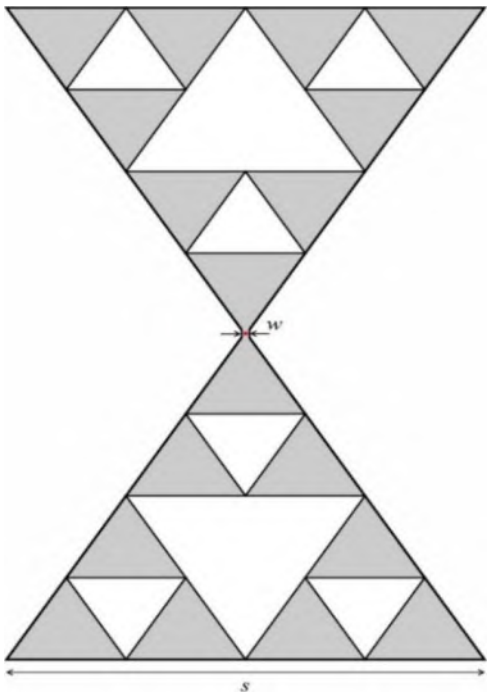


Fig. 1. Architecture of Fractal Antenna.

accordingly calculating the resonant frequency in this way is not enough, but it is also necessary to know the value of the S_{11} parameter at that frequency. Since there is no equation that defines the value of the S_{11} parameter at the resonant frequency for this design, an EM model was developed.

III. EM MODEL OF FRACTAL ANTENNA

EM simulators provide an opportunity to avoid complex mathematical calculations for obtaining primary and secondary antenna characteristics. The software package MATLAB [10] was used to analyze the fractal antenna. Antenna Toolbox, which is an integral part of MATLAB, uses the Method of Moments (MoM) to calculate the performance of the antenna. By using the *fractalGasket(Name,Value)* function, a fractal antenna was created, which is processed in this paper. The "*Name"* is a property of the antenna that can represent all the parameters of the fractal antenna, while the *"Value"* represents the value of those parameters. The parameters used to describe the antenna are: *NumIterations N* - number of iterations of fractal antenna, *Side s* - lengths for three sides of triangle and *NeckWidth w* – width at neck of fractal antenna. In this work, the property *NumIterations* iteration is taken as a constant value and N = 2. Figure 2 for the physical characteristics of the antenna w = 3.35 mm and s = 0.1 m shows the value of the parameter S_{11} as a function of frequency. The resonant frequency can be detected from this graph. The accompanying graphs of the 3D radiation diagram and directivity in the azimuthal and elevation planes at the previously discovered resonant frequency were created

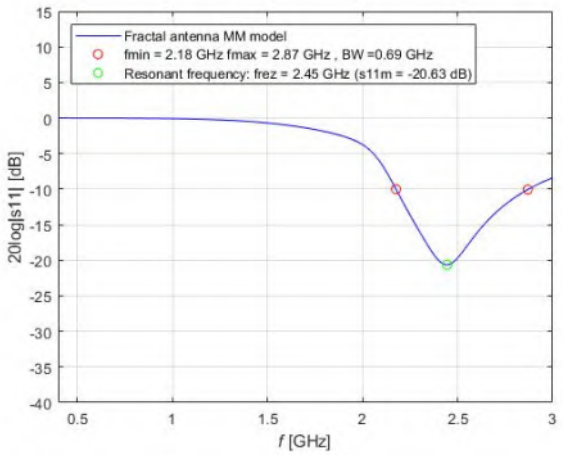


Fig. 2. Plotted frequency response of S_{11} antenna parameter.

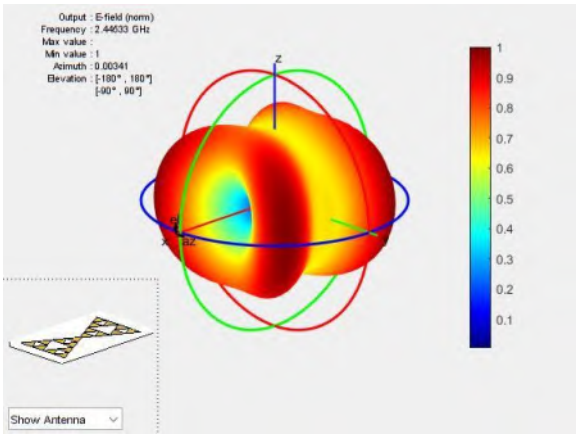


Fig. 3. 3D radiation pattern of the Fractal Antenna at f_r = 2.45 GHz.

using the *pattern*, *patternAzimuth*, and *patternElevation* functions and they are shown in Fig. 3, Fig. 4 and Fig. 5, respectively.

IV. NEURAL MODEL OF THE FRACTAL ANTENNA

The fractal antenna neural model (MLP_FA) is composed of one MLP neural network. The specific configuration and interconnection of neurons in the form of a network, which represents the architecture of a neural network is shown in Fig 6. MLP_FA serves to model the dependence of resonant frequency f_r and minimum value of S_{11} parameters S_{11min} on the physical parameters: neck width *w* and side *s*. MLP_FA network consists of four layers, input, output, and two hidden layers which have various numbers of neurons each. Due to the number of input and output parameters of the MLP_FA network, both the input and output layers contain two neurons each, but the hidden layers have a different number of neurons. In this specific case, there are 2 hidden layers and each hidden layer has between 6 and 22 neurons.

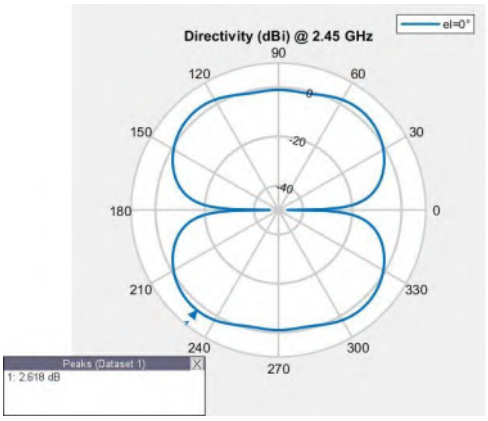


Fig. 4. Directivity of the fractal antenna at f_r = 2.45GHz in the azimuthal plane.

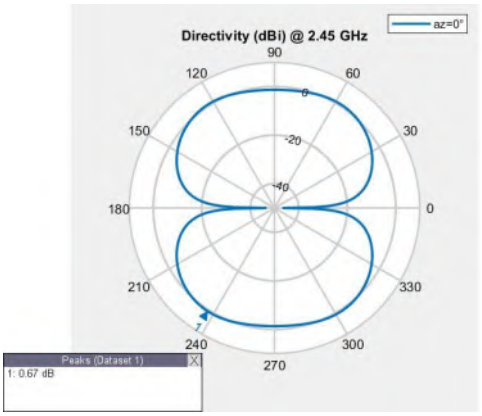


Fig. 5. Directivity of the fractal antenna at f_r = 2.45GHz in the elevation plane.

The vector of input variables used the model is $\boldsymbol{x}=[w\ \ s]^T$, while the vector of output variables is $\boldsymbol{y}=[f_r\ S_{11min}]$ in a matrix representation. Description of the MLP FA neural network is $y = y(x, W, B)$, where y is a network processing function, W is a set of connection weighting matrices w_i, $W = \{w_1, w_{2,}\ldots w_{H+1}\}$ and B is a set of bias vectors b_i, $B = \{b_1, b_{2,}\ldots b_{H+1}\}$ (The MLP network's total number of hidden layers is H). For defined input and output vectors and previously described parameters the MLP_FA neural network can be characterized as follows:

$$[f_r\ \ S_{11min}] = y([w\ s]^T, W, B) \tag{1}$$

The following function represents the output of the MLP's *l*-th hidden layer, $\boldsymbol{y}_l$

$$\mathbf{y}_l = F(\mathbf{w}_l \mathbf{y}_{l-1} + \mathbf{b}_l) \quad l = 1, 2, \ldots, H \tag{2}$$

where the (*l-1*)th hidden layer's output is represented by the vector y_{l-1}, w_l is a connection weight matrix between (*l-1*)-th and *l*-th hidden layer neurons, and b_l is a vector representing biases of *l*-th hidden layer neurons. The output of the input layer is represented by the vector $y_0 = x$. Neurons in hidden layer as activation function have the hyperbolic tangent sigmoid transfer function:

$$F(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}} \tag{3}$$

The output of the MLP network is as follows

$$\left[f_r\ \ S_{11\min}\right]^T = \mathbf{w}_{H+1}\mathbf{y}_H + \mathbf{b}_{H+1} \tag{4}$$

because the activation function of the neurons in the output layer is linear. In this paper, the notation MLP*H*-N_1-…-N_i-…-N_H is used for this specifically designed MLP_FA neural model where N_i is the number of neurons in the *i*-th hidden layer. For example, the MLP neural network, which consists

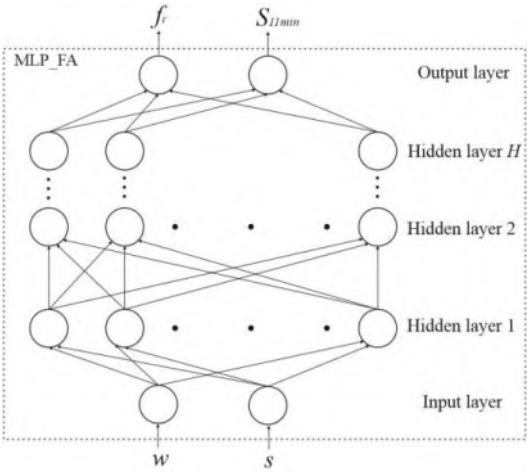


Fig. 6. Arhitecture of MLP_FA network.

of four layers, input, output, and two hidden layers, is denoted as MLP2-10-8. This network includes 10 neurons in the first hidden layer and 8 neurons in the second.

In the MATLAB software environment, the MLP_FA neural network is implemented. Using the EM model of the fractal antenna described in section III, we generated training and test sample sets for the network. A random generator was used to obtain input parameter data in a previously defined range. For each input pair, the software provides calculated output data in the form of the resonant frequency f_r and the value of the S_{11min} parameter at that resonant frequency. The input parameter range $w[mm] \in [0.2 - 2]$ and $s[mm] \in [70 - 250]$ ensures that the resonant frequency f_r=[500-3000] [MHz] is reached, so the output range of the model falls within the UHF range as specified by the International Telecommunication Union (ITU). Training and test sets with 5000 and 300 samples were obtained, respectively. The Levenberg-Marquardt method was used in the model training process with added performance so that the maximum number of iterations is N_{max}=500 while the mean square error of the MSE_{target}

training is set to 10^{-4}. The samples of training and testing sets were normalized, so the input values in this phase were in the range [-1,1]. To evaluate generalization skills, after training each network is tested using samples from the test set. The accuracy and generalization of the network are evaluated using the following test metrics: root mean squared error (RMSE) and Pearson product moment correlation coefficient (r^{PPM}). The root mean square error is shown for both network outputs. The Pearson product moment correlation coefficient was calculated using a theoretical formula that gives the correlation coefficient for n desired outputs, in this case for two desired outputs, namely the resonant frequency and the minimum value of the S_{11} parameter. Five MLP networks with the greatest testing performances after training are shown in Table I.

TABLE I
TESTING RESULTS FOR FIVE MLP_FA NETWORKS WITH THE BEST TEST STATISTICS

MLP networks	RMSE (f_r)	RMSE (S_{11min})	r^{PPM}
MLP2-17-13	0.2015	3.4017	0.9852
MLP2-19-19	0.2062	3.4234	0.9850
MLP2-13-21	0.2218	3.4635	0.9846
MLP2-13-14	0.1936	3.5121	0.9842
MLP2-15-16	0.2172	3.6236	0.9832

MLP2-17-13 is chosen to realize the MLP model of the fractal antenna because, during the testing process, it achieved the highest Pearson product moment correlation coefficient value. Although the root mean square error for resonant frequency of the MLP2-13-14 network is smaller than the selected one, the MLP2-17-13 network was chosen because its correlation coefficient is the highest among the other networks. The scattering diagrams of MLP2-17-13 neural network on the test set for both outputs f_r and S_{11min} are shown in Fig. 7 and Fig 8, respectively.

V. CONCLUSION

The numerical EM approach in antenna modeling produces results with a high degree of accuracy. While using artificial neural networks, it is possible to obtain a neural model with comparable accuracy that, once developed successfully, outperforms other classical numerical EM models in terms of speed. The optimization of this model will be the main focus of future work in order to offer the possibility of fast finding the physical parameters for the desired performance of antenna.

ACKNOWLEDGEMENT

This work was supported by the Ministry of Education, Science and Technological Development of Republic of Serbia (Grant No. 451-03-9/2021-14/200102).

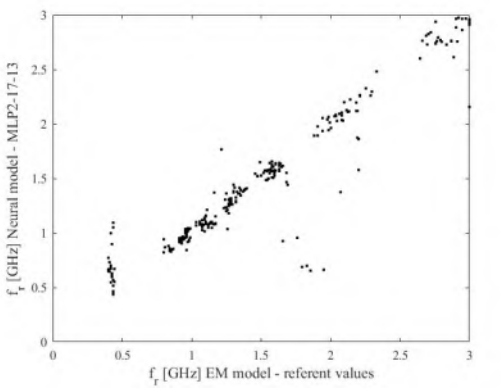


Fig. 7. Scattering diagrams of MLP2-17-13 neural model on the test set for f_r.

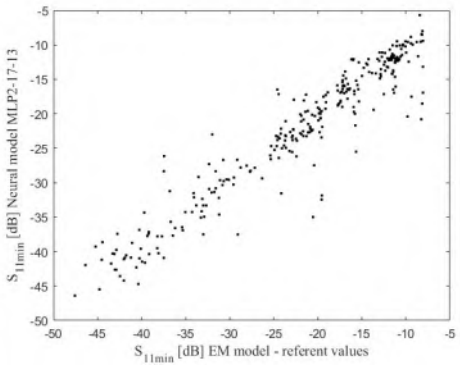


Fig. 8. Scattering diagrams of MLP2-17-13 neural model on the test set for S_{11min}.

REFERENCES

[1] Balanis, C. A., *Antenna Theory: Analysis and Design*, Wiley, New York, 2005.

[2] R. Waterhouse, *Printed Antennas for Wireless Communications*, John Wiley & Sons, Ltd, 2007.

[3] Matthew N. O. Sadik, *Numerical techniques in electromagnetics*, 2nd edition, CRC Press LLC, 2001.

[4] Levent Sevgi, *Electromagnetic Modeling and Simulation*, First Edition. The Institute of Electrical and Electronics Engineers, Inc. Published by John Wiley & Sons, Inc., 2014

[5] J. Joković, T. Dimitrijević, N. Dončov, "Efficient Modeling of a Circular Patch-Ring Antenna Using the Cylindrical TLM Approach", *IEEE Antennas and Wireless Propagation Letters*, Vol. 16, No. 4, pp. 2070-2073, 2017.

[6] S. Haykin, *Neural Networks*, New York, IEEE, 1994.

[7] Q. J. Zhang, K. C. Gupta, Neural networks for RF and microwave

[8] D. Parkash, T. Kundu, P. Kaur, *The RFID technology and its applications: A review*, International Journal of Electronics, Communication & Instrumentation Engineering Research and Development (IJECIERD), Vol.2, Issue 3 Sep 2012 109-120 © TJPRC Pvt. Ltd.

[9] https://www.researchgate.net/publication/304935624_Design_of_sierpinski_gasket_fractal_antenna_with_slits_for_multiband_application

[10] https://www.mathworks.com

Dozimetar jonizujućeg zračenja zasnovan na Arduino mikrokontroleru i senzoru BG51

Miloš Marjanović, Emilija Živanović, Danijel Danković, Goran Ristić

Sadržaj – U ovom radu predstavljen je dozimetar jonizujućeg zračenja zasnovan na Arduino mikrokontroleru i senzoru BG51. Pomoću mikrokontrolera vrši očitavanje poluprovodničkog senzora zračenja, kontrolu zvučne signalizacije u slučaju detekcije zračenja, proračun vrednosti brzine ekvivalentne doze i slanje podataka do računara serijskom komunikacijom. Dozimetar je realizovan na protoploči i iskorišćen za realizaciju edukativnih radionica za učenike srednjih škola.

I. UVOD

Elektromagnetni talasi predstavljaju kombinaciju električnih i magnetnih talasa koji zajedno putuju kroz prostor kao međusobno normalni talasi. Ovo zračenje može da se interpretira i kao talas i kao čestica. Prema dejstvu na materiju zračenje se može podeliti na nejonizujuće i jonizujuće (slika 1) [1]. U nejonizujuće zračenje spadaju radio i mikro talasi i optičko zračenje, koje obuhvata infracrveno, vidljivi deo spektra i ultraljubičasto zračenje. S druge strane, pod jonizujućim zračenjem podrazumeva se snop brzih naelektrisanih ili nenaelektrisanih čestica koje imaju dovoljnu energiju da izazovu jonizaciju sredine kroz koju prolaze. U jonizujuće zračenje spada elektromagnetno zračenje (deo UV zračenja, rendgensko (x-) i γ-zračenje) i čestično zračenje (α i β čestice, protoni, neutroni, …). α-zračenje ima najmanju, a γ – zračenje ima najveću prodornost.



Slika 1. Ilustracija izvora nejonizujućeg i jonizujućeg zračenja [1].

U okruženju izvora zračenja postoji potreba za detekciju i merenje intenziteta zračenja, u cilju praćenja i preduzimanja odgovarajućih akcija. Na primer, u najvećoj nuklearnoj katastrofi ikada, u Černobilu, brojači za merenje intenziteta zračenja nisu bili projektovani da prikažu tako visoke vrednosti zračenja u trenutku nesreće [2]. Neki od dozimetara su Gajger-Milerovi brojači, jonizacione komore, scintilacioni brojači, poluprovodnički brojači, itd [3]. Detektori se dele na aktivne i pasivne u zavisnosti od toga da li pri radu koriste napajanje ili ne; integracione i trenutne u zavisnosti da li sakupljaju informaciju o ukupnoj količini zračenja u toku vremena ili je daju odmah; dok se prema agregatnom stanju radne materije dele na gasne, tečne i čvrste.

U dozimetriji se najčešće koriste sledeće fizičke veličine: ekspoziciona, apsorbovana, ekvivalenta doza [3]. Ekspoziciona doza pokazuje moć jonizujućeg zračenja da jonizuje vazduh kroz koji prolazi, dok je apsorbovana doza mera interakcije jonizujućeg zračenja sa materijom i predstavlja energiju apsorbovanu po jedinici mase ozračene materije na mestu ozračivanja. Jedinica za apsorbovanu dozu je Grej (Gy). Ekvivalentna doza je proizvod srednje apsorbovane doze u tkivu ili organu, koje preda zračenje određene vrste i odgovarajućeg radijacionog težinskog faktora, pri čemu se sumiranje vrši prema vrsti zračenja. Jedinica za ekvivalentnu dozu je Sivert (Sv). Ekvivalentna doza ima manje fundamentalno značenje nego apsorbovana doza, ali ima veći biološki značaj [3].

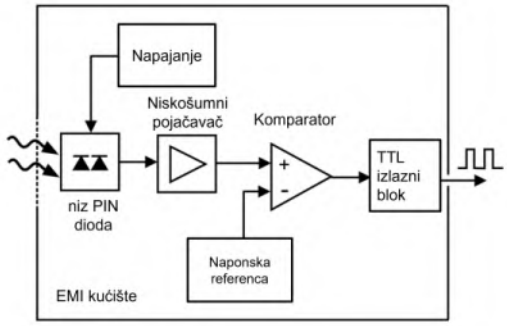
Detektori upravljani mikrokontrolerskom plaftormom otvorenog koda ArduinoUno prisutni su u literaturi [4-7]. Autori su u radu [5] predstavili digitalni detektor zračenja sa Gajger – Milerovom tubom LND7121 i Arduinom. Merenja otkucaja u minuti izvedeno je korišćenjem tajmera mikrokontrolera, na osnovu kojih se izračunava i prikazuje brzina ekvivalentne doze zračenja u μSvh^{-1}. Vrednost faktora konverzije određena iz tehničke dokumetacije poređena je sa vrednošću ovog parametra dobijenog na osnovu kalibracije. Eksperimentalno je pokazano da ovaj detektor ima linearni odziv za doze ispod 3500 μSvh^{-1}. U radu [6] opisan je sistem za monitoring zračenja zasnovan na Gajger – Milerovom brojaču SBM-20 i Arduinu. Sistem određuje akumuliranu dozu zračenja za određeni period i emituje alarmni signal ako se vrednost ekvivalentne doze premaši. Komercijalni proizvod, dodatak za Arduino razvojno okruženje, sa senzorom zračenja RD2007 predstavljen je u [7]. Dodatak ima integrisanu piezo-zujalicu i pinove za eksterno povezivanje sedmosegmentnih displeja za prikaz izmerenih rezultata.

U ovom radu biće prikazan dozimetar jonizujućeg zračenja zasnovan na senzoru BG51 [8] i ArduinoUno razvojnom okruženju. Arduino kontroliše očitavanje senzora, zvučnu signalizaciju u slučaju pojave zračenja, izračunavanje vrednosti brzine ekvivalentne doze i slanje izmerenih vrednosti do računara, putem serijske komunikacije. Sistem je realizovan na protoploči u laboratorijskim uslovima.

M. Marjanović, E. Živanović, D. Danković i G. Ristić su sa Katedre za mikroelektroniku, Elektronski fakultet, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: milos.marjanovic@elfak.ni.ac.rs.

II. SENZOR ZRAČENJA BG51

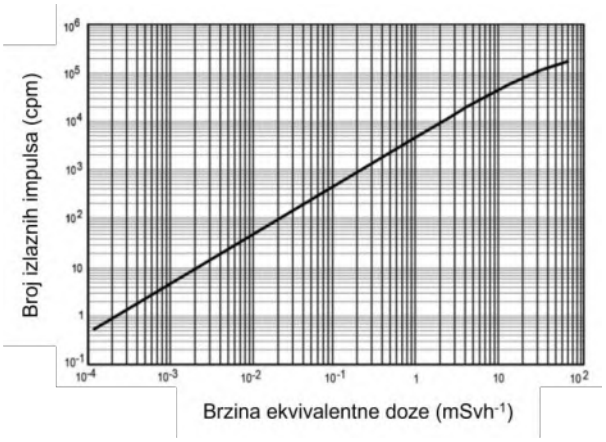
Senzor zračenja BG51 zasnovan je na nizu specijalno projektovanih PIN dioda [8]. PIN dioda je dioda sa širokom intrinstičnom (nedopiranom) oblašću između oblasti p- i n-tipa poluprovodnika. Osiromašena oblast postoji skoro u potpunosti unutar nedopirane oblasti, bez obzira na primenjen napon. Upadno jonizujuće zračenje dovodi do stvaranja nosilaca naelektrisanja unutar strukture diode, pa dolazi do porasta struje tame. Promena struje tame detektuje se i procesira određenim elektronskim kolom. Blok šema senzora prikazana je na slici 2. Ovaj senzor može da detektuje β- zračenje (elektrone), γ- zračenje (fotone) i rentgenske (x-) zrake. Senzor je aktivan, koristi DC napajanje od 2.5 V do 12 V, pri čemu je tipična vrednost struje 25 μA, što ga čini senzorom ultraniske potrošnje (*ultra low power*). Prema agregatnom stanju radne materije spada u čvrste (*solid-state*) poluprovodničke senzore. Opseg merenja brzine doze je od 0.1 μSvh^{-1} do 100 $mSvh^{-1}$, pri čemu je osetljivost senzora 5 $cpm/\mu Svh^{-1}$ (cpm - *counts per minute*). To znači da je senzor osetljiv jonizujuće zračenje čija je energija u opsegu od 50 keV do 2 MeV. Senzor je integrisan u EMI (*Electro Magnetic Interference*) kućište, što ga čini visoko otpornim na uticaje RF i elektrostatičkog polja. TTL blok obezbeđuje digitalni izlaz senzora. Naponski nivo izlaznih impulsa biće jednak naponu napajanja, pri čemu širina impulsa može varirati od 50 μs do 200 μs.



Slika 2. Blok šema senzora zračenja BG51 [8].

Na slici 3 prikazana je zavisnost broja izlaznih impulsa od brzine ekvivalentne doze zračenja za Cs-137 i Co-60 [8]. Brzina doze zračenja izražena je u $mSvh^{-1}$. Može se uočiti linearni odziv senzora BG51 u log-log razmeri, pri čemu proizvođač tvrdi da je odziv takav u širokom temperaturnom opsegu, od -30°C do 60°C.

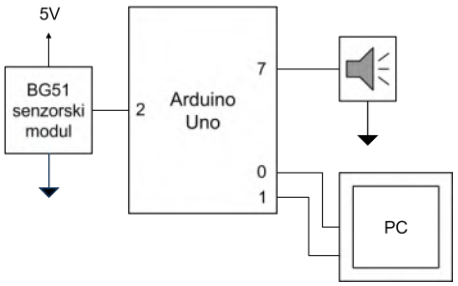
Na osnovu svih navedenih karakteristika, senzor BG51 je pogodan za primenu u različitim sistemima. Senzor se može koristiti u sistemima za detekciju radioaktivnosti u medicinskom okruženju, u sistemima nuklearne zaštite i bezbednosti, kao i gama-senzor za otkrivanje nedozvoljenog nuklearnog materijala, … Senzor BG51 pogodan je i za realizaciju edukativnih praktičnih laboratorijskih eksperimenata. U ovom radu korišćen je senzorski modul *Radiation Click*, firme Mikroelektronika [9], sa opisanim senzorom zračenja. Na *Radiation Click* pločici dostupni su pinovi za napajanje (5V, GND) i INT - izlazni pin na kome se generišu impulsi usled detekcije zračenja.



Slika 3. Zavisnost broja izlaznih impulsa (cpm) od ekvivalentne doze za Cs-137 i Co-60, senzora zračenja BG51 [8].

III. PRAKTIČNA REALIZACIJA

Blok šema povezivanja senzorskog modula i ostalih komponenti sistema na ArduinoUno razvojom okruženju prikazana je na slici 4. Izlazni pin senzorskog modula INT povezan je na pin 2 Arduina koji je predodređen za rad sa hardverskim prekidima. Za zvučnu signalizaciju prilikom detekcije zračenja iskorišćena je piezo-zujalica, povezana na pin 7 Arduina. Brzina ekvivalentne doze prikazuje se na ekranu računara. Veza između mikrokontrolera i računara uspostavljena je preko serijske UART (*Universal Asynchronous Receiver/Transmitter)* komunikacije. Na ArduinoUno ploči za UART se koriste pinovi 0 (Rx) i 1 (Tx).



Slika 4. Blok šema povezivanja komponenti detektora jonizujućeg zračenja sa ArduinoUno razvojnim okruženjem.

Listing koda za Arduino dat je u nastavku. Na početku su definisane labele koje označavaju pinove na koje su povezani senzorski modul i piezo-zujalica. U slučaju promene pina, potrebno je samo na jednom mestu u kodu promeniti broj pina. Sledi deklaracija promenljivih i konstanti korišćenih u programu. Promenljive *brojac*,

vreme, *odmerak*, *staroVreme* i konstante *kalibracioniFaktor* i *maxBrojac* koriste se u proračunu brzine ekvivalentne doze, čija se vrednost čuva u promenljivoj *zracenje*. Vrednost kalibracionog faktora određena je za laboratorijski izvor zračenja, korišćen u ovom eksperimentu. U okviru funkcije *void setup()* koja se izvršava samo jednom nakon što mikrokontroler dobije napajanje, izvršena su inicijalna podešavanja mikrokontrolera. Inicijalna podešavanja podrazumevaju deklaraciju izlaznog digitalnog pina na kome je povezana piezo-zujalica, podešavanje prekidne rutine i inicijalizaciju serijske komunikacije. Prekid (*Interrupt*) se definiše pinom na kome se očekuje signal koji će izazvati prekid (*digitalPinToInterrupt(Pin)*), nazivom prekidne rutine u koju će program ući usled prekida (*nazivISR*) i tipom ivice signala pri kojoj će program ući u prekidnu rutinu (*RISING/FALLING*). U prekidnu rutinu pod nazivom *detekcijaISR()* program će ući kada se na pinu *senzorPin* pojavi rastuća ivica (*RISING*). Serijska komunukacija inicijalizuje se prosleđivanjem bitske brzine (*baud rate*) funkciji *begin*. U ovom slučaju, bitska brzina je 9600 bps.

U prekidnoj rutini, pri nailasku svakog impulsa sa senzorskog modula vrednost bojača će se povećati za jedan i biće poslat impuls za aktiviranje piezo-zujalice u trajanju od 500 ms. U glavnoj programskoj petlji *void loop()* vrši se izračunavanje brzine ekvivalentne doze i slanje izračunate vrednosti na serijski terminal. Za merenje vremena iskorišćena je Arduino funkcija *millis()* koja vraća proteklo vreme u milisekundama, i ta vrednost se upisuje u promenljivu *vreme*. Kada je izbrojano najmanje *maxBrojac* = 10 impulsa koji su stigli sa senzorskog modula, vrši se izračunavanje brzine ekvivalentne doze. Prvo se u promenljivu *odmerak* upiše razlika vrednosti u promenljivoj *vreme* i *staroVreme*. Inicijalno je vrednost promenljive *staroVreme* postavljena na 0. Promenljiva *staroVreme* zatim uzima vrednost promenljive *vreme* i vrši se resetovanje brojača na 0. Na osnovu vrednosti promenljive *odmerak* i definisanih konstanti *maxBrojac*, *kalibracioniFaktor* izračunava se brzina ekvivalentne doze i upisuje u promenljivu *zracenje*. Dobijena vrednost brzine ekvivalentne doze izražena je u μSvh^{-1}. Korišćenjem funkcija *print* i *println* iz klase *Serial* vrši se slanje dobijene vrednosti na serijski terminal.

IV. Rezultati i diskusija

Opisani dozimetar jonizujućeg zračenja, realizovan na protoploči korišćenjem senzorskog modula BG51 i ArduinoUno razvojnog okruženja prikazan je na slici 5. Napajanje razvojnog okruženja i ostalih komponenata sistema dolazi sa računara. Pored zvučne signalizacije, rezultati očitavanja prikazuju se na serijskom terminalu u okviru ArduinoIDE.

Za tastiranje detektora jonizujućeg zračenja iskorišćen je laboratorijski školski izvor radijuma (Ra). Kada izvor zračenja nije u blizini senzora, piezo-zujalica nije aktivna i na serijskom terminalu se očitava vrednost 0 μSvh^{-1}. Kada se školski izvor radijuma prinese blizu senzora, piezo-zujalica se aktivira, čuje se određeni broj zvučnih signala i na serijskom terminalu se očitava izmerena vrednost. Na slici 6 grafički su ilustrovani rezultati merenja brzine ekvivalentne doze tokom vremena. Nakon izlaganja senzora izvoru zračenja očitana vrednost brzine ekvivalente doze bila je oko 400 μSvh^{-1}.

```
#define zujalicaPin 7
#define senzorPin 2

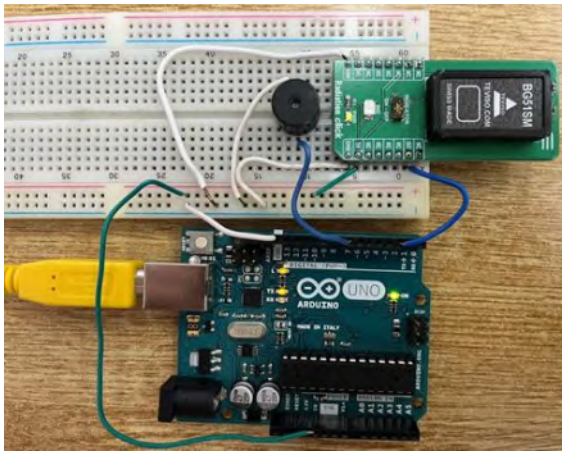
volatile int brojac=0;
unsigned long vreme;
unsigned long odmerak;
unsigned long staroVreme=0;
float zracenje=0.0;
const float kalibracioniFaktor=3.4;
int maxBrojac=10;

void setup() {
  pinMode(zujalicaPin,OUTPUT);
  attachInterrupt(digitalPinToInterrupt(senzorPin),
detekcijaISR,RISING);
  Serial.begin(9600);
}

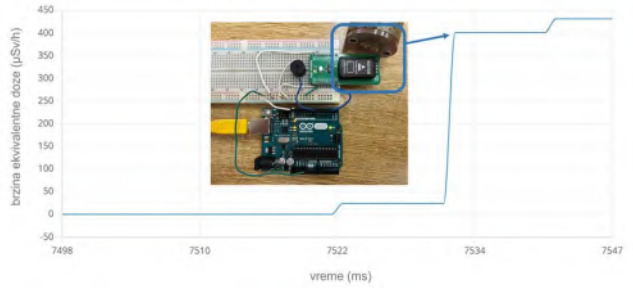
void loop() {
  vreme=millis();
  if(brojac>=maxBrojac){
    odmerak=vreme-staroVreme;
    staroVreme=vreme;
    brojac=0;
    zracenje=float(maxBrojac)*60.0*1000.0/float(odmera
k)/kalibracioniFaktor;

  }
  Serial.print("Brzina ekvivalentne doze je: ");
  Serial.print(zracenje);
  Serial.println(" uSv/h.");
}

void detekcijaISR(){
  brojac++;
  digitalWrite(zujalicaPin,HIGH);
  delay(500);
  digitalWrite(zujalicaPin,LOW);
}
```



Slika 5. Realizacija detektora jonizujućeg zračenja na protoploči.



Slika 6. Grafički prikaz rezultata merenja nakon izlaganja senzora laboratorijskom izvoru Ra.

Opisani prototip dozimetra jonizujućeg zračenja korišćen je u realizaciji pet radionica za učenike srednjih škola, koje su održane tokom aprila i maja 2022. godine. Na svakoj od radionica učenici su čuli kratko predavanje o zračenju i praktično su realizovali opisani dozimetar. U tabeli I dat je spisak škola i broj učenika koji su učestvovali u radionicama. Radionice su organizovane u okviru promocije Elektronskog fakulteta, projekta Elicsir [10] i deo su aktivnosti na projektu "Let STEM visit again IEEESTEC" [11], koji se realizuje u okviru 2022 IEEE Pre-University STEM Portal Grant programa. Posle svake od radionica organizovan je kviz, a učenici sa najvećim brojem ostvarenih poena pozvani su u finalni krug kviza koji je održan 24. maja 2022. godine. U kvizovima je učestvovalo ukupno 39 učenika iz svih pomenutih škola, a prosečan broj poena bio je 13.5/20. U finalnom kvizu učestvovalo je 10 odabranih učenika i prosečan broj poena bio je 15.6/20. Rezultati kviza ukazuju na veliku zainteresovanost učesnika radionica za ovu oblast istraživanja. Na slici 7 prikazane su fotografije sa održanih radionica i finalnog kviza.

V. Zaključak

U ovom radu opisan je prototip dozimetar jonizujućeg zračenja realizovan korišćenjem senzora BG51 i ArduinoUno razvojnog okruženja. Pored zvučne signalizacije, omogućen je prikaz brzine ekvivalentne doze na serijskom terminalu računara. Opisan sistem korišćen je u edukativne svrhe prilikom realizacije radionica za učenike 5 srednjih škola iz Niša i okoline, sa ciljem upoznavanja učenika sa jonizujućim zračenjem i mogućnošću njegovog merenja.

Zahvalnica

Ovaj rad podržan je od strane *European Union's Horizon 2020 research and innovation programme, grant agreement No.* 857558, kao i *2022 IEEE Pre-University STEM Portal Grant Program*-a.

Autori se zahvaljuju svim učesnicima radionica na interesovanju za programe koji se realizuju na Elektronskom fakultetu u Nišu.



Slika 7. Fotografije sa radionica za učenike srednjih škola na kojima je predstavljen i realizovan opisani dozimetar jonizujućeg zračenja.

TABELA I
SPISAK SREDNJIH ŠKOLA I BROJ UČENIKA NA REALIZOVANIM RADIONICAMA

Škola	Broj učenika
ETŠ "Mija Stanimirović" Niš	11
Gimnazija "Bora Stanković" Niš	13
Mašinsko-elektrotehnička škola Paraćin	70
Gimnazija "Svetozar Marković" Niš	5
ETŠ "Nikola Tesla" Niš	12

Literatura

[1] Elektromagnetno zračenje. Dostupno na: https://perunprotect.rs/elektromagnetno-zracenje/
[2] Chernobyl: Facts about the world's worst nuclear disaster, Dostupno na: https://www.livescience.com/39961-chernobyl.html
[3] Goran Ristić, "Dozimetrija i dozimetri, predavanja", Elektronski fakultet Niš, 2016.
[4] Stefan Ilić, Aleksandar Jevtić, Nikola Đikić, Goran Ristić, "Smart Geiger Muller counter", Seventh International Conference on Radiation in Various Fields of Research (RAD 2019), Herceg Novi, Montenegro, 2019.
[5] Nur Aira Abd Rahman et al, "Arduino based radiation survey meter", in *AIP Conference Proceedings*, vol. 1704, 030012, 2016.
[6] A. Holovatyy, et al. "Development of Microcontroller-Based System for Background Radiation Monitoring", *Sensors*, vol. 20, no. 24: 7322, 2020.
[7] The Radiation Measurement Shield „Tino" for the Arduino and Arduino-Compatible Microcontrollers. Dostupno na: http://opengeiger.de/TinoKitDescription.pdf
[8] Radiation Sensor BG51, datasheet. Dostupno na: https://www.teviso.com/file/pdf/bg51-data-specification.pdf
[9] Radiation Click. Dostupno na: https://www.mikroe.com/radiation-click
[10] ELICSIR project. Dostupno na: https://elicsir-project.eu/index.php
[11] Let STEM visit again IEEESTEC. Dostupno na: http://ieee.elfak.ni.ac.rs/stem-visits-ieeestec/

Detektor vlažnosti zemljišta zasnovan na Arduinu

Andrija Đorđević, Filip Filić, Mateja Nikolić

Sadržaj – U ovom radu predstavljen je detektor vlažnosti zemljišta zasnovan na Arduinu. Korišćenjem senzora vlažnosti vrši se očitavanje nivoa vlažnosti zemljišta i uključuje se piezo-zujalica ako je nivo vlažnosti ispod zadatog nivoa.

I. UVOD

Vlažnost zemljišta je količina vode u porama zemlje ili na njenoj površini. Sadržaj vlage u zemljištu zavisi od faktora kao što su vremenske prilike, vrsta zemljišta i biljke. Vlažnost je parametar od vitalnog značaja za praćenje poljoprivrednih aktivnosti. Biljke mogu stradati ako postoji nedostatak ili previše vode. Koreni biljaka prvo upijaju vodu, pa njihovo stanje direktno zavisi od njene količine i aeracije (slika 1). Optimalni opseg vlažnosti zemljišta za većinu biljnih vrsta je između 20 % i 60 % [1]. Zadržavanje vode poboljšavaju finija tekstura zemljišta sa više pora, veća koncentracija organske materije i niža temperatura.



Slika 1. Izgled biljaka u zavisnosti od vlažnosti zemljišta.

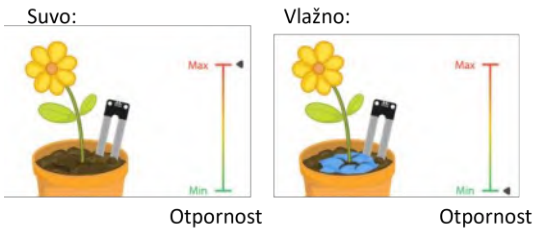
Postoji nekoliko metoda za merenje vlažnosti zemljišta: gravimetrijsko (ili direktno) merenje, analiza pomoću senzora i daljinsko ispitivanje [1]. Gravimetrijska detekcija vlage u zemljištu je metoda koja podrazumeva izdvajanje vode iz uzorka isparavanjem, ispiranjem i hemijskom reakcijom. Gravimetrijska vlažnost zemljišta se izračunava na osnovu merenja razlike između težine vlažnog i suvog uzorka. Postoje različiti senzori za merenje vlažnosti zemljišta, kao što su senzori za merenje zapreminskog sadržaja vode, senzori za merenje vodenog potencijala zemljišta, itd. Daljinsko ispitivanje podrazumeva satelitsko praćenje vlažnosti velikih površina zemlje.

A. Đorđević, F. Filić, M. Nikolić su učenici III razreda gimnazije "Bora Stanković", Vožda Karađorđa 27, 18000 Niš, Srbija, E-mail: andrijadjordjevic18@gmail.com, ficafilic2@gmail.com, mateja.nikolic.005@gmail.com.

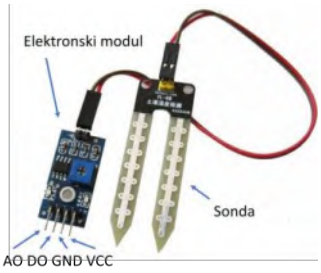
U ovom radu biće opisan detektor vlažnosti zemljišta realizovan pomoću Arduino mikrokontrolera. Merenje nivoa vlažnosti vrši se korišćenjem senzora vlažnosti zemljišta. Usled detekcije smanjene vlažnosti korisnik se alarmira aktiviranjem piezo-zujalice da treba da izvrši zalivanje biljke. Detektor je realizovan na protoploči i testiran u laboratorijskim uslovima.

II. SENZOR VLAŽNOSTI ZEMLJIŠTA

Senzor vlažnosti zemljišta radi na jednostavan način [2-4]. Sonda u obliku viljuške sa dva provodnika ponaša se kao promenljivi otpornik čija se otpornost menja sa sadržajem vlage u zemljištu. Ova otpornost se menja obrnuto srazmerno sa vlagom u zemljištu: što je više vode u zemljištu, to je bolja provodnost i manja je otpornost (slika 2). Senzor generiše izlazni napon srazmeran otpornošću, čijim merenjem se može odrediti nivo vlage u zemljištu. Korišćen senzor vlažnosti zemljišta sastoji se od dva dela: sonde i elektronskog modula. Sonda u obliku viljuške sa dva izložena provodnika se ubacuje u zemljište. Elektronski modul povezuje sondu sa Arduino mikrokontrolerom. Modul generiše izlazni napon na osnovu otpornosti sonde na pinu za analogni izlaz (AO). Isti signal se dovodi u komparator LM393, koji ga digitalizuje i čini dostupnim na pinu za digitalni izlaz (DO). Modul uključuje i potenciometar za podešavanje osetljivosti digitalnog izlaza. U ovom radu korišćen je analogni izlaz senzora. Izgled senzora vlažnosti zemljišta sa obeleženim pinovima prikazan je na slici 3. AO (analogni izlaz) generiše analogni izlazni napon proporcionalan nivou vlažnosti u zemljištu, tako da viši nivo rezultira višim naponom. DO (digitalni izlaz) će biti u stanju logičke nule kada nivo vlažnosti pređe graničnu vrednost podešenu potenciometrom. VCC napaja senzor. Senzor se može napajati naponom od 3.3 V do 5 V. Vrednost napona na analognom izlazu zavisiće od vrednosti napona napajanja. GND je pin za masu.



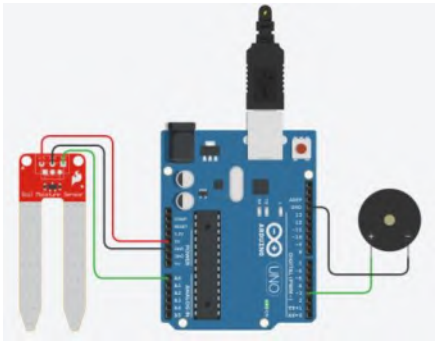
Slika 2. Princip rada senzora vlažnosti zemljišta.



Slika 3. Senzor vlažnosti zemljišta: sonda i elektronski modul.

III. PRAKTIČNA REALIZACIJA DETEKTORA

Detektor vlažnosti zemljišta praktično je realizovan korišćenjem ArduinoUno razvojnog okruženja, senzora i piezo-zujalice, povezanih na protoploči. Blok šema povezivanja komponenata detektora prikazana je na slici 4. Glavni deo detektora je Arduino mikrokontroler, ATMega 328P. Analogni izlaz senzorskog modula povezan je na analogni ulazni pin Arduina A0. Analogni signal sa senzora prevodi se u digitalni domen korišćenjem 10-bitnog analogno-digitalnog konvertora mikrokontrolera. Za napajanje senzorskog modula obezbeđeno je 5 V DC sa ArduinoUno razvojnog okruženja. Piezo-zujalica povezana je na digitalni izlazni pin Arduina 3. Za zvučnu signalizaciju, tj. aktiviranje piezo-zujalice koriste se PWM (*Pulse Width Modulation*) signali, koje je moguće proslediti iz Arduina preko pinova označenih tildom "~".



Slika 4. Blok šema povezivanja senzora vlažnosti zemljišta i piezo-zujalice na ArduinoUno razvojno okruženje.

Programski kod za Arduino mikrokontroler dat je u nastavku. Na početku se definisani nazivi pinova. U okviru funkcije *void setup()* inicijalizovana je serijska komunikacija između mikrokontrolera i računara, pozivanjem funkcije *begin* iz klase *Serial*. Ovoj funkciji prosleđena je vrednost bitske brzine od 9600 bps. U okviru beskonačne programske petlje *void loop()* vrši se očitavanje senzora korišćenjem funkcije *analogRead()*. U zavisnosti od vlažnosti zemljišta biće očitana celobrojna vrednost u opsegu od 0 do 1023. Korišćenjem funkcija *print* i *println* iz klase *Serial* vrši se štampanje izmerene vrednosti na serijskom monitoru u okviru Arduino IDE. Ako je očitana vrednost senzora manja od 1000 piezo-zujalica neće biti aktivna, što je definisano Arduino funkcijom *noTone()*. Vrednost 1000 određena je eksperimentalno, u procesu kalibracije detektora. U suprotnom, kada je vrednost sa senzora veća od 1000, aktiviraće se piezo-zujalica, koja treba da alarmira korisnika da je zemljište suvo. Piezo-zujalica emitovaće ton frekvencije 300 Hz, sve dok se senzorom ne očita vrednost manja od 1000, tj. dok korisnik ne izvrši zalivanje.

```
#define senzorPin A0
#define zujalicaPin 3
void setup() {
Serial.begin(9600);
}
void loop() {
int senzor=analogRead(senzorPin);
Serial.print("Ocitana vrednost je: ");
Serial.println(senzor);
delay(100);
if(senzor<1000){
  noTone(zujalicaPin);
}
else{
  tone(zujalicaPin,300);
}
}
```

IV. ZAKLJUČAK

U ovom radu prikazana je praktična realizacija jednostavnog detektora vlažnosti zemljišta. Detektor se sastoji iz senzora, Arduina i piezo-zujalice. Kada se pomoću senzora očita da je zemljište suvo, aktivira se piezo-zujalica koja alarmira korisnika da izvrši zalivanje. Ovaj projekat se može realizovati na štampanoj ploči, pri čemu bi se koristilo baterijsko napajanje. Projekat se može unaprediti dodavanjem LCD ekrana na kome bi se prikazivao nivo vlažnosti zemljišta i drugih modula koji bi omogućili slanje ovih informacije do drugih uređaja.

ZAHVALNICA

Ovaj rad nastao je kao rezultat realizacije projekta „Let STEM visit again IEEESTEC", finansiranog u okviru 2022 IEEE Pre-University STEM Portal Grant Program-a. Autori se zahvaljuju svim saradnicima sa Katedre za mikroelektroniku, Elektronskog fakulteta u Nišu, a posebno mentoru na ovom radu Milošu Marjanoviću.

LITERATURA

[1] Soil Moisture: How To Measure & Monitor Its Level. Dostupno na: https://eos.com/blog/soil-moisture/.
[2] How Soil Moisture Sensor Works and Interface it with Arduino. Dostupno na: https://lastminuteengineers.com/soil-moisture-sensor-arduino-tutorial/
[3] How to Use a Soil Moisture Sensor. Dostupno na: https://create.arduino.cc/projecthub/MisterBotBreak/how-to-use-a-soil-moisture-sensor-ce769b
[4] Complete Guide to Use Soil Moisture Sensor. Dostupno na: https://create.arduino.cc/projecthub/electropeak/complete-guide-to-use-soil-moisture-sensor-w-examples-756b1f

Mašinsko učenje u okviru kursa Informacioni sistemi korišćenjem *Weka* API za Javu

Nenad Petrović

Sadržaj – Mašinsko učenje je prepoznato kao jedan od krucijalnih faktora koji omogućava mnoge inovativne scenarije upotrebe i servise u okviru modernih informacionih sistema, analizom velike količine podatake i vršenjem predikcija vezanih za raznolike relevantne aspekte. U ovom radu, fokus je na implementaciji servisa mašinskog učenja u okviru kursa na trećoj godini smera Informatike i računarstvo Elektronskog fakulteta u Nišu. Kao ishod, prikazana je implementacija dve studije slučaja: 1) klasifikacija – detekcija odlazećih korisnika mobilnog operatera 2) regresija – predikcija potrošnje električne energije. Što se tehnologije tiče, koristi se *Weka* API za nadgledano mašinsko učenje u okviru Java *Enterprise Edition* (JEE) aplikacije.

I. Uvod

U današnje vreme, informacioni sistemi generišu ogromnu količinu podataka o različitim relevantnim aspektima – korisinicima, objektima, događajima. Prema tome, sam izlaz informacionih sistema daje prostora za dalje iskorišćavanje [1, 2]. Skladišteni podaci se mogu dalje analizirati sa ciljem detekcije pojedinih šablona ili obuke prediktivnih modela radi donošenja bitnih zaključaka. Odluke donete na osnovu predikcija mogu potencijalno doneti povećanje profita, smanjenje troškova ili pomoći u postizanju drugih benefita ili ostvarivanju poslovnih ciljeva organizacije od interesa [2]. Prema tome, servisi mašinskog učenja polako postaju neizostavan deo modernih informacionih sistema, što dovodi do odomaćivanja takozvanog sloja poslovne inteligencije u maltene svim oblastima primene – od trgovine, preko zabave do energetike, zdravstva i edukacije. Sl. 1 daje ilustraciju toka ključnih aktivnosti u informacionim sistemima koji se oslanjaju na rezultate predikcija metoda mašinskog učenja [2].



Slika 1. Iskorišćavanje predikcija u informacionim sistemima.

Ovaj rad prikazuje kako se *Weka* API [3] za nadgledano mašinsko učenje u Java programskom jeziku može uklopiti u arhitekturu Java *Enterprise Edition* (JEE) [4] projekta za višeslojne, skalabilne, bezbedne i pouzdane poslovne aplikacije iz predmeta Informacioni sistemi [1] na Elektronskom fakultetu, Univerzitet u Nišu. Predloženi pristup je evaluiran na javno dostupnim skupovima podataka, pokrivajući dve najčešće tehnike: 1) klasifikacija – studija slučaja detekcije odlazećih korisnika operatera mobilne telefornije 2) regresija – predviđanje potrošnje električne energije domaćinstva.

N. Petrović je student doktorskih studija i asistent na Katedri za računarstvo i informatiku, Elektronski fakultet, Univerzitet u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: nenad.petrovic@elfak.ni.ac.rs.

II. Osnove

A. Nadgledano mašinsko učenje

Nadgledano mašinsko učenje se odnosi na skup metoda i algoritama koji omogućavaju predviđanje ciljanih vrednosti posmatranjem ogromne količine instanci sa unapred poznatim tačnim ishodom [5]. Obično se parametri metoda mašinskog učenja dele u sledeće kategorije: 1) hiperparametri – upravljaju samim procesom obuke 2) težinski parametri – vrednosti određene kao ishod obuke (treninga). U ovom kontekstu, trening skup podataka predstavlja kolekciju uzoraka na osnovu kojih se tokom obuke podešavaju hiperparametri prediktivnog modela. Sa druge strane, skup test podataka se iskorišćava za evaluaciju prethodno utreniranog modela. Osim toga, imamo i funkciju greške, čiji je cilj da proceni koliko je predviđena vrednost daleko od očekivane, što se dalje koristi u procesu obuke za štelovanje težinskih parametara, proporcionalno veličini datih hiperaparametara. Dve najčešće tehnike nadgledanog mašinskog učenja su: 1) klasifikacija – predviđanje labele ili kategoričkog ishoda 2) regresija – određivanje kontinualnih brojnih vrednosti.

Nakon obuke prediktivnog modela, kvalitet predikcija koje on daje se može proceniti upotrebom različitih vrsta metrika. Najčešće metrike za dve pomenute tehnike su date u Tabeli I, dok oznake u formulama imaju sledeće značenje: TP – pravi pozitivni, predikcije čiji se ishod slaže sa očekivanum da pripada nekoj kategoriji; FP – lažno pozitivan, predikcija kaže da pripada nekoj kategoriji, a ne pripada; TN – pravi negativan, predikcije gde se predviđanje i očekivani rezultat da ne pripada kategoriji slažu; FN – lažno negativan, predviđeno da ne pripada kategoriji, a zapravo se očekuje da pripada; predikcija – vrednost predviđena od strane modela mašinskog učenja; očekivano – vrednost izmerena u stvarnom svetu.

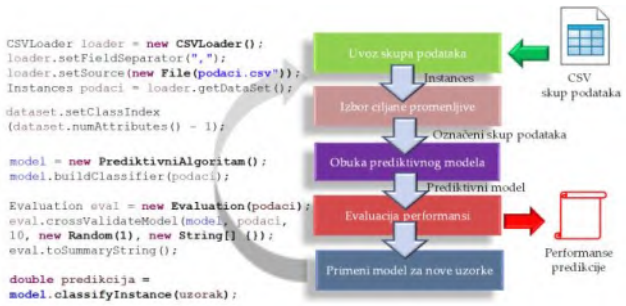
B. Weka API u Java programskom jeziku

Što se tiče implementacije elemenata mašinskog učenja u okviru projekta realizovanog kao deo predmeta Informacioni sistemi, oslanjamo se na *Waikato Environment for Knowledge Analysis* (*Weka*) [3]. Inicijalno, ovaj skup alata za prediktivnu analizu podataka je nastao 1993. sa ciljem obrade podataka iz oblasti poljoprivrede, a kasnije, od 1997. godine se orijentiše na Java ekosistem. *Weka* predstavlja kolekciju besplatnih alata, čiji je cilj analiza i vizuelizacija podataka po modelu GNU javne licence. Osim vizuelnih alata i pristupa SQL bazama podataka, takođe pruža i programski interfejs (API) u Javi [3] i obuhvata veliki katalog implementacija prediktivnih algoritama, kao što su oni zasnovani na stablima odlučivanja (*J48*, *DecisionTable*, *DecisionStump*), zatim k-najbližih suseda (*IBk*) i varijante linearne regresije. [5]. Odlučujemo se da koristimo *Weka* API zbog njegove intuitivnosti, portabilnosti (zahvaljujući Javi), besplatne upotrebe u sinergiji sa interfejsom visokog nivoa ka različitim algoritimima i tehnikama mašisnkog učenja.

TABELA I
METRIKE PERFORMANSI MODELA NADGLEDANOG MAŠINSKOG UČENJA

Tehnika	Metrika	Formula
Klasifikacija	Preciznost (*Precision*)	TP/(TP+FP)
	Opoziv (*Recall*)	TP/(TP+FN)
	Tačnost (*Accuracy*)	(TP+TN)/ (TP+FP+TN+FN)
	F1-score	2*(Precision*Recall)/ (Precision+Recall)
Regresija	Relativna greška	|(predikcija-očekivano)/očekivano| [%]
	Apsolutna greška	|očekivano-predikcija|
	Koren srednje kvadratne greške	sqrt($\sum_{i=1}^{N}$(predikcija-očekivano)^2/N)

Sl. 2 prikazuje tok procesa nadgledanog mašinskg učenja upotrebom *Weka* API za Java programski jezik. Prvo se skup podataka za obuku učitava u Comma Separated Values (CSV) formatu. Dalje, prethodno učitani skup podataka se konvertuje u pogodnu formu, koja predstavlja skup instanci uzoraka. Nakon toga, potrebno je razvrstati polja uzoraka u ulazne (nezavisne) i izlazne (ciljana, prediktovana vrednost). Posle toga, skup podataka se u označenom obliku prosleđuje kao ulazni argument metode koja vrši obuku modela. Po završetku procesa obuke, vrši se evaluacija performansi prediktivnog modela. Eventualno, moguće je modifikovati parametre prediktivnog modela ili procesirati ulaze samog skupa podataka, sa ciljem da se dobije bolji kvalitet predikcija. Konačno, kada budemo zadovoljni performansama, prethodno utrenirani model se može iskoristiti da predviđa vrednosti na osnovu potpuno novih uzoraka, koji nisu bili prisutni u skupu podataka za obuku. Sa druge strane, Tabela II prikazuje ključne komponente *Weka* API za implementaciju prediktivnih modela u Javi.



Slika 2. Proces nadgledanog mašinskog učenja uz pomoć *Weka* API u Java programskom jeziku.

TABELA II
PREGLED KLJUČNIH MODULA ZA NADGLEDANO MAINSKO UČENJE UZ POMOĆ *WEKA* API U JAVI

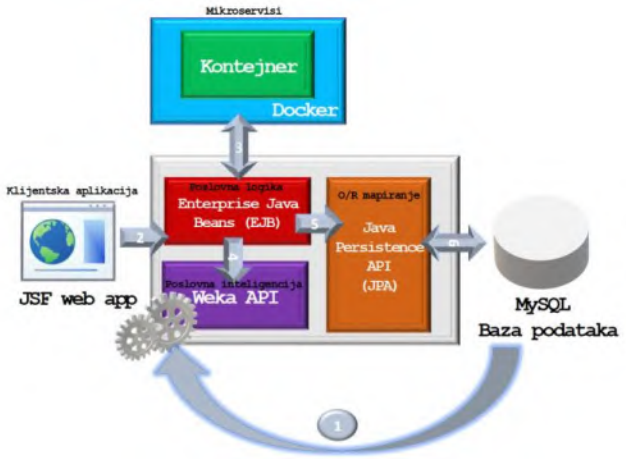
Modul	Opis	Relevantne metode
weka.core. Instances	Apstrakcija skupa podataka	dataset. setClassIndex(indeks_kolone): izbor ciljane (izlazne, zavisne) promenljive koju model prediktuje
weka.core. Instance	Pojedinačni uzorak iz skupa podataka	
weka..core. converters. CSVLoader	Pomoćna klasa za učitavanje skupa podataka iz CSV fajla	loader.setFieldSeparator(separator): podešavanje vrednosti separatora
		loader.setSource(new File("dataset.csv")): izbor CSV fajla sa diska
		Instances podaci = loader.getDataSet(): pribavljanje svih instanci iz skupa podataka
weka. classifier. functions.*	Kolekcija različitih algoritama nadgledanog mašinskog učenja	model. buildClassifier(trening_podaci): obuka prediktivnog modela nad datim skupom instanci
		model.classifyInstance(uzorak): predikcija za pojedinačni uzorak iz skupa podataka
weka. classifiers. evaluation. Evaluaton	Objekat koji vrši procenu performansi predikcije datog modela	Evaluation eval = new Evaluation(podaci): kreiranje novog evaluatora za dati skup instanci
		eval.crossValidateModel(model, podaci, broj_koraka, new Random(1), new String[] {}): evaluacija modela za dati skup podataka
		eval.toSummaryString(): štampanje rezimea evaluacije prediktivnih performansi modela

III. IMPLEMENTACIJA

A. Arhitektura aplikacije

Sl. 3 ilustruje arhitekturu praktičnog dela projekta iz predmeta Informacioni sistemi na Elektronskom fakultetu u Nišu, koji se odnosi implementaciju JEE aplikacija. Kao što se može videti, korisnik zadaje unose preko JSF klijentske aplikacije u web pretraživaču, a HTTP zahtevi se prosleđuju serveru, na kome se izvršava poslovna logika. Rezultati obrade mogu koristiti podatke iz baze kao ulaz, a

i sam ishod se može trajno skladištiti. Prilikom generisanja odgovora, poslovni sloj na serveru može dalje interagovati sa mikroservisima za pojedine operacije. Konačno, rezultat obrade se korisniku vraća kao HTTP odgovor i prikazuje odgovarajuća HTML stranica.



Slika 3. Arhitektura JEE aplikacije iz predmeta Informacioni sistemi: 1 – Obuka prediktivnog modela; 2 – Unos korisnika; 3 – 3 – Interakcija sa mikroservisima; 4 - Poziv prediktivnih servisa; 5 – Manipulacija poslovnim objektima 6 – Pribavljanje ulaznih podataka i skladištenje rezultata.

Dalje, sl. 4 prikazuje UML klasni dijagram, na kome se mogu izdvojiti različiti slojevi JEE aplikacije: 1) JPA entiteti – enkapsuliraju neophodne informacije o relevantnim objektima, preslikavaju se na tabele u bazi podataka uz pomoć *Hibernate* objektno-relacionog mapera; 2) EJB servisi poslovne logike – obrada podataka, logika i izračunavanja 3) JSF *managed beans* kontroler – prosleđuju unose korisnika sa web strana ka odgovarajućim servisima poslovne logike 4) *Weka* prediktori – klase koje predstavljaju interfejs vrlo visokog nivoa ka prediktivnim mogućnostima klasifikacije ili regresije nad zadatim skupom podataka. Kod koji se tiče implementacije dela za mašinsko učenje je dostupan u okviru javnog *GitHub* repozitorijuma: https://github.com/penenadpi/is2022_java_weka.



Slika 4. UML klasni dijagram JEE aplikacije koja se oslanja na *Weka* servise mašinskog učenja u okviru sloja poslovne logike.

Isečak koda *Weka* prediktora za klasifikaciju i regresiju je dat na sl. 5.

```
public WekaPredictor(String dataset_path)
{

     CSVLoader loader = new CSVLoader();
     loader.setFieldSeparator(",");
     loader.setSource(new File("podaci.csv"));
     Instances data = loader.getDataSet();

     // Obuka regresionog modela
     data.setClassIndex(data.numAttributes() - 2);
     recovery_predictor = new LinearRegression();
     model.buildClassifier(data);

     // Obuka klasifikatora
     data.setClassIndex(data.numAttributes() - 1);
     death_predictor = new DecisionTable();
     model.buildClassifier(data);

     //Evaluacija
     Evaluation eval = new Evaluation(data);

     eval.crossValidateModel(recovery_predictor, data, 10,
                                    new Random(1), new String[] {});
     System.out.println(eval.toSummaryString());
}
```

Slika 5. Isečak Java koda *Weka* prediktivnog modula za obuku modela.

Dodatno, sl. 6 daje isečak koda kojim se objašnjava primena *Weka* prediktivnog modula u Java EE aplikaciji. Prvo je potrebno utrenirati prediktivni model na relevantnom skupu podataka (obično CSV fajl). Zatim, instancira se servis poslovne logike, a uz pomoć njega kasnije može pribaviti zapis o nekom relevantnom entitetu iz baze podataka. Nakon toga, ovaj zapis se konvertuje u instancu (specifična reprezentacija ulaza u Weka prediktor), koja se dalje prosleđuje kao ulaz prediktivnom modelu. Konačno, prediktivni model daje rezultat, koji se dalje može iskoristiti za odlučivanje u sistemu od strane sloja poslovne logike. Sl. 6 daje primer kako se predikcija klasifikatora o tome da li korisnik napušta operatera dalje koristi za odluku o najboljoj ponudi koja može biti ponuđena kao preventivna mera za sprečavanje napuštanja.

```
//Obuka prediktivnog modela
WekaPredictor predictor=new WekaPredictor("dataset.csv");
BusinessLogicServiceImpl bl_service=new BusinessLogiServiceImpl();
JPAEntity entity = bl_service.getEntityById("e1");
entity_sample = predictor.entity_to_sample(entity)
//Poziv prediktivnog servisa za dati uzorak
int churn=predictor.predict_class(entity_sample);
int plan=offer_plan(churn);
```

Slika 6. Isečak Java koda koji prikazuje primenu Weka prediktivnog modula u okviru sloja poslovne logike.

B. Detekcija odlazećih korisnika u oblasti telekomunikacionih usluga

Predikcija da li će korisnik napustiti mobilnog operatera i eventualno preći kod drugog, predstavlja jedno od ključnih pitanja u odnosu sa potrošačima, što se tiče oblasti telekomunikacionih usluga [6]. Proaktivna detekcija potencijalno odlazećih korisnika može poslužiti kao osnova za preduzimanje koraka za zadržavanje ovakvih korisnika,

kao što se popusti, mobilni uređaji i pogodnosti. U ovoj studiji slučaja se koristi pristup binarne klasifikacije, pri čemu je ishod kategorički: da (1) ili ne (0). Korišćen je otvoren, javni skup podataka sa Kaggle servisa [7], čija je struktura rezimirana u Tabeli III.

TABELA III
SKUP PODATAKA ZA DETEKCIJU ODLAZEĆIH KORISNIKA

Promenljiva	Opis
Id korisnika	Jedinstveni pretplatnički broj kod operatera
Pol	Ženski ili muški
Penzioner	Spada u starije korisnike – da ili ne
Ima partnera	Da li je u braku ili ne
Izdržava drugo lice	Da li ima dece ili izdržava drugo lice
Broj meseci kod operatera	Koliko dugo koristi usluge
Telefonske usluge	Da li koristi mobilnu telefoniju
Više brojeva	Jedan ili više
Internet usluge	Koristi ili ne
Tip ugovora	Godina, dve, prepaid
Prosečni mesečni troškovi	Koliko mesečno u proseku iznosi račun
Odlazeći korisnik	Da li namerava da napusti operatera

C. Predviđanje energetske efikasnosti domaćinstva

Predviđanje potrošnje električne energije za domaćinstva tokom različitih sezona je od ključne važnosti za efikasnu distribuciju u pametnim mrežama. Druga studija slučaja ima za cilj da predvidi koliko energije troši domaćinstvo zimi za grejanje, a leti za rashladu, zavisno od karakteristika prostorije. U ovoj studiji slučaja se problem tretira kao regresija, s obzirom da predviđamo kontinulanu numeričku vrednost. Korišćen je otvoren, javni skup podataka za 768 domaćinstva [9], čija je struktura rezimirana u Tabeli IV.

TABELA IV
SKUP PODATAKA ZA PREDVIĐANJE POTROŠNJE ENERGIJE

Promenljiva	Opis
Relativna kompaktnost	Odnos zapremine i površine prostorije
Veličina stana	Površina u m^2
Veličina zidova	
Zastakljene površine	
Veličina krova	
Visina kuće	Koliko je visoka građevina, u m
Orijentacija	Koliko dugo koristi usluge
Distribucija staklenih površina	Jedan ili više
Opterećenje grejanja	Koliko domaćinstvo troši na grejanje zimi
Opterećenje hlađenja	Letnja potrošnja

IV. EVALUACIJA

Tabela V prikazuje rezultate evaluacije iz dva ugla: kvalitet predikcija na trening skupu (procenat tačno svrstanih uzoraka za klasifikaciju, srednja relativna greška za regresiju) i vreme potrebno za obuku. Što se klasifikacije tiče, najbolje rezultate daje tablica odlučivanja, ali uz najduže vreme procesiranja. Za svrhu evaluacije je korišćen *Acer Nitro* 5 laptop, sa sledećim karakteristikama: Intel Core i5-10300H quad-core CPU 2.50GHz, 8GB DDR4 RAM i 512 GB M1 SSD disk.

TABELA V
SKUP PODATAKA ZA PREDVIĐANJE POTROŠNJE ENERGIJE

Problem	Algoritam	Performanse	Vreme obuke [s]
Klasifikacija	J48	78.62% Accuracy	4.028
	iBk	71.92% Accuracy	3.66
	Decision Table	79.09% Accuracy	13.639
	Decision Stump	73.46% Accuracy	1.055
Regresija	Linear Regression	22.86% MRE 29.282% RMSE	1.087

V. ZAKLJUČAK

U ovom radu je prikazana primena *Weka* API za implementaciju prediktivnih modula u okviru projekta iz predmeta Informacioni sistemi na trećoj godini smera Računarstvo i informatika, Elektronski fakultet u Nišu. Na osnovu predstavljenih rezultata, možemo zaključiti da ova tehnologija ne samo da je prihvatljiva za početnike u oblasti mašinskog učenja, već i daje praktično upotrebljive rezultate za relativno kratko vreme procesiranja (red veličine 10 sekundi u našim primerima), koji se mogu dalje iskoristiti u okviru sloja poslovne logike informacionog sistema.

LITERATURA

[1] N. Petrović, "COVID-19 Prevention Systems within Hardware and Software Design Computer Science Courses", 2022 57th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST), 2022, pp. 173-176. https://doi.org/10.1109/ICEST55168.2022.9828724

[2] N. Petrović, V. Roblek, N. Papachashvili, "Decision Support Based on Data Mining for Post COVID-19 Tourism Industry", SAUM 2021, pp. 29-32, 2021.

[3] Weka 3: Machine Learning Software in Java [online], available on: https://www.cs.waikato.ac.nz/ml/weka/, last accessed: 14/09/2022.

[4] D. Heffelfinger, Java EE 8 Application Development, Packt Publishing, 2017.

[5] M. Mehryar, A. Rostamizadeh and A. Talwalkar, Foundations of Machine Learning, The MIT Press, 2nd edition, 2018.
[6] N. Petrovic, "Churn Prediction in Telco Industry Leveraging Call Center Data", IcETRAN 2019, 845-850, 2019.
[7] Telco Customer Churn [online], dostupno na: https://www.kaggle.com/datasets/blastchar/telco-customer-churn, poslednji pristup: 04/11/2022.
[8] N. Petrović, Đ. Kocić, "Data-driven Framework for Energy Efficient Smart Cities", Serbian Journal of Electrical Engineering, Vol. 17, No. 1, Feb. 2020, pp. 41-63, 2020.
[9] Energy efficiency Data Set [online], dostupno na: https://archive.ics.uci.edu/ml/datasets/Energy+efficiency, poslednji pristup: 04/11/2022.

MOS tranzistor sa plivajućim gejtom kao dozimetar ultraljubičastog zračenja u fototerapiji

Stefan Ilić, Milija Sarajlić, Goran Ristić

Sadržaj – U ovom radu su obrađene karakteristike električno programabilnog MOS tranzistora sa plivajućim gejtom kao dozimetra ultraljubičastog zračenja za potrebe fototerapije. Pokazano je da komercijalni MOS tranzistor sa plivajućim gejtom u dekapsuliranom kućištu može zabeležiti primljenu dozu UVB zračenja kao promenu napona praga. Proces očitavanja je nedestruktivan, nije potrebno napajanje tokom ozračivanja i moguće je čuvati infomaciju o primljenoj dozi duži period.

I. UVOD

Fototerapija ili lečenje svetlošću predstavlja vrstu lečenje određenih kožnih oboljenja, ali i nekih drugih bolesti (npr. rahitisa), dejstvom fotona iz ultraljubičastog spektra elektromagnetnog zračenja. Ultraljubičasta svetslost se koristi u terapijske svrhe za lečenje psorijaze, atopijskog termatitisa, ekcema i vitiliga. Takođe, pored ovih oboljenja primenjuje se u lečenju pitirijasis lihenoides, pitirijasis rozea, generalizovanog pruritusa (svraba), mikozis fungoides (ranog stadijuma kožnog T ćelijskog limfoma), parapsorijaze i lihen planusa [1].

Veštački izvori ultraljubičaste svetlosti koji se koriste u fototerapiji su najčešće živine lampe koje pored korisnog dela elektromagnetnog zračenja sadrže i druge delove spektra. Poboljšanjem tehnologije razvijene su uskopojasne (eng. narrowband) lampe koje imaju najveći intenzitet u željenom delu spektra, dok je uticaj ostatka spektra minimizovan, ali ipak postojan. Razvojem ultraljubičastih LE dioda postoji težnja da se živine lampe izbace iz terapijske upotrebe zbog rizika oslobađanja žive u čovekovu okolinu incidentnim oštećivanjem lampe, takođe i zbog reciklaže živinih sijalica. Međutim, živine lampe su i dalje najzastupljeniji izvor ultraljubičastog zračenja u lečenju kožnih oboljenja. U novije vreme pod fototerapijom se ubrajaju i terapije infracrvenim zracima, kao i laserke terapije malih snaga [2].

S. Ilić je istraživač pripravnik pri Departmanu za mikroelektronske tehnologije, Instituta za hemiju, tehnologiju i metalurgiju, i student treće godine doktorskih studija Elektronskog fakulteta, Univerziteta u Nišu, E-mail: stefan.ilic@nanosys.ihtm.bg.ac.rs.

M. Sarajlić, viši naučni saradnik pri Departmanu za mikroelektronske tehnologije, Instituta za hemiju, tehnologiju i metalurgiju, Univerziteta u Beogradu, Njegoševa 12, 11000, Beograd, Srbija, E-mail: milijas@nanosys.ihtm.bg.ac.rs.

G. S. Ristić je redovni profesor na Katedri za mikroelektroniku, Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: goran.ristic@elfak.ni.ac.rs.

Pokazano je da fotoni koji pomažu u lečenju kožnih oboljenja imaju talasnu dužinu iz UVB spektra koji iznosi od 280 do 315 nm. Tokom izlaganja sunčevoj svetlosti, 7-dehidroholesterol u koži apsorbuje UVB zračenje i pretvara se u previtamin D3, koji se zauzvrat izomerizuje u vitamin D3. Previtamin D3 i vitamin D3 takođe apsorbuju UVB zračenje i pretvaraju se u razne fotoproizvode, od kojih neki imaju jedinstvena biološka svojstva [3].

Kontraindikacije tretmanima ultraljubičastog zračenja na čoveka su najčešće fototoksični solarni dermatitis, fotoalergijske reakcije, solarna elastoza i keratoza (karcinom kože) i fotokeratitis (katarakta). Zbog toga je potrebno vršiti precizno doziranje ultraljubičastog zračenja svakom pacijentu individualno, zbog različitih tipova i osetljivosti kože. U nekim slučajevima, važnije od određivanja doze je vršiti proveru primljene (absorbovane) doze pacijenta, kao bi se sa sigurnošću potvrdila primljena doza, i izvršile potrebne korekcije ako se uoče neke od kontraindikacija.

Za navedene potrebe merenja primljenje doze ultraljubičastog zračenja potrebno je napraviti pasivan dozimetar malih dimenzija (veličine flastera) sa brzim, jednostavnim i nedestruktivnim načinom očitavanja doze. Zadate uslove može zadovoljiti jedino (do sada poznato) poluprovodnička struktura sa zarobljenom određenom količinom naelektrisanja koju fotoni ultraljubičastog zračenja mogu smanjiti i time zabeležiti (upisati) vrednost doze. Takva poluprovodnička struktura je MOS tranzistor sa plivajućim gejtom.

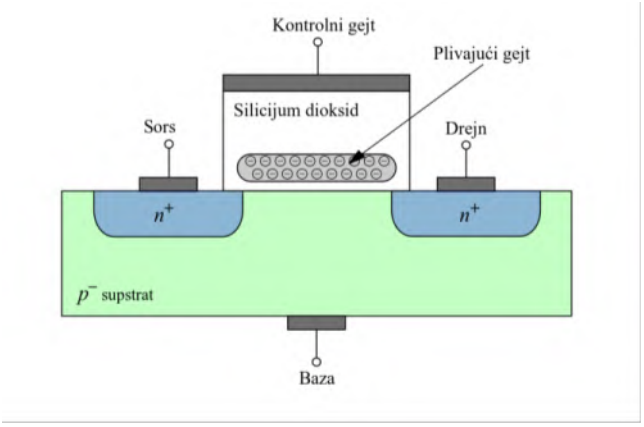
U nastavku će biti predstavljena struktura MOS tranzistora sa plivajućim gejtom, obrađene karakteristike i prikazani rezultati MOS tranzistora sa plivajućim gejtom kao dozimetrom ultraljubičastog zračenja.

II. MOS TRANZISTOR SA PLIVAJUĆIM GEJTOM

MOS tranzistori sa plivajućim gejtom su srž svake moderne stalne memorije u računarima [4]. MOS struktura sa plivajućim gejtom predstavlja MOS tranzistor koji se sastoji od dva gejta: plivajući i kontrolni gejt (Sl. 1). Plivajući gejt je okružen silicijum dioksidom i time izolovan od ostatka MOS strukture, napravljen je od polisilicijuma. Kontrolni gejt ima funkciju gejta kod običnog MOS tranzistora.

Na slici 1 je prikazan n-kanalni MOS tranzistor sa napunjenim plivajućim gejtom. Proces punjenja plivajućeg gejta elektronima se realizuje dovođenjem pozitivnog napona na kontrolni gejt (tipično oko 12 V) kojim se

privlače elektroni iz kanala prema plivajućem gejtu. Kako bi elektroni prevazišli potencijalnu barijeru silicijum dioksida i došli do plivajućeg gejta potrebno je da imaju dovoljnu energiju koja zavisi od električnog polja sa kontrolnog gejta. Elektroni se injektuju u plivajući gejt po principu vrućih nosioca i tako oštećuju silicijum dioksid, što ograničava maksimalni broj mogućih punjenja plivajućeg gejta pri kojima je naelektrisanje stabilno, tj. ne curi tokom vremena [6]. Komercijalni MOS tranzistori sa plivajućim gejtom mogu čuvati naelektrisanje do 10 godina [7], što ovoj komponenti daje veliku prednost za potrebe dozimetra ultraljubičastog zračenja jer se mogu uvek proveriti vrednosti doze koje je pacijent primio tokom terapije i čuvati u arhivi kao dokaz.



Slika 1. N-kanalni MOS tranzistor sa plivajućim gejtom [5].

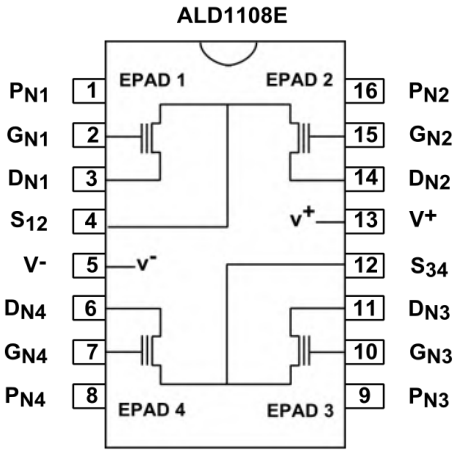
Kada se plivajući gejt napuni elektronima dolazi do promene napona praga MOS tranzistora jer za formiranje kanala je potrebno dovesti veću vrednost napona kako bi se poništio uticaj električnog polja sa plivajućeg gejta nego vrednost napona koja je potrebna za formiranje kanala kada je plivajući gejt bio nenapunjen elektronima. Dakle, sa povećanjem naelektrisanja na plivajućem gejtu raste napon praga n-kanalnog MOS tranzistora [8].

Pražnjenje plivajućeg gejta je moguće električnim putem dovođenjem negativnog napona na kontrolni gejt i uzemljenjem ostalih izvoda tranzistora. U ovom slučaju elektroni bivaju odvučeni električnim poljem sa kontrolnog gejta prema sorsu ili drejnu. Dakle, punjenjem i pražnjenjem plivajućeg gejta moguće je realizovati vrednost logične nule ili jedinice što čini osnovu današnjih permanentnih memorija. Starija poluprovodnička tehnologija nije omogućivala električno pražnjenje plivajućeg gejta, već samo njegovo punjenje elektronima. Dok je praženjenje bilo realizovano UV ozračivanjem memorije određeni period [8].

Kada fotoni pogode strukturu MOS tranzistora sa napunjenim plivajućim gejtom dolazi do predaje energije elektronima na plivajućem gejtu. Ako je ta energija veća od energije potencijalne barijere (silicijum dioksid/silicijum) koju je potrebno da elektroni prevaziđu kako bi napustili plivajući gejt i došli do kanala ili kontrolnog gejta u tom slučaju dolazi do pražnjenje plivajućeg gejta. Energija fotona je veća od potencijalne barijere kada su u pitanju talasne dužine iz oblasti UVB i UVC spektra što znači da ultraljubičasta svetlost može dovesti do pražnjena plivajućeg gejta, ali poluprovodnička struktura kao i kućište integrisanog kola mora biti tako realizovano (sa kvarcnim prozorom) da fotoni ultraljubičaste svetlosti mogu dopreti do plivajućeg gejta.

III. KARAKTERISTIKE ELEKTRIČNO PROGRAMABILNOG ANALOGNOG UREĐAJA

U ovom radu je korišćena komercijalna komponenta koja se naziva električno programabilni analogni uređaj (eng. *Electrically Programmable Analog Device*, skraćeno EPAD). EPAD predstavlja n-kanalni MOS tranzistor sa plivajućim gejtu. Integrisano kolo ALD1108E se sastoji od četiri EPAD-a. Na slici 2 je prikazan raspored pinova ovog integrisanog kola.



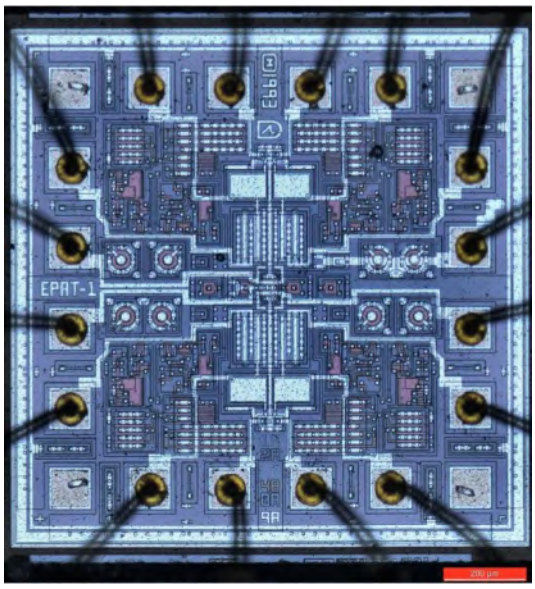
Slika 2. Integrisano kolo ALD1108E sa četiri MOS tranzistora sa plivajućim gejtom [7].

Kolo sadrži četiri identična EPAD-a, svaki od njih ima poseban pin za kontrolu punjenja plivajućeg gejta koji prilikom korišćenja tranzistora može da bude vezan za masu ili da pliva. EPAD1 i EPAD2 imaju zajednički sors, kao i EPAD3 i EPAD4 [7]. Jedan EPAD-a se sastoji od glavnog i pomoćnog MOS tranzistora. Uloga pomoćnog tranzistora je da vrši punjenje plivajućeg gejta koji je zajednički za oba tranzistora. Struktura EPAD-a je opisana u patentu proizvođača [9].

Na čipa ALD1108E svi EPAD-i imaju inicijalno napunjen plivajući gejt elektronima tako da im je vrednost napona praga tačno 1 V. Ovi tranzistori imaju mogućnost jako finog podešavanja vrednosti napona praga i zato imaju primenu u analognoj elektronici kada je potrebno upariti tranzistore tako da imaju identične karakteristike. Punjenje EPAD se realizuje dovođenjem napona na gejt u vrednosti od 12 V i propuštanjem struje kroz kanal tranzistora. Međutim, zbog zaštitne diode između gejta i supstrata nije moguće dovesti negativne napone, čime je pražnjenje plivajućeg gejta onemogućeno.

Do sada je istraživano ponašanje EPAD-a u polju jonizujućeg zračenja, gde gama fotoni dovode do

pražnjenja plivajućeg gejta [10, 11]. Saradnjom sa Irskim nacionalnim Institutom „Tyndall" izvršena je profesionalna dekapsulacija plastičnih kućišta IC ALD1108E pri čemu su svi bondovi ostali neoštećeni, čime su zadržane mogućnosti merenja karakteristika EPAD-a, kao i njihovog električnog punjenja (Sl. 3).

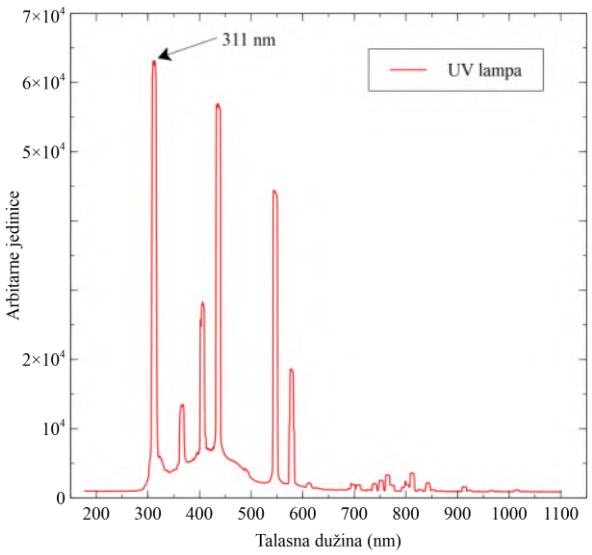


Slika 3. Dekapsulirano integrisano kolo ALD1108E . U centru čipa se nalaze četiri MOS tranzistora sa velikim oblastima plivajućeg gejta.

Kako je plastika iznad čipova dekapsulacijom uklonjena, omogućeno je UV ozračivanje EPAD-a.

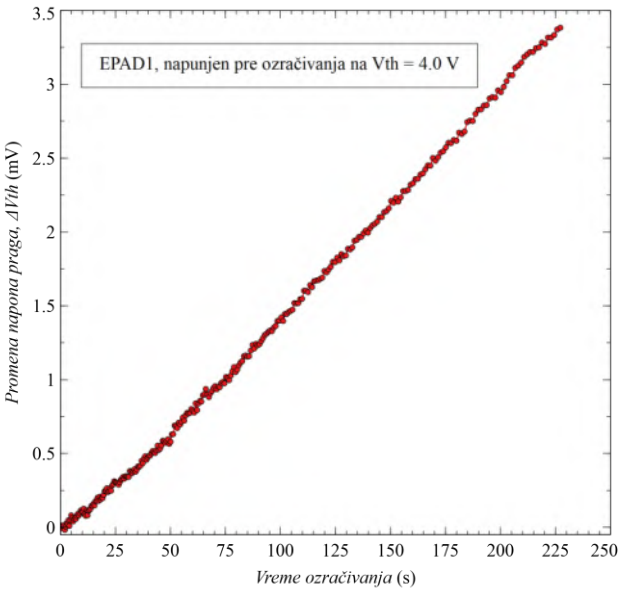
IV. Rezultati i diskusija

U ovom radu je korišćena uskopojasna UV lampa sa dve sijalice od 9 W koje se koristi u fototerapiji (Sl. 4).



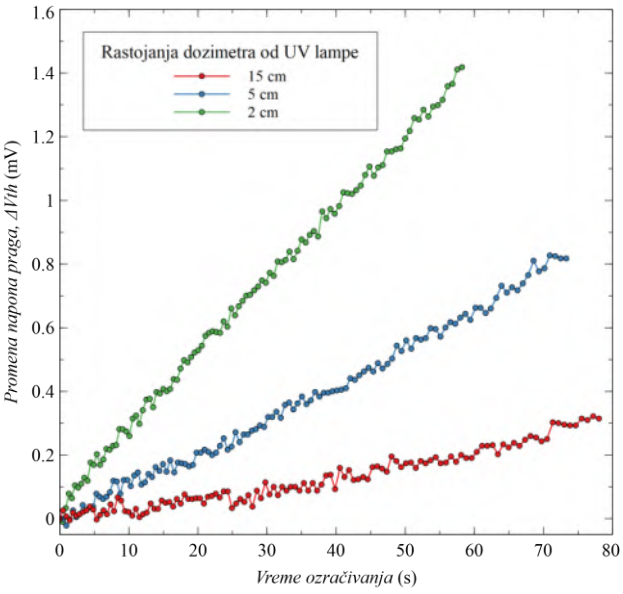
Slika 4. Spektar UV lampe koja se koristi u fototerapiji.

Na slici 4 prikazan je spektar UV lampe na kome možemo uočiti pik oko 311 nm koji se koristi u fototerapiji. Dekapsuliran EPAD, čiji je plivajući gejt prethodno napununjen, je ozračivan dozama ultraljubičastog zračenja koja se koriste najčešće u fototerapiji. Na slici 5 je prikazana promena napona praga EPAD-a tokom ozračivanja. Plivajući gejt je pre ozračivanja napunjen elektronima tako da je vrednost napona praga Vth = 4 V.



Slika 5. Promena napona praga EPAD-a tokom UV ozračivanja.

Možemo uočiti linearnu zavisnost promene napona praga sa vremenom ozračivanja. Linearnost za male doze je takođe primećena kod gama ozračivanja EPAD-a [11]. Na slici 6 je prikazana promena napona praga EPAD-a za različita rastojanja od UV lampe tokom ozračivanja.



Slika 6. Promena napona praga EPAD-a za različita rastojanja od izvora ultraljubičaste svetlosti.

Uočavamo da se promena napona praga za određeno vreme ozračivanja (osetljivost) smanjuje sa rastojanjem od UV izvora. Vrednosti osetljivosti za različita rastojanja su dati u Tabeli I.

TABELA I
OSETLJIVOST EPAD-A ZA RAZLIČITA RASTOJANJA OD ULTRALJUBIČASTOG IZVORA SVETLOSTI (UV LAMPA)

Broj	Rastojanje od izvora (cm)	Osetljivost (mV/s)
1	15	8.45937×10^{-5}
2	5	8.95601×10^{-5}
3	2	1.12405×10^{-4}

Napon praga EPAD-a nakon ozračivanja ima stabilnu vrednost sa vremenom, što ukazuje na mogućnost čuvanja informacije o primljenoj dozi ultraljubičastog zračenja.

Tokom ozračivanja MOS tranzistor sa plivajućim gejtom je u funkciji pasivnog dozimetra bez potrebe za napajanjem (nulta polarizacija na kontrolnom gejtu). Prilikom očitavanja doze dozimetar bi se povezivao sa uređajem koji meri napon praga i na osnovu razlike vrednosti napona praga pre i posle ozračivanja određivala bi se vrednost primljene doze.

V. ZAKLJUČAK

U fototerapiji postoji potreba za pasivnim dozimetrom malih dimenzija koji bi pomogao u preciznom određivanju doze za potrebe terapije. Predstavljen je princip rada i karakteristike MOS tranzistora sa plivajućim gejtom. Pokazano je da MOS tranzistor sa plivajućim gejtom ima karakteristike koje zahteva potencijalni dozimetar za potrebe fototerapije. Informaciju o vrednosti primljene doze moguće je čuvati dug period (do 10 godina), dozimetar nema potrošnju energije tokom ozračivanja i očitavanje doze je nedestruktivno. Moguće je ponavljati očitavanje doze koju je primio pacijent, kako se bi se potvrdila ili dokazala informacija o dozi. Prateći savremene tehologije, moguće je voditi elektronski dnevnik svakog pacijenta o primljenoj dozi ultraljubičastog zračenja.

ZAHVALNICA

Zahvaljujemo se Irskom nacionalnom Institutu „Tyndall” u Korku na odličnoj saradnji i pomoći u ovom istraživanju. Rad je realizovan u okviru projekata broj 43011 i broj 451-03-68/2022-14/200026 finansiranih od strane Ministarstva prosvete, nauke i tehnološkog razvoja Republike Srbije, kao i Evropskog projekta, WIDESPREAD-2018-3-TWINNING, grant broj 857558 – ELICSIR.

LITERATURA

[1] Krutmann, Jean, and Herbert Hönigsmann. Dermatological phototherapy and photodiagnostic methods. Eds. Craig A. Elmets, and Paul R. Bergstresser. Vol. 8. Berlin:: Springer, 2009.
[2] Kemény, Lajos, Emese Varga, and Zoltan Novak. "Advances in phototherapy for psoriasis and atopic dermatitis." Expert review of clinical immunology 15.11 (2019): 1205-1214.
[3] Wacker, Matthias, and Michael F. Holick. "Sunlight and Vitamin D: A global perspective for health." Dermato-endocrinology 5.1 (2013): 51-108.
[4] Pavan, Paolo, Luca Larcher, and Andrea Marmiroli. Floating gate devices: operation and compact modeling. Springer Science & Business Media, 2007.
[5] Ilić, Stefan, et al. "Floating-gate MOS transistor with dynamic biasing as a radiation sensor." Sensors 20.11 (2020): 3329.
[6] Edgecock, R., et al. "Evaluation of commercial programmable floating gate devices as radiation dosimeters." Journal of Instrumentation 4.02 (2009): P02002.
[7] Advanced Linear Devices, Inc. Dostupno na: https://www.aldinc.com/pdf/ALD1110E.pdf
[8] Ilić, Stefan and Miloš Marjanović, “Ispitivanje debljine oksida u Flash EEPROM ćeliji pomocu TCAD alata”, IEEESTEC – 10th Student Projects Conference, vol. 10, pp. 243-246, 2017.
[9] Chao, Robert L. "Integrated electrically adjustable analog transistor device." U.S. Patent No. 5,309,009. 3 May 1994.
[10] Ilić, Stefan, et al. "Feasibility of applying an electrically programmable floating-gate MOS transistor in radiation dosimetry." 2019 IEEE 31st International Conference on Microelectronics (MIEL). IEEE, 2019.
[11] Ilić, Stefan D., et al. "Recharging process of commercial floating-gate MOS transistor in dosimetry application." Microelectronics Reliability 126 (2021): 114322.

U susret jubileju: naša petnaesta godina kroz radionice i druženje

Emilija Živanović, Miloš Marjanović, Sandra Veljković, Nikola Mitrović, Jana Vračar, Danijel Danković

Sadržaj – Cilj ovog rada je da prikaže aktivnosti koje smo realizovali između dve IEEESTEC konferencije studentskih projekata. Kroz te aktivnosti moći će da se prati saradnja Elektronskog fakulteta sa relevantnim institucijama iz zemlje i sveta.

I. UVOD

Kad se nešto radi sa puno entuzijazma i srcem, rezultati su neminovni. To potvrđuje i IEEESTEC konferencija studentskih projekata koja se po 15. put održava ove godine na Elektronskom fakultetu u Nišu. IEEESTEC konferenciju studentskih projekata organizuju IEEE studentski ogranak Niš [1], EESTEC lokalni komitet Niš [2] i Elektronski fakultet, Univerziteta u Nišu [3]. Ko su sve pokrovitelji konferencije od osnivanja do danas, kao i detalji o tome kako je tekao njen razvoj, koliko je projekata međunarodnih i domaćih realizovano da bi ona bila moguća i tako uspešna, koliko je učenika i studenata prohađalo brojne radionice u okviru njih, kako su same radionice realizovane, može se pročitati u već publikovanim radovima [4]-[8]. Cilj ovog rada je prezentovati aktivnosti koje su realizovane u poslednjih godinu dana i koje su prethodile ovom našem malom jubileju.

Poznato je da je u prethodnih 14 godina tradicija bila da svaka konferencija bude nečemu posvećena, tako da ovogodišnju posvećujemo nama samima i našem jubileju. Koliki trud i energija su zaista uloženi, šta nas motiviše da i dalje radimo na ovom „projektu" videće se u onome što sledi.

II. AKTIVNOSTI

IEEE Electron Devices Society (EDS) [9] i IEEE Solid-State Circuits Society (SSCS) [10] Srbija i Crna Gora, kao i EDS/SSC studentski ogranak Univerziteta u Nišu (EDS/SSC SB) organizovali su nekoliko aktivnosti čiji je krajnji cilj bilo motivisanje mladih da realizuju kroz projekte neke svoje ideje koje bi prezentovali na ovogodišnjoj IEEESTEC konerenciji.

E. Živanović, M. Marjanović, S. Veljković, N. Mitrović, J. Vračar i D. Danković su sa Katedre za mikroelektroniku, Elektronskog fakulteta, Univerziteta u Nišu, Aleksandra Medvedeva 14, 18000 Niš, Srbija, E-mail: emilija.zivanovic@elfak.ni.ac.rs.

Prvi od događaja koji je izazvao veliko interesovanje bio je Galaksija kup 2022. koji je održan 26. marta 2022. godine u Tehničkoj školi u Zaječaru, po peti put. Učenici srednjih tehničkih škola Srbije su učesnici ovog takmičenja. Učenici su za realizaciju svojih projekata koristili mikrokontrolere i mikroračunarske platforme. Ovogodišnje takmičenje je imalo 18 timova iz 12 različitih škola. Predsednik IEEE EDS/SSCS ogranka, prof. dr Danijel Danković, i predsedavajući IEEE SB Univerziteta u Nišu, Miloš Marjanović, bili su članovi stručnog žirija. Svaki učesnik dobio je predlog kako da unapredi svoj projekat i prilagodi ga učešću na 15. IEEETSEC konferenciji.

9. aprila 2022. Naučno-tehnološki park u Nišu [11], bio je domaćini jubilarne 10. konferencije Technology 'n' Trends. Moderna web tehnologija bila je u fokusu ovogodišnje konferencije. Miloš Marjanović je održao predavanje na popularnu temu Internet of Things inspirišući grupu mladih inovatora da neki od svojih projekata prikažu na ovogodišnjoj IEEESTEC konferenciji.

Pored toga, aprila meseca je Nikola Mitrović studentima treće godine Građevinsko-arhitektonskog fakulteta Univerziteta u Nišu [12] održao predavanje o pametnim kućama (Slika 1). Nakon kratkog uvoda o pametnim kućama i IoT tehnologiji, razmatrana je ideja pametnog staklenog prozora i razgovaralo se o pametnom utikaču, kontroli temperature, nivou CO_2, a pomenuto je i osvetljenje. Zasnovano na primerima datim na predavanju, studenti su izneli neke ideje koje bi se mogle realizovati kao projekti za 15. IEEESTEC konerenciju.



Slika 1. Predavanje na temu Internet of Things.

Jedna vrlo važna aktivnost kako za motivisanje mladih za učešće na konferenciji, tako i samih organizatora bilo je

održavanje finalne runde IEEE Region 8 Student Paper Contest [13]. Takmičenje je održano u okviru IEEE R8 vodeće konferencije MELECON 2022, u italijanskom gradu Palermu, od 14. do 16. juna 2022. godine.

S obzirom na visok nivo kvaliteta radova i prezentacija, žiri nije imao lak zadatak da odabere pobednike takmičenja, koje je posle dve godine održano uživo. Veliko priznanje pripalo je Emiliji Ćojbašić, studentu Elektronskog fakulteta, koja je osvojila treću nagradu na ovom takmičenju (Slika 2). Njen rad nosio je naziv "Machine learning as an aid to predicting clinical outcome after stroke". Treba istaći da je rad odabran kao najbolji na 14. IEEESTEC konferenciji studentskih projekata [14], i kao takav dobio mogućnost učestvovanja na ovom prestižnom takmičenju.



Slika 2. Prezentovanje rada i potvrda o osvojenom trećem mestu.

Radove je ocenjivao žiri u sastavu, Prof. Paul Micallef sa Malte, Prof. Mike Hinchey iz Irske, Prof. Giambattista Gruosso iz Italije i prof. dr Vera Marković sa Elektronskog fakulteta u Nišu, koja je i koordinator ovog takmičenja.

Vrlo značajna aktivnost koju treba istaći jeste i poziv za učešće na dva online sastanka studentskom ogranaku Univerziteta u Nišu, u okviru IEEE Sekcija Srbija i Crna Gora i Sandra Veljković je imala tu čast da bude naš predstavnik (Slika 3).



Slika 3. Zahvalnica IEEE SSCS društva na učešću na online sastanku i prezentovanju aktivnosti.

Prvi sastanak, pod nazivom "Inspiring and Grooming Tomorrow's Circuit Stars" je organizovalo IEEE SSC društvo, dok je drugi sastanak bio u hibridnoj formi "EDS SRC R8 Chapters Meeting and MQ in Tarragona" u organizaciji IEEE ED društva. Tema oba sastanka je bila predstavljanje realizovanih aktivnosti i projekata u okviru Ogranaka, ali i najava budućih aktivnosti. Bilo je veoma inspirativno čuti druge ogranke, slušati o realizovanim radionicama širom sveta, a posebno je bila naglašena posvećenost studenata i učenika, budućih inženjera, radu i učenju.

Alvin Loke i Abira Altvater iz IEEE SSCS-a su uputili poziv za učešće na online sastanku, dok su Benjamin Iñiguez i Lluis F. Marsal iz IEEE EDS-a uputili poziv za učešće u hibridnom sastanku. Oba skupa su bila održana krajem juna, i svim organizatorima dugujemo veliku zahvalnost na ukazanom poverenju.

Nakon niza radionica u okviru vrlo uspešno realizovanog projekta "STEM visits IEEESTEC" [15] i kviz takmičenja učenika srednjih škola, održan je i završni krug kviza na Elektronskom fakultetu u Nišu. Kviz i završna radionica su održani 24. maja 2022.

Desetak studenata iz tehničke škole „Mija Stanimirović" [16], elektrotehničke škole „Nikola Tesla" [17], gimnazije „Svetozar Marković" [18] i Tehničke škole iz Vlasotinca [19], učestvovalo je u kvizu (Slika 4).



Slika 4. Svi predavači i učesnici završnog kviza.

Elektronski fakultet je omogućio prigodne nagrade za najbolje od njih. U okviru ove manifestacije prof. dr Danijel Danković i njegovi saradnici održali su popularno predavanje o diodama uz prateću praktičnu realizaciju, dok je prof. dr Goran Ristić održao je predavanje o jonizujućem i nejonizujućem zračenju. Prof. dr Emilija Živanović, Miloš Marjanović, Jana Vračar, Neda Stanojević, i Marija Živković takođe su dali svoj doprinos u realizaciji ovih aktivnosti (Slika 5).

III. PROJEKTI

Kako je vrlo uspešno realizovan sličan program 2021. godine koji je ocenjen kao kreativan i vrlo inovativan "STEM visits IEEESTEC", kada su studenti realizovali projekte koristeći Arduino platformu, ove godine nastavnicima i saradnicima Katedre za mikroelektroniku odobren je projekat "Let STEM visit again IEEESTEC" u

okviru inicijative TryEngineering IEEE Pre-University STEM Portal Grant for 2022 [20] (Slika 6).



Slika 5. Realizatori i nagrađeni učesnici finalne runde kviza.

Slika 6. TryEngineering IEEE Pre-University STEM Portal Grant program.

Projekat se realizuje na Elektronskom fakultetu Univerziteta u Nišu sa učenicima srednjih škola. Našim preduniverzitetskim STEM programom, nauka (Science), tehnologija (Technology), inžinjerstvo (Engineering) i matematika (Mathematics), kroz interaktivne radionice i finalno takmičenje učenici uče da rade u timu, razmenjuju ideje, nauče da brzo identifikuju i reše problem i dođu do krajnjeg rezultata - praktične implementacije projekta. Tokom realizacije projekta u različitim fazama obuhvaćeno je oko 100 srednjoškolaca iz 5 različitih škola (Gimnazija „Bora Stanković“ Niš [21], ETŠ „Mija Stanimirović“ Niš [16], ETŠ „Nikola Tesla“ Niš [17], TŠ „Rade Metalac“ Leskovac [22] i Tehnička škola Vlasotince [19]) koji će kroz brojne radionice učestvovati u njegovoj realizaciji. Ove godine program će biti posvećen ARM mikrokontrolerima, zbog njegove velike važnosti u svetu integrisanih sistema. U implementaciji će se koristiti razvojna ploča *Nucelo*, a biće obrađene sve osnovne teme: digitalni pinovi, analogno-digitalna konverzija, prekidači, PWM signali, implementacija komunikacionih protokola. Pored već pomenutog razvojnog okruženja, koriste se matične ploče i dodatne elektronske komponente: senzori, LE diode, motori, LCD ekrani. Poseban značaj radionica ogleda se u kombinaciji hardvera i softvera. Osnovni cilj radionica je da osposobe učenike da: identifikuju primene odabranih elektronskih komponenata, tumače tehničku dokumentaciju sa podacima o komponentama, povezuju i testiraju određena elektronska kola. Pored tog cija, kada učenici na kraju kursa završe radionicu o pisanju naučnog rada, biće spremni za pisanje svog rada za 15. IEEESTEC konferenciju u čemu će im pomoći realizatori projekta.

Pored ovog već pomenutog, od strane Centra za Promociju nauke [23] (Slika 7) podržan je i projekat „U susret 15. IEEESTEC-u“.



Slika 7. Centar za promociju nauke finansira ovaj projekat.

Tim povodom Centar za promociju nauke organizovao je 11. jula 2022. dodelu priznanja dobitnicima projekata promocije nauke, u sklopu obeležavanja Dana nauke. Priznanje je u ime Elektronskog fakulteta, kao rukovodilac projekta primio prof. dr Danijel Danković (Slika 8).



Slika 8. Sa dodele priznanja.

Projekat Elektronskog fakulteta jedan je od 24 dodeljena projekta koji promovišu nauku, u konkurenciji od 150 prijavljenih projekata. Centar za promociju nauke je odobrio i 15 projekata koji će biti realizovani u naučnim klubovima širom Srbije [24].

Na skupu su se dobitnicima priznanja obratili Branko Ružić, prvi potpredsednik Vlade Republike Srbije i ministar prosvete, nauke i tehnološkog razvoja, prof. dr Vladan Đokić, rektor Univerziteta u Beogradu, dr Marko Krstić, v.d. direktora CPN-a i dr Tanja Adnađević, rukovodilac sektora programskih aktivnosti CPN-a.

Ovaj projekat naučnog komuniciranja omogućiće učesnicima učenicima srednjih škola da realizuju sopstvene projekte iz praktične elektronike i urade naučnu analizu rezultata. Učenici učestvovanjem u interaktivnim radionicama unapređuju svoja znanja u oblasti hardvera i softvera na jedan interesantan način. Usput rade na

unapređenju svojih veština pisanja, prezentovanja, rada u timu, učeći da brzo uočavaju, definišu i rešavaju probeleme. Pripremanjem rada za Konferenciju učenici razvijaju takmičarski duh koji ih podstiče da unaprede postojeće i razvijaju nove projekte. Cilj projektnih aktivnosti biće motivisanje učesnika da uz prevazilaženje straha od praktičnog rada, pored toga rade i na razvoju svojih ideja koje će kasnije doprineti i samom razvoju društva.

Sa realizacijom ovih projekata krenulo se početkom septembra 2022. godine na Elektronskom fakultetu u Nišu, prvom u nizu planiranih radionica. Učesnici prve radionice bili su petnaestoro učenika gimnazije „Bora Stanković“ iz Niša [21] (slika 9). Kroz radionicu su se učenici upoznali sa mogućnostima koje pruža *Arduino* i realizovali svoj prvi projekat - semafor.



Slika 9. Učesnici prve radionice.

U sklopu aktivnosti obeležavanja IEEE dana [25] na Elektronskom fakultetu u Nišu, 6. oktobra 2022. godine organizovana je radionica u okviru projekta “Let STEM visit again IEEESTEC”. U okviru ove aktivnosti učenici gimnazije „Bora Stanković“ iz Niša radili su sa svojim mentorima na praktičnoj realizaciji projekata za IEEESTEC konferenciju. Rađeno je na realizaciji detektora vlažnosti zemljišta, koncentracije gasova, uređaju za merenje temperature i vlažnosti vazduha i istraživana je primena *touch* senzora u aplikacijama kontrole osvetljenja i motora jednosmerne struje. U okviru radionice učenicima je održano predavanje o pravilima pisanja naučnoistraživačkih radova. Mentori učenika su saradnici sa Katedre za mikroelektroniku: Miloš Marjanović, Milan Stojanović, Jana Vračar, Aleksandra Stojković, Sandra Veljković i Neda Stanojević. Kroz niz realizovanih radionica i individualni rad mentora i učenika realizovani su projekti i napisani radovi za IEEESTEC konferenciju.

IV. ZAKLJUČAK

Kontinuitet u održavanju IEEESTEC konferencije jača našu motivaciju i omogućuje uspešnije premošćavanje jaza između nauke i mladih. Druženjem i razmenom ideja tokom radionica i pripremom za samu konferenciju, mladi pokazuju da je nauka posebno interesantna tek kada nađe svoju praktičnu primenu kroz realizaciju projekata iz oblasti elektrotehnike. Veoma je korisno to što se svaki od učenika nauči da „počne od malog”. To je važno za njihovo samopouzdanje i budući izbor zanimanja za koji ne očekuju da će na početku karijere biti savršen, ali ono što moraju da nauče je da budu uporni u rešavanju problema. To je od velike važnosti za razvoj učeničkih veština, a stečena znanja su im veliki zalog za budućnost. Koristeći iskustvo stečeno kroz radionice, vežbanje i takmičenja, učenici prepoznaju koje veštine im je potrebno više da razviju. Jedan od vrlo važnih rezultata svega ovoga je i dobra prilika za međusobno upoznavanje učenika sa istim

interesovanjima, ali i mogućnost za sticanje novih prijateljstava i saradnju.

Dajemo sve od sebe da naša misija ove godine i po petnaesti put bude uspešno realizovana. Kruna našeg dugogodišnjeg rada može se najbolje ogledati u publikovanju rada [26]. Članovi organizacionog odbora IEEESTEC konferencije će i dalje vredno raditi i truditi se da uvođenjem novih aktuelnih stvari Konferenciju učine još interesantnijom. Na taj način Konferencija će privući još veći broj mladih učesnika čime će direknto učestvovati na popularizaciji naučno-istraživačkog rada.

Treba napomenuti da će se i po završetku 15. IEEESTEC konferencije intenzivno nastaviti sa radionicama i mentorskim radom sa učenicima. Na ovaj način omogućiće se učenicima više različtih škola Južne Srbije da se intenzivno pripremaju za različita takmičenja, sajmove, konferencije, simpozijume... Takođe, intenzivna saradnja nastaviće se i sa profesorima srednjih tehničkih škola i gimanzija, a sve u cilju promocije i popularizacije nauke i tehnike.

ZAHVALNICA

Zahvaljujemo svim ustanovama i institucijama koje podržavaju IEEESTEC konferenciju.

Zahvaljujemo svim članovima Katedre za mikroelektroniku Elektronskog fakulteta Univerziteta u Nišu, koji nam daju ogromnu podršku prilikom organizacije IEEESTEC konferencije.

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] M. Marjanović, i D. Danković, "Deset godina Konferencije studentskih projekata IEEESTEC", *in Proc. 10th IEEESTEC 2017*, Niš, Serbia, 2017, pp. 1-4.

[5] M. Marjanović, D. Danković, i D. Danković, "Projekat Centra za promociju nauke "Pretočimo znanje u praktičan rad" realizovan u Naučnim klubovima Niš i Leskovac", *in Proc. 12th IEEESTEC 2019*, Niš, Serbia, 2019, pp. 105-108.

[6] D. Danković, i M. Marjanović, "Projekat Centra za promociju nauke "Za IEEESTEC se spremi" na Elektronskom fakultetu u Nišu", *in Proc. 13th IEEESTEC 2020*, Niš, Serbia, 2020, pp. 199-200.

[7] D. Danković, M. Marjanović, J. Vračar, N. Mitrović, S. Veljković, i E. Živanović, "IEEESTEC na-S-paja", *in Proc. 14th IEEESTEC 2021*, Niš, Serbia, 2021, pp. 283-286.

[8] S. Veljković, M. Marjanović, i D. Danković, "IEEE Electron Devices Society (EDS)/ Solid-State Circuits Society (SSCS) Joint Serbia and Montenegro Chapter and EDS/SSCS University of Niš Student Branch Chapter Recent Activities", *IEEE Solid-State Circuits Magazine*, Vol. 13, No. 3, 2021, pp. 54-55.

[9] IEEE Electron Devices Society (EDS). Dostupno na: https://eds.ieee.org/

[10] IEEE Solid-State Circuits Society (SSCS). Dostupno na: https://ethw.org/IEEE_Solid-State_Circuits_Society_History

[11] Naučno tehnološki park Niš. Dostupno na: https://ntp.rs/

[12] Građevinsko arhitektonski fakultet Niš. Dostupno na: https://gaf.ni.ac.rs/

[13] IEEE Region 8 Student Paper Contest. Dostupno na: https://ieeer8.org/student-activities/sa-announcements/student-paper-contest-2022/

[14]14. IEEESTEC konferencija studentskih projekata. Dostupno na: http://ieee.elfak.ni.ac.rs/sr/ieeesteclive-rs/

[15] Projekat "STEM visits IEEESTEC". Dostupno na: http://ieee.elfak.ni.ac.rs/stem-visits-ieeestec/

[16] ETŠ Mija Stanimirović Niš. Dostupno na: https://www.etsmijastanimirovic.edu.rs/

[17] ETŠ Nikola Tesla Niš. Dostupno na: http://nikola.etstesla.ni.ac.rs/

[18] Gimnazija Svetozar Marković. Dostupno na: https://gsm-nis.edu.rs/

[19] Tehnička škola Vlasotince. Dostupno na: https://tsvlasotince.edu.rs/

[20] IEEE TRYEngineer ing. [OnLine] Dostupno na: https://tryengineering.org/news/announcing-the-2021-ieee-pre-university-stem-portal-grant-recipients/

[21] Gimnazija Bora Stanković Niš. Dostupno na: https://www.borastankovic.edu.rs/

[22] TŠ Rade Metalac Leskovac. Dostupno na: https://www.rademetalac.edu.rs/

[23] Centar za promociju nauke. [OnLine] Dostupno na: http://www.cpn.rs/o-centru/

[24] Naučni klubovi. [OnLine] Dostupno na: http://www.cpn.rs/prograi/mreza-naucnih-klubova/

[25] IEEE dan. [OnLine] Dostupno na: https://ieeeday.org/

[26] D. Danković, M. Marjanović, N. Mitrović, E. Živanović, M. Danković, A. Prijić, Z. Prijić, "The Importance of Students' Practical Work in High Schools for Higher Education in Electronic Engineering", *IEEE Transaction on Education*, pp(99):1-10, 2022.